

В. Д. Руденко, Н. В. Речич, В. О. Потієнко

Інформатика

**ДЛЯ ЗАГАЛЬНООСВІТНІХ НАВЧАЛЬНИХ ЗАКЛАДІВ
З ПОГЛИБЛЕНИМ ВИВЧЕННЯМ ІНФОРМАТИКИ**

**Підручник для 9 класу
загальноосвітніх навчальних закладів**

Рекомендовано Міністерством освіти і науки України

Харків
Видавництво «Ранок»
2017

УДК [004:37.016](075.3)
P83

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 20.03.2017 № 417)

Видано за рахунок державних коштів. Продаж заборонено

Експерти, які здійснили експертизу підручника під час проведення конкурсного відбору проектів підручників для 9 класу загальноосвітніх навчальних закладів і зробили висновок про доцільність надання підручнику грифа «Рекомендовано Міністерством освіти і науки України»:

К. М. Брозинська, учитель інформатики Чернівецької ЗОШ І–ІІІ ступенів № 24 ім. Ольги Кобилянської, Чернівецької міської ради Чернівецької області, учитель-методист;

В. О. Носкін, методист Кременчуцького міського науково-методичного центру Кременчуцької міської ради Полтавської області;

Д. А. Покришень, завідувач кафедри інформатики та інформаційно-комунікаційних технологій в освіті Чернігівського обласного інституту післядипломної педагогічної освіти імені К. Д. Ушинського, доцент, канд. пед. наук

Руденко В. Д.
P83 Інформатика для загальноосвітніх навчальних закладів з поглибленим вивченням інформатики : підруч. для 9 кл. загальноосвіт. навч. закл. / В. Д. Руденко, Н. В. Речич, В. О. Потієнко. — Харків : Вид-во «Ранок», 2017. — 240 с. : іл.

ISBN 978-617-09-3355-3

УДК [004:37.016](075.3)



Інтернет-підтримка
Електронні матеріали до підручника розміщено на сайті interactive.ranok.com.ua

ISBN 978-617-09-3355-3

© Руденко В. Д., Речич Н. В., Потієнко В. О., 2017
© ТОВ Видавництво «Ранок», 2017

Дорогі учні!

У дев'ятому класі ви завершуєте перший етап вивчення основ інформатики. Це досить відповідальний період. Саме в цей час формуються не тільки основні знання й навички володіння комп'ютерними та інформаційними засобами, а й ключові компетентності, зокрема ІКТ. Дехто з вас ще не один рік системно вивчатиме інформатику і стане професійним ІТ-спеціалістом, дехто завершить її вивчення у школі і почне готуватися до оволодіння іншими спеціальностями. Але знання і навички з інформатики, набуті у 5–9 класах, допоможуть вам у подальшому успішному навчанні, самостійному оволодінні інформаційними технологіями, а також у повсякденній життєдіяльності.




Цього року ви будете працювати з новими програмними засобами. Ознайомтеся із сутністю інформаційного суспільства, етапами становлення інформаційних технологій і розвитком інформаційних систем. Вперше будете вивчати сутність електронного документообігу і комп'ютерних публікацій. Навчитесь розробляти найпростіші бази даних навчального призначення і працювати в комп'ютерних мережах. Ви також будете створювати персональне навчальне середовище, продовжите вивчення основ алгоритмізації та програмування.

Інформаційні технології — це величезне досягнення людства. Вміння працювати з інформацією буде корисним для вашого успішного навчання.

Бажаємо вам успіхів, *автори*

Підручник, який ви тримаєте в руках, — ваш надійний помічник. У ньому ви знайдете практичні роботи, виконуйте їх на комп'ютері з натхненням, повторюйте теоретичний матеріал і викладайте основні положення на папері.

Різноманітні питання для перевірки знань і завдання для самостійного виконання відповідають рівням навчальних досягнень:

-  — початковий і середній рівні;
-  — достатній рівень;
-  — високий рівень.

У тексті використано також позначки:



— означення, висновок



— питання на повторення



— зверніть увагу



— цікаво знати



— завдання для виконання й обговорення в парах або групах



— вправи для домашнього виконання



Розділ 1. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У СУСПІЛЬСТВІ

1.1. Інформатика та її вплив на розвиток суспільства



Пригадайте означення понять «інформація», «дані». Назвіть сфери використання комп'ютерів.



Термін «інформатика» виник у 60-х роках ХХ ст. у Франції. Він походить від слів *information* (інформація) і *automatique* (автоматика) й означає автоматизоване опрацювання інформації. Широко вживаний також англomовний варіант цього терміна — *computer science* (наука про комп'ютерну техніку).



Рис. 1. Комп'ютер — атрибут сучасного навчання



Слово «технологія» походить від грец. *techné* — майстерність, уміння і *logos* — учення, поняття. У матеріальному виробництві воно означає сукупність засобів і методів виготовлення продукту.

Сьогодні важко уявити роботу з великими обсягами даних без комп'ютера. Його винайдення за лічені десятиліття змінило життя людства, визначило нові способи діяльності в усіх сферах виробничого й суспільного життя. Все це сприяло зародженню нового наукового напрямку — *інформатики*.

Інформатика як наука сформувалася лише у другій половині ХХ ст., та вже стала необхідною складовою освіти сучасної людини. Нині існує чимало означень інформатики, що обумовлено багатогранністю її функцій, форм і методів. У широкому сенсі інформатика являє собою поєднання різних галузей науки, техніки, виробництва.

Як **фундаментальна наука** інформатика вивчає структуру і загальні властивості інформації, методи й засоби інформаційних процесів. Вона охоплює багато наукових напрямів.

Як **прикладна дисципліна** інформатика вивчає закономірності в процесах накопичення, опрацювання й розповсюдження інформації, моделює об'єкти (процеси та явища) в різних галузях людської діяльності, розв'язує наукові проблеми створення й упровадження комп'ютерних засобів.

Як **галузь людської діяльності** інформатика пов'язана з процесами опрацювання інформації за допомогою комп'ютера в різних сферах його застосування (рис. 1).

Головна функція інформатики полягає в розробці методів і засобів опрацювання даних та їхньому використанні в різних галузях людської діяльності.

Основні завдання інформатики спрямовані на:

- дослідження інформаційних процесів;
- розробку засобів комп'ютерної техніки й створення новітніх технологій опрацювання даних;
- вирішення наукових та інженерних проблем щодо ефективного використання апаратно-програмного забезпечення у всіх галузях людської діяльності.

Сучасне розуміння інформатики тісно пов'язане з поняттям інформаційної технології (ІТ), а також інформатизації. Інформатика є теоретичною базою для розвитку та становлення ІТ.

Існують традиційні матеріальні види ресурсів, такі як нафта, газ та інші корисні копалини. Сьогодні ж одним із найголовніших ресурсів суспільства є інформація. Процес опрацювання інформації за аналогією з обробкою матеріальних ресурсів також називають технологією.



Інформаційні технології — це сукупність методів пошуку, збирання, накопичення, зберігання та опрацювання даних за допомогою засобів обчислювальної й комунікаційної техніки.

Класифікують ІТ за різними ознаками. Найбільш розповсюдженими є такі:

- **за сферою застосування:** наука, виробництво, освіта, тощо;
- **за функціональними можливостями:** інструментальні (розробка інших технологій та їх обслуговування), прикладні (розв'язання задач певної предметної галузі), комунікативні (зв'язок, комунікації та спілкування);
- **за способом взаємодії з користувачем:** ІТ з командним рядком, із інтерфейсами WIMP (забезпечує діалог користувача з комп'ютером за допомогою графічних образів — меню, вікон та ін.) і SILK (наближений до звичайної форми спілкування людини з комп'ютером; знаходиться на стадії експериментальних досліджень) та ін.

ІТ увійшли в сучасне життя настільки, що застосування комп'ютера стало буденною справою. Вони впливають не лише на характер і темпи розвитку наукових і промислових галузей, а й на соціальну сферу, взаємовідносини між людьми та суспільство в цілому.

Створення глобального інформаційного простору сприяє ефективній інформаційній взаємодії людей, їх доступу до інформаційних ресурсів та різноманітних даних з електронних джерел усього світу (рис. 2).



Таким чином, інформатизація суспільства — це глобальний соціальний процес, особливість якого полягає в домінуванні такого виду діяльності у сфері суспільного виробництва, як збирання, накопичення, опрацювання, зберігання, передавання й використання даних.



Термін «інформаційна технологія» з'явився наприкінці ХХ ст. завдяки бурхливому розвитку персональних комп'ютерів і мереж. До їх появи основними засобами для роботи з повідомленнями були папір, олівець, фарба, книга, телефон, телеграф тощо.

Країнами з розвинутою інформаційною індустрією є Швеція, Норвегія, США, Німеччина, Японія. Наша країна — на шляху до інформаційного суспільства. Сьогодні за числом програмістів, які одержали сертифікати у сфері ІТ, Україна посідає четверте місце у світі.



Рис. 2. Виробнича нарада з використанням інформаційних технологій



Запитання для перевірки знань

- 1 У чому полягають головна функція та завдання інформатики?
- 2 Чому інформатику відносять до різних галузей: науки, техніки, виробництва?
- 3 Що означає поняття «інформаційні технології»?
- 4 За якими ознаками класифікують інформаційні технології?
- 5 Поясніть, чому у світі зростає кількість працівників, зайнятих в інформаційній сфері.
- 6 Чи сформувалося в нашій країні інформаційне суспільство? Відповідь обґрунтуйте.

1.2. Інформаційні процеси та інформаційні системи



Поясніть, що таке інформація. Наведіть приклади пристроїв для передавання повідомлень. У якому вигляді, на вашу думку, відбувається передавання повідомлень каналами зв'язку?



26 листопада — Всесвітній день інформації, який відзначається з 1994 р. за ініціативою Міжнародної академії інформатизації.

Протягом останніх десятиліть інформаційні процеси — процеси пошуку, зберігання, передавання та опрацювання інформації — здійснюються в основному засобами обчислювальної техніки й комп'ютерних мереж.

Пошук і збирання. Значну роль у цьому відіграють інформаційні пошукові системи (рис. 1), які дозволяють миттєво отримати будь-яку інформацію з електронних джерел усього світу. Особливою популярністю користується нині пошукова система Google.

Зберігання й накопичення. Дуже тривалий час знання передавалися в усній формі (у переказах, народних піснях тощо), потім — за допомогою матеріальних носіїв інформації (наскельних малюнків, глиняних дощочок, папіруса тощо). Окремо слід виділити книжки — книгодрукування мало величезний вплив на розвиток цивілізації. Сьогодні інформацію зберігають не тільки на паперових, але й на електронних носіях (рис. 2) — оптичних і магнітних дисках, картках пам'яті тощо; її накопичують в Інтернеті, який уже відіграє роль глобального носія інформації.

Передавання й поширення. Сучасні засоби передавання й поширення інформації фактично почали розвиватися з середини XIX ст., коли для цього на практиці стали застосовувати електричні сигнали. Величезний вплив на їх розвиток мав телеграфний апарат Морзе, за допомогою якого з 1844 року розпочалося трансатлантичне передавання телеграм. Потім з'явилися радіо, телебачення, інші засоби зв'язку; особливу роль у розвитку засобів передавання й поширення повідомлень відіграв супутниковий зв'язок (рис. 3), без якого нині неможливо уявити функціонування Інтернету.

Для передавання будь-яке повідомлення кодується — перетворюється з початкового вигляду до вигляду, придатного для передавання каналами зв'язку, після отримання приймачем сигнал декодується — перетворюється до початкового вигляду, зрозумілого й доступного людині.

Опрацювання. У широкому розумінні опрацювання даних — це процес їх перетворення з одного вигляду в інший. Для опрацювання даних розроблено різноманітні програмні засоби. Під час роботи з даними важливого значення набуває **захист інформації** — комплекс правових, організаційних і технічних заходів, спрямованих на зберігання інформації, запобігання випадковим і навмисним пошкодженням, несанкціонованому доступу.



Рис. 1. Приклади інформаційних пошукових систем



Рис. 2. Приклади носіїв даних



Рис. 3. Пристрої для передавання повідомлень



Всесвітньо відомий вислів «Хто володіє інформацією, той володіє світом» не втратив актуальності й нині. Але в умовах глобалізації суспільства важливого значення набуває захист інформації. Тому до цього вислову є сенс додати: «Хто захищає інформацію — той захищає себе й країну».



Для захисту даних у комп'ютері від випадкових загроз (збій напруги, помилка в роботі апаратних засобів тощо) створюють резервні копії файлів, а для тривалого їх зберігання — архіви.

Якщо комп'ютерна система має вихід у мережу Інтернет, дані слід захищати від намагань зловмисників і шахраїв отримати доступ до приватної інформації, наприклад паролів. Програми та дані потрібно захищати також від зараження комп'ютерними вірусами. Для цього існує низка програмних засобів (Захисник Windows, Зілля, Avast! та ін.), які під час вмикання комп'ютера здійснюють автоматичну перевірку наявності шкідливих програм та їх видалення.

Що означає поняття «хакер»? Шляхом пошуку в Інтернеті дослідіть питання про найвідоміші у світі хакерські атаки.

Інформаційні процеси досліджують шляхом їх моделювання на основі засобів математичної логіки, алгебраїчних та інших методів. Для цього широко застосовують принцип «чорного ящика», за яким досліджують вихідні дані як реакцію на зміну вхідних. Процес моделювання інформаційних процесів дозволяє отримати найголовніші їх характеристики, наприклад надійність, швидкість реалізації.

Удосконалення методів і засобів інформаційних процесів із часом вплинули на розвиток інформаційних технологій (ІТ).

Історія розвитку ІТ налічує кілька етапів.

- **Писемність** — фактично з її появою починається розвиток процесів зберігання, накопичення й передавання знань від одного покоління до іншого.
- **Книгодрукування** — розпочалося у XV ст. та сприяло стрімкому поширенню знань.
- **Електричні засоби зв'язку** — поява наприкінці XIX ст. телеграфа, телефона, радіо сприяла миттєвому передаванню повідомлень на велику відстань. Із першої половини XX ст. розпочалася ера телебачення.
- **Засоби обчислювальної техніки** — їх поява принципово прискорила й змінила методологію опрацювання інформації.
- **Всесвітня мережа Інтернет** — докорінно змінилися методи пошуку інформації, суттєво підвищилася швидкість і ефективність передавання великих обсягів даних на необмежені відстані (рис. 4).

Інформаційні процеси відбуваються в інформаційних системах (ІС). Навколо нас існує безліч різних ІС, наприклад, комп'ютер (технічна ІС), людина (біологічна ІС), суспільство (соціальна ІС) тощо.

Склад і функції ІС значною мірою визначаються тією предметною галуззю, у якій використовується система, а також завданнями, які необхідно розв'язувати за її допомогою.



Рис. 4. Інформація — найцінніший ресурс суспільства

Деякі вчені, аналізуючи ІТ, які використовуються в Інтернеті, порівнюють його з нейронною мережею та обговорюють питання про виникнення і розвиток нейронної мережі планети і становлення планетарного розуму.

Розглянемо основні складові сучасної ІС (рис. 5).

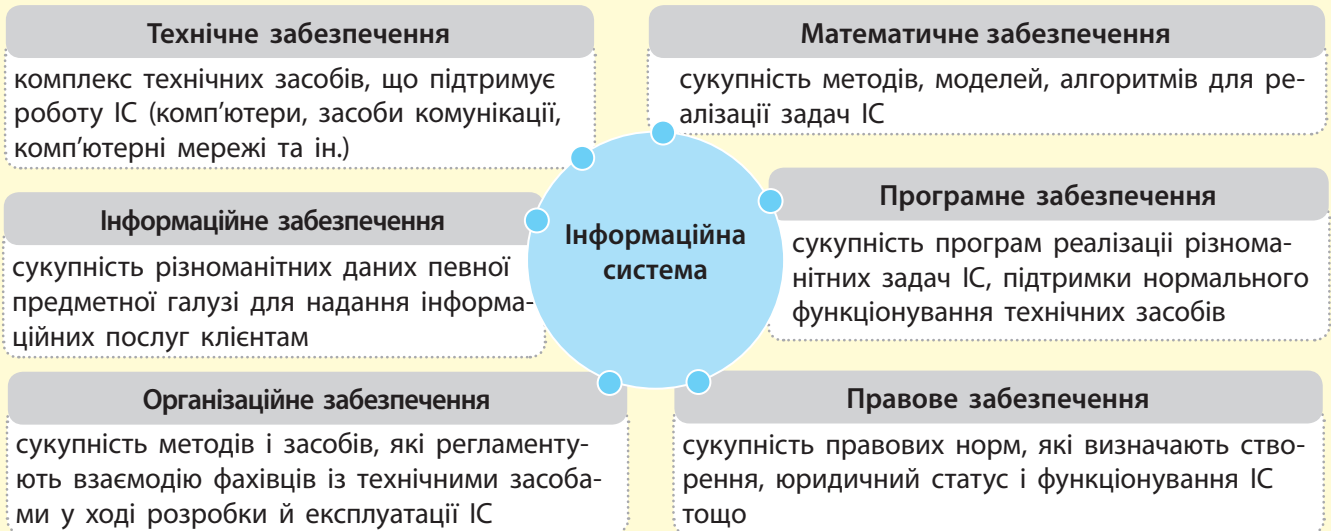


Рис. 5. Типовий склад ІС



Обсяг пам'яті людини невідомий. Але відомо, що американська піаністка Барбара Мур із 25 жовтня до 14 листопада 1988 року напам'ять програла 1852 музичні твори.



Під технічною інформаційною системою будемо розуміти програмно-апаратний комплекс для зберігання даних, виконання над ними операцій, обумовлених специфікою предметної галузі, забезпечення зручного інтерфейсу для доступу до необхідних даних та їх опрацювання.

Обов'язковими **складовими апаратного забезпечення ІС** є комп'ютер і засоби комунікації, а **складові програмного забезпечення ІС** залежать від функцій, які вона виконує. У більшості застосовуються операційні системи, СУБД, браузері (рис. 6).

Класифікують ІС за різними ознаками. До основних можна віднести такі:

- **за структурою даних, якими оперують ІС**: фактографічні (структуровані дані, наприклад особисті відомості людини — ПІБ, адреса тощо, склад комп'ютера) та документальні (наприклад, наказ, стаття, закон);
- **за сферою застосування**: економіка, освіта, медицина тощо;
- **за призначенням у професійній діяльності**: обчислювальні, пошуково-довідкові, управління персоналом тощо;
- **за рівнем автоматизації**: ручні, автоматизовані, автоматичні.



Рис. 6. Зовнішній вигляд комп'ютеризованого робочого місця



Запитання для перевірки знань

- 1 Поясніть поняття та сутність інформаційних процесів.
- 2 Які етапи розвитку інформаційних технологій ви знаєте?
- 3 Які основні переваги мають сучасні інформаційні системи?
- 4 Наведіть приклади галузей, у яких використовують інформаційні системи.
- 5 За якими основними ознаками класифікують інформаційні системи?
- 6 Які загальні риси мають інформаційні системи?

1.3. Поняття інформаційної культури й інтелектуальної власності

Поясніть, що означають поняття «культура», «культура мовлення», «художня культура». Як розвиток інформаційних технологій вплинув на розвиток людини, на її культуру?

Інформатизація суспільства впливає на всі сфери життєдіяльності людини, на розвиток її загальної культури, збагачує її новою складовою — інформаційною культурою.



Слово «культура» (від латин. *colo, colere* — вирощувати, доглядати, обробляти; *culturare* — вирощений, доведений до досконалості) спочатку стосувалося землеробства. Згодом стало вживатись у ширшому значенні — стосовно духовно-практичної діяльності людини.



Інформаційна культура як елемент загальної культури людини є характеристикою особистості, яка проявляється в умінні цілеспрямованої роботи з інформацією, знанні нових засобів інформаційно-комунікаційних технологій, морально-етичних норм під час створення й використання інформаційних матеріалів.

Інформаційна культура визначає здатність і вміння людини жити й працювати в інформаційному суспільстві.



Рівень сформованості інформаційної культури залежить від **інформаційної грамотності** людини — певного рівня знань щодо можливостей інформаційних і комунікаційних технологій та вміння їх використовувати в процесі роботи з інформацією для розв'язування різноманітних задач у своїй життєдіяльності.

Інформаційна грамотність, інформаційна культура людини впливають на її навчання, роботу, на успішне життя в інформаційному суспільстві (рис. 1). Сьогодні велику роль відіграє інформаційна компетентність, яку також називають ІКТ-компетентністю (від ІКТ — інформаційно-комунікаційні технології). У широкому розумінні — це обізнаність у світі інформації, знання сучасних засобів її опрацювання (рис. 2).



Рис. 1. Інформаційна грамотність — необхідна складова освіти



ІКТ-компетентність — це здатність особи самостійно шукати, аналізувати, опрацьовувати та передавати інформацію за допомогою ІКТ та доцільно використовувати її у своїй життєдіяльності.

ІКТ-компетентність учнів складають:

- уміння доцільно використовувати ІКТ в навчанні та повсякденному житті;
- уміння розробляти інформаційні моделі, реалізовувати їх за допомогою ІКТ, досліджувати, аналізувати отримані результати й робити висновки;
- практичні навички володіння ІКТ;
- знання основних правил і дотримання етичних норм у процесі використання ІКТ та участі в публічних дискусіях, соціальних мережах тощо.

Компетентності

Ключові

спрямовані на соціально продуктивну діяльність

Базові

застосовуються у різних професійних сферах

Предметні

призначені для конкретних дій

Рис. 2. Види ІКТ-компетентностей



Якими засобами ви користуєтесь під час підготовки реферату або виконання завдання, самостійного опанування навчального матеріалу? Чи звертались ви до сайта «Вікіпедія», сайтів електронних бібліотек?

Електронні освітні ресурси

- Електронні підручники, посібники
- Електронні демонстраційні, методичні матеріали
- Віртуальні бібліотеки
- Комп'ютерні тести, тренажерні системи
- Освітні сайти
- Мультимедіа-засоби навчання
- Курси дистанційного навчання та ін.

Рис. 3. Основні види електронних освітніх ресурсів

Для самостійного отримання знань із певного навчального предмета можна користуватися спеціально розробленими програмними засобами навчального призначення (освітніми електронними ресурсами) (рис. 3). Від традиційних їх відрізняє наявність зручних у використанні засобів унаочнення навчального матеріалу. Навчальні матеріали подаються на електронних носіях у вигляді текстових, графічних, звукових і відеоданих та їх комбінацій, що сприяє підвищенню інтересу до навчання.

Освітні електронні ресурси є у вищих та загальноосвітніх навчальних закладах. Їх мають бібліотеки, наприклад Національна бібліотека України імені В. І. Вернадського (nbuv.gov.ua) (рис. 4), Державна наукова педагогічна бібліотека України імені В. О. Сухомлинського (bnpb.gov.ua), а також освітні установи, видавництва тощо.

Програмний навчальний засіб, як і будь-яке програмне забезпечення, розробляється автором або авторським колективом. Автор має право інтелектуальної власності на розроблений ним програмний засіб так само, як і автор літературного чи художнього твору.



Право інтелектуальної власності — це право особи на результат своєї інтелектуальної або творчої діяльності.

Власність називають інтелектуальною, якщо вона є результатом роботи інтелектуальної та творчої діяльності особи або кількох осіб (наприклад, запатентовані винаходи, літературні, художні та наукові твори, фотографії, фільми, картини тощо).

Право інтелектуальної власності становлять немайнові та (або) майнові права. Особисті немайнові права інтелектуальної власності (наприклад, право на визначення особи творцем об'єкту власності) не залежать від майнових прав.



Авторське право — це право інтелектуальної власності на літературний, художній або інший твір, до яких належать і комп'ютерні програми.

Комп'ютерні програми, захищені авторським правом, мають ліцензію — правовий документ, який визначає порядок використання й розповсюдження програмного засобу. Ліцензія не тільки захищає авторські права, а й гарантує клієнтам безпечність та правильність функціонування програмного засобу. Користувачу, який має ліцензію, надається право отримувати технічну підтримку з боку розробника або офіційних розповсюджувачів програмного продукту.

Авторське право і суміжні права представлено в Законі України «Про авторське право і суміжні права». Закон ґрунтується



Рис. 4. Головна сторінка сайта Національної бібліотеки України імені В. І. Вернадського

на міжнародних нормах, враховує сучасні тенденції правового регулювання як традиційних, так і нових об'єктів авторського права — наприклад, комп'ютерних програм, баз даних.

У навчальному процесі можна використовувати фрагменти матеріалу без згоди автора. Якщо мета використання не містить комерційної складової, можна цитувати, використовувати як ілюстрацію, наприклад у рефератах тощо, але з обов'язковим посиланням на автора і джерело.

Програмне забезпечення, яке копіюється без дозволу власника, є «піратським». Проблема захисту програм та авторських прав їх розробників у всьому світі надзвичайно актуальна.

? Запитання для перевірки знань

- 1 Поясніть поняття інформаційної культури особистості.
- 2 Поясніть сутність інтелектуальної власності.
- 3 Від чого залежить формування інформаційної культури особистості?
- 4 Як ви розумієте ІКТ-компетентність учня?
- 5 Назвіть шляхи використання інформаційних технологій в освіті.
- 6 Що спричинило виникнення права інтелектуальної власності?

Практична робота № 1

Тема. Освітні інформаційні ресурси й системи.

Завдання: ознайомитися з освітніми інформаційними ресурсами, навчитися користуватися пошуковими системами.

Обладнання: комп'ютер, під'єднаний до мережі Інтернет.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Визначте, чи є на форумі інформатиків України (informatic.org.ua) посилання на сайт вашої школи (Головна → Корисні сайти → Сайти шкіл).
 - 2 Визначте, які освітні інформаційні ресурси знаходяться на сайті вашої школи або іншому за вказівкою вчителя.
 - 3 Ознайомтеся з навчальними мультфільмами на сайті форуму (Головна → Допомога користувачам → Навчальні мультфільми).
 - 4 На сайті Інституту модернізації змісту освіти МОН України (imzo.gov.ua) ознайомтеся з переліком електронних версій підручників (Освіта → Середня освіта і підручники → Електронні версії підручників). Чи є серед них такі, якими користуєтеся ви?
 - 5 Ознайомтеся з переліком олімпіад та конкурсів на сайті «Острів знань» (ostriv.in.ua): На головну → Школа → Для розумників → Олімпіади та конкурси. Про які з них ви щойно довідалися?
 - 6 На сайті «Розумники» (rozumniki.net) перегляньте перелік електронних підручників з інформатики для 9 класу. На сайті e-pidruchnyky.net ознайомтеся з переліком навчальної літератури з інформатики. Знайдіть електронну версію посібника «Операційні системи» та ознайомтеся зі змістом.
 - 7 Відвідайте сайт «Інтерактивне навчання» (interactive.ranok.com.ua). Виконайте пошук електронних версій підручників та навчальних посібників з інформатики для 9 класу, підручників з інших предметів.
 - 8 За допомогою пошукової системи Google виконайте пошук кількох дистанційних курсів (наприклад, із навчання програмування, іноземної мови). Які з них вас зацікавили?
 - 9 Повідомте вчителя про результати роботи.
- **Зробіть висновки:** які є освітні інформаційні ресурси, як знаходити навчальну інформацію в Інтернеті.

Розділ 2

ЕЛЕКТРОННИЙ ДОКУМЕНТООБІГ

2.1. Поняття електронного документообігу



Пригадайте, що таке текстовий документ. Які програми призначені для роботи з текстовими документами?



Слово «документ» походить від лат. *documentum* — зразок, взірць. Документування інформації пройшло тривалий шлях від документа, що виник майже одночасно з появою писемності, до електронного документа, історія якого пов'язана з виникненням комп'ютера.



Рис. 2. Засоби зберігання даних



Порядок обміну електронними документами, правовий статус електронного цифрового підпису та відносини, що виникають під час використання такого підпису, визначено Законами України «Про електронні документи та електронний документообіг», «Про електронний цифровий підпис», а також низкою нормативно-правових актів.

Робота з документами полягає у створенні оптимальних умов для всіх різновидів робіт, які можуть бути автоматизовані в офісі,— від моменту створення або отримання документів до моменту знищення або передачі в архів. Існують обов'язкові вимоги до створення та опрацювання несекретних офіційних документів незалежно від їхнього вигляду.

На рис. 1 наведено загальну схему роботи з документами.

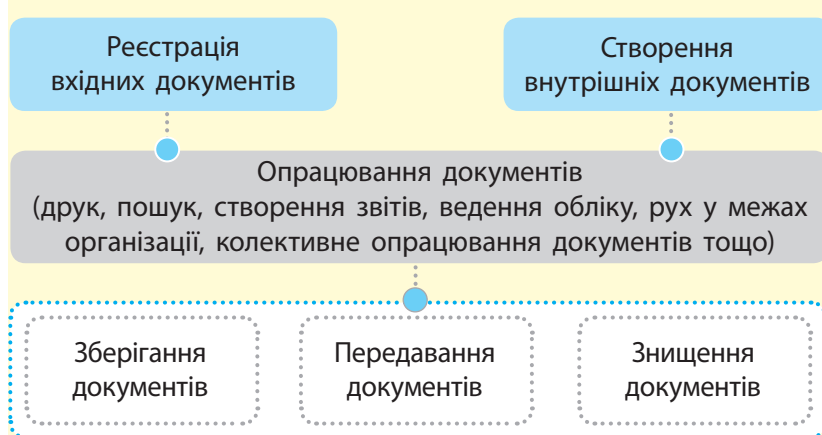


Рис. 1. Схематичне зображення процесу роботи з документами

В основу забезпечення роботи організації передусім покладено опрацювання документів.

Файл, повідомлення електронної пошти, сторінка сайту, база даних — все, що використовує користувач у своїй діяльності,— з технічної точки зору є *електронним документом* (ЕД). Носіями таких документів є засоби зберігання даних — від перфокарт до сучасних електронних засобів (рис. 2).

У зв'язку з розвитком інформаційно-комунікаційних технологій більшість робіт із документами сьогодні здійснюється за допомогою різноманітних технічних засобів (копіювальні апарати, сканери тощо). Обмін ЕД між підрозділами та організаціями здійснюють по комп'ютерній мережі.



Електронний документ — документ, інформацію в якому зафіксовано у вигляді електронних даних, включаючи обов'язкові реквізити.

ЕД складається з двох частин: *реквізитної* і *змістовної*.

Реквізитна частина містить атрибути, які надають документу юридичної сили (електронний цифровий підпис, час створення та ін.).

Змістовна частина містить текстові, графічні дані, що опрацьовуються як одне ціле й становлять інформаційну сутність документа.

Оригіналом ЕД вважають електронний примірник документа з обов'язковими реквізитами, у тому числі з електронним підписом.

Електронний цифровий підпис — це особливий реквізит документа (рис. 3), який формується за спеціальними правилами криптографічного перетворення даних, надає йому юридичної сили та дозволяє встановити відсутність спотворення інформації в електронному документі.

Подавши відповідну заяву та пакет документів, електронний цифровий підпис та підтверджувальний сертифікат можна отримати у відповідних центрах сертифікації.



Рис. 3. Електронний цифровий підпис — обов'язковий атрибут ЕД



Електронний документообіг (обіг електронних документів) — сукупність процесів створення, опрацювання, відправлення, передавання, одержання, зберігання, використання та знищення електронних документів, які виконуються із застосуванням перевірки цілісності та в разі необхідності — з підтвердженням факту одержання таких документів.

Переваги впровадження електронного документообігу:

- доступність інформації, прозорість руху документів, відсутність потреби роздруковувати та зберігати паперові копії;
- пришвидшення процесу опрацювання документів;
- забезпечення оперативної й одночасної колективної роботи з документами, коли один і той самий документ може опрацьовуватися синхронно кількома працівниками.

Електронний документообіг регулюється певними нормативними документами як на національному, так і на міжнародному рівні.

Надалі в рамках знайомства з документообігом під ЕД будемо розуміти саме його змістовну частину. Під створенням ЕД розумітимемо створення нового документа за допомогою відповідних програмних засобів (наприклад, у текстовому процесорі), імпортування готового ЕД, перетворення паперового документа на його електронну копію (рис. 4).

Особливості роботи з ЕД

- Скорочення часу на обмін документами між підрозділами та організаціями
- Автоматизація потоків документів
- Автоматизація створення, управління, опрацювання, розповсюдження документів

Рис. 4. Переваги роботи з електронним документообігом



Запитання для перевірки знань

- 1 Дайте означення електронного документа.
- 2 Поясніть поняття електронного документообігу.
- 3 Який обов'язковий реквізит надає електронному документу юридичної сили?
- 4 Опишіть життєвий цикл електронного документа.
- 5 Чим регламентуються порядок і організація електронного документообігу?
- 6 Наведіть приклади використання електронних документів у професійній діяльності і побуті.

2.2. Апаратні засоби для забезпечення електронного документообігу



З яким документом працювати зручніше: поданим у паперовому чи електронному вигляді? Пригадайте пристрої, які використовуються для введення даних у комп'ютер.



Поява електронної документації є основною рисою розвитку інформаційного суспільства. У країнах Європи та в США багато років працює електронна система обігу документів. В Україні впровадження цих світових тенденцій (офіційно з 2003 року) стає дедалі більш актуальним, у першу чергу у сфері надання послуг.



Рис. 1. Сканер та багатофункціональний комплекс

Незважаючи на значне збільшення обсягу інформації, кількість документів на паперових носіях невпинно зменшується, а на електронних — зростає. Це пояснюється розвитком сучасних технологій роботи з інформацією.

Для опрацювання ЕД використовують різні апаратні засоби: комп'ютери, об'єднані в мережу, — для передавання та сумісного опрацювання документів, принтери — для виведення документів на друк, а також пристрої для копіювання, ламінування та знищення документів.

Для переведення паперового документа в електронну форму використовують різні апаратні та програмні засоби. Процес зчитування інформації сканером називають *скануванням*.



Сканер (від англ. *scanner, scan* — пильно розглядати) — пристрій для створення графічного образу текстового документа та графічних зображень, поданих зазвичай на папері, та переведення в електронну копію.

За способом сканування сканери поділяють на планшетні, аркушопротяжні, ручні, барабанні, слайдові, планетарні. Найпоширенішими є планшетні сканери. Їх механізм вміщено в герметичний корпус зі склом зверху, на якому й розміщують об'єкт для сканування (наприклад, аркуш паперу або сторінка книжки) сканованою поверхнею вниз.

Назви сканерів складаються з назв фірм, які їх випускають. Найбільш відомими є Xerox, Hewlett Packard (HP), Canon, Konica Minolta.

Планшетні сканери існують як окремі пристрої, так і в складі багатофункціональних комплексів: *принтер — копіювальний апарат — сканер — факс* (рис. 1). Такі комплекси є найбільш поширеними.



Пригадайте, що таке *роздільна здатність принтера, роздільна здатність екрана монітора*.



У 1857 році було винайдено апарат «пантелограф». У 1902 році німецький фізик Артур Корн одержав патент на технологію фотоелектричного сканування, яка стала основою сучасних факсів та інших приладів.

Основними характеристиками сканера є *оптична роздільна здатність, глибина кольору, швидкість сканування, формат сканованого документа*.

Оптична роздільна здатність вимірюється в точках на дюйм — dpi. Зрозуміло, що чим більша роздільна здатність сканера, тим якісніші зображення можна отримати. Так, для сканування друкованого тексту достатньо встановити 300 dpi, що становить приблизно 12 точок на міліметр (значення 600 і 1200 dpi дають гарний результат сканування).

Глибина кольору вимірюється в бітах. Цей показник відіграє дуже важливу роль у випадку, коли потрібно відсканувати кольорове зображення. Він вказує, скільки кольорових відтінків може розпізнати сканер. Сучасні сканери характеризуються 48-бітною глибиною кольору, тобто можуть розпізнати 2^{48} кольорових відтінків.

Для сканування можна встановлювати один із трьох режимів: *кольоровий, відтінки сірого, чорно-білий*.

Швидкість сканування вимірюється в кількості сторінок, які сканер може просканувати за одну хвилину, або в кількості секунд, потрібних сканеру для опрацювання однієї сторінки.

До характеристик сканера можна віднести і **розмір поверхні для розміщення документа**, що обмежує розмір аркуша паперу, з якого можна відсканувати документ.

Під час під'єднання сканера до комп'ютера треба встановити та налаштувати драйвер сканера. Сучасні операційні системи дають змогу автоматизувати цей процес. Розглянемо *алгоритм під'єднання сканера до комп'ютера* (ОС Windows за версією 8 і вище).



Матриця сканера — це той світлочутливий елемент, який приймає світло, відбите від об'єкта сканування. Властивості матриці суттєво впливають на характеристики сканера взагалі.

Іноді за типом матриці називають і типи сканерів: CIS-сканери (*Contact Image Sensor* — контактний датчик зображення), CCD-сканер (*Charge-Coupled Device* — пристрій із зарядовим зв'язком (ПЗЗ)).

Крок 1	Під'єднайте сканер до комп'ютера (зазвичай через порт USB), до живлення та ввімкніть кнопку живлення на сканері, якщо вона існує. Комп'ютер розпізнає сканер як новий пристрій і запустить драйвер сканера. Якщо з якихось причин комп'ютер не бачить сканер, вручну пропишіть місцезнаходження драйвера й запустіть його.
Крок 2	Покладіть документ для сканування на скло сканера (вниз потрібною стороною).
Крок 3	Відкрийте вікно Панель керування , виберіть Пристрої та принтери . У вікні, що відкривається, в розділі Принтери виберіть потрібний пристрій (рис. 2). Зелена галочка біля пристрою показує, що саме він вибраний для друку і сканування — за замовчуванням.
Крок 4	Викличте контекстне меню пристрою (рис. 3) та виберіть команду Почати сканування .

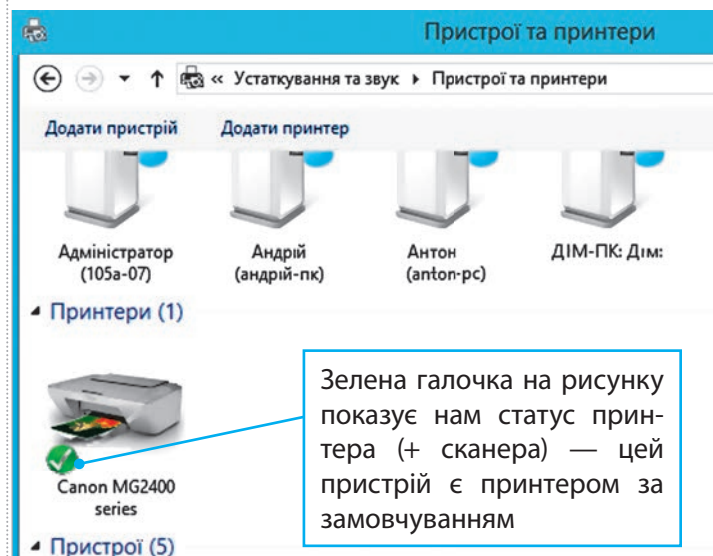


Рис. 2. Вікно пристроїв із під'єднаним пристроєм

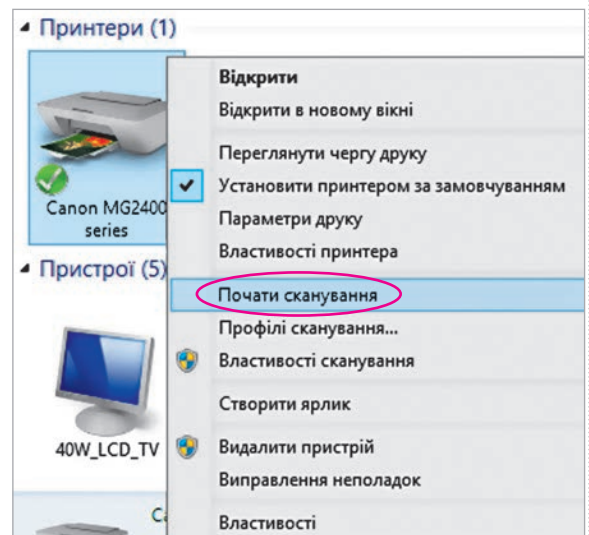


Рис. 3. Контекстне меню сканера

У вікні, що відкрилося, налаштуйте параметри сканування (рис. 4).

У прикладі вибрано такі налаштування:

- формат кольору — кольорове зображення;
- тип збереженого графічного файлу — JPG (BMP, JPG, PNG або TIF);
- роздільна здатність — 300 dpi;
- яскравість і контрастність (збільшити за потреби).

Крок 5

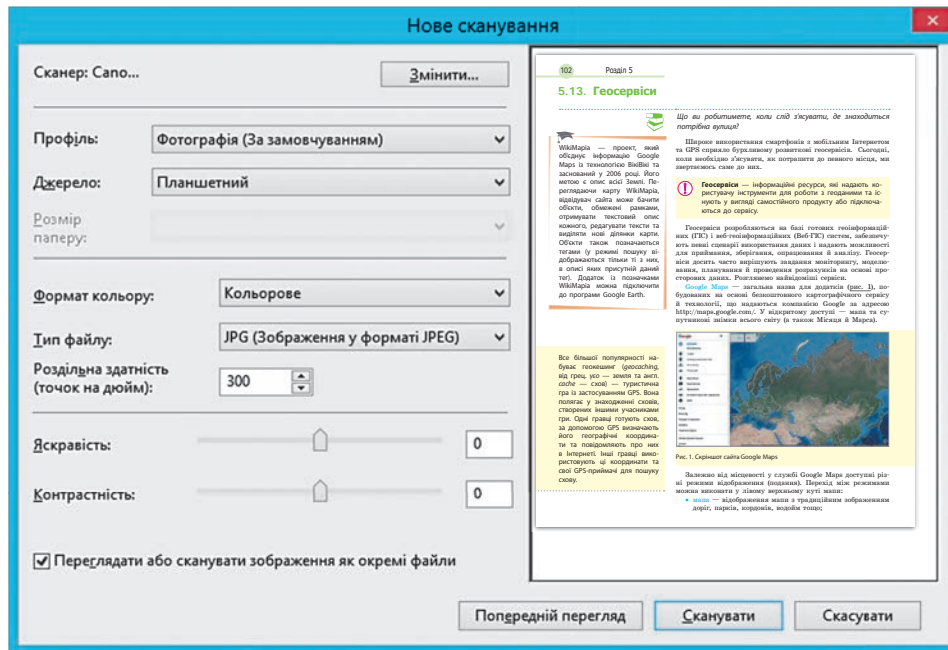


Рис. 4. Вікно налаштування параметрів сканування

Крок 6

Налаштувавши параметри, почніть сканувати, клацнувши кнопку **Сканувати**.

Крок 7

Збережіть відскановане зображення у файлі — введіть ім'я файлу у вікні, яке з'являється після завершення сканування.

Крок 8

Запустіть команду **Попередній перегляд** для перегляду відсканованого зображення.

Сьогодні цифрові копії документів можна отримати не лише за допомогою сканера. Заміною сканера стають смартфони, якщо вони мають потужну фотокамеру та програму для сканування. Можливо, такі документи й поступаються якістю, але цей спосіб швидкий і мобільний.

Графічний образ паперового документа в електронному вигляді також можна отримати з цифрового фотоапарата.

? Запитання для перевірки знань

- 1 Які пристрої призначені для введення текстових і графічних даних у комп'ютер?
- 2 Яку роль відіграють сканери в процесі документообігу?
- 3 У якому форматі зберігаються відскановані документи?
- 4 Назвіть характеристики сканера.
- 5 На що впливає така характеристика сканера, як глибина кольору?
- 6 Чи є різниця між файлами, в яких зберігається один і той самий документ після сканування та фотографування?

2.3. Програмне забезпечення електронного документообігу

Відсканували два аркуші: один із фотографією, другий із віршем. Чи будуть відрізнятися формати збережених після сканування файлів?

Процес переведення текстового паперового документа в електронну форму складається з етапів *сканування, розпізнавання, перевірки електронного варіанта на відповідність паперовому та збереження*. Для цього існують різні програми розпізнавання тексту (рис. 1).



Найбільш популярними програмами розпізнавання тексту є ABBYY Fine Reader, Solid Converter PDF, CuneiForm (безкоштовне програмне забезпечення), Microsoft Office Document Imaging. Існує онлайн-версія програми ABBYY Fine Reader, що дає можливість безкоштовно розпізнати і зберегти текст.

! **Програми розпізнавання тексту** — програмні засоби, призначені для перетворення тексту, відсканованого з паперового носія, в електронний документ.

Вікна програм оптичного розпізнавання тексту мають типовий вигляд вікон офісних програмних засобів. На рис. 2 наведено вікно програми розпізнавання CuneiForm. Такі самі інструменти для перетворення паперового документа в електронний є в усіх програмах розпізнавання. У робочій зоні вікна міститься документ, який підлягає розпізнаванню.

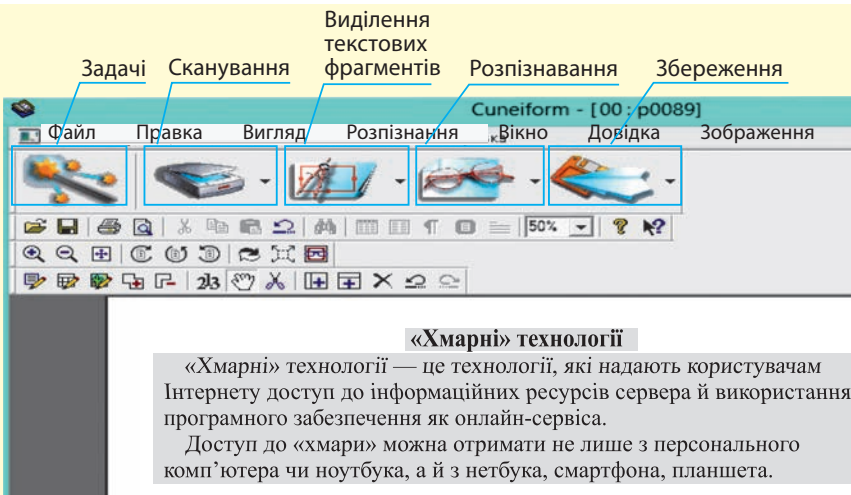


Рис. 2. Панель інструментів програми CuneiForm

Після виконання команди Розпізнати (інструмент Розпізнавання) у вікні Текст відображається розпізнаний текст. Символи, графічні зображення, які програма не може однозначно визначити, в текстовому документі позначено кольором, який відрізняється від кольору решти символів (рис. 3).

Якщо у вікні Текст неправильно відображаються символи (наприклад, на місці літери стоїть символ «?»), це означає, що поточний шрифт не підтримується обраною мовою розпізнавання. Треба змінити тип шрифту на закладці Оформлення (меню Властивості — Параметри) у групі Шрифти і розпізнати документ заново.



Рис. 1. Піктограми деяких програм розпізнавання тексту

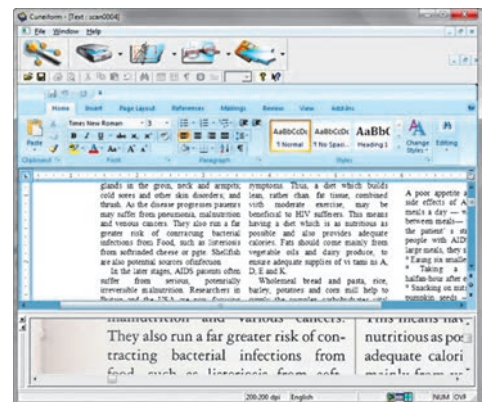


Рис. 3. Результат розпізнавання текстового документа

Під час сканування книги використовують пакетний режим: спочатку сканують всі сторінки — за допомогою команди Сканувати кілька сторінок із меню Файл, а надалі всі їх розпізнають, скориставшись командою Розпізнати з меню Процес.

У вікні Текст із розпізнаним текстом можна працювати ще до збереження, як у текстовому редакторі,— виділяти, вставляти, видаляти, копіювати. Відсканований документ можна зберігати як файл у форматі RTF (із графічними об'єктами), TXT (текст без графічних зображень), HTM (гіпертекстова сторінка).

? Запитання для перевірки знань

- 1 Яке призначення програм розпізнавання тексту?
- 2 Які програми розпізнавання ви знаєте?
- 3 У яких форматах можна зберегти розпізнаний текстовий документ?
- 4 Чи можна опрацьовувати програмою розпізнавання кілька сторінок тексту?
- 5 Як виділити ділянку, яка має бути в розпізаному документі незмінною?
- 6 Від чого залежить вибір формату файла з відсканованим документом?

Практична робота № 2

Тема. Сканування графічних зображень (малюнків, ілюстрацій, фотографій, схем тощо). Використання програм оптичного розпізнавання тексту.

Завдання: відсканувати документ та розпізнати в різний спосіб.

Обладнання: комп'ютер зі встановленим програмним засобом для розпізнавання тексту; сканер; документи для сканування; файли з відсканованими документами.

Примітка. Бажано підготувати для сканування два документи: один із текстом в одну колонку, другий — із текстом у кілька колонок та ілюстраціями. Результати сканування зберегти у файлах файл1 та файл2 відповідно для подальшого розпізнавання.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Відкрийте програму розпізнавання тексту, дотримуючись указівок учителя.
 - 2 Відкрийте файл1, який є результатом сканування першого документа.
 - 3 Виберіть команду Виділення фрагментів тексту.
 - 4 Виберіть команду Розпізнати.
 - 5 Збережіть отриманий документ.
 - 6 Відкрийте файл2, який є результатом сканування другого документа.
 - 7 Запустіть команду автоматичного виділення текстових фрагментів, розпізнайте та збережіть документ.
 - 8 Поверніться до команди Виділення фрагментів тексту. Виділіть фрагмент однієї з колонок тексту за допомогою миші — так, як виділяють прямокутні фрагменти в середовищі текстових редакторів.
 - 9 Розпізнайте виділений фрагмент.
 - 10 Збережіть отриманий документ.
 - 11 Закрийте програму. Проаналізуйте якість розпізнавання тексту в трьох отриманих документах.
- **Зробіть висновки:** в яких випадках можна запускати автоматичне виділення фрагментів, а в яких краще виділяти за допомогою миші.

2.4. Програми автоматизованого перекладу текстів

Поміркуйте, чи можна створити робота, що володіє багатьма мовами. Якщо так, то де його можна використувати?



Робота з документами в організації не обмежується переведенням паперових документів в їх електронний аналог. Чимало організацій мають закордонних партнерів, що зумовлює потребу в перекладі тексту документа на інші мови. Такий переклад можна доручити не лише професійному перекладачу, а й комп'ютерній програмі (рис. 1).



Рис. 1. Піктограми деяких програм автоматизованого перекладу тексту



Автоматизованим перекладом називається процес перекладання тексту документа з однієї мови на іншу, який здійснюється спеціальними програмами.

Автоматизований переклад використовують, коли потрібно скласти уявлення про загальний зміст іншомовного тексту (інструкції, листи з електронної пошти, веб-сторінки з Інтернету), перекласти великий за обсягом фрагмент тексту з однієї тематики. Результати такого перекладу потребують подальшої перевірки та редагування людиною.

Автоматизований переклад класифікують за способом перекладання на машинний, статистичний, гібридний.

Машинний переклад має за основу словники та аналіз граматичних правил конкретних мов. Лінгвістичний опис двох мов дає можливість перекласти текст з однієї мови на іншу.

Статистичний машинний переклад побудований на наявності речень однією мовою й аналогічних речень іншою (так званих паралельних текстів). Якість перекладу залежить від об'єму бази паралельних текстів.

Гібридний машинний переклад поєднує можливості машинного (основа — слова і словники) та лінгвістичного перекладу (фрази, речення).

Технологія «Пам'ять перекладів» базується на порівнянні тексту, який потрібно перекласти, з готовими перекладами, які зберігаються в попередньо створеній базі перекладів. У загальному масиві текстів система знаходить сегменти, які вже одного разу перекладено, і бере переклад із бази перекладів.

У Microsoft Office для перекладання тексту передбачені команди стрічки панелі Рецензування, де, крім команд для перекладання, є команди для перевірки правопису на визначеній мові. Налаштувати панель Рецензування можна за допомогою команд Файл → Параметри → Налаштувати стрічку, а потім у зоні Основні вкладки вибрати Рецензування.

Під час перекладання виникає потреба в перекладі як окремого слова, так і всього тексту, фрагмента або абзацу (рис. 2).



Ідею автоматизованого перекладу в сучасній формі розвинув Мартін Кей, піонер комп'ютерної лінгвістики. У 1980 році він висунув тезу, що комп'ютер бере на себе рутинні операції і звільняє людину для операцій, що потребують людського мислення.



Рис. 2. Зміст листівки можна зрозуміти за допомогою машинного перекладу

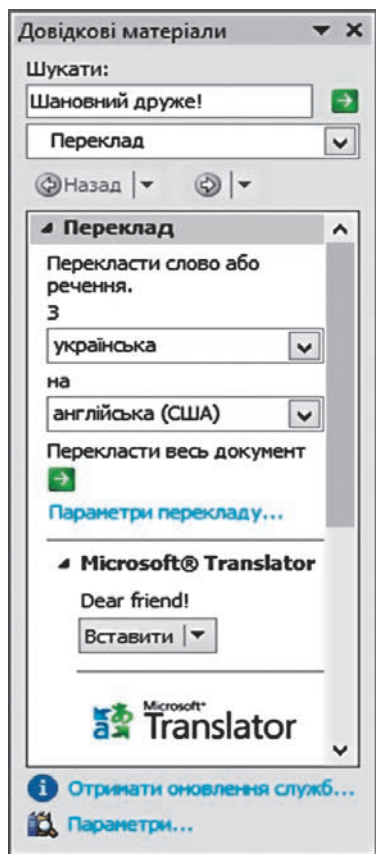


Рис. 4. Область завдань Довідкові матеріали

Щоб здійснити переклад, слід виділити текст, запустити команду Перекласти з групи Мова панелі Рецензування (рис. 3) або із контекстного меню для виділеного тексту відкрити область завдань Довідкові матеріали й вибрати з неї команду Переклад.

Ознайомимося зі списком команд режиму перекладання:

- запуск послуги онлайн-перекладання (для всього документа);
- режим миттєвого перекладання;
- режим запуску області завдань Довідкові матеріали (рис. 4) для налаштування параметрів перекладання та перегляду результату.

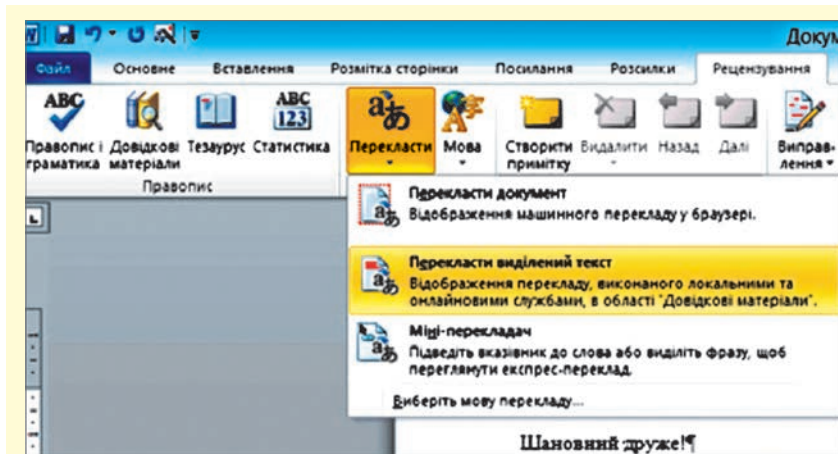


Рис. 3. Стрічка Рецензування: запуск команди Перекласти

В області завдань Довідкові матеріали можна:

- вибрати мову, з якої перекладають і на яку необхідно перекласти текст (зона Переклад зі списками вибору мов);
- переглянути перекладений текст;
- задати режим вставлення перекладеного тексту: замінити виділений текст або скопіювати і вставити в будь-яке місце документа (запускається командою Вставити);
- перевірити правопис.

Запитання для перевірки знань

- 1 Як ви розумієте словосполучення «автоматизований переклад», «машинний переклад»?
- 2 Навіщо потрібні програми-перекладачі?
- 3 Які технології використано в програмах-перекладачах?
- 4 Опишіть алгоритм перекладання тексту.
- 5 Як викликають вікно Довідкові матеріали?
- 6 Наведіть приклади тексту, переклад якого можна автоматизувати, та приклад для перекладання людиною (поміркуйте над

прикладом — українською скоромовкою: «За піщаною косою сірий косий впав під гострою косою косої баби з косою»).

- 📌 Перекладіть наведений текст на одну з іноземних мов. Покажіть переклад вчителю іноземної мови.

Історія українського народу дуже давня й розповідає про основну рису українців — любов до свободи та волі. Вона знайшла своє відображення не тільки в історичних документах, а й у піснях та художніх творах.

Практична робота № 3

Тема. Використання програм автоматизованого перекладу текстів.

Завдання: перекласти текст за допомогою програми, порівняти машинний переклад із власним.

Обладнання: комп'ютер зі встановленою програмою для автоматизованого перекладу тексту, текстовий файл із віршами українських поетів та їх перекладом на одну з іноземних мов.

Примітка: бажано підготувати переклад тексту з домашнього завдання за § 2.4 (с. 20).

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Запустіть програму Microsoft Word.
- 2 Введіть фразу «Шановний друже» і виділіть її.
- 3 На стрічці Рецензування клацніть кнопку Переклад.
- 4 У вікні, що відкриється, виберіть команду Перекласти виділений текст. В області робочого поля з'явиться область завдань Довідкові матеріали.
- 5 У зоні Переклад області завдань Довідкові матеріали виберіть мову, на яку ви перекладатимете звернення до друга.
- 6 Із списку Вставити виберіть команду Вставити — і ви побачите у своєму документі перекладену фразу.
- 7 Наберіть у робочому полі такий текст:
Історія українського народу дуже давня й розповідає про основну рису українців — любов до свободи та волі. Вона знайшла своє відображення не тільки в історичних документах, а й у піснях та художніх творах.
- 8 Перекладіть набраний текст на іноземну мову, яку ви вивчаєте, і порівняйте з вашим перекладом (див. домашнє завдання до § 2.4). Чи збігається перекладений вами текст і текст, отриманий після машинного перекладу?
- 9 Перекладений текст послідовно перекладіть на будь-які іноземні мови, а останній отриманий варіант — на українську. Наприклад, послідовність мов перекладу може мати такий вигляд:
українська → англійська → польська → німецька → українська.
- 10 Чи отримали ви в кінці «ланцюжка» перекладів початковий варіант тексту? Чи задовольняє вас вихідний варіант? Як ви вважаєте, чому отримано саме такий результат?
- 10 Завантажте файл із віршем Т. Г. Шевченка. Якщо такого файла на вашому диску немає, відкрийте новий документ та введіть текст:
*Рече та стогне Дніпр широкий,
Сердитий вітер завива,
Додолю верби гне високі,
Горами хвилю підійма.*
- 11 Перекладіть текст на англійську мову й порівняйте з перекладом Івана Федорівича Вив'юрського, відомого під іменем Джон Вір.
*The mighty Dnieper roars and bellows,
The wind in anger howls and raves,
Down to the ground it bends the willows,
And mountain — high lifts up the waves.*
- 12 Відкрийте новий документ і наберіть текст, який містить відомості про вас:
*Мене звали (ім'я та прізвище).
Я народився (народилась) у (місце й дата народження).
Я навчаюсь у (назва навчального закладу).*
- 13 Перекладіть текст на англійську мову, а потім навпаки — на українську. Збережіть документ.
- **Зробіть висновки:** який текст перекладається автоматично без спотворень; переклад якого тексту потребує редагування людиною; переклад якого тексту необхідно повністю доручити людині.

Розділ 3.

КОМП'ЮТЕРНІ ПУБЛІКАЦІЇ

3.1. Поняття комп'ютерної публікації. Програмні засоби для створення публікацій



Пригадайте, які програмні засоби є в пакеті Microsoft Office. У яких програмах опрацьовують текстові дані? Які програмні засоби для створення та опрацювання графічних зображень вам відомі?



Публікація (від лат. *publico* — оголошувати публічно) — це оприлюднення деяких відомостей. Цим терміном називають також єдиний за формою та вмістом документ, який опубліковано.

Як відомо, текстовий документ може містити не тільки текст, а й графічні об'єкти — ілюстрації до тексту. Це сприяє підвищенню наочності документа, а також акцентує увагу на певних об'єктах.

У кожного з вас є досвід створення листівок, оголошень, запрошень тощо. Усі тримали в руках газети, журнали, книжки. Для їх створення можна використовувати знайомі вам програмні засоби — текстовий процесор, графічні редактори. Але існують програмні засоби, які спеціально розроблені для створення публікацій.

Донедавна публікаціями вважали документи, виконані поліграфічним способом (рис. 1). Тепер статус публікацій мають також електронні документи, розташовані, зокрема, на веб-серверах, які можна переглядати за допомогою браузера.

Залежно від способу збереження публікації розподіляють на друковані та електронні (рис. 2).



Рис. 1. Приклади друкованих публікацій



Електронне видання (англ. *electronic publication*) — електронний документ (кінцевий продукт), що пройшов редакторсько-видавничу обробку, призначений для розповсюдження.

Публікації

Друковані

Публікації, відтворені на папері

Електронні

Публікації, збережені на електронних носіях, розміщені в Інтернеті

Рис. 2. Види публікацій

Більшість публікацій створюють, редагують і форматують, використовуючи певні комп'ютерні програми. У такому випадку вживають терміни «комп'ютерна публікація», «електронна публікація». Наприклад, електронні публікації створюють у середовищах програмних засобів, призначених для створення сайтів, блогів тощо.

Для підготовки друкованих публікацій використовують **видавничі системи** — комплекс апаратних і програмних складових. Часто так називають тільки програми з підготовки текстових документів для поліграфії.

Розглянемо складові видавничої системи (рис. 3).

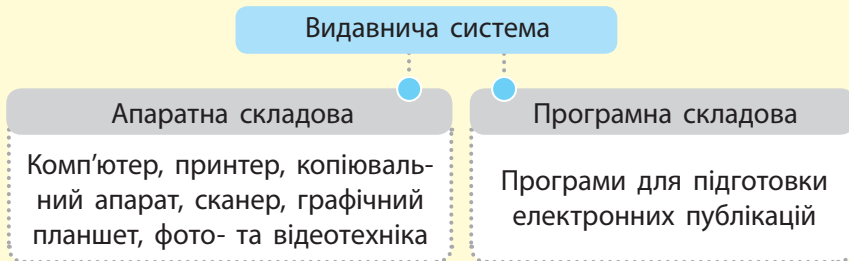


Рис. 3. Складові видавничої системи

У публікаціях велика увага приділяється взаємному розміщенню текстового матеріалу та ілюстрацій до нього.



Макетом публікації є зразок розміщення текстових та графічних об'єктів публікації.

Версткою називають процес компонування текстового та ілюстративного матеріалу відповідно до розробленого макета публікації з дотриманням принципів дизайну та технічних вимог. Підготовлені до друку публікації (оригінал-макети) направляють до поліграфічних підприємств.

Програми, що входять до складу видавничих систем, є професійними і настільними. Вони забезпечують:

- створення оригінал-макетів публікацій;
- опрацювання текстових і графічних об'єктів;
- імпортування об'єктів з інших програм;
- попередній перегляд публікації на екрані монітора з урахуванням характеристик пристрою друку;
- можливість друку публікацій кожним кольором окремо;
- збереження публікації у файлах різних форматів тощо.

Професійні програми використовують для підготовки публікації з великим тиражем (від тисячі до кількох сотень тисяч).



Видавничу систему називають **настільною** (від англ. *desktop publishing* — настільні публікації), оскільки її складові можна розмістити на одному столі (рис. 4).

Для створення публікацій із невеликою тиражною кількістю використовують настільні видавничі системи: офісні додатки (Microsoft Publisher) та інші програми, наприклад Page Plus, Avery DesignPro, Paraben's Label Builder тощо.



Запитання для перевірки знань

- 1 Що означає поняття «публікація»?
- 2 Які програми призначені для створення електронних публікацій?
- 3 Що таке макет публікації?
- 4 Що таке верстка публікації?
- 5 Назвіть особливості видавничих систем.
- 6 З'ясуйте, що таке оригінал-макет? Чим він відрізняється від макета?



Першою настільною видавничою системою є PageMaker компанії Aldus (1985 рік). Формат графічних файлів tiff, розроблений разом із Microsoft, є стандартом у видавничій справі. Надалі компанія стала частиною корпорації Adobe, а програмний засіб отримав назву Adobe PageMaker.

Прикладами професійних програмних засобів верстки є QuarkXPress, Adobe PageMaker, Adobe InDesign тощо.



Рис. 4. Настільна видавнича система

3.2. Види та структура публікацій. Основні складові публікацій



Чи можна використати текстовий процесор для створення публікацій? Чим відрізняються режими створення шаблону документа від режиму створення звичайного документа у Microsoft Word? У файлах із яким розширенням зберігаються шаблони текстових документів у Microsoft Word?

Види публікацій

- Бланк
- Брошура
- Буклет
- Газета
- Інформаційний бюлетень
- Візитівка
- Листівка
- Оголошення
- Реклама
- Реферат

Рис. 1. Види публікацій

- Буклет (від фр. *bouquette* — завиток)
- Брошура (від фр. *broc hure* від *brother* — лишити)
- Інформаційний бюлетень (від лат. *bullā* — документ із печаткою, італ. *bulletino* — записка, листок)
- Реклама (від лат. *reclamare* — викрикувати)
- Реферат (від нім. *referat*, лат. *refere* — доповідати)
- Газета (від італ. *gazzetta* — назва дрібної монети)

Загальний вигляд публікації залежить від її призначення. Розглянемо деякі види публікацій (рис. 1).

Буклет — переважно багатокольорова публікація, подана на одному аркуші в кілька колонок (зі згином по межі між ними). Найчастіше використовується під час проведення публічних заходів. На різних сторінках (колонках) буклету зазначають назву заходу, час проведення, призначення, програму, відомості про заклад, який проводить захід; як доїхати до місця проведення заходу, контактні дані організаторів тощо.

Газета — щоденне, щотижневе або двотижневе періодичне видання у вигляді складених аркушів, що має постійну назву. Може містити оперативну інформацію, фотографії, матеріали з актуальних громадсько-політичних, наукових, виробничих та інших питань.

Брошура — твір у вигляді книжки, обсягом не більше 48 сторінок. Брошури використовують у сфері освіти, рекламному бізнесі тощо.

Інформаційний бюлетень — видання, що має на меті інформування групи людей щодо окремих питань. Зазвичай містить добірку актуальної інформації.

Оголошення — публікація, яка інформує про захід, подію та за необхідності містить стислі відомості про назву, час та (або) місце проведення.

Реклама — публікація для довільного кола осіб, спрямована на привертання уваги до об'єкта рекламування, формування або підтримку інтересу до нього.

Бланк — документ, який містить поля для заповнення, а також назви та призначення таких полів.

Реферат — твір, у якому проаналізовано, узагальнено й подано відомості з різних джерел.

Візитівкою є документ, у якому надано відомості про особу або організацію, заклад.

Кожний вид публікації має своє призначення, чим зумовлений його зовнішній вигляд, розмір, структура тощо (див. рис. 2). Наприклад, інформаційний бюлетень про проведення олімпіади з певного предмета містить опис того, на що має звернути увагу учасник, та посилання на літературу. Якщо б його зміст було обмежено місцем і часом проведення олімпіади, то це було б оголошення, а якщо б містило заклик до участі в олімпіаді та пропозицію з додатковими бонусами

учасникам і переможцям,— то рекламою. Візитівка має адресу, телефон, логотип організації або фотографію особи тощо.

У настільних видавничих системах створення різних видів публікацій максимально автоматизовано.



Шаблон (від фр. *echantillon* — зразок) — приклад, який наслідують; готовий зразок, за яким створюють документ певного виду.

На основі шаблонів, які визначають загальну структуру певного виду публікації, створюються макети, що мають певний дизайн. Зазвичай для кожного шаблону є кілька варіантів макетів.

Для створення кожного виду публікацій існують бібліотеки професійних макетів. Користувачу потрібно тільки наповнити необхідним вмістом запропоновані макетом блоки.

Структура публікації складається з *текстових* і *графічних* блоків, що заповнюються користувачем, а також *стандартних* блоків (рис. 3).

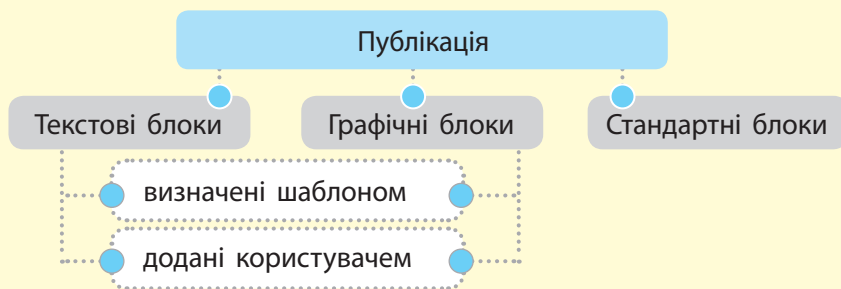


Рис. 3. Структура публікації

Текстові та **графічні блоки** в публікації обмежені спеціальною рамкою, призначеною для зручного введення тексту або додавання графічного зображення.

Стандартні блоки — це фрагменти, які можна використати в різних видах публікацій, наприклад заголовки, календарі тощо. Вони зберігаються в окремих колекціях.

Шаблони різного виду публікацій мають свої особливості. Більшість із них містять спеціальні місця для логотипу організації (графічний блок) і введення назви, за потреби — дати випуску (текстові блоки).

Програмними засобами передбачено можливість вставляння додаткових блоків для введення тексту та рисунків за бажанням користувача.



Запитання для перевірки знань

- 1 Які види публікацій розрізняють?
- 2 Що означає поняття «шаблон публікації»?
- 3 Який зв'язок між поняттями «вид публікації», «шаблон», «макет публікації»?
- 4 Які структурні блоки мають публікації?
- 5 Поясніть використання стандартних блоків у відомих вам видах публікацій.
- 6 З'ясуйте, які відомості мають бути у візитівці приватної особи.



Рис. 2. Публікації різних видів

3.3. Створення комп'ютерної публікації



Пригадайте, як запустити офісну програму на виконання. Як створити текстовий документ?

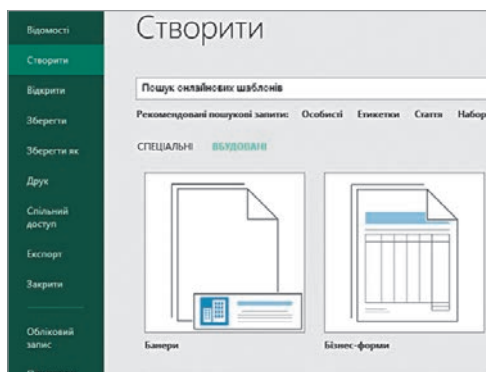


Рис. 1. Створення нової публікації в Microsoft Publisher

Для того щоб створити тестовий документ, електронну таблицю чи комп'ютерну презентацію, можна скористатися певною офісною програмою, наприклад Microsoft Word, Microsoft Excel, Microsoft PowerPoint з пакету Microsoft Office. Після запуску будь-якої з цих програм у робочій області вікна програми з'являється новий документ, створений за певним шаблоном.

До складу Microsoft Office також входить програма Microsoft Publisher — настільна видавнича система, яка має інтуїтивно зрозумілий інтерфейс. І хоч ця програма має обмежену кількість функцій порівняно з професійними видавничими системами, наявність великої кількості шаблонів і макетів публікацій значно спрощує створення публікацій різних видів.

Розглянемо особливості настільних видавничих систем, їхні можливості щодо створення та редагування публікацій на прикладі Microsoft Publisher.

Процес створення нової комп'ютерної публікації відрізняється від створення нового документа в інших додатках пакета Microsoft Office. Наявність великої кількості шаблонів і макетів зумовила деякі особливості початку роботи з новою публікацією (рис. 1).

Розглянемо алгоритм створення нової комп'ютерної публікації.

Крок 1

Запустіть програму Microsoft Publisher за допомогою ярлика програми або команд Пуск → Microsoft Office → Microsoft Publisher.

Виберіть шаблон публікації (рис. 2) відповідно до виду публікації.

Крок 2

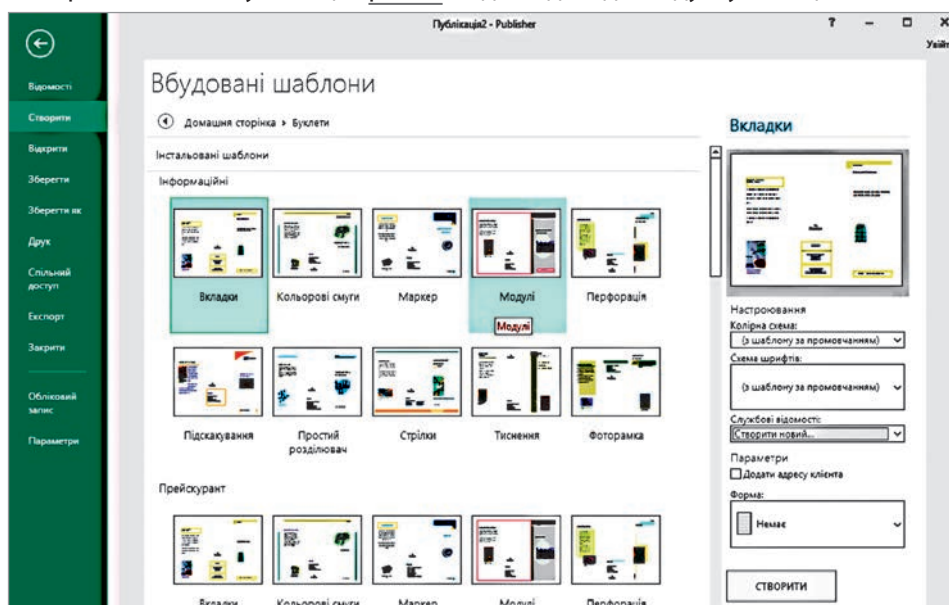


Рис. 2. Вибір шаблону публікації

	Шаблон вибирають із бібліотеки Microsoft Publisher (вбудовані шаблони) або з Інтернету. Вбудовані типи шаблонів згруповано в групи Порожня публікація (книжкова, альбомна), Мої шаблони , Популярні публікації (булетети, бюлетені, календарі, листівки, поштові картки тощо), Інші шаблони .
Крок 3	Відповідно до вибраного шаблону виберіть макет публікації (див. рис. 2).
Крок 4	Після вибору макета в робочій області вікна програми з'явиться новий документ (публікація) із розміткою кожної сторінки згідно з вибраним макетом.
Крок 5	Наповніть запропоновані блоки текстом та ілюстраціями. За потреби вставте додаткові блоки та наповніть їх.
Крок 6	Збережіть публікацію (тип документа Microsoft Publisher .pub).

Таким чином, усі сторінки документа, створеного у текстовому чи табличному процесорі, мають однаковий вигляд. Сторінки різних видів публікацій за виглядом відрізняються одна від одної (рис. 3).

У робочій області також з'явиться Область завдань з командами для роботи з публікацією. Наприклад, для вибору поточної сторінки, що дозволяє опрацьовувати сторінки у довільному порядку.

У групі Параметри сторінки містяться команди для встановлення орієнтації та полів сторінки, розміру паперу, в групі Фон сторінки — команди встановлення фону.

У процесі роботи з публікацією є можливість змінити один вид публікації на інший, панель Макет сторінки містить команду Замінити шаблон. Команди групи Схеми цієї панелі призначено для вибору або створення власної кольорової схеми макета публікації та шрифтів для різних текстових блоків макета (рис. 4).

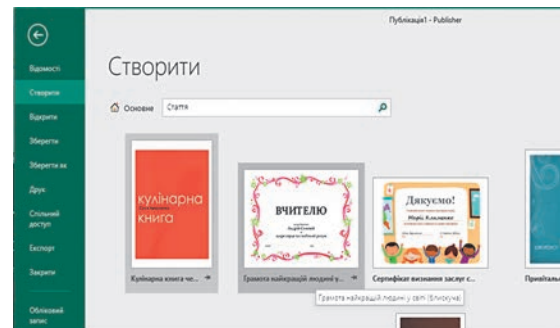


Рис. 3. Приклади публікацій, створених у Microsoft Publisher

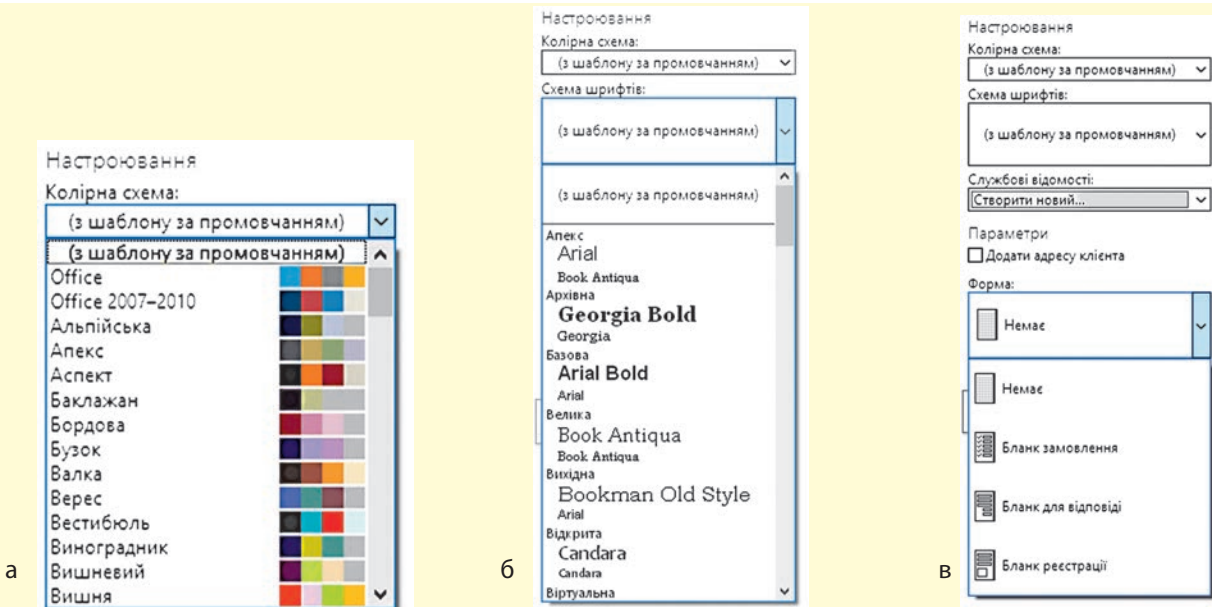


Рис. 4. Налаштування параметрів макета

Вікно програми Microsoft Publisher містить стандартні елементи вікон додатків Microsoft Office: рядок заголовка, стрічкову панель, робочу область (рис 5). А крім них містить ліворуч вузьку вертикальну панель для роботи з об'єктами і область задач з командами для розв'язування задачі створення швидкої публікації відповідно до зразка макета чи шаблону.

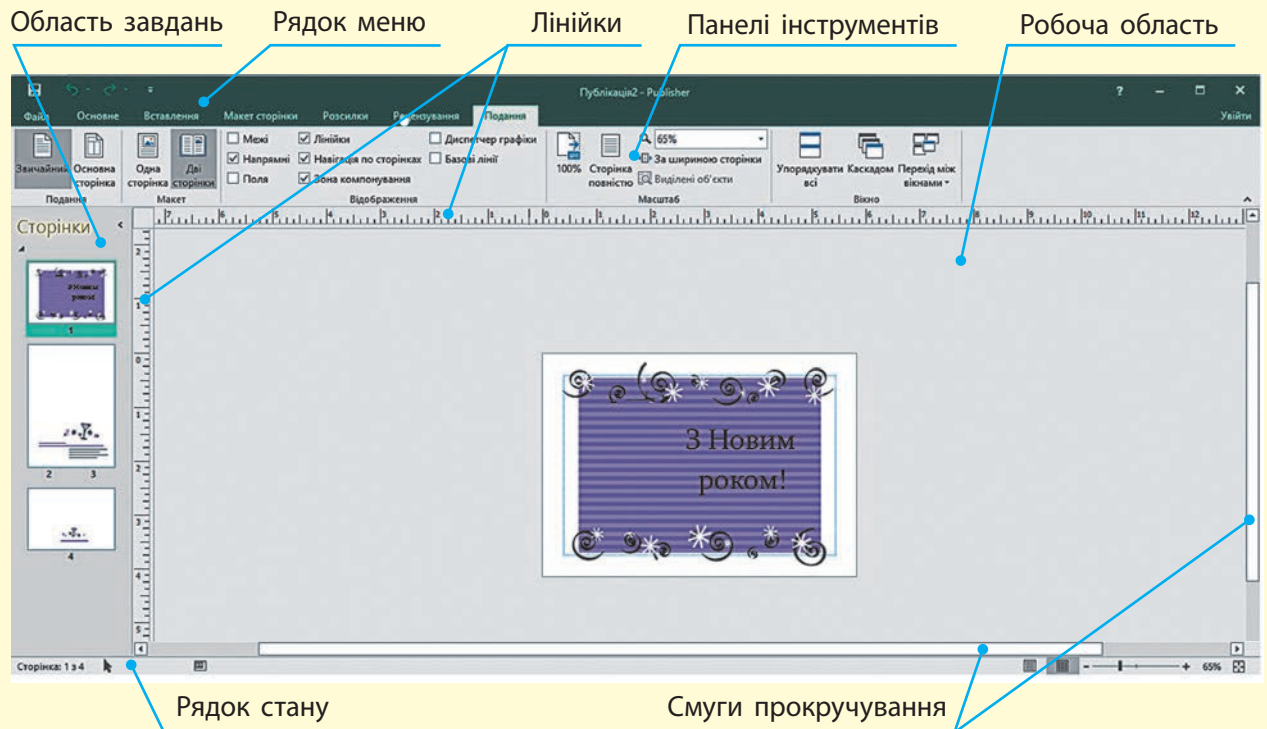


Рис. 5. Інтерфейс програми Microsoft Publisher

У вікні Microsoft Publisher за замовчуванням відображаються дві панелі інструментів: Стандартна та Форматування. Керування виведенням чи приховуванням цих панелей здійснюється у звичний спосіб.

Запитання для перевірки знань

- 1 Як запустити програму Microsoft Publisher на виконання?
- 2 Як створити нову публікацію?
- 3 Де можна знайти шаблони публікацій?
- 4 Назвіть елементи вікна програми Microsoft Publisher.
- 5 Як змінити макет вже створеної публікації?
- 6 Яке розширення має файл публікації?

Завдання для самостійного виконання

- 1 Створіть листівку для привітання друга з днем народження.
- 2 Запустіть програму Microsoft Publisher.
- 3 У вікні, що відкрилося, виберіть макет листівки (наприклад, день народження).
- 4 У текстовий блок уведіть привітання друга з днем народження.
- 5 Збережіть публікацію у форматі Publisher, вибравши необхідну папку.
- 6 У вікні, що відкрилося, виберіть шаблон публікації (наприклад, вітальні листівки).

Практична робота № 4

Тема. Створення буклета з використанням шаблону.

Завдання: створить інформаційний буклет для учасників і гостей конкурсу.

Обладнання: комп'ютер зі встановленою програмою **Microsoft Publisher**, файл із логотипом (емблемою), дані про навчальний заклад, відомості про конкурс.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Відкрийте програму Microsoft Publisher звичним для вас способом.
 - 2 Виберіть шаблон Буклети з групи шаблонів Популярні.
 - 3 Виберіть один із макетів інформаційного буклета і виконайте команду Створити (при подвійному клацанні на ескіз макета команда Створити запускається автоматично).
У вікні, що відкрилося, в робочій області розміщено макет першої сторінки буклета. Буклет за замовчуванням має альбомну орієнтацію з трьома колонками розміщення даних. Після роздрукування аркуш згортають по межах між колонками — і буклет набуває вигляду ширми.
 - 4 Заповніть текстові та графічні блоки даними:
 - 1) на титулі буклета (третя колонка першої сторінки) у блоці «Назва організації» введіть «Міністерство освіти і науки України», на наступному рядку — назву вашого закладу;
 - 2) додайте нове текстове поле (панель Вставлення, команда Додати текстове поле), розмістіть його по центру титульної сторінки; вставте назву заходу — «Конкурс-захист науково-дослідницьких робіт учнів — членів МАН України»;
 - 3) додайте нове текстове поле, розмістіть його внизу титульної сторінки; введіть дату проведення конкурсу.
 - 5 На зворотній сторінці буклета (друга колонка першої сторінки) у блок із назвою «Організація» введіть реквізити вашого закладу: адресу, контактні телефони; в графічний блок додайте емблему закладу.
 - 6 У першій колонці першої сторінки, яка містить текстове поле з назвою Заголовок задньої панелі, розмістіть інформацію про заклад.
 - 7 У першій колонці другої сторінки додайте план проведення конкурсу:
 - 1) змініть назву в текстовому блоці на «План конкурсу»;
 - 2) вилучіть вміст текстового блоку (виділіть і скористайтесь клавішею видалення);
 - 3) перейдіть на панель Вставлення, в групі Таблиці виберіть зручний для вас спосіб вставлення таблиці з трьох стовпців і декількох рядків;
 - 4) за допомогою миші розмістіть таблицю під введеною назвою;
 - 5) введіть заголовки таблиці та заповніть її за зразком:

№ з/п	Час, хв	Доповідач, ПІБ
 - 8 В інші колонки другої сторінки введіть інформацію про роботу секцій МАН (секції виберіть на власний розсуд).
 - 9 Збережіть публікацію.
 - 10 Покажіть публікацію вчителіві.
 - 11 Закрийте програму.
- **Зробіть висновки:** до яких подій у вашому шкільному житті краще підготувати буклети; наведіть приклади; у яких випадках можна застосувати інші види публікацій.

3.4. Робота з текстовими об'єктами. Зв'язки між об'єктами публікації



Пригадайте, що таке текст, з яких об'єктів він складається.

Основним об'єктом публікації є **текстовий блок**, який складається з текстової рамки та області, в яку вводиться текст. На одній сторінці може міститися декілька текстових блоків, визначених макетом публікації.

Текстові блоки можна додавати, вилучати, переміщувати, змінювати розміри для надання публікації потрібного вигляду.

Для **додавання текстового блоку** можна скористатися командою Вставлення → Накреслити текстове поле (рис. 1).

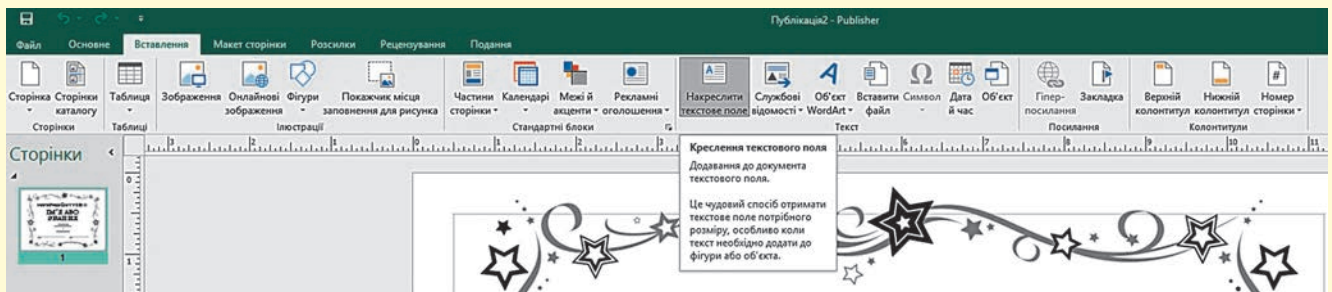


Рис. 1. Стрічкова панель Вставлення: група Текст



Властивості текстового блоку: розмір, кут розташування, колір і тип контура (або без контура), колір і тип заливки (або без заливки).

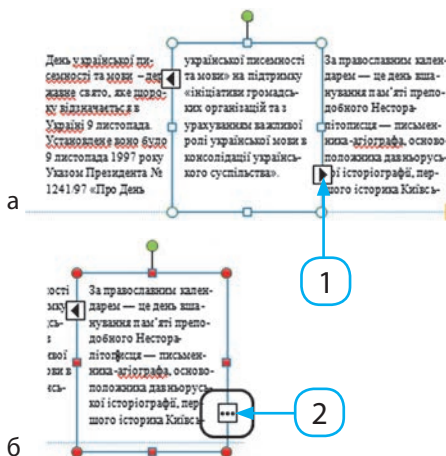


Рис. 2. Перетікання тексту між текстовими блоками (а); переповнення текстового блоку (б)

Переміщувати текстові блоки та змінювати їхній розмір зручно за наявності **направляючих** — допоміжних ліній на сторінці, які встановлюються за допомогою команд групи Макет.

Для **вилучення текстового блоку** його потрібно виділити, клацнувши рамку, а потім натиснути відповідну клавішу клавіатури (Delete, Backspace).

Для **переміщення об'єктів** потрібно навести вказівник миші на межу об'єкта (але не на маркери), де вказівник змінює вигляд на хрестоподібний зі стрілками. Потім об'єкт перетягують у потрібне місце з натиснутою лівою кнопкою миші.

Для **змінення розміру об'єкта** слід навести вказівник миші на один із маркерів на межі об'єкта. Коли вказівник змінює вигляд на двобічну стрілку, можна перетягувати в потрібному напрямку відповідний маркер рамки з натиснутою лівою клавішею миші.

Якщо текстовий блок переповнений або текст перетікає на наступну сторінку, блоки з'єднують у ланцюжок. У випадку коли текстовий фрагмент перевищує розмір текстового блоку, текст автоматично розміщується у наступному зі зв'язних текстових блоків.



Текстові блоки, в яких може розміщуватися один текстовий фрагмент, називають **зв'язаними**.

На зв'язність текстових блоків вказують **спеціальні маркери** (1) (рис. 2, а). Якщо текст не вміщується в текстовий блок, у нижній правій частині цього блоку з'являється

маркер переповнення (2) (рис. 2, б) — квадратик із трьома крапками. У випадку, коли текст не вміщується в блок, потрібно збільшити текстову рамку, перетягуючи рамку блоку, або зв'язати поточний блок з новим текстовим блоком для перетікання в нього тексту. У нового текстового блока на лівому краї з'являється стрілка, яка вказує, що текст у блоці є продовженням тексту з іншого блоку.

Для створення зв'язку між блоками потрібно навести вказівник миші на маркер переповнення; клацнути лівою кнопкою миші — маркер набуває вигляду глечика; перемістити вказівник миші в інший текстовий блок і клацнути його.

У ході редагування тексту (видалення, додавання тексту) слова перетікатимуть з одного текстового блоку до іншого. Якщо для текстового фрагмента двох текстових блоків не вистачає, можна налагодити зв'язок ще з одним текстовим блоком, і текст перетікатиме вже по трьох блоках.

Згадаємо, що процес змінення зовнішнього вигляду документа називається форматуванням (рис. 3). **Форматування текстових блоків** складається з форматування рамки та тексту в блоці. Для форматування призначена вкладка Формат (Знаряддя для тестових полів), яка з'являється, щойно текстове поле стає активним.

Розглянемо групи команд для форматування тексту:

Група Текст	Містить команди По розміру тексту (часто використовують команду Збільшення для зміни розміру текстового поля під розмір текстового фрагмента), Зміна напрямку тексту та Розстановка переносів для використання або відмови від переносів
Група Вирівнювання	Містить команди, що дозволяють відформувати розміщення тексту в блоці по вертикалі та горизонталі
Група Зв'язування	Містить команди для встановлення (або розриву) зв'язку тексту між текстовими блоками та перетікання між ними

Розглянемо групи команд для форматування символів:

Група Шрифт	Містить команди для змінення вигляду символів і міжсимвольного інтервалу.
Група Ефекти	Містить команди, що дозволяють застосувати стиль WordArt і текстові ефекти (тінь, підсвічування, рельєфність) до символів.
Група Оформлення	Містить команди для художнього виокремлення виділених символів у тексті.

? Запитання для перевірки знань

- 1 Що являє собою текстовий блок?
- 2 Як переміщувати текстовий блок, змінювати його розміри?
- 3 Для чого під час макетування сторінки використовують направляючі? Поясніть на прикладах.
- 4 Які групи команд використовуються для форматування всього тексту блоку?
- 5 Як додати новий текстовий блок?
- 6 Які текстові блоки називають зв'язними? Поясніть, як зв'язати текстові блоки.

Форматування об'єктів

Форматування текстових об'єктів

символ (установлення шрифту, розміру, кольору, накреслення)

абзац (установлення відступів, міжрядкового інтервалу, вирівнювання)

сторінка (встановлення полів, орієнтації)

Форматування графічних об'єктів

Рис. 3. Форматування об'єктів

3.5. Робота з графічними об'єктами



Які види графічних об'єктів можна додавати в текстовий документ у програмах офісного пакету Microsoft Office?

Як ви знаєте, крім текстових блоків, структура публікації містить стандартні блоки. Для додавання стандартних блоків використовують групу команд Стандартні блоки панелі Вставлення (рис. 1). Команди вибирають із бібліотеки блоків, а ті, у свою чергу,— у відкритому вікні групи або на панелі.

Для додавання графічних об'єктів до публікації призначені команди групи Ілюстрації стрічкової панелі Вставлення.

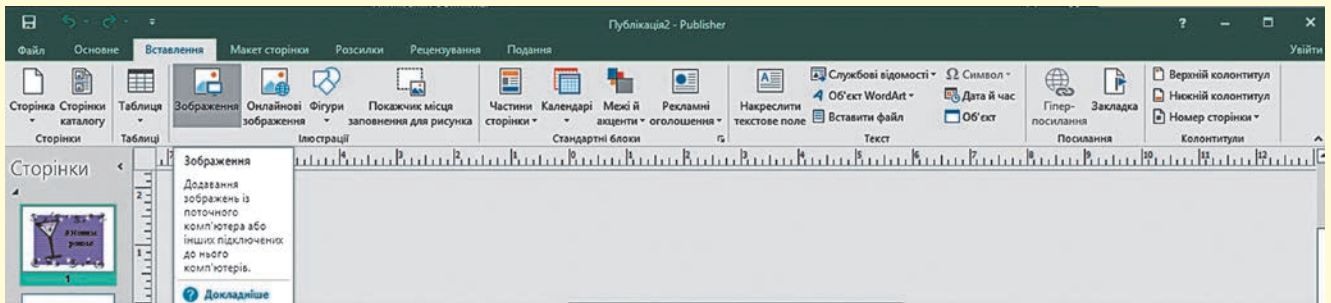


Рис. 1. Стрічкова панель Вставлення: групи Таблиці, Ілюстрації, Стандартні блоки

Розглянемо можливості додавання графічних зображень.

Для додавання рисунка з файлу використовують команду Зображення. Після цього відкривається діалогове вікно Вставлення зображення, в якому в потрібній папці вибирають потрібний файл із рисунком. Вставлення рисунка відбувається без зміни його початкового розміру.

Для додавання рисунка з колекції Microsoft Office використовують команду Онлайнові зображення, для додавання фігур векторного редактора, вбудованого в Microsoft Office,— команду Фігури.

Для додавання рамки для зображення у групі Ілюстрації є команда Показчик місця заповнення для рисунка. З її допомогою можна позначити місце для майбутнього рисунка в публікації.

Додавши рамки для зображення, можна змінити її розміри, а також перемістити в інше місце сторінки (рис. 2).

Для вставлення рисунка в рамку потрібно навести вказівник миші на схематичне зображення рисунка всередині рамки (він підсвітиться) і клацнути — відкриється вікно додавання рисунка, як і під час виконання команди Вставлення → Зображення. Рисунок займатиме місце, визначене рамками. Як тільки рисунок буде додано, автоматично активується команда Обтинання.

Щоб завершити вставлення зображення, слід клацнути кнопкою миші поза рамкою.

Форматування графічного об'єкта здійснюється командами стрічкової панелі Формат (Знаряддя для зображень), яка з'являється, якщо об'єкт виділено.

Стрічкова панель Формат має різний вигляд — залежно від типу графічного об'єкта, який виділено (див. рис. 3).

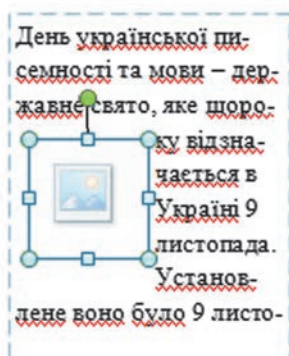


Рис. 2. Фрагмент публікації з рамкою для вставлення рисунка

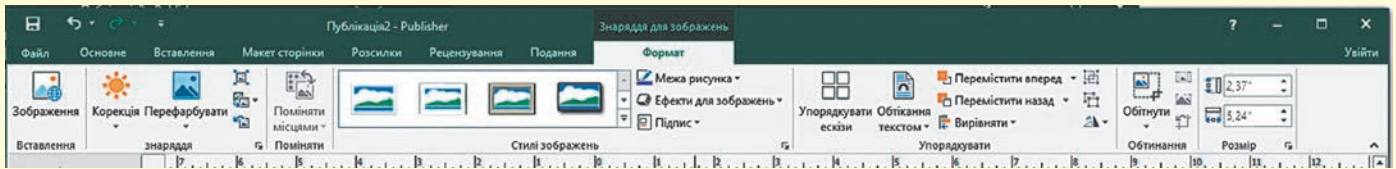


Рис. 3. Стрічкова панель **Формат (Знаряддя для зображення)** для форматування рисунка

Стрічкова панель **Формат** містить такі групи команд:

Група Знаряддя	Містить команди для опрацювання зображення: корегування кольору, установлення ступеня прозорості кольору зі списку команд Перефарбувати та ін.
Група Стилі зображень	Містить команди, що дозволяють вибрати вигляд межі рисунка, його форму, а також вставити підпис до зображення. До рисунка можна застосувати тіньові ефекти завдяки командам однойменної групи.

Команди решти груп стрічкової панелі **Формат** ідентичні знайомим вам командам для форматування зображень у текстовому процесорі.

Для форматування фігур панель **Формат** (рис. 4) має такі самі команди, як і відповідна панель програми MS Word.

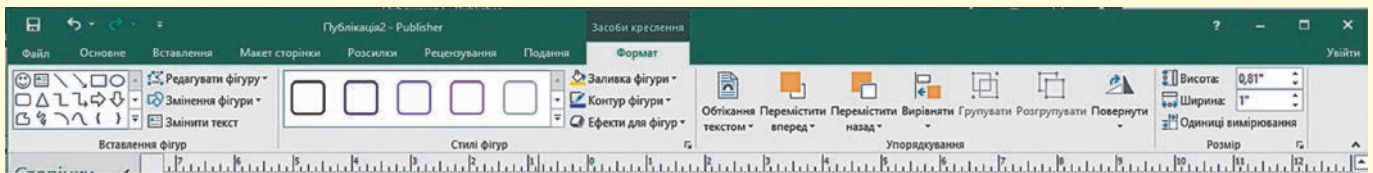


Рис. 4. Стрічкова панель **Формат (Засоби креслення)** для опрацювання векторних фігур

Аналогічно зображенню в публікацію можна вставити таблицю і відформатувати її. Робота з таблицями здійснюється у звичний для Microsoft Word спосіб.

? **Запитання для перевірки знань**

- 1 Які команди використовують для вставлення графічних об'єктів у публікацію?
- 2 Наведіть приклади стандартних блоків.
- 3 Як форматують у публікаціях малюнки?
- 4 Які особливості має графічне зображення, розміщене в попередню вставлену рамку?
- 5 Поясніть, як вставити та відформатувати таблицю в публікації.

💻 **Завдання для самостійного виконання**

- 1 Створіть бюлетень із рисунком, відформатованим за наведеним планом.
- 1 Запустіть програму Microsoft Publisher. Виконайте команду **Файл** → **Створити**.
- 2 Виберіть тип публікації **Бюлетені**.
- 3 Виберіть макет бюлетеня на свій розсуд.
- 4 Виконайте **Вставлення** → **Зображення**, додайте рисунок та залиште його активним.
- 5 Активуйте панель **Формат**, виберіть із групи **Перефарбувати**, у списку, що відкриється — **Установити прозорий колір**.
- 6 Наведіть змінений вказівник миші на будь-який колір на рисунку й клацніть його.

3.6. Етапи роботи з публікацією



Пригадайте, як створити, зберегти, відкрити та надрукувати документ Word.

Створення публікації дещо нагадує роботу над проектом. Звичайно, на створення вітальної листівки буде затрачено небагато часу. Але буклет, брошура, інформаційний бюлетень і навіть реклама потребують попередньої роботи з планування та добору матеріалу.

Розглянемо основні етапи роботи з публікацією.

I. Аналіз призначення та тематики публікації, визначення виду публікації.

З'ясуйте, який вид публікації доцільно створити, що в свою чергу визначить вибір шаблону публікації.

II. Планування та добір матеріалів.

Відповідно до заданої теми спланувати публікацію та підібрати матеріал (текст, ілюстрації), продумати, чи будуть у публікації стандартні блоки і які.

У процесі роботи з публікацією можна вводити текст, додавати рисунки, створені у видавничій системі. Але основне наповнення краще копіювати із заздалегідь підготовлених файлів.

III. Створення комп'ютерної публікації.



Крок 1

Запустіть програму створення публікацій одним зі знайомих вам способів.

Крок 2

Виберіть шаблон і макет публікації.

Крок 3

Наповніть згідно з макетом текстові та графічні блоки підготовленими заздалегідь текстами та ілюстраціями. За необхідності вставте додаткові блоки.

Крок 4

Відредагуйте та відформатуйте публікацію.

Крок 5

Збережіть публікацію після закінчення роботи. Публікацію можна зберегти у файлах Microsoft Publisher (*.pub), а також у форматі PDF, як веб-сторінку, у форматі текстового документа.

IV. Розповсюдження комп'ютерної публікації.

Роздрукувати презентацію або розмістити в Інтернеті.



Запитання для перевірки знань

- 1 Назвіть етапи створення комп'ютерної публікації.
- 2 Які дії потрібно зробити після запуску програми Microsoft Publisher перед введенням даних у публікацію?
- 3 Чи можна скористатись можливостями Інтернету для визначення шаблону публікації?
- 4 Як вибрати поточну сторінку публікації для наповнення та редагування.
- 5 Як змінити колірну схему макета?
- 6 Для чого використовують направляючі під час макетування сторінки публікації?

Практична робота № 5

Тема. Створення бюлетеня з використанням шаблону та стилів оформлення.

Завдання: створити бюлетень до Дня української писемності та мови (відзначається 8 і 9 листопада).

Обладнання: комп'ютер зі встановленою програмою Microsoft Publisher; матеріали про історію української писемності, видатних українських поетів, письменників, їхні фотографії.

Примітка: зазначені матеріали можна знайти в мережі Інтернет у ході виконання роботи.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Відкрийте програму Microsoft Publisher.
 - 2 Виберіть шаблон Бюлетені з групи шаблонів Популярні.
 - 3 Виберіть один із макетів бюлетеня на власний розсуд.
 - 4 У робочій області вікна, що відкрилося, розміщено макет першої сторінки бюлетеня. (Бюлетень подано чотирма сторінками: *першою, заключною*, між ними на розвороті *дві внутрішні сторінки*.)
 - 5 Змініть колірну схему бюлетеня: відкрийте панель Макет сторінки, в групі Схеми виберіть колірну схему макета на власний розсуд.
 - 6 Виберіть шрифтову схему для оформлення буклета: на панелі Макет сторінки відкрийте список Шрифти і виберіть шрифтову схему.
 - 7 Заповніть графічні та текстові блоки матеріалами з урахуванням назв текстових блоків та можливістю перетікання тексту між текстовими блоками:
 - *перша* сторінка містить назву бюлетеня, дату та номер випуску, основну статтю (про День української писемності та мови), додаткову статтю, а також зміст (на якій сторінці розташовані статті бюлетеня);
 - заповніть *внутрішні* сторінки бюлетеня, якщо маєте матеріали для заповнення, за час, відведений на практичну роботу. Якщо таких матеріалів немає, перейдіть в область навігації по сторінках, виділіть внутрішні сторінки (*другу і третю*) та вилучіть їх командою з контекстного меню;
 - *заклучна* сторінка має містити дані про авторів, анонс наступного випуску бюлетеня (введіть текст на власний розсуд).
 - 8 Збережіть публікацію.
 - 9 Покажіть публікацію вчителю.
 - 10 Закрийте програму.
- **Зробіть висновки:** як створити бюлетень із використанням шаблону та стилів оформлення.

Практична робота № 6

Тема. Проектування та створення комп'ютерної публікації для подання результатів самостійного дослідження.

Завдання: створити інформаційний бюлетень для подання результатів самостійного дослідження.

Обладнання: комп'ютер зі встановленою програмою Microsoft Publisher, текстові та графічні матеріали за темою дослідження.

Примітка. Якщо самостійне дослідження не було здійснено, дослідіть питання про наявність курсів (наприклад, із програмування, основ живопису або інших — на власний вибір) за допомогою мережі Інтернет.

Хід роботи

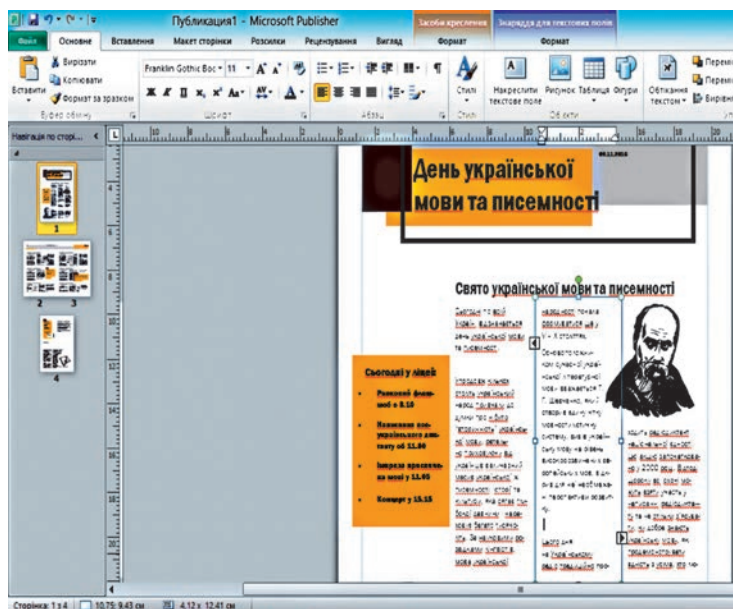
Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Відкрийте програму Microsoft Publisher.
- 2 Виберіть один із макетів бюлетеня на власний розсуд.
- 3 На першій сторінці виконайте такі дії:
 - 1) уведіть назву в область заголовка;
 - 2) перейменуйте область «Важливі деталі» на «Етапи дослідження» і заповніть список назвами етапів;
 - 3) заповніть область *першої* статті назвою головної статті про дослідження та текстом про загальну тематику дослідження (текст скопіюйте з документа, який ви попередньо створили в ході виконання дослідження);
 - 4) заповніть область *другої* статті заголовком, наприклад, «Чому я вибрав (вибрала) _____ (галузь дослідження)», та опишіть, чому саме ця тема вас зацікавила. Якщо місця на першій сторінці не вистачає, створіть перетікання тексту на другу (внутрішню) сторінку;
 - 5) відредагуйте текст області «У цьому виданні» так, щоб назви статей відповідали сторінкам, на яких вони розташовані.
- 4 Створіть рамку для вставлення графічного зображення — виділіть зображення командою Показчик місця заповнення.
- 5 Перемістіть рамку зображення в праву нижню область *першої* статті.
- 6 Уставте в рамку ілюстрацію, яка б відповідала темі головної статті.
- 7 Уставте в публікацію таблицю, що має 5 стовпців, та заповніть:
 - 1) перший рядок таблиці — назвами стовпців: № з/п; Назва курсу; Тривалість; Дати проведення; Адреса;
 - 2) інші рядки таблиці — даними відповідно до запропонованих заголовків стовпців (для дослідження про курси).
- 8 Перейдіть до заповнення внутрішніх сторінок. Вилучіть зайві текстові блоки.
- 9 Заповніть текстом і графічними зображеннями області статей внутрішніх сторінок, висвітливши в одній із них:
 - 1) відомості про вашу участь у дослідженні (якщо ви брали участь у колективному дослідженні);
 - 2) актуальність вашого дослідження;
 - 3) аналіз інформаційних матеріалів з тематики вашого дослідження;
 - 4) ваші очікування від проведеного дослідження.
- 10 Присвятіть *другу* статтю власне процесу дослідження.

Якщо у вас є числові дані дослідження, проаналізуйте їх у табличному процесорі, створіть діаграму, скопіюйте й додайте в публікацію. (Для дослідження про курси такими даними можуть бути вартість курсів або їх тривалість.)

- 11 Розмістіть на *заклучній* сторінці висновки, дані про автора (авторів).
- 12 Збережіть публікацію.
- 13 Покажіть публікацію вчителю.
- 14 Закрийте програму.

Зробіть висновки: як спроектувати та створити комп'ютерну публікацію для подання результатів самостійного дослідження.



Розділ 4. Бази даних

4.1. Поняття бази даних. Реляційна модель даних

Чи доводилося вам раніше працювати з будь-якими базами даних? Чи можна підручник з інформатики назвати базою даних?



Нині важко уявити роботу великого супермаркету, системи резервування й продажу авіаквитків, банку, морського порту, готельного комплексу тощо без використання інформаційних систем, основою яких є бази даних.



База даних — це сховище організованої сукупності даних різного типу, які відображують стан об'єктів певної предметної галузі та зв'язки між ними.

Предметною галуззю називають сферу застосування конкретної БД, наприклад школа, банк, фірма, поліклініка, районний відділ освіти тощо.

Об'єктом предметної галузі є те, про кого або про що зберігаються дані в БД, наприклад учні, книжки, футболісти, класи школи тощо.

Кожен об'єкт характеризується сукупністю властивостей (атрибутів). Так, властивостями об'єкта **УЧЕНЬ** можуть бути *прізвище, ім'я, домашня адреса, дата народження, зріст*.

Далі об'єкти записуватимемо великими літерами, а в дужках зазначатимемо їхні властивості. Наприклад, об'єкт **УЧИТЕЛЬ** можна позначити так: **УЧИТЕЛЬ** (прізвище, ім'я, по батькові, рік народження, предмет), а об'єкт **БОРОШНО** так: **БОРОШНО** (назва, виробник, сорт, вага, ціна).

Кожна властивість об'єкта має сукупність значень — неподільних **елементів даних**. Так, властивість сорт об'єкта **БОРОШНО** може мати такі значення: *вищий, перший, другий*, а властивість предмет об'єкта **УЧИТЕЛЬ** — *математика, історія, хімія* та інші. Повний набір елементів даних про об'єкт називають екземпляром об'єкта. Наприклад, екземпляром об'єкта **УЧЕНЬ** є: (Котенко, Іван, Петрович, 2017, 9 клас).

Сукупність властивостей об'єкта та їхніх значень є **даними конкретної БД**. Дані можуть бути різних типів: *числові, символні* та ін. Властивості та їхні значення мають між собою зв'язки. Так, властивість *прізвище учня* пов'язана з конкретним



Вперше термін *database* (база даних) з'явився на початку 60-х років ХХ ст. і був уведений у вжиток на симпозиумах, організованих фірмою System Development Corporation (США) у 1964 і 1965 роках. Широкого розповсюдження в сучасному розумінні цей термін набув у 1970-ті роки.



В Ісландії створена БД, яка містить інформацію про родинні зв'язки усіх ісландців, починаючи з ХVІІІ ст.

Кожна предметна галузь містить багато об'єктів, між якими існують певні зв'язки. Причому, якщо предметною галуззю є *школа*, то її об'єктами можуть бути *класи, вчителі, учні, предмети*, між якими існують зв'язки. Так, вчителі викладають певні предмети, учні навчаються в конкретних класах.

прізвищем. Звідси випливає, що дані, які зберігаються в базі, мають певну логічну структуру, тобто описуються деякою моделлю подання даних, або просто моделлю даних.

Способи відображення зв'язків між даними, тобто моделі даних, можуть бути різними. Класичними є такі моделі даних: **ієрархічна**, **мережева** та **реляційна** (рис. 1).

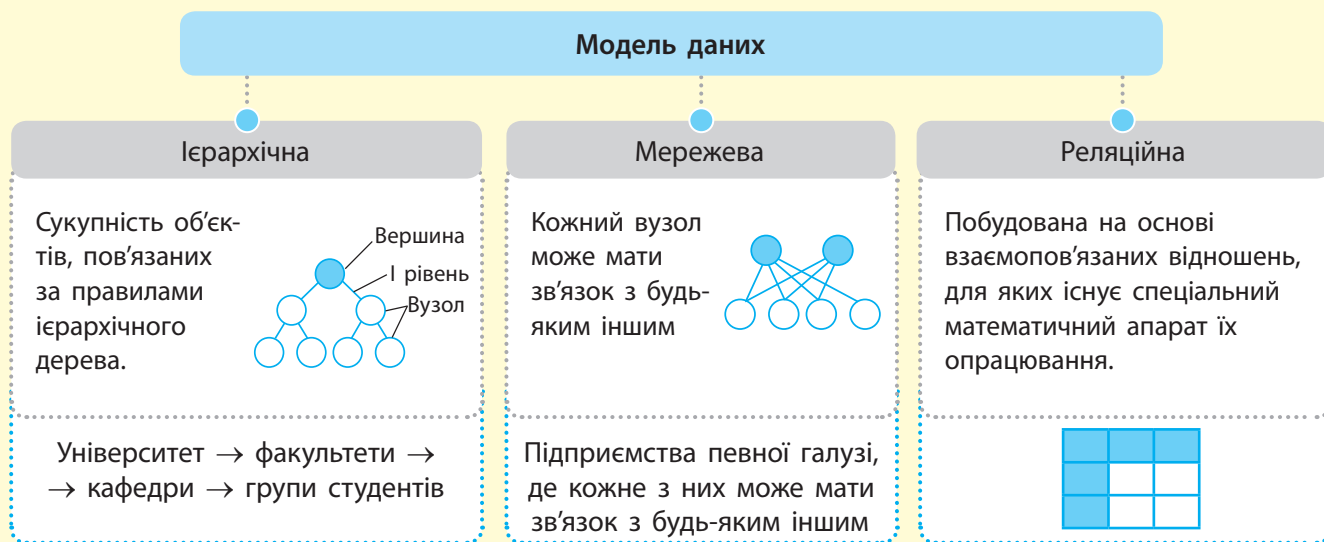


Рис. 1. Моделі даних

Отже, модель даних визначає, яким способом відбувається об'єднання даних у структурі, встановлює обмеження на значення даних і можливі операції над ними. Найпоширенішою зараз є **реляційна модель даних**. Активно розвиваються об'єктно-реляційна й об'єктно-орієнтована моделі даних, які ґрунтуються на класичних.

За характером даних БД поділяють на дві основні групи: **документальні** й **фактографічні** (рис. 2). Далі розглядатимемо в основному фактографічні БД.

Реляційна модель — це фактографічна БД, що є набором взаємопов'язаних таблиць.

Найпростіша БД містить одну таблицю, а складні — декілька десятків і навіть сотні таблиць. Розглянемо основні поняття реляційних БД на прикладі таблиці УЧНІ (табл. 1).

Таблиця 1. УЧНІ

Номер	Прізвище	Дата народження	Зріст	Адреса
1	Топов А. І.	25.06.2001	172	вул. Кам'яна, 8
2	Носко І. П.	10.03.2002	168	вул. Березина, 5, кв. 20

У таблиці наведено деякі дані про учнів, а саме: порядковий номер, прізвище, дата народження, зріст і домашня адреса.

Бази даних

Документальні

зберігають бібліографічні, реферативні й повнотекстові документи (*закони, статті, рецепти* тощо). Пошук може здійснюватися за фрагментом тексту або ключовими словами

Фактографічні

зберігають дані про об'єкти предметної галузі у вигляді фактів (*дані про результати ЗНО* тощо). На запит видаються дані про об'єкт (*прізвища учнів класу, які народилися в травні, тощо*)

Рис. 2. Поділ БД за характером даних

Основними елементами таблиці є поле (стовпець) і запис (рядок) (рис. 3).

Поле має ім'я (наприклад, Зріст) і певне значення (наприклад, прізвище Топов А. І.).

Дані в одному полі можуть бути тільки одного типу. Наприклад, у полі Прізвище тип даних символний, а в полі Дата народження — тип дата.

Запис — це множина значень усіх полів таблиці, на перестріжці яких він знаходиться.

Не кожна таблиця може бути об'єктом БД. Для того щоб таблиця була об'єктом БД, вона повинна відповідати таким основним вимогам:


- У таблиці не може бути однакових записів, вони повинні відрізнитися значенням хоча б одного поля.
- Імена полів у таблиці повинні бути різними. Одне поле не може об'єднувати два і більше імен.
- Таблиця повинна мати ключ.

Зазвичай у таблиці є унікальне поле або кілька полів, які однозначно ідентифікують записи. Такі поля називають **ключем** (*ключовим полем*). Він використовується для швидкого пошуку і зв'язування даних із різних таблиць. Ключ, який містить тільки одне поле, називають **простим**, а ключ, який містить декілька полів, — **складним** (рис. 4).

Зважаючи на те, що в одній таблиці може бути декілька ключів, тільки один із них можна визнати первинним. Найкраще первинним ключем вибрати простий ключ (бажано, щоб він мав цілочисловий тип). У цьому випадку операції з опрацювання даних будуть виконуватися швидше.

У таблиці часто використовується поле, яке називається **лічильником**. Лічильник використовується для того, щоб зробити кожний запис унікальним. Крім того, лічильник забезпечує нумерацію записів.

Реляційна БД може містити лише одну таблицю, але реально вона складається із сукупності взаємопов'язаних між собою таблиць. Зазвичай уміст таблиць формується за функціональною ознакою. Це спрощує модифікацію БД і має інші переваги. Наприклад, у БД підприємства оптової торгівлі в одній таблиці можуть зберігатися дані про замовників товару, у другій — про співробітників підприємства, у третій — про відвантажені протягом доби товари тощо.

 Зв'язки можуть існувати між двома, трьома й більшою кількістю таблиць. Але найчастіше вони встановлюються між двома таблицями.

Для встановлення зв'язків між двома таблицями одну з них вибирають **основною** (батьківською), а другу — **допоміжною** (дочірньою). В основній таблиці вибирають **первинний ключ**,



У 1973 році американський вчений Чарльз Бахман отримав премію Тюрінга за керування роботою Data Base Task Group (робоча група по базах даних, США), яка розробила стандартну мову опису даних і маніпулювання даними.

Номер	Прізвище	Дата народження	Зріст
1	Топов А. І.	25.06.2001	172
2	Носко І. П.	10.03.2002	168

Diagram labels: "Поле" points to the header row, "Значення поля" points to the data rows, "Запис" points to the second row.

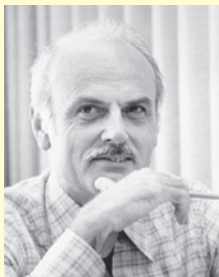
Рис. 3. Структура таблиці

Номер	Прізвище	Дата народження	Зріст
1	Топов А. І.	25.06.2001	172
2	Носко І. П.	10.03.2002	168

Diagram labels: "Простий ключ" points to the "Номер" column, "Складний ключ" points to the "Дата народження" and "Зріст" columns.

Рис. 4. Ключове поле

Таблиця може мати декілька ключів. У табл. 1 УЧНІ поля Прізвище і Дата народження можуть бути визнані ключем, оскільки неможливо, щоб в одній школі були два чи більше учнів з однаковими прізвищами, іменами, по-батькові, які народилися в один день.



У 1970-х роках американський математик Е. Кодд розробив теоретичні основи реляційної моделі даних, яка швидко посіла домінуюче становище на ринку БД. У 1981 році за свій внесок у теорію і практику вечний отримав премію Тюрінга.

а в допоміжній — **спеціальне поле** (або сукупність полів) для забезпечення зв'язку між таблицями, яке називається **зовнішнім ключем** (рис. 5).

Зовнішній ключ повинен однозначно визначати поле основної таблиці, на яку він посилається. У ньому не має бути даних, відсутніх у первинному ключі, інакше посилання буде некоректним. Розглянемо приклад.



Приклад. Нехай у БД вступних іспитів до інституту є дві таблиці: АБІТУРІЄНТИ (табл. 2) і РЕЗУЛЬТАТИ (табл. 3).

Таблиця 2. АБІТУРІЄНТИ

Код абітурієнта	ПІБ	Телефон
023	Іванов П. Н.	21-355
024	Сидоров Є. М.	30-240
025	Опришко М. В.	21-621

Таблиця 3. РЕЗУЛЬТАТИ

Предмет	Бали	Код абітурієнта
Математика	57	023
Математика	56	024
Математика	54	025
Фізика	52	024

У табл. 2 АБІТУРІЄНТИ первинним ключем є поле з іменем Код абітурієнта. У табл. 3 РЕЗУЛЬТАТИ це поле не є унікальним, тому що результати в ньому дублюються (той самий абітурієнт здає іспити з декількох предметів). Але це поле є зовнішнім ключем, і за його даними можна встановити зв'язок із табл. 2. Отже, зв'язок між цими двома таблицями

встановлюється за допомогою поля Код абітурієнта. Якщо потрібно дізнатись прізвище й домашній телефон абітурієнта, який отримав із математики 56 балів, а з фізики — 52 бали, то слід із табл. 3 РЕЗУЛЬТАТИ вибрати його код (024) і за значенням цього коду знайти в табл. 2 АБІТУРІЄНТИ прізвище Сидоров Є. М., телефон 30-240.

Первинний ключ

Код абітурієнта	ПІБ	Телефон
023	Іванов П. Н.	21-355
024	Сидоров Є. М.	30-240
025	Опришко М. В.	21-621

Зовнішній ключ

Предмет	Бали	Код абітурієнта
Математика	57	023
Математика	56	024
Математика	54	025
Фізика	52	024

Рис. 5. Зв'язані таблиці

Як бачимо, зв'язки дозволяють отримати дані з декількох таблиць (див. рис. 5).



Зв'язки між таблицями забезпечують захист даних і автоматичну зміну даних у декількох таблицях, якщо сталися зміни в одній таблиці.

Наприклад, припустимо, що зв'язок між табл. 2 і 3 не встановлено. З якоїсь причини із табл. 2 видалено запис із прізвищем Сидоров Є. М., а в табл. 3 всі дані збереглися (рис. 6). Тоді невідомо, кому належать результати іспитів із математики й фізики абітурієнта, що має код 024. Вважається, що у цьому випадку трапилося порушення *цілісності даних*.

Залежно від того, у якому співвідношенні перебувають ключові поля з полями зв'язку, між двома таблицями можуть існувати чотири види зв'язку (рис. 7).

Код абітурієнта	ПІБ	Телефон
023	Іванов П. Н.	21-355
025	Опришко М. В.	21-621

Предмет	Бали	Код абітурієнта
Математика	56	024
Математика	54	025
Фізика	52	024

Рис. 6. Порушення цілісності даних



Рис. 7. Види зв'язку між двома таблицями

Невід'ємною складовою БД є система, що забезпечує весь цикл робіт від її створення до отримання даних, яка має назву **система управління базами даних (СУБД)**.

Основними функціями СУБД є створення бази даних, опрацювання даних за відповідними алгоритмами, пошук потрібних даних, забезпечення цілісності даних, захист даних і забезпечення їх таємниці.

? Запитання для перевірки знань

- 1 На які основні групи поділяють БД?
- 2 Які класичні моделі даних існують?
- 3 Яке поле називають ключовим?
- 4 Для чого використовують зовнішній ключ?
- 5 Наведіть означення бази даних.
- 6 Поясніть сутність реляційної моделі даних.
- 7 Яким основним вимогам повинні відповідати таблиці баз даних?
- 8 Поясніть сутність об'єкта предметної галузі.
- 9 Які основні функції виконує система управління базами даних?
- 10 Які типи зв'язків існують між таблицями баз даних?

4.2. Загальні відомості про систему Access 2016



Які чинники, на вашу думку, сприяли бурхливому розвитку баз даних в останні декілька десятиліть?

СУБД Access 2016 функціонує під керуванням ОС Windows. Бажано, щоб процесор мав частоту не менше 800 МГц, оперативна пам'ять була не менше 512 Мб і вільна пам'ять на жорсткому диску — не менше 2 Гб.

Історія розвитку баз даних і систем управління ними складається з кількох етапів. За цей час розроблено багато СУБД, наприклад Dbase, FoxPro, Oracle 8.4, MS SQL Server 7.0, SQL Base, MS Access 7 та ін. Всі вони по-різному працюють з різними об'єктами і мають різні функціональні можливості. Та попри все, більшість із них спирається на єдиний комплекс основних понять. Це дає нам можливість розглянути одну систему й узагальнити її поняття, прийоми й методи на весь клас СУБД. Далі розглядатимемо одну з найпоширеніших сьогодні СУБД — Access 2016.

Access 2016 входить до пакета Microsoft Office та призначена для створення й роботи з реляційними БД. Ця система працює з такими об'єктами, як таблиці, запити, форми, звіти (рис. 1). Усі об'єкти найчастіше зберігаються в одному файлі. Однак інколи таблиці зберігаються в одному файлі, а інші об'єкти — в іншому файлі.



Рис. 2. Стартове вікно системи Access 2016

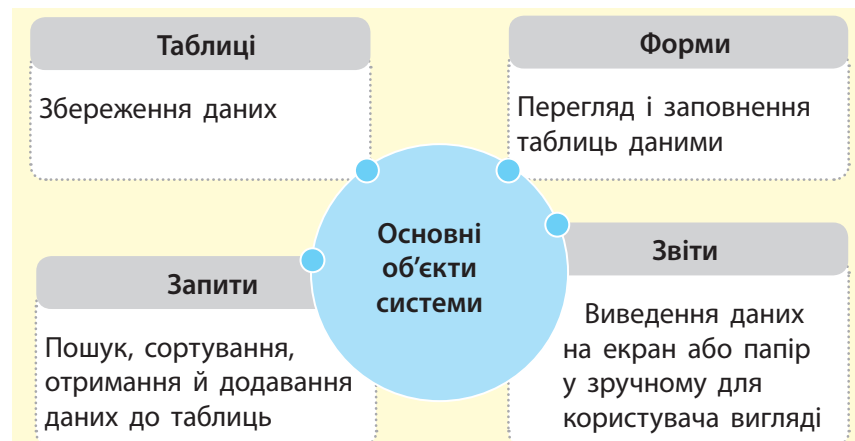


Рис. 1. Об'єкти СУБД та їх призначення

Способи створення БД у середовищі MS Access 2016

- за допомогою порожньої БД
- з використанням шаблонів таблиць
- з використанням копії вже існуючої БД

Рис. 3. Створення БД

Запуск системи Access можна здійснити стандартними способами, що передбачені в ОС Windows. Після запуску системи на екрані монітора з'явиться її стартове вікно (рис. 2). Із цього вікна починається робота вже з існуючою БД або створення нової (рис. 3).

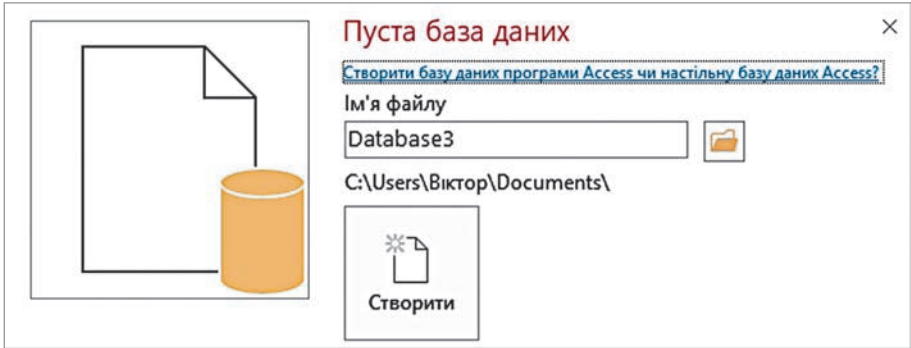
У лівій області вікна відображено імена БД, з якими користувач працював останнім часом. У правій області вікна розташовано шаблони різноманітних БД, які можна наповнити власними даними, що суттєво прискорює створення БД. Скориставшись полем пошуку, можна викликати нові шаблони.

Серед шаблонів в Access є Пуста база даних, яка слугує для створення нової БД. У подальшому ми будемо використовувати саме цей спосіб.

Ознайомимося з **алгоритмом створення нової порожньої БД**.

Крок 1

Клацнути поле **Пуста база даних**.
Відкриється вікно, зображене на рис. 4.



Крок 2

У рядок **Ім'я файлу** ввести ім'я файла майбутньої БД, наприклад **osvita**, натиснути кнопку **Знайти розташування для бази даних**, що розташована праворуч від цього рядка.

Крок 3

У вікні, що відкриється, вибрати місце збереження файла БД, наприклад диск F:, натиснути кнопку **ОК**, а потім — кнопку **Створити**.

У результаті цих дій файл БД osvita буде зареєстровано у кореневому каталозі диска F:, а на екрані з'явиться вікно для створення таблиці 1. Після цього можна приступити до створення таблиць БД або закрити систему Access.

Тепер розглянемо початкове вікно відкритої БД на вкладці Поля (рис. 5).

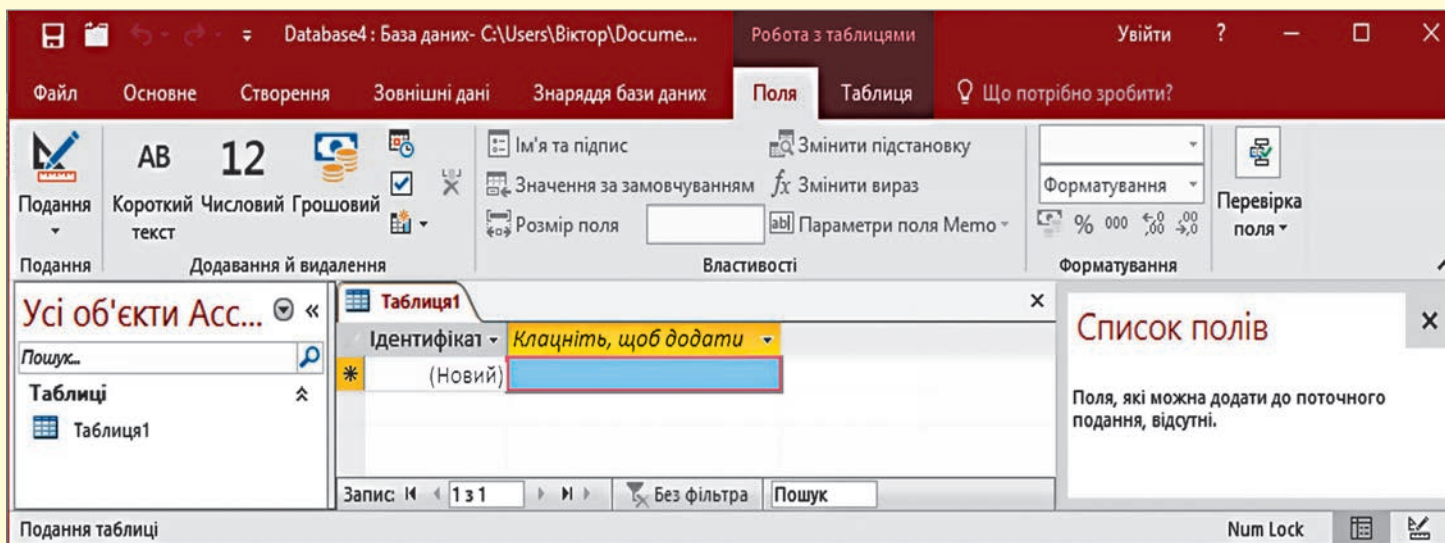


Рис. 5. Початкове вікно бази даних



У кожний момент часу Access 2016 підтримує роботу тільки з однією БД. Число користувачів, які одночасно працюють із БД, може досягати 255. Імена об'єктів можуть включати до 64 символів, максимальний обсяг файлу БД — 2 Гб.



Щоб згорнути й розгорнути стрічку, слід скористатися відповідними командами кнопки **Налаштувати панель швидкого доступу**, яка міститься на панелі швидкого доступу, або натиснути праву кнопку миші в будь-якому її місці та виконати команду **Згорнути стрічку** в контекстному меню, що з'явиться.

Основним об'єктом вікна є стрічка, на якій розташовано команди й інструменти Access 2016. У верхній частині стрічки розміщено вкладки, зокрема Файл, Основне, Створення, Зовнішні дані, Знаряддя бази даних. Зміст команд та елементів керування, які відображаються на стрічці, залежить від того, яка в даний момент відкрита вкладка. Усі команди й елементи керування згруповані в розділах, назви яких виведені у нижній частині стрічки.

У нашому випадку (див. рис. 5) на стрічці розташовано розділи Подання, Додавання й видалення, Властивості та інші, а над ними містяться відповідні кнопки команд. Якщо відкрити вкладку Створення, то будуть виведені інші розділи й відповідні команди.

Стисло ознайомимося з призначенням вкладок.

Вкладка Основне містить команди й елементи керування, які найчастіше виконуються в процесі роботи з БД. Це команди для роботи з буфером обміну, форматування тексту, сортування й фільтрування даних тощо.

За допомогою команд вкладки Створення створюють таблиці, запити, форми, звіти й макроси, за допомогою команд вкладки Зовнішні дані здійснюють експортування, імпортування й збирання даних.

Вкладка Знаряддя бази даних містить в основному команди, що стосуються адміністрування БД (команди аналізу й розміщення даних між програмами та ін.).

У лівій області екрана вміщено Усі об'єкти Access — панель переходів, на якій можуть відображатися назви всіх створених об'єктів. За допомогою цієї панелі можна здійснювати перехід між об'єктами БД, змінювати параметри області переходів.

Праворуч від панелі переходів є область редагування, у якій можуть одночасно відображатися таблиці, запити та інші об'єкти. Щоб закрити БД, слід скористатися командою Закрити базу даних у меню швидкого доступу.



Запитання для перевірки знань

- 1 Як запустити систему Access 2016?
- 2 Назвіть основні об'єкти системи Access 2016.
- 3 Які є способи створення файлів бази даних?
- 4 Яке призначення вкладок стрічки початкового вікна системи Access 2016?
- 5 Поясніть порядок створення файлів бази даних.



Завдання для самостійного виконання

- 1 Запустіть систему Access 2016. Створіть на жорсткому диску файл БД з іменем **Mybasa**. Переконайтеся, що файл зареєстровано.
- 2 Проаналізуйте призначення об'єктів початкового вікна БД, відкриваючи різні вкладки.
- 3 Відкрийте та проаналізуйте призначення команд кнопки **Налаштувати панель швидкого доступу**.
- 4 Відкрийте і проаналізуйте зміст вкладки **Основне**.

4.3. Таблиці

Приступаючи до розробки БД, необхідно ретельно виконати її проектування. Проектування передбачає детальний аналіз поставленого завдання, визначення вимог до документів, які потрібно отримати. Особливу увагу слід приділити визначенню кількості таблиць, структурі кожної з них, зв'язкам між ними.

4.3.1. Створення структури таблиць

Які основні складові, на вашу думку, повинна містити база даних? Наведіть приклади найпростіших баз даних.



У середовищі Access існує кілька способів створення таблиць (рис. 1). Найбільш потужним і універсальним способом створення таблиць є режим конструктора, який розглянемо нижче.

Порядок і методику створення таблиць розглянемо на такому прикладі.



Приклад. Нехай БД торговельної корпорації з назвою corporation містить дві таблиці — ФІРМИ і ПРАЦІВНИКИ (табл. 1 і 2 відповідно).

Основною вважатимемо табл. 1, а табл. 2 — допоміжною.

Способи створення таблиць

- за допомогою майстра таблиць
- у режимі таблиці
- у режимі конструктора

Рис. 1. Способи створення таблиць

Таблиця 1. ФІРМИ

Фірма	Адреса	Директор	Телефон	Працівники	Провізори	Примітка
14	вул. Зоряна, 3	Зімін І. Ф.	241-750	800	60	Директор — новопризначений
12	вул. Морська, 20	Ясин В. М.	350-621	400	30	
13	вул. Заводська, 15	Бойко М. І.	400-502	700	50	Директор — заслужений працівник

Таблиця 2. ПРАЦІВНИКИ

Справа	Прізвище	Посада	Народження	Освіта	Стаж	Тиждень	Фірма
1020	Ахтирко В. В.	аналітик	1960	вища	25	48	12
1005	Носко М. Г.	диспетчер	1972	середня спеціальна	15	21	13
1140	Сороко Я. М.	експерт	1980	вища	5	46	14
1075	Мерез Ф. Б.	диспетчер	1970	середня	17	40	14
1232	Варава А. А.	диспетчер	1975	середня спеціальна	10	36	12
1206	Грач П. О.	аналітик	1966	вища	16	46	14
1230	Зоря Ш. Ш.	диспетчер	1961	середня	22	48	12

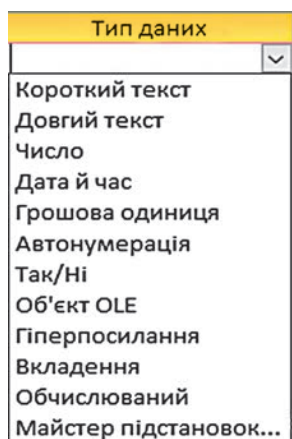


Рис. 2. Типи даних системи

Створити структуру таблиці означає визначити й увести в комп'ютер назви й типи її полів, а також властивості типів. У середовищі Access 2016 використовують типи даних, перелік яких наведено на рис. 2.

Кожний із цих типів даних має власний набір властивостей. Окремі властивості є унікальними, тобто містяться тільки в одному конкретному типі даних. Але переважна їх більшість — загальні, тобто містяться у різних типах даних. Далі будемо використовувати в основному такі типи даних: *Короткий текст*, *Довгий текст*, *Число*, *Дата й час*.

Тип *Короткий текст* — це послідовність символів довжиною від 0 до 255, а тип *Довгий текст* — це послідовність символів до 65 536. У полі типу *Число* зберігаються числа. Розмір і конкретний тип чисел визначаються значенням властивості *Розмір поля*. У полях типу *Дата й час* зберігаються дати й час різних форматів.

Для більш зручної роботи у ході введення структури табл. 1 і 2 доцільно створити на папері додаткові таблиці (табл. 3 і 4) з іменами полів, типом даних та їх описом. Назви полів залишимо ті самі, тип даних і властивості полів виберемо з урахуванням можливостей системи Access 2016.

Таблиця 3. Структура таблиці ФІРМИ

Ім'я стовпця	Тип даних	Опис	Властивості поля
Фірма	Число	Первинний ключ	
Адреса	Короткий текст		50
Директор	Короткий текст		40
Телефон	Число		20
Працівники	Число	Станом на 1 вересня	
Провізори	Число		
Примітка	Довгий текст	Додаткові відомості про директора фірми	

Таблиця 4. Структура таблиці ПРАЦІВНИКИ

Ім'я стовпця	Тип даних	Опис	Властивості поля
Справа	Число	Первинний ключ	
Прізвище	Короткий текст		20
Посада	Короткий текст		30
Народження	Число		
Освіта	Короткий текст		30
Стаж	Число	Станом на 1 січня	
Тиждень	Число		
Фірма	Число		

Тепер перейдемо безпосередньо до створення структур таблиць у режимі конструктора, користуючись нашими табл. 1–4.

1. Завантажимо систему Access і відкриємо БД corporation, для чого в області Останні натиснемо клавішу миші на імені БД.
2. У вікні, що відкриється, активізуємо вкладку Створення.

Відкриється вікно зі стрічкою (рис. 3).

3. На цій стрічці натиснемо кнопку Конструктор таблиць.

У результаті до БД додасться порожня таблиця (рис. 4).

Слід звернути увагу на те, що вміст стрічки змінився. Тепер на екрані відкрито вкладку Конструктор (рис. 5).

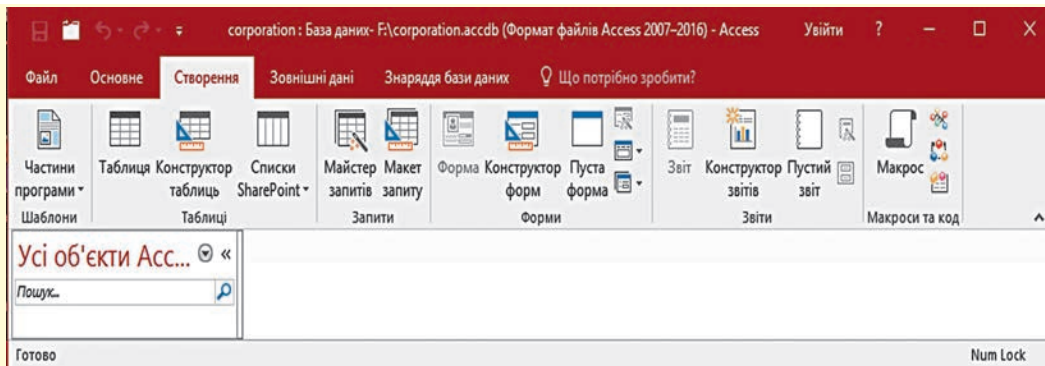


Рис. 3. Стрічка з активованою вкладкою Створення

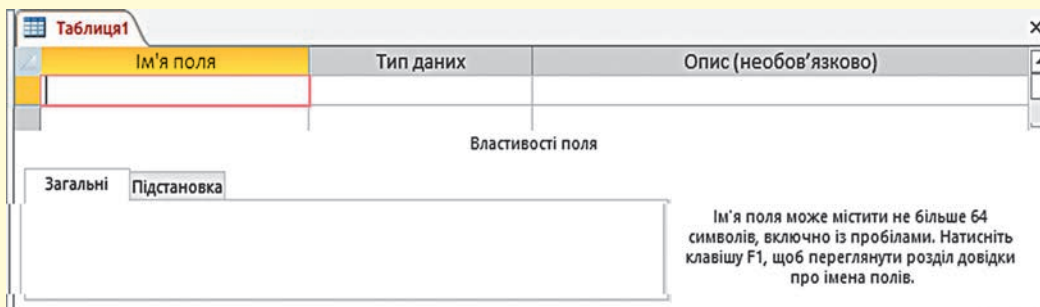


Рис. 4. Порожня таблиця в режимі конструктора

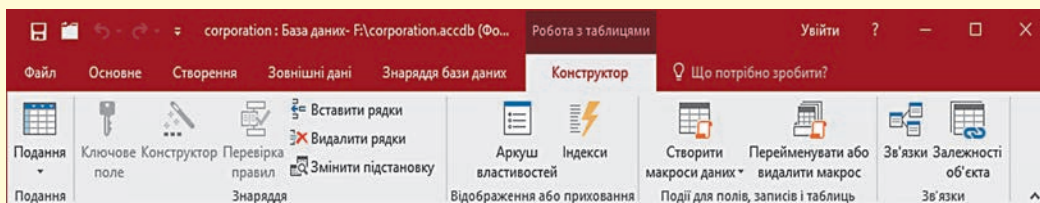


Рис. 5. Стрічка з активованою вкладкою Конструктор

4. Введемо імена полів. Їхнє введення здійснюється стандартним способом: потрібно встановити курсор миші в необхідну клітинку таблиці й за допомогою клавіатури ввести певні імена.

Типи полів краще вводити шляхом вибору їх зі списку типів (див. рис. 1).

Щоб відкрити цей список, необхідно у певній клітинці поля Тип даних клацнути кнопку миші на кнопці Прапорець і вибрати необхідний тип. Опис полів не обов'язковий. Його використовують для вказування призначення поля й допустимих його значень.

5. Уведемо всі дані з табл. 3 Структура таблиці ФІРМИ у таблицю з іменем ФІРМИ і збережемо її. Для цього слід на панелі швидкого доступу натиснути кнопку

Зберегти або клавіші Ctrl + S, увести ім'я таблиці й натиснути кнопку ОК. На екран буде виведено попередження (рис. 6).

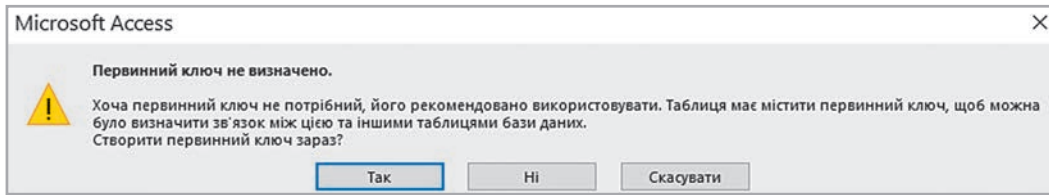


Рис. 6. Вікно попередження про невизначений первинний ключ

6. Натискання кнопки Так у цьому вікні приведе до створення первинного ключа типу *Лічильник*. На цьому етапі не обов'язково встановлювати первинний ключ, тому натиснемо кнопку Ні. Таблиця буде

збережена, а її ім'я з'явиться в області переходів. Закриємо табл. 1 ФІРМИ, для чого відкриємо її контекстне меню та виконаємо команду Закрити. Аналогічно створимо структуру табл. 2 ПРАЦІВНИКИ.

Модифікування структури таблиці передбачає

- вилучення полів
- вставлення нових полів
- змінення порядку розміщення полів
- змінення імен полів
- змінення розмірів полів
- перетворення типів даних

Рис. 7. Операції з модифікування структури таблиці

Створену структуру таблиці можна модифікувати, виконавши певні операції (рис. 7). Виконання цих операцій відбувається аналогічно операціям у ОС Windows.

Для вилучення поля таблиці в режимі конструктора слід відкрити контекстне меню цього поля й виконати команду Видалити рядки. Якщо поле, яке вилучають, містить дані, то буде видано попередження про те, що дані цього поля будуть втрачені. Якщо після вилучення поля з'ясується, що це вилучення було помилковим, слід натиснути кнопку Скасувати на панелі швидкого доступу. Поле буде повернуто в таблицю. Однак, якщо після вилучення поля були виконані будь-які операції, відновити поле буде неможливо.



Поля, які є первинним або зовнішнім ключем, вилучити неможливо доти, поки не буде вилучено зв'язок між таблицями.



Запитання для перевірки знань

- 1 Які способи створення таблиць існують у системі Access?
- 2 Назвіть основні типи даних у системі Access.
- 3 Як установити типи полів?
- 4 Як встановити властивості полів?
- 5 Які операції можна виконувати над створеними структурами таблиць?



Завдання для самостійного виконання

- 1 Створіть на папері таблицю МОЇ ДРУЗІ з полями **Номер**, **Прізвище**, **Ім'я**, **Телефон**, **День народження** й **Адреса**.
- 2 Розробіть на папері структуру таблиці з типами її полів і властивостями.
- 3 Самостійно на папері заповніть таблицю даними шістьох своїх друзів.

4.3.2. Ключові поля. Зв'язування таблиць

Пригадаємо, які основні дії необхідно виконати в процесі створення структури таблиці. Які операції можна виконувати в таблиці? Чи можна виконати ці операції, якщо БД містить не одну таблицю?



Ключі у БД відіграють важливу роль — за їх допомогою СУБД ідентифікує об'єкти.

Згадаємо, що ключове поле (ключ) — це поле таблиці, значення якого не повторюється в жодному іншому записі.

Для створення первинного ключа слід відкрити таблицю в режимі конструктора, виділити поле, яке використовується як первинний ключ, і натиснути кнопку Ключове поле, що знаходиться в розділі Знаряддя вкладки Конструктор.

Створити первинний ключ можна також за допомогою контекстного меню певного поля. Скориставшись цим способом, у табл. 1 ФІРМИ визначимо як первинний ключ поле Фірма. Поряд із назвою цього поля з'явиться малюнок ключа (рис. 8). Збережемо таблицю ФІРМИ.

Ім'я поля	Тип даних	Опис (необов'язково)
Фірма	Число	Первинний ключ
Адреса	Короткий текст	50
Директор	Короткий текст	40
Телефон	Короткий текст	20
Працівники	Число	Станом на 1 вересня
Провізори	Число	
Примітка	Довгий текст	Додаткові відомості про директора фірми

Рис. 8. Структура таблиці ФІРМИ з ключовим полем

Для створення первинного ключа типу Лічильник у відкритій таблиці в режимі конструктора необхідно додати до таблиці поле, надати йому тип Лічильник (Автонумерація), виділити його й натиснути кнопку Ключове поле. Крім того, первинний ключ для поля цього типу може бути встановлений і в тому разі, якщо під час створення таблиці взагалі ключ не визначено.

Розглянемо тепер алгоритм встановлення зв'язків між таблицями на прикладі таблиць ФІРМИ і ПРАЦІВНИКИ.



Якщо деяке поле в конструкторі оголошене як Лічильник, то воно за замовчуванням стає ключовим.

1. Відкриваємо БД corporation, потім вкладку Знаряддя бази даних та натискаємо кнопку Зв'язки.

Відкриється вікно Відображення таблиці.

Якщо останнього вікна немає, слід натиснути кнопку Відобразити таблицю, повинно з'явитися вікно (рис. 9).

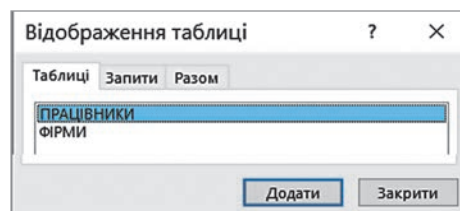


Рис. 9. Вікно з переліком таблиць бази даних

2. У схему даних додаються таблиці, які необхідно зв'язати. Для цього у відкритому вікні Відображення таблиці виділяємо імена таблиць, що зв'язуються.
3. Натискаємо кнопку Додати й закриваємо вікно Відображення таблиці.
4. Установлюємо курсор на первинному ключі головної таблиці (ФІРМИ), натискаємо

кнопку миші і, не відпускаючи, переміщаємо в поле зовнішнього ключа (Фірми) таблиці ПРАЦІВНИКИ.

5. Відпускаємо кнопку миші. У результаті відкриється вікно Редагування зв'язків (рис. 10).

Рис. 10. Діалогове вікно для редагування зв'язків

6. Для зручності в подальшій роботі слід увімкнути прапорець Забезпечення цілісності даних. Після цього стануть доступними Каскадне оновлення пов'язаних полів і Каскадне видалення пов'язаних записів. Вмикаємо останній прапорець.

Сутність каскадного оновлення пов'язаних записів: за будь-якої зміни значення первинного ключа запису в основній таблиці автоматично оновиться значення відповідного поля у всіх зв'язаних таблицях.

Сутність каскадного видалення пов'язаних записів: під час видалення будь-якого запису з головної таблиці автоматично

видаляються зв'язані записи й у зв'язаної таблиці.

Таким чином, каскадне оновлення і каскадне видалення прискорюють роботу з БД і сприяють підвищенню надійності її функціонування.

7. У вікні Редагування зв'язків натискаємо кнопку Створити, в результаті чого у вікні Зв'язки з'явиться лінія зв'язку між певними полями таблиць. За описаною методикою слід встановити зв'язок між таблицями ФІРМИ і ПРАЦІВНИКИ. Якщо все буде виконано правильно, з'явиться схема, зображена на рис. 11. Далі слід виконати команду Зберегти.

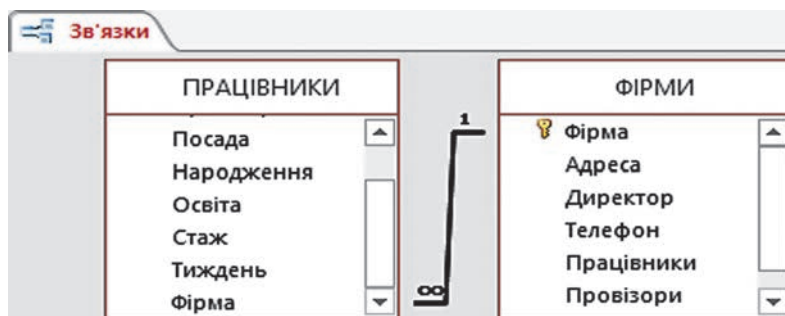


Рис. 11. Зв'язок типу 1:Б між таблицями

8. Повертаємося ще раз до вікна Редагування зв'язків (якщо в даний час його немає на екрані, слід натиснути кнопку Змінити зв'язки в області Знаряддя на вкладці Конструктор).

Натискаємо у цьому вікні кнопку Тип об'єднання... У результаті з'являється вікно Параметри об'єднання (рис. 12).

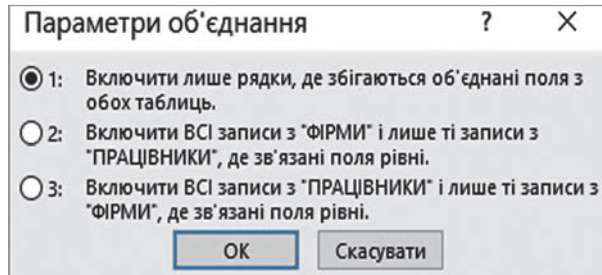


Рис. 12. Вікно для встановлення параметрів об'єднання таблиць

9. За замовчуванням встановлюється перший тип об'єднання. Як правило, розробники БД користуються цим типом

об'єднання. Далі потрібно натиснути у цьому вікні кнопку ОК і закрити вікно Редагування зв'язків.

Для скасування зв'язку між таблицями необхідно відкрити контекстне меню лінії зв'язку й виконати команду Видалити.

? Запитання для перевірки знань

- 1 Як створити первинний ключ?
- 2 Як створити первинний ключ типу *Лічильник*?
- 3 Як можна скасувати зв'язок між таблицями?
- 4 Поясніть порядок установлення зв'язків між таблицями.

🖨 Завдання для самостійного виконання

- 1 Створіть таблицю УСПІХИ ДРУЗІВ із полями Прізвище, Навчання, Спорт, Музика.
- 2 Створіть ключове поле в таблиці МОЇ ДРУЗІ.
- 3 Заповніть таблицю УСПІХИ ДРУЗІВ необхідними даними.
- 4 Виконайте зв'язування таблиць МОЇ ДРУЗІ та УСПІХИ ДРУЗІВ.

4.3.3. Введення даних у таблицю

До цього часу ми створювали лише структури таблиць. Яким повинен бути, на вашу думку, наступний крок створення БД?



Після задання структури й імені таблиці можна вводити дані. Розглянемо алгоритм введення даних у таблицю на прикладі таблиці ФІРМИ.

1. Відкриємо таблицю ФІРМИ в режимі таблиці. Для цього встановимо курсор в області переходів на її імені й двічі

натиснемо кнопку миші. Також можна скористатися контекстним меню.

2. Введемо дані першого запису таблиці ФІРМИ в рядок, позначений зірочкою. Щойно будуть уведені дані в запис, курсор переміститься на наступний запис, що визначає готовність наступного рядка до введення нового запису.

Слід пам'ятати, що вводити в поля можна тільки ті типи даних, які співпадають із визначеним типом поля.

3. Вводимо всі записи таблиці ФІРМИ і отримуємо вміст таблиці (рис. 13). Зберігаємо таблицю.

Фірма	Адреса	Директор	Телефон	Працівники	Провізори	Примітка
12	вул. Морська, 20	Ясин В.М	350-621	400	30	
13	вул. Заводська, 15	Бойко М.І.	400-502	700	50	Директор-заслужений працівник
14	вул. Зоряна, 3	Зімін І.Ф.	241-750	800	60	Директор-новопризначений
*	0			0	0	

Рис. 13. Вміст таблиці ФІРМИ



У процесі введення даних автоматично перевіряються такі типи даних: числові, грошові, дати й часу, логічні. На вкладці **Основне** у групі **Форматування тексту** розташовані елементи, за допомогою яких можна змінити розмір шрифту тексту та інші його параметри.

Якщо в таблиці є поле типу Лічильник, воно формується автоматично. У випадках, якщо на екрані не вміщаються всі записи, слід скористатися вертикальною смугою прокручування, а якщо не вміщуються поля — горизонтальною смугою прокручування.

За замовчуванням записи виводяться на екран відсортованими в порядку значення первинного ключа, а поля — у послідовності, визначеній структурою таблиці. У режимі таблиці можна змінювати деякі параметри виведення даних, наприклад розмір шрифту, ширину стовпців, висоту рядків.

Для здійснення навігації по записах у нижній частині вікна таблиці розміщені кнопки навігації, призначені для переміщення курсору в перший, сусідній або останній запис. Навігацію в таблиці можна також здійснювати за допомогою миші, смуг прокручування, сполучення деяких клавіш.

Для додавання нового запису в таблицю треба натиснути кнопку Створити запис на панелі навігації й увести дані.



Запитання для перевірки знань

- 1 У якому порядку виводяться записи таблиці після її створення?
- 2 Як можна здійснити навігацію по записах у таблиці?
- 3 Поясніть порядок введення даних у таблицю.
- 4 Як додати новий запис у таблицю?



Завдання для самостійного виконання

- 1 Уведіть і збережіть дані таблиці МОЇ ДРУЗИ.
- 2 Уведіть і збережіть дані таблиці УСПІХИ ДРУЗІВ.

4.3.4. Пошук, заміна і фільтрування даних у таблицях

Нехай створена таблиця УЧНІ вашого класу. Які дані ви б хотіли отримати з цієї таблиці? Що для цього слід зробити?



Потрібний запис у таблиці можна знайти за значенням будь-якого його поля або за фрагментом його значення. Розглянемо алгоритм здійснення пошуку запису в таблиці на прикладі табл. 2 ПРАЦІВНИКИ.

1. У відкритій таблиці ПРАЦІВНИКИ в режимі таблиці виділяємо поле, за значенням якого треба шукати запис (наприклад, поле Прізвище), і натискаємо кнопку Знайти в розділі Пошук.

Відкриється вікно Пошук і заміна (рис. 14).

2. У поле Знайти вікна Пошук і заміна вводимо потрібне значення, наприклад Сороко Я. М., і натискаємо кнопку Знайти далі. У результаті курсор встановиться на записі з цим іменем.

У полі Зіставити можна вибрати одне зі значень, що зменшує необхідність використання метасимволів:

- **будь-яку частину поля** (зразок може знаходитися всередині поля, наприклад зразок ере буде знайдено в запису з прізвищем Мереж Ф. Б.);
- **усе поле** (зразок без метасимволів повинен збігатися з усім значенням поля);
- **початок поля** (будуть знайдені тільки ті поля, які починаються зі зразка).

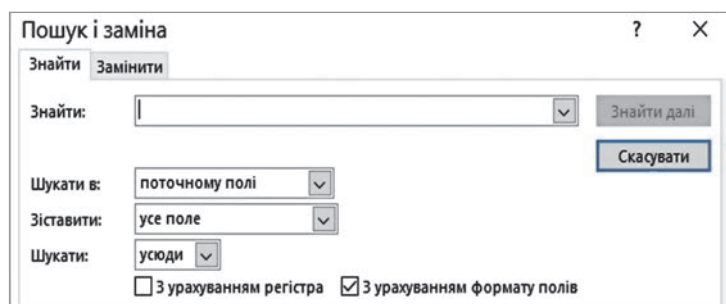


Рис. 14. Діалогове вікно для пошуку й заміни даних



Для прискорення пошуку можна в поле **Знайти** ввести такі метасимволи:

- * — довільна кількість будь-яких символів
- ? — один довільний символ
- ≠ — одна довільна фраза

Записи з таблиці можна копіювати, вирізати до буфера обміну й потім за допомогою кнопки Вставити, яка знаходиться в розділі Буфер обміну, вставляти в іншу таблицю, а також у документи Word і Excel.

Для **вилучення запису з таблиці** необхідно його позначити й натиснути клавішу Del або кнопку Видалити та в меню, що відкриється, виконати команду Видалити запис. Відкриється вікно, де потрібно підтвердити вилучення.



Слід пам'ятати, що у зв'язаних таблицях із встановленим прапорцем **Забезпечення цілісності даних** видалити запис не завжди вдається.

Записи в таблиці можна також *сортувати* й *фільтрувати*.



Сортування записів — це їх впорядкування за значеннями одного або кількох полів.



Іноколи виникає потреба приховати деякі поля. Тоді слід виділити певні поля й виконати у групі **Записи** команду **Додатково** → **Приховати поля**. Для відновлення прихованих полів потрібно у цьому ж розділі виконати команду **Додатково** → **Відобразити поля** й увімкнути прапорець імені відповідного поля.

Сортування

- за значенням одного поля
- за значенням кількох полів

Рис. 15. Способи сортування записів

Розглянемо різні способи сортування записів (рис. 15).

Для сортування за значенням одного поля слід виділити це поле і натиснути кнопку За зростанням (А → Я) або За спаданням (Я → А). Можна також скористатися контекстним меню поля.

Для сортування за значенням декількох полів треба виділити ці поля й скористатися одним із вказаних способів.

У першу чергу сортування записів виконується за значенням лівого поля. Якщо в ньому є поля, які збігаються, то певні записи сортуються за значенням наступного поля. Наприклад, після сортування записів у порядку спадання значень у полі Прізвище таблиці ПРАЦІВНИКИ отримуємо розміщення записів, зображене на рис. 16.

Справа	Прізвище	Посада	Народженн	Освіта	Стаж	Тиждень	Фірма
1140	Сороко Я.М.	експерт	1980	вища	5	46	14
1005	Носко М.Г.	диспетчер	1972	середня спеці	15	21	13
1075	Мерез Ф.Б.	диспетчер	1970	середня	17	40	14
1230	Зоря І.І.	диспетчер	1961	середня	22	48	12
1206	Грач П.О.	аналітик	1966	вища	16	46	14
1232	Варава А.А.	диспетчер	1975	середня спеці	10	36	12
1020	Ахтирко В.В.	аналітик	1960	вища	25	48	12
*	0		0		0	0	0

Рис. 16. Вміст таблиці ПРАЦІВНИКИ, упорядкований за полем Прізвище

Фільтрування

- за виділенням
- за формою

Рис. 17. Способи фільтрування записів

БД можуть містити тисячі записів. Щоб відшукати потрібні дані, слід скористатися фільтром. Наприклад, якщо здійснити фільтрацію записів у таблиці ПРАЦІВНИКИ за значенням *більше або дорівнює 22* в полі Стаж, то будуть відібрані записи з прізвищами Ахтирко В. В. і Зоря І. І.



Фільтрування записів — це відбір із таблиці записів, які містять задане значення в обраних полях.

Розглянемо способи фільтрування записів (рис. 17).

Для фільтрування за виділенням слід відібрати записи на основі значень поточного поля. Для цього спочатку треба впорядкувати записи за значенням поля, яке задано у фільтрації, встановити курсор на тому значенні поля, за яким буде виконуватися фільтрування, та виконати команду Виділення в групі Сортування й фільтр.

Приклад. Відфільтрувати записи таблиці ПРАЦІВНИКИ за значенням *аналітик* поля Посада.

1. Після впорядкування записів за значенням поля Посада встановлюємо курсор на назві *аналітик* і виконуємо команду Виділення у групі Сортування й фільтр. Відкриється меню з умовами: *дорівнює аналітик; не дорівнює аналітик; містить аналітик; не містить аналітик.*

2. Виконаємо першу умову. Отримаємо запис, у полі Посада якого є назва *аналітик*, тобто записи з прізвищами Ахтирко В. В. і Грач П. О.

У нижній частині таблиці праворуч від кнопки навігації висвітиться напис Відфільтровано (або Не відфільтровано), що означає, що записи відфільтровані (або не відфільтровані). За допомогою кнопки Видалити фільтр у групі Сортування й фільтр можна вимкнути або

увімкнути фільтр. До відфільтрованих даних можна застосувати ще кілька фільтрів.

- Відфільтруємо записи в таблиці ПРАЦІВНИКИ за значенням *вища* поля Освіта. Отримаємо результат (рис. 18).
- У фільтрах за формою можна вводити критерії в поля таблиці. Для цього у відкритій таблиці в розділі Сортуння й фільтр необхідно відкрити меню кнопки Параметри розширеного фільтра і виконати команду Розширений фільтр → Сортуння... Відкриється перелік полів таблиці,

а в нижній частині вікна — таблиця конструктора.

- У запис Поле першого поля перенесемо ім'я поля, за яким будемо виконувати фільтрування.
- У запис Критерії введемо потрібний критерій і в групі Сортуння й фільтр натиснемо кнопку Застосувати фільтр.

За описаною методикою для таблиці ПРАЦІВНИКИ виконаємо фільтрування за полем Стаж із критерієм > 15.

Отримаємо результат, як показано на рис. 19.

Справа	Прізвище	Посада	Народження	Освіта	ІУ	Стаж	Тиждень	Фірма
1020	Ахтирко В.В.	аналітик	1960	вища		25	48	12
1206	Грач П.О.	аналітик	1966	вища		16	46	14
1140	Сороко Я.М.	експерт	1980	вища		5	46	14
*	0		0			0	0	0

Запис: 4 з 4 | Відфільтровано | Пошук

Рис. 18. Вміст таблиці ПРАЦІВНИКИ, відфільтрований за значенням поля Освіта

Справа	Прізвище	Посада	Народженн	Освіта	Стаж	ІУ	Тиждень	Фірма
1020	Ахтирко В.В.	аналітик	1960	вища		25	48	12
1075	Мерез Ф.Б.	диспетчер	1970	середня		17	40	14
1206	Грач П.О.	аналітик	1966	вища		16	46	14
1230	Зоря І.І.	диспетчер	1961	середня		22	48	12
*	0		0			0	0	0

Запис: 5 з 5 | Відфільтровано | Пошук

Рис. 19. Вміст таблиці ПРАЦІВНИКИ, відфільтрований за полями Посада і Тиждень



Запитання для перевірки знань

- Які операції можна виконувати над записами таблиці?
- Як вилучити запис із таблиці?
- Поясніть порядок сортування записів у таблиці.
- Поясніть порядок пошуку даних у таблиці.
- Які операції виконують над таблицями?
- Як можна приховати поля таблиці?
- Поясніть порядок фільтрування записів.



Завдання для самостійного виконання

- Виконайте пошук необхідних записів у таблиці МОЇ ДРУЗИ.
- Виконайте редагування даних у таблиці УСПІХИ ДРУЗІВ.
- Виконайте сортування записів у таблиці МОЇ ДРУЗИ за значенням різних полів.
- Виконайте фільтрування записів у таблиці МОЇ ДРУЗИ за різними значеннями даних двох-трьох полів.

4.4. Запити



Таблиця є основним об'єктом бази даних. Але існують й інші об'єкти, наприклад запити. Чому не можна обмежитися тільки таблицями?

Запит

- на вибірку даних
- на змінення
- з параметрами
- з обчисленнями

Рис. 1. Типи запитів даних

Запит

- простий (записи вибирають за значенням одного поля однієї таблиці)
- складний (дані формуються за значеннями кількох полів кількох таблиць)

Рис. 2. Запити за складністю



Запити не містять даних. Під час кожного нового виконання запиту формуються необхідні дані з тих таблиць, на основі яких його створено.

Для БД Access можна створити кілька типів запитів (рис. 1). Кожний із них може бути виконаний самостійно у будь-який час. Але найчастіше запити виконуються як складова форм і звітів.

Запити класифікують за кількома ознаками:

- **кількістю використаних таблиць** (однотабличні й багатабличні);
- **складністю функцій**, які виконує запит (прості й складні) (рис. 2);
- **призначенням** (вибірки даних і заміни даних);
- **способом вибірки й опрацювання даних** (із параметрами, з обчисленнями тощо).

4.4.1. Запити на вибірку даних



Відомо, що деякі дані можна отримати безпосередньо з таблиць. Але зазвичай для цього використовуються запити. Які вони надають переваги?

Запити, які забезпечують вибір необхідних даних із таблиць, називають **запитами на вибірку**. Такий тип запитів часто використовують у формах і звітах та є найпростішим.

Розглянемо алгоритм створення запиту на вибірку даних для однієї таблиці та приклади.

Крок 1	Відкрити БД, активувати вкладку Створення й у розділі Запити натиснути кнопку Макет запиту , який фактично є конструктором запиту. У результаті відкриється вікно запиту й вікно Відображення таблиці , у якому містяться імена всіх таблиць цієї бази.
Крок 2	Вибрати у цьому вікні необхідну таблицю. У результаті відкриється перелік полів цієї таблиці.
Крок 3	Створити на основі вмісту таблиці потрібний запит.



Приклад 1. Створити запит з іменем запит1, за допомогою якого з таблиці ПРАЦІВНИКИ виводяться дані про співробітників зі стажем понад 15 років. Результуючий набір записів повинен мати поля: Справа, Прізвище, Народження, Стаж і Фірма.

1. Для створення запиту відкриваємо БД corporation і виконуємо команду Створення → Макет запиту. Відкриється вікно Відображення таблиці, у якому містяться назви всіх таблиць БД. Вибираємо таблицю ПРАЦІВНИКИ, для чого встановлюємо курсор на імені таблиці й натискаємо кнопку Додати. Закриваємо вікно Відображення таблиці. На екрані залишиться вікно, зображене на [рис. 3](#).
2. У записі Поле таблиці конструктора запитів послідовно розміщуємо вказані імена

полів таблиці ПРАЦІВНИКИ. Для цього достатньо двічі натиснути кнопку миші на імені певного поля цієї таблиці.

3. У записі Критерії поля Стаж вводимо вираз «> 15». Для збереження запиту на панелі швидкого доступу натискаємо кнопку Зберегти. У вікні, що відкриється, вводимо ім'я запит1 і натискаємо кнопку ОК. Ім'я цього запиту з'явиться в області переходів.
 4. Виконуємо запит. Для цього на смузі в групі Результати натискаємо кнопку Запуск!
- Отримуємо результат, наведений на [рис. 4](#).
5. Щоб закрити запит1, відкриваємо контекстне меню запиту й виконуємо команду Закрити.

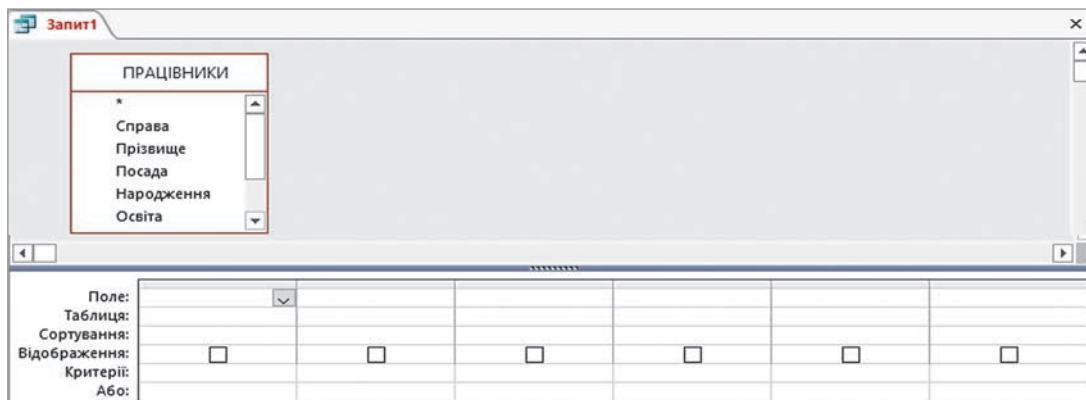


Рис. 3. Вікно конструктора запитів

Справа	Прізвище	Народженн	Стаж	Фірма
1020	Ахтирко В.В.	1960	25	12
1075	Мerez Ф.Б.	1970	17	14
1206	Грач П.О.	1966	16	14
1230	Зоря І.І.	1961	22	12
* р		0	0	0

Запис: 5 з 5 | Без фільтра | Пошук

Рис. 4. Дані про працівників зі стажем роботи понад 15 років



Методика створення запиту для декількох таблиць багато в чому схожа на методику створення запиту для однієї таблиці. Однак слід враховувати, що таблиці обов'язково повинні мати між собою зв'язок.

Приклад 2. Створити запит2, за допомогою якого в результуючу множині включаються прізвища диспетчерів із полями Фірма і Телефон таблиці ФІРМИ й дані полів Прізвище і Посада таблиці ПРАЦІВНИКИ. Записи треба упорядкувати в алфавітному порядку прізвищ працівників.

1. На вкладці Створення в групі Запити виконуємо команду Макет запиту і додаємо у вікно конструктора запиту вказані таблиці. Після цього вікно Відображення таблиці слід закрити.
2. У запис Поле таблиці конструктора запиту переносимо імена полів Фірма і Телефон таблиці ФІРМИ і поля Прізвище і Посада таблиці ПРАЦІВНИКИ.

3. У запис Критерії поля Посада вводимо назву *диспетчер*, а в запису Сортування поля Прізвище встановлюємо значення За зростанням.
4. Зберігаємо запит з іменем запит2 і виконуємо його. На екрані повинні з'явитися записи, наведені на рис. 5.

Фірма	Телефон	Прізвище	Посада
12	350-621	Варава А.А.	диспетчер
12	350-621	Зоря І.І.	диспетчер
14	241-750	Мерез Ф.Б.	диспетчер
13	400-502	Носко М.Г.	диспетчер

Рис. 5. Дані про диспетчерів, вибрані з таблиць



Компанія Oracle є абсолютним лідером на ринку систем управління БД. Їй належить 45 % ринку

Створений запит можна перейменувати, редагувати.

У процесі редагування можна виконувати такі дії:

- додавати поля у запит із таблиці;
- видаляти поля;
- додавати нові поля;
- змінювати розмір поля;
- змінювати послідовність розміщення полів;
- змінювати критерії відбору записів, порядку їх сортування й порядку виведення (невиведення) полів;
- перейменувати поля запиту.



Запитання для перевірки знань

1. Які основні функції виконують запити?
2. Назвіть основні типи запитів системи Access.
3. Поясніть порядок створення запиту для однієї таблиці.
4. Поясніть особливості створення запиту для кількох таблиць.
5. Для чого призначені запити на вибірку даних?



Завдання для самостійного виконання

1. Створіть запит з іменем **запит101**, за допомогою якого з таблиці **МОЇ ДРУЗІ** можна вибрати записи з однаковими іменами учнів. Запит повинен містити поля **Прізвище**, **Дата народження** й **Адреса**.
2. Створіть запит з іменем **запит102**, за допомогою якого з таблиці **УСПІХИ ДРУЗІВ** можна вибрати записи з однаковими успіхами учнів у спорті. Запит повинен містити поля **Прізвище** і **Спорт**.
3. Самостійно додайте у **запит102** нові поля. Змініть порядок їх розташування та здійсніть інші види редагування. Відновіть початковий зміст запиту.

4.4.2. Запити з параметрами і з полями, що обчислюються

Часто виникає потреба отримати з таблиць узагальнені дані, наприклад кількість учнів класу, що мешкають на вул. Лісова. Саме такі типи запитів розглядаються нижче.



Пошукова система Google була створена у 1998 році. За перші два роки її існування у БД цієї пошукової системи було зібрано дані про мільярд веб-сторінок, а у 2008 році — близько трильону.



Запит із параметрами — це запит, у процесі виконання якого пропонується ввести деякі дані, наприклад умову, яку потрібно вставити в поле. Їх ще називають запитими зі змінними критеріями.

За запитими із параметрами на початку їх виконання на екран видаються повідомлення про необхідність введення нового критерію (виразу). Методика створення запиту такого типу несуттєво відрізняється від методики створення звичайного запиту.

Розглянемо алгоритм створення запитів із параметрами на прикладі.



Приклад 1. Розробити запит з іменем запит3, за яким із БД corporation вибиратимуться прізвища працівників за спеціальністю *диспетчер* із фірм, у яких кількість працівників більше 400 осіб і менше 800 осіб. Результуючі записи повинні складатися з поля Фірма таблиці ФІРМИ і полів Прізвище і Посада таблиці ПРАЦІВНИКИ.

На початку виконання запиту необхідно в запис Критерії поля Посада таблиці ПРАЦІВНИКИ увести назву посади (*диспетчер*), а в цей же запис поля Працівники таблиці ФІРМИ — вираз, на основі якого вибираються певні записи.

1. Відкриваємо на екрані порожній бланк запиту.
2. Із таблиці ФІРМИ переносимо в таблицю конструктора поля Фірма і Працівники (для останнього поля — зняти прапорець Відображення), а з таблиці ПРАЦІВНИКИ — поля Прізвище і Посада.
3. У клітинку на перетині запису Критерії і поля Працівники вводим вираз: >[Більше?] AND <[Менше?], а на перетині

цього запису і поля Посада — такий текст [Назва посади?].

4. Зберігаємо й виконуємо запит3. На екрані з'явиться вікно Введіть значення параметра (рис. 6).

Рис. 6. Етап виконання запиту зі змінними критеріями

5. Вводим значення критерію (400) і натискаємо кнопку ОК. Система запросить увести значення *Менше*. Вводим 800, потім — назву посади *диспетчер* і натискаємо кнопку ОК. У результаті на екрані з'являться дані про працівника Носко М. Г. Якщо ще раз запустити запит3 й увести, наприклад, 300, 800 і слово *диспетчер*, то на екрані з'являться дані про працівників Вараву А. А., Зорю І. І. і Носка М. Г.



Запити, за допомогою яких у результуючий набір записів виводяться не тільки поля таблиць, але й поля, у яких містяться обчислювальні значення на основі даних полів, називають **запитами з полями, що обчислюються**.

Наприклад, на основі даних таблиці ПРАЦІВНИКИ можна підрахувати коефіцієнт доплати до зарплати, який залежить від стажу й кількості годин робочого тижня. Припустимо, що посадовий оклад працівника визначається на основі 36-годинного робочого тижня, а за кожну додаткову годину нараховується один відсоток від зарплати, тобто відсоток доплати обчислюється за виразом (Тиждень – 36), де Тиждень — це кількість реальних робочих годин на тиждень. Крім того, нараховується

коефіцієнт надбавок до зарплати за стаж: за кожний рік після 5-річного стажу зарплата збільшується на один відсоток, тобто цей коефіцієнт обчислюється за формулою: (Стаж – 5).

Отже, сумарний коефіцієнт надбавки можна підрахувати за формулою:

Відсоток = (Стаж – 5) + (Тиждень – 36).

Так, для працівника Носка М. Г. відсоток надбавки до зарплати дорівнює:
 $(15 - 5) + (42 - 36) = 16 \%$.

Приклад 2. На основі таблиці ПРАЦІВНИКИ створити запит з іменем запит4, за яким будуть виводитися всі записи таблиці з полями Прізвище, Стаж, Тиждень і Відсоток (значення поля Відсоток обчислюється за виразом, наведеним вище).

Правила створення запиту такого типу принципово не відрізняються від правил створення запитів, описаних у прикладі 1.

1. У таблицю конструктора запитів у запис Поле переносимо імена полів Прізвище, Стаж і Тиждень таблиці ПРАЦІВНИКИ, а в наступне поле вводимо вираз:

Відсоток: $([\text{Стаж}] - 5) + ([\text{Тиждень}] - 36)$.

Імена полів, які входять у вираз, наводяться у квадратних дужках.

- Установлюємо в записі Сортування поля Прізвище значення *За зростанням* для того, щоб прізвища виводилися в алфавітному порядку.
- Зберігаємо запит з іменем запит4. Після виконання запиту має з'явитися результат, зображений на рис. 7. Закриваємо запит4.

Прізвище	Стаж	Тиждень	Відсоток
Ахтирко В.В.	25	48	32
Варава А.А.	10	36	5
Грач П.О.	16	46	21
Зоря І.І.	22	48	29
Мезез Ф.Б.	17	40	16
Носко М.Г.	15	21	-5
Сороко Я.М.	5	46	10
*	0	0	

Рис. 7. Результат виконання запиту з полем, що обчислюється



У системі Access є вбудовані функції, що дають змогу узагальнити дані деяких полів. Такими функціями, зокрема, є: **Count** (Кількість), **Sum** (Сума), **Avg** (Середнє), **Max/Min** (максимум/мінімум). Запити з використанням таких функцій можна створювати за допомогою конструктора різними способами.

Розглянемо приклад створення запиту з функціями з використанням кнопки Підсумки, яка розташована на вкладці Конструктор у групі Відобразити або приховати.

Приклад 3. Створити запит з іменем запит5, за яким у таблиці ПРАЦІВНИКИ серед працівників зі стажем понад 5 років і тривалістю робочого тижня більше 36 годин відшукується найстарший рік (народження) працівника.

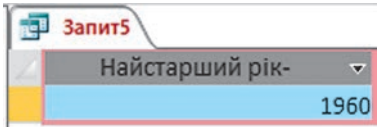
1. Звичайним способом відкриваємо порожній бланк запиту для таблиці ПРАЦІВНИКИ (для цього слід активізувати вкладку Створити, натиснути кнопку Макет запиту, виділити таблицю ПРАЦІВНИКИ, виконати команду Додати і натиснути кнопку Закрити).
2. У запис Поле таблиці конструктора запитів переносимо поля Стаж і Тиждень і знімаємо для них прапорці у записі Відображення. Натискаємо кнопку Підсумки в групі Відображення або приховання.

Поле:	Стаж	Тиждень	Найстарший рік-: ((Народження))
Таблиця:	ПРАЦІВНИКИ	ПРАЦІВНИКИ	
Підсумок:	Вираз	Вираз	Min
Сортування:			
Відображення:	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Критерії:	>5	>36	
Або:			

Рис. 8. Будова запиту з функцією

У результаті в таблиці конструктора запиту з'явиться новий запис Підсумок. У цьому записі для поля Стаж потрібно встановити значення *Вираз*, у запис Критерії ввести: > 5, а у полі Тиждень також встановити значення *Вираз* (для цього клацнути це поле, натиснути кнопку трикутника й у списку, що розкриється, вибрати *Вираз*) і ввести в запис Критерії: > 36.

3. У наступному полі запису Поле вводимо: *Найстарший рік-:([Народження])*, а в запису Підсумок цього поля вибираємо функцію Min і встановлюємо прапорці Відображення (рис. 8).
4. Зберігаємо й виконуємо запит5. Результат виконання запиту наведено на рис. 9.



Запит5	Найстарший рік-	1960
--------	-----------------	------

Рис. 9. Результат виконання запиту з функцією

? Запитання для перевірки знань

1. Які запити називають запитом з параметрами?
2. Які запити називають запитом з полями, що обчислюються?
3. Поясніть порядок створення запиту з параметрами.
4. Які функції використовуються для узагальнення значень полів?
5. Поясніть порядок створення запитів із функціями.
6. Наведіть приклад запиту з полями, що обчислюються.

Завдання для самостійного виконання

1. Створіть запит з іменем запит103, за допомогою якого можна вибрати прізвища учнів, які мають середній бал успішності навчання більше 9. Результуючі записи повинні містити поля Прізвище і Навчання таблиці УСПИХИ ДРУЗІВ і поле Телефон таблиці МОЇ ДРУЗІ.
2. Створіть запит з іменем запит104, за допомогою якого з БД MyBaza можна відшукати наймолодшого серед учнів, який має середній бал успішності більше 9.

4.4.3. Запити на змінення



Чи можна, наприклад, з двох таблиць отримати нову таблицю? Як, на вашу думку, це можна зробити?



Порядок створення запити на змінення даних у таблиці суттєво не відрізняється від розглянутих раніше. Принципова різниця полягає в тому, що в групі **Тип запити** натискають кнопку **Тип запити: оновлення**. Правила створення запитів на вилучення також принципово не відрізняються від правил створення інших запитів на змінення.



Запит на змінення — це запит, за допомогою якого в таблиці вносяться зміни.

Існують різні типи запитів на змінення: запити на створення таблиці, вилучення записів, оновлення записів і додавання нових записів.

Розглянемо лише запит на створення таблиці. Запити на змінення на панелі переходу відображаються знаком оклику. Такі запити необхідно використовувати обережно, щоб не втратити дані в таблицях.

За запитом на створення таблиці вибираються дані з однієї або кількох таблиць і з них формується нова таблиця. Ця таблиця може бути розміщена як у поточній, так і в іншій БД, ім'я якої вказується в процесі створення запити цього типу. Нова таблиця не має зв'язку з тими таблицями, з яких вона створена. Тому дані в ній автоматично не оновлюються в разі зміни в таблицях-джерелах.



Приклад. Розробити запит з іменем запитб, за яким на основі даних таблиць ФІРМИ і ПРАЦІВНИКИ створюється нова таблиця з іменем ДОДАТКОВА, у якій містяться особисті номери й прізвища працівників тих фірм, де кількість працівників перевищує 500.

1. У режимі конструктора створюємо запит на вибірку. У таблицю конструктора запити з таблиці ФІРМИ переносимо поле Працівники, а з таблиці ПРАЦІВНИКИ — поля Справа і Прізвище. У запис Критерії поля Працівники вводимо умову > 500 і знімаємо прапорець Відображення цього поля.

2. Виконуємо створений запит і переконуємося, що формується правильний результат. У режимі Конструктор перетворюємо запит на вибірку в запит на створення таблиці. Для цього на вкладці Конструктор у групі Тип запити слід натиснути кнопку Створення таблиці. З'явиться вікно Створити таблицю.

3. У поле Ім'я таблиці вводимо ім'я таблиці ДОДАТКОВА й вмикаємо перемикач Поточна база даних, оскільки таблиця надалі буде збережена у відкритій базі corporation (рис. 10). Для збереження виконаних налаштувань таблиці слід натиснути кнопку ОК.

Рис. 10. Етап виконання запити на створення нової таблиці

4. Зберігаємо запит з іменем запит6.
5. Виконуємо створений запит.
З'явиться вікно, зображене на рис. 11.
6. Для збереження таблиці в поточній БД натискаємо кнопку Так. В області переходів з'явиться ім'я таблиці ДОДАТКОВА.
7. Закриваємо запит6, викликаємо таблицю ДОДАТКОВА і переконуємося, що вона сформована правильно.
Вміст цієї таблиці зображено на рис. 12.

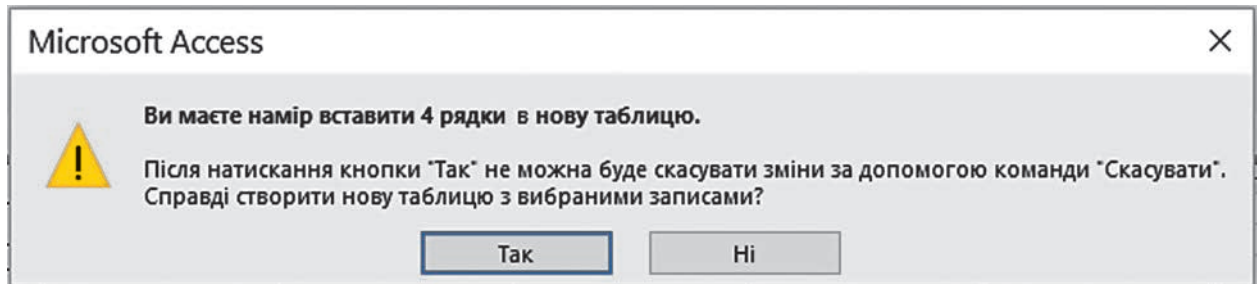


Рис. 11. Вікно підтвердження вставлення записів у таблицю

ДОДАТКОВА	
Справа	Прізвище
1140	Сороко Я.М.
1075	Мерез Ф.Б.
1206	Грач П.О.
1005	Носко М.Г.
*	

Запис: 5 з 5 Без фільтра Пошук

Рис. 12. Вміст створеної таблиці ДОДАТКОВА

? Запитання для перевірки знань

- 1 Які функції виконують запити на змінення?
- 2 У якій базі даних можна розмістити нову створену таблицю?
- 3 Чому в новій створеній таблиці не можна автоматично оновлювати дані?
- 4 Поясніть алгоритм створення запиту для створення нової таблиці.
- 5 Поясніть особливості створення запиту на додавання.

📁 Завдання для самостійного виконання

- 1 Створіть запит з іменем **запит105**, за яким на основі таблиць **МОЇ ДРУЗІ** й **УСПІХИ ДРУЗІВ** створюється нова таблиця з іменем **НОВА1**, у якій містяться поля **Прізвище**, **Ім'я**, **Телефон**, **Навчання** тих учнів, які займаються музикою.
- 2 Створіть запит з іменем **запит106**, за яким на основі таблиць **МОЇ ДРУЗІ** й **УСПІХИ ДРУЗІВ** створюється нова таблиця з іменем **НОВА2**, у якій містяться поля **Прізвища**, **Адреса** і **Телефон** тих учнів, які займаються спортом і мають середній бал успішності не менше 10.

4.5. Форми

Форма — це об'єкт бази даних, який використовується як для введення даних у таблицю, так і для їх виведення. Найчастіше форми для введення даних розробляються для таблиць, а джерелом даних форм для виведення є запити.

4.5.1. Засоби створення форм та елементи їх керування



Запит є досить потужним об'єктом опрацювання даних у БД. Чому, на вашу думку, в Access 2016 використовуються ще й форми?

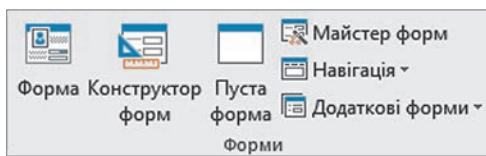


Рис. 1. Засоби для створення форм

У системі Access 2016 є різні засоби для створення форм, які містяться у групі Форми вкладки Створення (рис. 1).

Найпотужнішим засобом для створення форм є конструктор форми, решта — прості засоби. Вони забезпечують створення форми фактично одним натисненням відповідної кнопки.

Розглянемо створення форм за допомогою двох простих засобів: Форма і Розділена форма.

- Засіб Форма. За допомогою цього засобу можна створити форму на основі таблиці або запиту.

Приклад 1. Створити форму з іменем форма1 для таблиці ФІРМИ.

1. Виберемо таблицю ФІРМИ (клацнувши це ім'я на панелі Усі об'єкти).
2. На вкладці Створення у групі Форми натиснемо кнопку Форма. У результаті форма набуде вигляду, наведеного на рис. 2.

Як бачимо, на формі виведено перший запис таблиці ФІРМИ, тобто фірми 12, а під нею — усі записи таблиці ПРАЦІВНИКИ з прізвищами працівників, які працюють на цій фірмі. Це обумовлено тим, що між вказаними таблицями встановлено зв'язок.

3. Збережемо форму з іменем форма1. Для цього можна скористатися командою Зберегти контекстного меню цієї форми.

Справа	Прізвище	Посада	Народженн	Освіта	Стаж	Тиждень
1020	Ахтирко В.В.	аналітик	1960	вища	25	48
1232	Варава А.А.	диспетчер	1975	середня спеці	10	36
1230	Зоря І.І.	диспетчер	1961	середня	22	48
*	0		0		0	0

Рис. 2. Форма для таблиці ФІРМИ, створена засобом Форма

- Засіб Розділена форма. Цей засіб знаходиться у меню кнопки Додаткові форми. За його допомогою можна створювати форму також на основі таблиці або запиту, виділеного (або відкритого) на панелі переходів. За цим засобом одночасно подаються всі записи таблиці або запиту, а також один запис, який є виділеним.

Приклад 2. Створити і зберегти форму з іменем форма2 для запиту2 за допомогою засобу Розділена форма. Результат виконання форми для запиту подано на рис. 3.

Можна виділити будь-який запис, розташований у нижній частині форми, і редагувати його в її верхній частині.

Фірма	Телефон	Прізвище	Посада
12	350-621	Варава А.А.	диспетчер
12	350-621	Зоря І.І.	диспетчер
14	241-750	Мерез Ф.Б.	диспетчер
13	400-502	Носко М.Г.	диспетчер

Рис. 3. Результат виконання форми для запиту

У системі Access 2016 є великий арсенал елементів керування їхніх властивостей для налаштування форм і оформлення їх зовнішнього вигляду. Для виведення складу елементів керування формою слід відкрити форму в режимі конструктора і натиснути кнопку Елементи керування. Відкриється склад елементів (рис. 4).

Назва елементів з'являється після встановлення на них вказівника миші. У табл. 1 наведено призначення деяких елементів керування, які застосовуються найчастіше.

Таблиця 1. ЕЛЕМЕНТИ КЕРУВАННЯ ФОРМИ ТА ЇХ ПРИЗНАЧЕННЯ

Тип	Призначення
Текстове поле	Виводить дані на екран і дозволяє вводити і редагувати дані з клавіатури
Підпис	Виводить текст на екран
Кнопка	Натискання на малюнку призводить до виконання певної команди (наприклад, виконується макрос)
Прямокутник	Зазвичай використовується для виділення елементів керування
Лінія	Пряма лінія певного кольору й товщини. Використовується для розмежування частин форми
Прапорець	Елемент керування з двома станами (включений або знятий)
Перемикач	Має два стани. Відрізняється від прапорця тим, що в групі може бути встановлений тільки один прапорець

Щоб додати елемент керування на форму можна, наприклад, у режимі конструктора клацнути необхідний елемент, а потім встановити курсор у потрібне місце форми і натиснути кнопку миші.

Вікно властивостей має п'ять вкладок: Формат, Дані, Подія, Інші, Усі.

Вкладка Усі містить повний набір параметрів елемента керування, а інші вкладки — скорочений перелік. Ці самі елементи керування використовуються й у звітах.

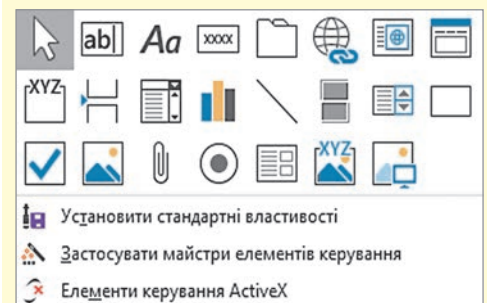


Рис. 4. Елементи керування форми



Кожний елемент керування має власний набір властивостей. Для виведення на екран його треба виділити й натиснути кнопку **Аркуш властивостей**. Але цей набір рідко використовується в повному обсязі.

**Запитання для перевірки знань**

- 1 Для чого призначені форми?
- 2 На основі яких об'єктів створюються форми?
- 3 Які способи створення форм існують?
- 4 Поясніть сутність засобу **Форма**.
- 5 Поясніть сутність засобу **Розділити форму**.
- 6 Назвіть основні елементи керування формою.
- 7 Як викликати **Аркуш властивостей** елементів керування формою?

**Завдання для самостійного виконання**

- 1 Створіть форму з іменем **форма101** для таблиці **МОЇ ДРУЗІ** за допомогою засобу **Форма**.
- 2 Створіть форму з іменем **форма102** для таблиці **УСПІХИ ДРУЗІВ** за допомогою засобу **Розділена форма**.

4.5.2. Створення форм за допомогою конструктора форм

У чому полягають, на вашу думку, переваги і недоліки засобів **Форма** і **Розділена форма**?

Розглянемо **способи створення форми за допомогою конструктора форм** на прикладі.



Приклад. Створити для бази даних corporation форму з іменем форма3 на основі таблиці ФІРМИ з полями: Фірма, Адреса, Телефон, Директор. Перед першими трьома полями розмістити текст *Адреса фірм*, а перед останнім — *Керівник*.

1. Виділяємо в області переходів таблицю ФІРМИ, активізуємо вкладку Створення й у групі Форми виконуємо команду Конструктор форм. На екрані з'явиться порожня форма, а також відкриється вкладка Конструктор. Для налаштування конфігурації й зовнішнього вигляду форми слід відкрити її контекстне меню (кляцанням правої кнопки миші усередині поля форми), зміст якого зображено на рис. 5.
2. Розміщуємо на формі колонтигули та знімаємо сітку, для чого виконуємо команди Сітка і Колонтитули. Вибираємо на смузі елемент керування Підпис і в області верхнього колонтитула вводимо назву *ФІРМИ МІСТА*, після чого натискаємо клавішу Enter. Стандартним способом можна змінити місце розташування поля заголовка форми.
3. Аналогічно вводимо у верхню частину області Подобиці текст *Адреса фірм*. Якщо на екрані відсутнє поле Список полів, слід виконати команду Додавання наявних полів у групі Знаряддя. Має відкритися поле Список полів, у якому треба виконати команду Відобразити всі таблиці. Ліворуч від назви таблиці ФІРМИ вмикається перемикач з ознакою (+). Відкриється список полів цієї таблиці.
4. Переміщуємо назву поля Фірма в те місце області даних, куди необхідно помістити це поле. У результаті на формі з'явиться група з двох пов'язаних елементів — власне поле введення і підпис для нього. Аналогічно переміщуємо у бланк форми поля Адреса і Телефон і розміщуємо їх одне під одним.
5. За аналогією під назвами трьох перших полів за допомогою елемента Підпис розміщуємо ще одне поле, у яке вводимо текст *Керівник*. Нижче від цього поля розміщуємо поле Директор. Елементи на формі можна розмістити, наприклад, так, як показано на рис. 6. Після цього можна закрити вікно Список полів.

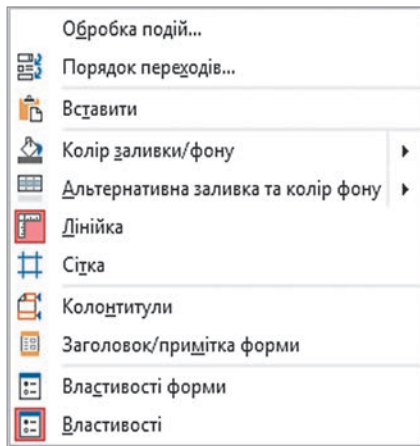


Рис. 5. Контекстне меню форми

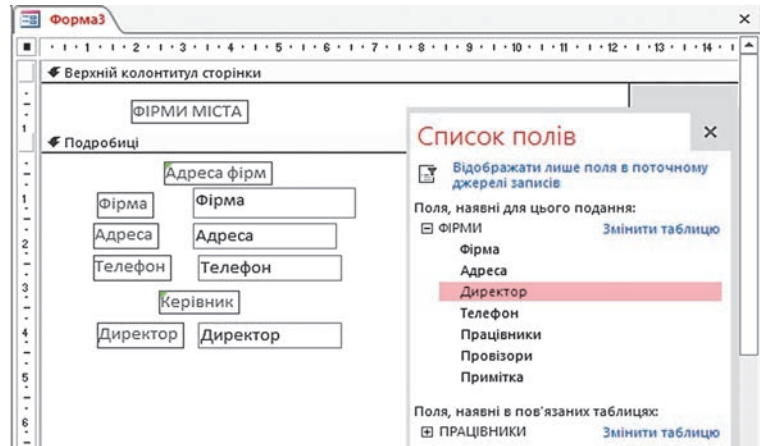


Рис. 6. Форма, створена за допомогою конструктора форм

6. Зберігаємо створену форму з іменем форма3. Назва форми з'явиться в області переходів.

Зовнішній вигляд створеної форми можна змінювати. Наприклад, щоб змінити колір заголовка форми, потрібно відкрити контекстне меню заголовка форми, встановити курсор на команді Колір заливки/фону. Відкриється вікно кольорів, у якому можна вибрати необхідний колір. Аналогічно можна змінити колір інших областей форми, а також колір тексту й фону надписів *Адреси фірм* і *Керівник*.

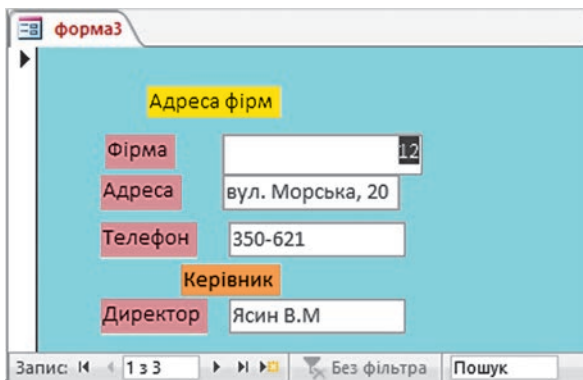


Рис. 7. Результат виконання форми в режимі форми

7. Після виконаних налаштувань зберігаємо форму, закриваємо її, а потім знову відкриваємо в режимі форми. Результат зазначених дій показано на рис. 7.

У наведеному прикладі форма виведена з одним (першим) записом. За допомогою кнопок навігації, розміщених у нижній частині форми, можна перейти до будь-якого запису. Результат зазначених дій показано на рис. 8.

Зазначимо, що правила створення форми в режимі конструктора на основі запиту практично нічим не відрізняються від правил її створення на основі таблиць.

Фірма	Адреса	Телефон	Директор
12	вул. Морська, 20	350-621	Ясин В.М
13	вул. Заводська, 15	400-502	Бойко М.І.
14	вул. Зоряна, 3	241-750	Зімін І.Ф.
*	0		

Рис. 8. Результат виконання форми в режимі подання форми

Форму бажано проектувати так, щоб потім не потрібно було вносити зміни до її структури. У процесі модифікації форми можна виконувати такі дії: змінювати параметри відображення форми; змінювати властивості форми; додавати нові елементи керування; змінювати властивості елементів керування; видаляти елементи керування.



Запитання для перевірки знань

- 1 Як можна змінити колір фону й тексту на формі?
- 2 Які основні дії можна виконати за допомогою контекстного меню форми?
- 3 Як викликати у вікні форми список полів таблиці?
- 4 Які основні операції можна виконати в процесі модифікації форми?
- 5 Поясніть порядок створення форми за допомогою конструктора форм.



Завдання для самостійного виконання

- 1 За допомогою конструктора форм створіть на основі таблиці **МОЇ ДРУЗІ** форму з іменем **форма103** з полями **Прізвище**, **Ім'я**, **День народження**, **Телефон**, **Адреса**. Перед першими трьома полями розмістіть текст *Основні дані*, а перед рештою — *Контакти*.
- 2 Використайте елементи керування Access 2016 для зміни зовнішнього вигляду **форми103**, надавши формі привабливого зовнішнього вигляду.

4.5.3. Використання форм для введення й редагування даних



Форми забезпечують наочне і якісне відображення даних. Але їх застосовують ще й для введення даних. Яка необхідність у цьому?

Форма є зручним засобом відображення даних, які зберігаються в таблицях або формуються запитом. Водночас форму зручно використовувати для введення й редагування даних. Розглянемо на прикладах способи *введення*, *редагування* й *видалення даних*.

- **Уведення нових записів**



Уведені у форму нові записи будуть автоматично додані до таблиці, на базі якої створено форму.

Приклад 1. Відкрити форму1 в режимі форми й увести в бланк цієї форми такі дані: Фірма — 15; Адреса — *вул. Лісова, 55*; Директор — *Івасько І. І.*; Телефон — *313-222*; Працівники — *500*; Провізори — *12*.

1. Відкриваємо форму1 на панелі навігації.

2. Натискаємо кнопку Створити запис і вводимо перелічені раніше дані нового запису.

3. Зберігаємо форму, закриваємо її, відкриваємо таблицю ФІРМИ. Результат додавання даних наведено на рис. 9.

Фірма	Адреса	Директор	Телефон	Працівники	Провізори	Примітка
12	вул. Морська, 20	Ясин В.М	350-621	400	30	
13	вул. Заводська, 15	Бойко М.І.	400-502	700	50	Директор-заслужений працівник
14	вул. Зоряна, 3	Зімін І.Ф.	241-750	800	60	Директор-новопризначений
15	вул. Лісова, 55	Івасько І.І.	313-222	500	12	
* 0				0	0	

Запис: 5 з 5 | Без фільтра | Пошук

Рис. 9. Таблиця ФІРМИ після додавання нового запису

• Редагування записів

Для редагування записів у таблиці за допомогою форми відкриваємо її в режимі форми, знаходимо цей запис у формі та вносимо зміни в поля даних. Пошук запису можна здійснити за допомогою кнопок на панелі навігації.

• Видалення записів

Слід враховувати, що для зв'язаних таблиць видалити запис не завжди можливо. Якщо, наприклад, запис не має підлеглих записів у зв'язаних таблицях або дозволено каскадне вилучення записів, то видалення можливе. У такому випадку відкриваємо вікно, у якому подано текст із проханням підтвердити вилучення. Для підтвердження вилучення натискаємо кнопку Так, інакше буде виведено повідомлення про неможливість вилучення. У такому випадку спочатку треба вилучити з інших таблиць усі записи, які пов'язані з даним записом. Можна також на час видалення записів спочатку вилучити зв'язки між таблицями, а після вилучення їх відновити.

Загальний порядок **видалення записів** із таблиці за допомогою форми такий: слід знайти у формі запис, який потрібно видалити, у групі Записи вкладки Основне відкрити меню кнопки Видалити і виконати команду Видалити запис.

Приклад 2.

1. Відкриваємо форму1 в режимі форми.
2. Знаходимо запис із прізвищем Івасько І. І. Уводимо в поле Адреса нову адресу: *Подільська, 7*.
3. Зберігаємо й закриваємо форму. Після цього відкриваємо таблицю ФІРМИ і переконаємося, що зміни внесені.

Приклад 2. Видалити за допомогою форми1 з таблиці ФІРМИ запис, де є прізвище Івасько І. І.

1. Відкриваємо форму1 в режимі форми, знаходимо запис із прізвищем Івасько І. І.
2. У групі Записи встановлюємо курсор на трикутнику, розташованому біля кнопки Видалити.
3. Натискаємо кнопку миші й у меню, що відкриється, виконуємо команду Видалити запис. У результаті на екрані з'явиться вікно, у якому потрібно ввести підтвердження на видалення поточного запису.
4. Для збереження форми зі змінами натискаємо кнопку Зберегти, закриваємо форму1, відкриваємо таблицю ФІРМИ і переконаємося, що запис дійсно вилучено.



Запитання для перевірки знань

- 1 Як видалити записи за допомогою форм?
- 2 Як здійснити редагування записів за допомогою форм?
- 3 Обґрунтуйте, чому форми доцільно використовувати для введення даних.
- 4 У якому режимі необхідно відкрити форму для редагування записів?
- 5 Як додати нові записи за допомогою форм?



Завдання для самостійного виконання

- 1 Додайте в таблицю **МОЇ ДРУЗІ** новий запис за допомогою **форми101**. Видаліть цей запис. Переконайтеся в правильному виконанні операції.
- 2 Виконайте в таблиці **МОЇ ДРУЗІ** редагування записів за допомогою **форми101**. Відновіть первинний вміст таблиці **МОЇ ДРУЗІ**.

4.6. Звіти



Звіт є специфічним об'єктом системи Access 2016. Які дії можуть виконуватися за допомогою звітів?

Джерелом даних для звіту є таблиці або записи, отримані в результаті виконання запити. До звіту можна включати малюнки, діаграми, таблиці, у ньому можна обчислювати відповідні значення для згрупованих записів (наприклад, за професією тощо).



Звіт — це об'єкт бази даних, призначений для виведення даних у зручному для користувача вигляді.

На початку створення звіту необхідно чітко визначити структуру й зовнішній вигляд документа, який необхідно отримати. Бажано його структуру подати на папері.

Для кожної БД може бути створено кілька звітів різної форми і з різними даними. Наприклад, один звіт може бути створено для бухгалтерії, інший — для відділу кадрів тощо. Засоби для створення звітів розміщені в групі Звіти вкладки Створити (рис. 1).

Найпростішим є засіб Звіт. За його допомогою звіт створюється автоматично, фактично натисненням однієї кнопки. Але отриманий у цьому випадку звіт містить усі поля однієї таблиці. Якщо необхідно вибрати дані з кількох таблиць, то для цього потрібно розробити відповідний запит.

Найпотужнішим засобом для створення звіту є Конструктор звітів. Розглянемо методику створення звіту в режимі конструктора на прикладі.

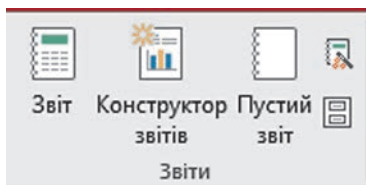


Рис. 1. Засоби для створення звіту

Приклад. Створити звіт на базі таблиць ФІРМИ і ПРАЦІВНИКИ. З таблиці ПРАЦІВНИКИ включити у звіт поля Прізвище, Посада і Стаж, а з таблиці ФІРМИ — поля Фірми і Директор. Згрупувати записи за значенням

поля Посада і впорядкувати їх за збільшенням стажу. У результаті виконання звіту необхідно отримати орієнтовно такий результат, як наведено на рис. 2.

ПРАЦІВНИКИ ФІРМ ЗА ПОСАДАМИ				
Фахівці за посадами <input type="text" value="аналітик"/>				
<input type="text" value="Грач П.О."/>	<input type="text" value="аналітик"/>	<input type="text" value="16"/>	<input type="text" value="14"/>	<input type="text" value="Зімін І.Ф."/>
<input type="text" value="Ахтирко В.В."/>	<input type="text" value="аналітик"/>	<input type="text" value="25"/>	<input type="text" value="12"/>	<input type="text" value="Ясин В.М."/>
Фахівці за посадами <input type="text" value="диспетчер"/>				
<input type="text" value="Варава А.А."/>	<input type="text" value="диспетчер"/>	<input type="text" value="10"/>	<input type="text" value="12"/>	<input type="text" value="Ясин В.М."/>
<input type="text" value="Носко М.Г."/>	<input type="text" value="диспетчер"/>	<input type="text" value="15"/>	<input type="text" value="13"/>	<input type="text" value="Бойко М.І."/>
<input type="text" value="Мерез Ф.Б."/>	<input type="text" value="диспетчер"/>	<input type="text" value="17"/>	<input type="text" value="14"/>	<input type="text" value="Зімін І.Ф."/>
<input type="text" value="Зоря І.І."/>	<input type="text" value="диспетчер"/>	<input type="text" value="22"/>	<input type="text" value="12"/>	<input type="text" value="Ясин В.М."/>
Фахівці за посадами <input type="text" value="експерт"/>				
<input type="text" value="Сороко Я.М."/>	<input type="text" value="експерт"/>	<input type="text" value="5"/>	<input type="text" value="14"/>	<input type="text" value="Зімін І.Ф."/>

Рис. 2. Орієнтовний результат виконання звіту для таблиць ФІРМИ і ПРАЦІВНИКИ

1. Вибираємо назви таблиць ФІРМИ і ПРАЦІВНИКИ. Активуємо вкладку Створення і в групі Звіти натискаємо кнопку Конструктор звітів. На екрані з'явиться порожній бланк звіту, а на смузі відкриється вкладка Конструктор.
2. Уводимо заголовок звіту. Для цього натискаємо кнопку Підпис у групі Елементи керування, встановлюємо курсор в область Верхній колонтитул сторінки і клацаємо кнопку миші. З'явиться текстове поле, у яке потрібно ввести текст заголовка звіту, наприклад *ПРАЦІВНИКИ ФІРМ ЗА ПОСАДАМИ*, і натиснути клавішу Enter. Далі натискаємо кнопку Додавання наявних полів у групі Знаряддя і кнопку Відобразити всі таблиці.
3. Перетягуємо з вікна Список полів у розділ Подобиці поля Прізвище, Посада, Стаж таблиці ПРАЦІВНИКИ і поля Фірми і Директор таблиці ФІРМИ. Під час перетягування кожного поля з'являються два зв'язаних поля: ліворуч — назва поля, а праворуч — поле для виведення даних. Кожне ліве поле можна вилучити, для чого достатньо вибрати його й натиснути клавішу Delete.
4. Згрупуємо записи, які будуть включені до звіту за посадою, тобто за значенням

поля Посада. Для цього на вкладці Конструктор натискаємо кнопку Групування та сортування, що міститься в групі Групування та підсумки...

У результаті в нижній частині екрана з'явиться область Групування, сортування і підсумок, у якій слід виконати команду Додати групу. Висвітяться всі імена вибраних полів.

Вибираємо поле Посада, за яким виконується групування записів. У результаті в області звіту з'явиться нова область Верхній колонтитул. У цю область перетягуємо поле Посада, натискаємо кнопку Підпис і ліворуч від цього поля вводимо текст *Фахівці за посадами*, після чого натискаємо клавішу Enter.

5. Установлюємо сортування згрупованих записів за значенням поля Стаж, для чого спочатку натискаємо кнопку Додати сортування, а потім — поле Стаж. Виконані налаштування відобразяться у вікні, поданому на [рис. 3](#).
6. Закриваємо область Групування, сортування й підсумок. Остаточний вміст звіту, створеного в режимі конструктора, який відповідає поставленим умовам, подано на [рис. 4](#).
7. Збережемо звіт2, виконаємо його в режимі Подання звіту і переконаємося, що видається вміст (див. [рис. 2](#)).

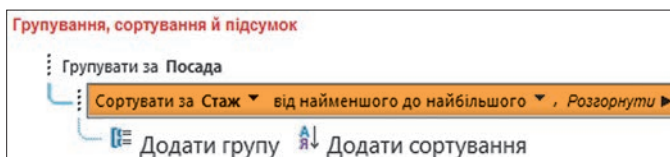


Рис. 3. Діалогове вікно для групування й сортування записів

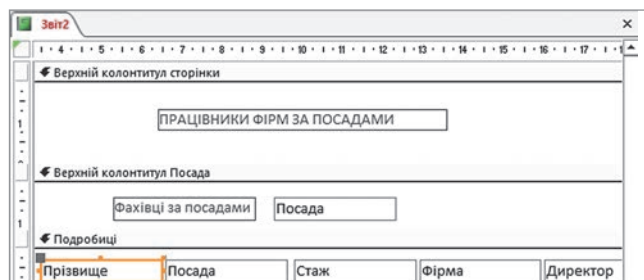


Рис. 4. Варіант структури звіту в режимі конструктора



До звіту можна додати ще деякі елементи керування, розмістити елементи по-іншому, виділити окремі елементи керування іншим шрифтом, кольором, товщиною ліній тощо. Усі ці операції принципово не відрізняються від аналогічних дій, що виконувалися в процесі розробки форм.



Щоб модифікувати звіти, можна виконати такі операції:

- змінити параметри звіту
- вибрати новий стиль форматування звіту
- додати нові елементи керування
- змінити властивості елементів керування
- вилучити елементи керування

Щоб надати звіту більш естетичного вигляду, можна змінити колір елементів керування або використати стандартні стилі оформлення звітів. Для цього потрібно відкрити звіт у режимі розмітки, на вкладці **Формат** розгорнути меню кнопки **Форматування елементів керування** й вибрати бажаний колір заливки фігур, контур фігури, а в групі **Шрифт** — шрифт тексту.

Можна також підрахувати кількість записів у кожній групі й у звіті в цілому. Для цього в режимі розмітки на вкладці Конструктор у групі Групування та підсумки потрібно відкрити меню кнопки Підсумки й виконати команду Кількість записів. Приклад такого звіту наведено на рис. 5.

ПРАЦІВНИКИ ФІРМ ЗА ПОСАДАМИ				
Фахівці за посадами аналітик				
Грач П.О.	аналітик	16	14	Зімін І.Ф.
Ахтирко В.В.	аналітик	25	12	Ясин В.М
		2		
Фахівці за посадами диспетчер				
Варава А.А.	диспетчер	10	12	Ясин В.М
Носко М.Г.	диспетчер	15	13	Бойко М.І.
Мерез Ф.Б.	диспетчер	17	14	Зімін І.Ф.
Зоря І.І.	диспетчер	22	12	Ясин В.М
		4		
Фахівці за посадами експерт				
Сороко Я.М.	експерт	5	14	Зімін І.Ф.
		1		
		7		

Рис. 5. Вміст звіту з підрахунком кількості записів

До звіту можна додати ще деякі елементи керування, розділити елементи по-іншому, виділити окремі елементи керування іншим шрифтом, кольором, товщиною ліній тощо. Усі ці операції принципово не відрізняються від аналогічних дій, що виконувалися в процесі розробки форм.



Запитання для перевірки знань

- 1 Як виконати групування й сортування записів у звіті?
- 2 Як підрахувати записи в групах звіту?
- 3 Поясніть порядок створення звіту за допомогою конструктора звітів.



Завдання для самостійного виконання

- 1 У режимі конструктора створіть звіт101 для таблиць **МОЇ ДРУЗИ** і **УСПІХИ ДРУЗІВ**. Із таблиці **МОЇ ДРУЗИ** включіть у звіт поля **Прізвище** й **Адреса**, а з таблиці **УСПІХИ ДРУЗІВ** — поля **Навчання** й **Музика**.
- 2 Використайте елементи керування системи Access 2016 для самостійного налаштування зовнішнього вигляду звіту101.

Практична робота № 7

Тема. Робота з готовою базою даних. Введення і редагування даних.

Завдання: відкрити готову базу даних, ввести та відредагувати дані за наведеним планом; створити нову базу даних, установити зв'язок між таблицями, ввести та відредагувати дані.

Обладнання: комп'ютер зі встановленою програмою система управління базами даних; готова база даних.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Запустіть програму Access. Відкрийте готову БД corporation і виконайте в ній такі дії.
 - 1) Видаліть із таблиці ПРАЦІВНИКИ два будь-які записи.
 - 2) Уведіть у таблицю новий запис і відредагуйте його.
 - 3) Виконайте створені раніше запити, звіти й форми та переконайтеся в правильності отриманих результатів.
 - 4) Відновіть попередній зміст таблиці ПРАЦІВНИКИ й переконайтеся в правильності виконання інших об'єктів цієї БД.
- 2 Створіть нову БД за таким планом.
 - 1) Створіть БД з іменем nova з таблицями АВТОРИ і КНИГИ та збережіть її в кореневому каталозі диска D:.
 - 2) Установіть зв'язок між таблицями.
 - 3) Перевірте правильність введених даних у таблицях і наявність зв'язку між таблицями.

Зробіть висновки: які трудовитрати на розроблення об'єктів бази даних; як створити нову базу даних, установити зв'язок між таблицями, ввести та відредагувати дані.

Таблиця АВТОРИ

Прізвище	Посада	Стаж	Кількість видань
Бех Н. А.	доцент	20	7
Невін К. К.	професор	15	15
Носов П. А.	інженер	17	1
Сокіл І. І.	доцент	12	4

Таблиця КНИГИ

Інвентарний №	Назва	Автор	Рік	Ціна	Тираж	Реалізовано
20100	Фізика	Бех Н. А.	2000	75	3000	2100
20101	Механіка	Сокіл І. І.	2005	84	1000	800
20102	Інформатика	Бех Н. А.	2008	60	4000	3500
20103	Інформатика	Невін К. К.	2006	70	2500	2000
20104	Механіка	Пека П. А.	2008	75	5000	4000
20105	Інформатика	Бех Н. А.	2010	90	3000	1900
20106	Фізика	Сокіл І. І.	2006	75	3000	2400

Практична робота № 8

Тема. Пошук даних у базі даних. Використання запитів і фільтрів.

Завдання: здійснити в БД пошук даних за зазначеними умовами.

Обладнання: комп'ютер зі встановленою програмою система управління базами даних; база даних, створена на попередньому уроці.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Запустіть програму Access. Відкрийте створену БД нова.
- 2 Знайдіть у таблиці КНИГИ всі записи, що містять книги з роком видання після 2000 року, ціна яких перевищує 75.
- 3 Знайдіть у таблиці АВТОРИ всі записи з авторами, які є доцентами та професорами.
- 4 Упорядкуйте записи таблиці КНИГИ за значенням полів Рік і Автор. Відкрийте цю таблицю в режимі таблиці й переконайтеся, що записи впорядковано правильно.
- 5 Із таблиці КНИГИ виберіть книжки з інформатики й фізики, видані після 2005 року.
- 6 Упорядкуйте записи таблиці АВТОРИ за значеннями полів Прізвище і Видання. Переконайтеся, що впорядкування виконано правильно.
- 7 Переіменуйте таблицю КНИГИ в таблицю ВИДАННЯ. Після цього відновіть попереднє ім'я таблиці.
- **Зробіть висновки:** яка ефективність фільтрів і запитів для пошуку даних; яким із них ви надаєте перевагу.

Практична робота № 9

Тема. Запити та їх використання.

Завдання: створити запити для знаходження в БД даних за певними критеріями відбору.

Обладнання: комп'ютер зі встановленою програмою система управління базами даних; база даних, створена на попередньому уроці.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Запустіть програму Access. Відкрийте БД нова і створіть запит на основі таблиці КНИГИ з полями Назва, Автор, Рік і Реалізовано, за допомогою якого виводяться дані про книжки, що мають реалізацію понад 2000 примірників. Збережіть запит з іменем запит200 і виконайте його. Проаналізуйте результат і закрийте запит.
- 2 Створіть запит на основі таблиці КНИГИ, за допомогою якого здійснюється пошук книг з інформатики автора Беха Н. А. Збережіть запит з іменем запит201 і виконайте його. Перевірте результат і закрийте запит.
- 3 Відкрийте запит200 в режимі конструктора й додайте до нього поле Тираж. Збережіть цей запит з іменем запит202. Виконайте його і перевірте результат.
- 4 Створіть запит, за яким виводяться дані про книги з фізики й механіки. Створіть

запит на основі таблиць АВТОРИ і КНИГИ. До запиту включіть поля Назва і Автор таблиці КНИГИ і поля Посада і Стаж таблиці АВТОРИ. Упорядкуйте записи в алфавітному порядку прізвищ авторів. Збережіть запит з іменем запит203, виконайте його і перевірте результат. Закрийте запит.

- 5 Створіть запит, за допомогою якого підраховується загальний тираж і загальна

реалізація книг із назвами «Фізика» і «Механіка». Збережіть запит з іменем запит204 і виконайте його.

- 6 Самостійно сформулюйте і створіть завдання для розроблення запиту для таблиці АВТОРИ і КНИГИ.

Зробіть висновки: яка ефективність запитів різних типів; яким з них ви надаєте перевагу.

Практична робота № 10

Тема. Створення звітів.

Завдання: створити звіти для аналізу даних в БД за певними умовами.

Обладнання: комп'ютер зі встановленою програмою система управління базами даних; база даних, створена на попередніх уроках.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Запустіть програму Access. Створіть на основі запиту200 звіт. Згрупуйте записи за значенням поля Автор. Записи в кожній групі упорядкуйте за значенням поля Рік. Підрахуйте для кожного автора загальну кількість реалізованих книжок. Виконайте звіт і переконайтеся в правильному результаті.
- 2 Модифікуйте звіт, створений у п. 1. Згрупуйте записи не за значенням поля Автор, а за назвами книг. Підрахуйте кількість книг кожного автора та загальну кількість реалізованих книг усіх авторів.
- 3 Збережіть звіт, виконайте його і перевірте результат.
- 4 За допомогою конструктора звітів створіть звіт на основі запиту203. Згрупуйте

записи за значенням поля Посада. Підрахуйте кількість книжок, виданих кожним автором. Уведіть назву звіту й назву кожної групи. Збережіть звіт, виконайте його і перевірте результат.

- 5 Модифікуйте звіт. Установіть нові елементи, виберіть їхні властивості й стиль оформлення. Надайте звіту естетичного вигляду. Переконайтеся, що він дійсно має привабливий вигляд. Видаліть додані елементи й закрийте звіт.
- 6 Самостійно сформулюйте завдання для розроблення звіту на основі запиту203. Створіть і виконайте цей звіт.

Зробіть висновки: яка доцільність і ефективність використання звітів.

Розділ 5. МЕРЕЖЕВІ ТЕХНОЛОГІЇ

5.1. Класифікація комп'ютерних мереж



Однією із суттєвих причин, які прискорили появу комп'ютерів, була потреба в розв'язуванні дуже широкого спектра задач. Пригадайте сфери використання комп'ютерів.



У 1961 році американський інженер українського походження Леонард Клейнрок запропонував ідею пакетної комутації, яка наразі є основою передавання даних мережею. А в 1964 році виклав основні принципи та розробив теорію.



Комп'ютерна мережа — сукупність пристроїв, з'єднаних каналами передавання даних, для спільного користування апаратними, програмними та інформаційними ресурсами під керуванням спеціального програмного забезпечення.

Пригадаємо, який пристрій у мережі називається вузлом (точкою з'єднання).



Вузол мережі (англ. *Node*) — пристрій, з'єднаний з іншими пристроями через мережу. Вузлами можуть бути комп'ютери, мобільні телефони, кишенькові комп'ютери та спеціальні мережні пристрої.

Призначенням комп'ютерних мереж є забезпечення:

- швидкого обміну даними між окремими комп'ютерами мережі;
- спільного використання комп'ютерних програм і даних;
- спільної роботи користувачів над проектами;
- віддаленого керування комп'ютерами;
- спільного доступу до периферійних пристроїв (принтерів, сканерів, зовнішньої пам'яті);
- спільного доступу до інформаційних ресурсів.

У комп'ютерній мережі комп'ютери можуть виконувати різні функції.



Комп'ютер, який керує розподілом ресурсів мережі, називають **сервером** (від англ. *server* — той, хто подає); комп'ютери, які користуються ресурсами мережі, називають **клієнтами**, або **робочими станціями**.

Залежно від завдань, які виконують комп'ютери, мережі розрізняють за територією, типом операційної системи, розподілом функцій, інфраструктурою та місцем розташування технічних засобів, які входять у мережу, та ін.



Американського вченого Джозефа К. Р. Ліклайдера часто називають духовним батьком Інтернету. У 1962 році в низці статей він виклав свою концепцію «Галактичної мережі» — прообраз сучасного Інтернету.

Схему класифікації комп'ютерних мереж за різними ознаками наведено на рис. 1.

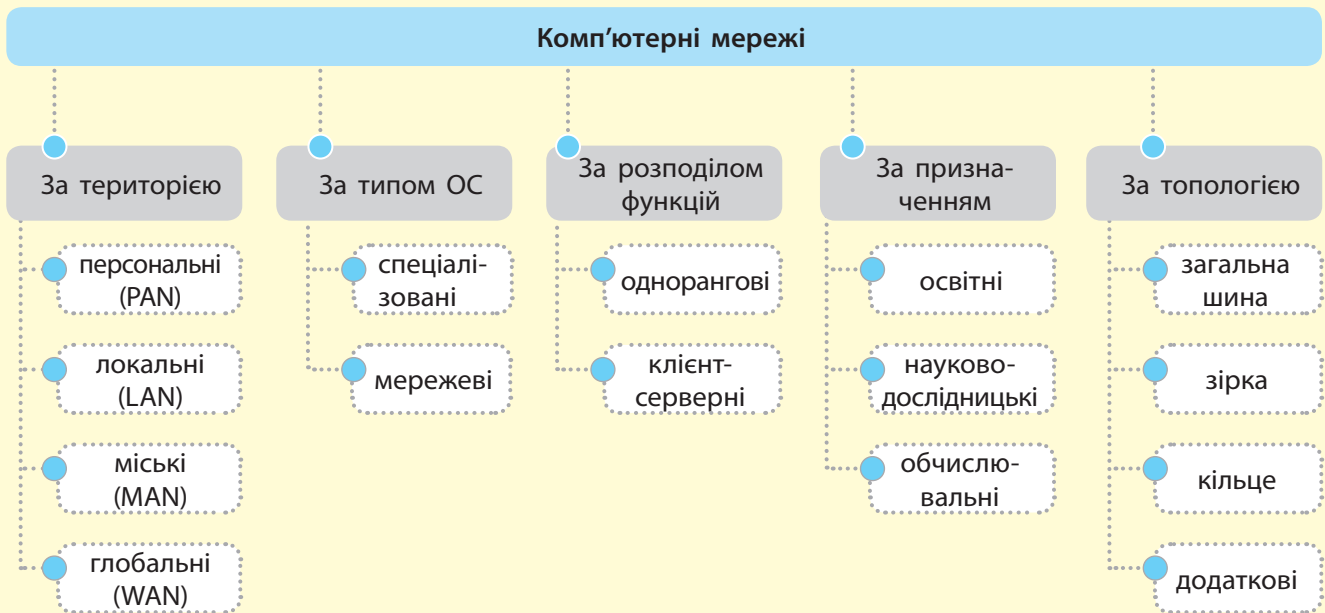


Рис. 1. Класифікація комп'ютерних мереж

Розглянемо класифікацію комп'ютерних мереж детально. За територією мережі поділяються таким чином.

- **Персональні** (PAN, від англ. *Personal Area Network* — мережа особистого простору, персональна мережа) — мережі для взаємодії пристроїв, що належать одній людині та об'єднують її власні електронні пристрої: персональні комп'ютери, ноутбуки, планшети, смартфони, комунікатори.
- **Локальні** (LAN, від англ. *Local Area Network* — мережа локального простору) — з'єднують пристрої, розташовані на порівняно невеликій відстані один від одного, зазвичай у межах однієї або кількох сусідніх будівель, наприклад мережа навчального закладу.
- **Міські, регіональні** (MAN, від англ. *Metropolitan Area Network* — мережа міського простору) — обласні й національні мережі. Приміром, www.ukr.net — це українська національна мережа.
- **Глобальні** (WAN, від англ. *Wide Area Network* — мережа широкого простору) — об'єднують комп'ютерні мережі. Найвідомішою глобальною мережею є Інтернет.

Сучасні операційні системи (ОС) поділяються на спеціалізовані та мережеві (рис. 2).

Спеціалізовані ОС призначені для роботи з мережевим обладнанням певної компанії. Так, Cisco IOS (англ. *Internetwork Operating System* — міжмережева ОС) працює виключно з маршрутизаторами й комутаторами компанії Cisco, а ZyNOS — ОС компанії ZyXEL, працює з маршрутизаторами Prestige. (ОС, що забезпечують роботу у мережі, буде розглянуто в § 5.2.)



Набуває популярності нова комп'ютерна мережа — «натільна» (BAN, від англ. *body area network*). Її пристрої можуть бути зафіксовані на поверхні тіла (фітнес-трекери, розумні браслети) або з'єднані з пристроями, які люди носять у кишнях, на руці або імплантовані в тіло (технологія перебуває у початковій стадії дослідження).



Рис. 2. Логотипи мережевих ОС



Рис. 3. Топологія «загальна шина»

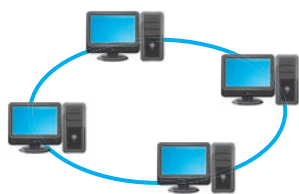


Рис. 4. Топологія «кільце»

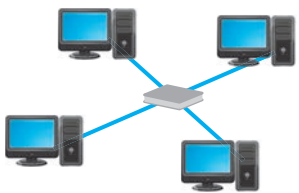


Рис. 5. Топологія «зірка»

За розподілом функцій між комп'ютерами мережі поділяють на *однорангові* й *клієнт-серверні*. (Детальніше такі мережі будуть розглянуті в § 5.2.)

Комп'ютерні мережі поділяються також за топологією.



Фізичне розташування вузлів мережі один відносно одного та способи їхнього з'єднання лініями зв'язку називають **топологією**.

Існують три базові топології («загальна шина», «кільце», «зірка») та додаткові, що є модифікацією або поєднанням базових, наприклад топологію «дерево» можна розглядати як комбінацію декількох «зірок».

Кожна топологія накладає певні вимоги.

Топологія «загальна шина» передбачає використання одного кабелю, до якого під'єднуються всі комп'ютери мережі (рис. 3). Надіслане з будь-якого комп'ютера мережі повідомлення поширюється на всі інші комп'ютери мережі. Кожний із них перевіряє, кому адресовано повідомлення. Опрацьовує повідомлення лише той комп'ютер, якому воно адресоване. Комп'ютери можуть передавати дані лише послідовно, оскільки лінія зв'язку одна і спільна. Всі комп'ютери мають рівні права, все обладнання є ідентичним.

Топологія «кільце» — топологія, в якій кожен комп'ютер з'єднано лініями зв'язку лише з двома іншими (рис. 4): від одного він тільки отримує інформацію, а іншому тільки передає. Комп'ютери в «кільці» не є повністю рівноправними: одні обов'язково отримують інформацію від комп'ютера, який надсилає повідомлення в цей момент, раніше, а інші — пізніше.

У **топології «зірка»** всі комп'ютери мережі приєднано до центрального вузла (рис. 5), через який весь обмін інформацією йде від одного комп'ютера до іншого. Як центральний вузол можуть виступати або концентратор чи комутатор — таку топологію називають пасивною «зіркою», або потужний комп'ютер, на який покладається дуже велике навантаження, — таку топологію називають активною «зіркою».



Запитання для перевірки знань

- 1 Що таке комп'ютерна мережа?
- 2 Яке призначення комп'ютерних мереж? Наведіть приклади.
- 3 Поясніть поняття «клієнт», «сервер».
- 4 Що означають скорочення LAN, PAN, WAN?
- 5 Які ви знаєте види комп'ютерних мереж? Поясніть, чим вони різняться.
- 6 Які мережі називають клієнт-серверними?
- 7 Які мережі називають одноранговими?
- 8 Які мережі називають локальними? Наведіть приклади таких мереж.
- 9 Назвіть переваги й недоліки базових топологій.
- 10 Зробіть повідомлення про одну з додаткових топологій.
- 11 Запропонуйте власну схему класифікації за територією, яка більш наочно демонструє відмінність таких комп'ютерних мереж, як LAN, PAN, MAN.

5.2. Апаратне і програмне забезпечення мереж

Пригадайте, які складові має інформаційна система.

Будь-яка комп'ютерна мережа має *апаратні* та *програмні* ресурси.

Апаратні ресурси мережі, або **апаратне забезпечення**, — це під'єднані до мережі комп'ютери, канали передавання даних (канали зв'язку) та мережеве обладнання. (Зі складовими апаратного забезпечення ми ознайомимось у § 5.3 і 5.4.)

Системне програмне забезпечення — це службові та системні програми, драйвери пристроїв мережі, мережеві компоненти операційної системи, які використовують для обміну даними між комп'ютерами в мережі.

Як ви вже знаєте, за розподілом функцій розрізняють *однорангові* та *клієнт-серверні* мережі (рис. 1). Розглянемо програмне забезпечення цих мереж.

В **однорангових мережах** (рис. 2) усі комп'ютери є рівноправними, кожен комп'ютер функціонує і як клієнт, і як сервер. Кожен користувач сам визначає, до яких ресурсів його комп'ютера матимуть доступ інші користувачі. Тобто у комп'ютерів немає чіткої спеціалізації щодо підтримки тієї чи іншої мережевої служби.

Для створення однорангових мереж не потрібне спеціальне мережеве програмне забезпечення. Для їх роботи достатньо набору мережевих функцій, який мають усі сучасні операційні системи.

У **клієнт-серверних мережах** (рис. 3) на відміну від однорангових одні комп'ютери, зазвичай найпотужніші, відіграють роль *серверів* і забезпечують надання певних послуг (служб), а інші, робочі станції, є *клієнтами*, тобто споживачами цих послуг. Тому, щоб підкреслити цю різницю, такі мережі називають також **багаторанговими**.

Для забезпечення нормального функціонування клієнт-серверних мереж потрібне спеціальне програмне забезпечення, яке складається з двох частин: *серверної* та *клієнтської*.

На серверах установлюють серверну частину: серверні операційні системи, що здатні опрацювати запити великої кількості користувачів, спеціальні програмні засоби, необхідні для функціонування клієнт-серверних мереж. Такі засоби забезпечують блокування файлів та записів, що необхідно для їх спільного використання, супровід довідників імен мережевих ресурсів, опрацювання запитів віддаленого доступу до файлової системи та бази даних сервера, керування чергою запитів віддалених користувачів до мережевих периферійних пристроїв (принтерів, сканерів, факсів тощо).

На робочих станціях установлюють клієнтську частину, яка надає можливість користувачу звертатися до ресурсів



Комп'ютерна мережа

- однорангова (комп'ютери виступають одночасно і як сервери, і як клієнти)
- клієнт-серверна (мережеві ресурси сконцентровані на одному комп'ютері)

Рис. 1. Типи мереж за розподілом функцій

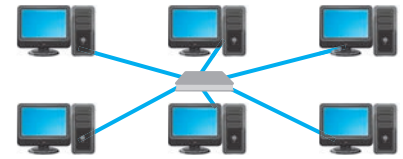


Рис. 2. Однорангова мережа

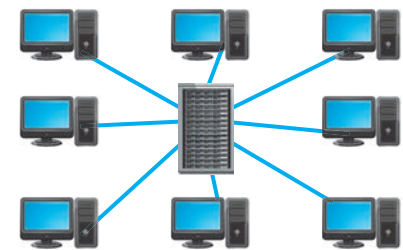


Рис. 3. Клієнт-серверна мережа



29 жовтня 1969 року вперше було проведено сеанс передавання даних на відстань 640 км. Цей день вважається офіційним днем народження Інтернету.



Рис. 4. Логотипи серверних ОС

сервера. Вона забезпечує процес розпізнавання запитів користувачів та перенаправлення відповідей від сервера.

Наведену технологію побудови програмного забезпечення називають клієнт-серверною.

Серед серверних операційних систем виділяють Unix, Linux (GNU/Linux) та Windows Server (рис. 4), кожна з яких має певні особливості.

Unix — сімейство багатозадачних та багатокористувацьких операційних систем. Це одна з перших серверних операційних систем, яка використовувалась на так званих мейнфреймах — великих багатокористувацьких комп'ютерах та корпоративних серверах.

Linux — загальна назва Unix-подібних операційних систем, які базуються на ядрі Linux і системних бібліотеках GNU. Linux є операційною системою з відкритим кодом.



Особливістю сучасних операційних систем є служби (наприклад, Windows SharePoint Services), які можуть створювати веб-вузли для здійснення обміну даними, що особливо важливо для забезпечення роботи з файлами у режимі спільної роботи з даними.

Крім того, на серверах встановлюють спеціалізоване програмне забезпечення, потрібне для підтримки роботи певної служби або кількох служб. У клієнт-серверних мережах підтримку мережевих служб здійснюють сервери, причому вони можуть мати різну спеціалізацію (виконувати різні функції) (рис. 5).

Windows Server наразі є найбільш поширеною операційною системою для серверів. Комп'ютер, на якому встановлено таку операційну систему, може виконувати функції:

- файлового сервера;
- сервера служби веб-додатків;
- сервера терміналів;
- поштового сервера;
- сервера віддаленого доступу;
- служби DNS (доменних імен);
- служби каталогів;
- сервера потоків мультимедіа та ін.

Наразі всі сучасні ОС мають вбудовані мережеві функції.

Сервер

- Веб-сервер — забезпечує доступ до сайтів
- Файловий сервер — зберігає великі обсяги інформації
- Поштовий сервер — підтримує роботу з електронною поштою
- Сервер резервного копіювання — відповідає за безпечне зберігання інформації

Рис. 5. Спеціалізація серверів у клієнт-серверній мережі



Запитання для перевірки знань

- 1 Що таке вузол мережі?
- 2 Який комп'ютер може бути сервером?
- 3 Опишіть принцип побудови програмного забезпечення у клієнт-серверній мережі.
- 4 За яких умов роботи краще налаштувати однорангову мережу?
- 5 Які функції може виконувати сервер у клієнт-серверних мережах?
- 6 Яке програмне забезпечення потрібне для функціонування однорангової мережі?
- 7 Знайдіть відомості в Інтернеті й зробіть повідомлення про серверні операційні системи.

5.3. Канали зв'язку

Пригадайте, що таке комп'ютерна мережа.

Усі апаратні складові мережі налаштовують таким чином, щоб несправність окремих комп'ютерів, фізичний вихід із ладу або від'єднання окремих ділянок мережі не впливали на функціонування мережі. Канали зв'язку забезпечують з'єднання приймача та передавача (рис. 1).

За способом передавання даних мережі поділяють на *кабельні (дротові)* і *бездротові*.



Мережу називають **кабельною** (дротовою), якщо середовищем передавання даних є кабель. У такому середовищі дані передаються електричними або оптичними сигналами.

На сучасному етапі розвитку комп'ютерних мереж використовують такі типи кабелів: кручена пара, коаксіальний та оптоволоконний кабелі.

Кручена пара — це декілька пар скручених мідних дротів у кольоровій пластиковій ізоляції (рис. 2). Пучки скручених пар дротів захищає зовнішнє обплетення. Такий кабель використовують у телефонному зв'язку та в більшості мереж Ethernet (від англ. *ether* — ефір і *net* — мережа) — це пакетна технологія передачі даних, яка застосовується при побудові комп'ютерних мереж. Залежно від типу кабелю максимальна відстань передавання даних без підсилення сигналу становить від 15 до 100 м, а швидкість передавання даних може досягати 100 Гбіт/с.

Коаксіальний кабель — це кабель із ізольованою мідною жилою, оточеною металевією оболонкою-екраном (рис. 3). Такий кабель використовують для під'єднання комп'ютерів до мережі та поширення сигналів телебачення. Максимальна відстань передавання даних без підсилення сигналу становить 500 м, максимальна швидкість передавання даних може досягати 10 Мбіт/с.

Оптоволоконний кабель — це скляна або пластикова нитка, що використовується для перенесення світла за допомогою повного внутрішнього відображення (рис. 4).

Структура оптоволоконного кабелю схожа на структуру коаксіального кабелю. Але замість центрального мідного дроту в такому кабелі використовується тонке (діаметром близько 1–10 мкм) оптоволокно, а замість внутрішньої ізоляції — скляна або пластикова оболонка, що не дозволяє світлу виходити за межі оптоволокна.

Застосування цього кабелю дозволяє реалізувати найшвидший на сьогодні спосіб передавання даних. Відстань передавання даних без підсилення сигналу становить 50 км, а швидкість передавання даних сягає від 10 Гбіт/с до 4–8 Тбіт/с.



Канали зв'язку

- середовище передавання (головна характеристика)
- швидкість передавання даних (кількість бітів даних, які передаються за секунду)
- максимальна відстань передавання даних (без спотворення сигналу)

Рис. 1. Характеристики каналів зв'язку



Рис. 2. Кручена пара



Рис. 3. Коаксіальний кабель



Рис. 4. Оптоволоконний кабель

Триває розробка оптоволоконного кабелю нового типу, який складається з багатожильних оптичних світловодів. На кабелі таких жил сім. Передавання даних кабелем завдовжки кілометр через 50 різних потоків дозволило досягти рекордного показника швидкості в 255 Тбіт/с. Це набагато швидше нинішнього стандарту в 4–8 Тбіт/с.



Рис. 5. Логотипи стандартів бездротових мереж

Технологія Bluetooth (англ. *blue tooth* — синій зуб) отримала назву від імені данського короля Харольда Синьозубого (X ст.) на прізвище Об'єднувач.

Поява мобільного Інтернету стала можливою завдяки розвитку бездротових мереж (англ. *wireless*).



Бездротовою називають мережу, в якій дані передаються радіосигналами.

Стандартами бездротових мереж є Wi-Fi, LTE, Bluetooth, WiMAX та ін. (рис. 5).

Wi-Fi (від англ. *Wireless Fidelity* — бездротова точність) — стандарт для обладнання бездротових мереж і торгова марка консорціуму Wi-Fi Alliance, до якого входять найбільші виробники комп'ютерного устаткування та обладнання Wi-Fi. Мобільні пристрої (телефони, смартфони, ноутбуки), які оснащені приймачами-передавачами Wi-Fi, можна під'єднувати до локальної бездротової мережі та Інтернету. Wi-Fi має обмежений радіус дії (до 300 м) та швидкість передавання даних до 300 Мбіт/с.

WiMAX, Mobile WiMAX, Mobile-Fi — технології бездротових мереж, які призначено для використання разом із технологією Wi-Fi (або замість неї) із метою розширення бездротових мереж. Зокрема, мережа WiMAX забезпечує кращий доступ до Інтернету, ніж Wi-Fi, і має більшу площу покриття. Наразі максимальна відстань становить 80 км, а швидкість передавання даних зросла до 75 Мбіт/с.

LTE (від англ. *Long-Term Evolution* — довготривалий розвиток, часто позначається як 4G LTE) — стандарт бездротової високошвидкісної передачі даних для мобільних телефонів і інших терміналів, що працюють із даними. Швидкість передавання даних може досягати 360 Мбіт/с на відстань до 100 км (наразі використовується швидкість 1 Мбіт/с на відстані до 13 км).

Bluetooth — стандарт для бездротових персональних мереж. Технологія забезпечує обмін даними між кишеньковими та стаціонарними комп'ютерами, мобільними телефонами, ноутбуками, принтерами, цифровими фотокамерами тощо. Зв'язок підтримується на відстані від 10 м до 100 м (відстань залежить від наявності перешкод) зі швидкістю до 25 Мбіт/с.



Запитання для перевірки знань

- 1 Назвіть характеристики каналів зв'язку.
- 2 На яку відстань можливо передавати дані кабельними мережами без спотворення якості сигналу?
- 3 Для чого використовується кручена пара?
- 4 Де використовують технологію Bluetooth?
- 5 Яка з технологій бездротових мереж забезпечує максимальну швидкість передавання даних?
- 6 Який фізичний принцип лежить в основі оптоволоконного каналу передавання даних?
- 7 Складіть таблицю основних характеристик кабельних та бездротових мереж.
- 8 Знайдіть і підготуйте повідомлення про категорії крученої пари.
- 9 Ви хочете переглянути фільм у режимі онлайн. Яким середовищем передавання даних ви скористаєтесь?
- 10 Який стандарт бездротового зв'язку використовується у мобільних телефонах?

5.4. Мережеві пристрої

Які ви знаєте мережеві пристрої?



За даними аналітичних агентств (IDC), лідером продаж мережевого устаткування у 2016 році традиційно є компанія Cisco Systems (56%), друге місце посідає Huawei (7%), на третьому місці Hewlett Packard (5%). До п'ятірки лідерів також входять Arista Networks та Juniper.



MAC-адреса — унікальний ідентифікатор, який присвоюється більшості мережевого обладнання.

Унікальність MAC-адреси забезпечується тим, що кожен виробник отримує в координуючому комітеті IEEE Registration Authority свій діапазон із шістнадцяти мільйонів (2^{24}) адрес. За трьома старшими байтами MAC-адреси можна визначити виробника.

Мережеві пристрої забезпечують транспортування даних між пристроями користувача. Вони подовжують і об'єднують кабельні з'єднання, перетворюють дані з одного формату в інший і керують передаванням даних.

До мережевих пристроїв належать повторювачі, концентратори, мости, комутатори і маршрутизатори. Розглянемо їхні особливості.

Повторювач (англ. *repeater*) — це пристрій (рис. 1), призначений для підсилення мережевих сигналів, що дозволяє передавати їх середовищем на більшу відстань. Причому повторювач не переглядає іншу інформацію, яка міститься в пакеті.

Концентратор (англ. *hub* — центр уваги) — це один із видів мережевих пристроїв (рис. 2), які можна встановлювати на рівні доступу мережі Ethernet. На ньому є кілька портів для під'єднання вузлів до мережі.

Концентратор не визначає, якому вузлу призначено конкретне повідомлення. Він просто приймає електронні сигнали одного порту й відтворює їх для всіх інших портів. Для передавання та отримання повідомлень всі порти концентратора Ethernet під'єднуються до одного і того самого каналу.



Повторювачі й концентратори мають схожі характеристики, тому концентратори часто називають багатопортовими повторювачами (англ. *multiport repeater*). Різниця між ними полягає лише в кількості кабелів, приєднаних до пристрою. Якщо повторювач має тільки два порти, концентратор зазвичай має від 4 до 24 і більше портів.



Рис. 1. Повторювач



Рис. 2. Концентратор



Рис. 3. Міст



Рис. 4. Комутатор



Рис. 5. Маршрутизатор



Рис. 6. Вбудована мережева карта



Рис. 7. Мережевий інтерфейс для бездротових мереж

Міст (англ. *bridge* — міст) — це пристрій (рис. 3), призначений для фільтрування потоків даних у локальній мережі для того, щоб локалізувати передавання даних і разом із тим зберегти можливість зв'язку з іншими частинами мережі для перенаправлення туди потоків даних. Міст збирає інформацію про те, на якому порті знаходиться конкретна MAC-адреса, і приймає рішення про пересилку даних на підставі відповідного списку MAC-адрес. Мости здійснюють фільтрацію потоків даних, базуючись лише на MAC-адресі вузлів, тому можуть швидко пересилати дані.

Комутатор (англ. *switch* — перемикач) — це пристрій, який можна назвати «розумним» концентратором (рис. 4), тому що він передає дані тільки безпосередньо отримувачу.

У таблиці комутатора, яка називається таблицею MAC-адрес, міститься список активних портів і MAC-адреси під'єднаних до них вузлів. Коли вузли обмінюються повідомленнями, комутатор перевіряє, чи є в таблиці ця MAC-адреса. Якщо так, комутатор установлює між портом джерела й портом приймача тимчасове з'єднання — канал. Цей новий канал є каналом, по якому два вузли обмінюються даними.

Інші вузли, під'єднані до комутатора, працюють на інших смугах пропускання каналу і не приймають чужі повідомлення. Для кожного нового з'єднання між вузлами створюється новий канал.

Маршрутизатори (англ. *router*) — це пристрої об'єднаних мереж, які пересилають пакети між мережами на основі адрес. Маршрутизатор (рис. 5) здатний вибирати найкращий шлях у мережі для переданих даних.

Маршрутизатор може приймати рішення на основі мережевих адрес замість використання індивідуальних MAC-адрес другого рівня. Завдяки цій здатності маршрутизатори стали основною магістраллю глобальної мережі Internet.

Мережева карта (мережевий інтерфейс) — пристрій, яким оснащують комп'ютер для під'єднання до мережі за допомогою мережевого кабелю чи радіоканалу (рис. 6). Для під'єднання до бездротової мережі можуть використовуватися не тільки мережеві карти, а й спеціальні пристрої.

Мережеві інтерфейси виготовляють у вигляді плат або окремих пристроїв — для бездротових мереж (рис. 7). Тип мережевого інтерфейсу має відповідати типу середовища передавання.



Запитання для перевірки знань

- 1 Що таке MAC-адреса?
- 2 Хто надає MAC-адресу? Чи можна її змінити?
- 3 Поясніть різницю між повторювачем та концентратором.
- 4 Коли є потреба у повторювачі?
- 5 Чи можна використовувати замість маршрутизатора комутатор? Якщо так, то в якому випадку?
- 6 У вас вдома налаштовано персональну локальну мережу, в яку під'єднано ноутбук, планшет, смартфон і стаціонарний персональний комп'ютер. Які мережеві пристрої для цього потрібні?

5.5. Адресація в мережах

Пригадайте, що локальна мережа може бути одноранговою або клієнт-серверною (з виділеним сервером).



В одноранговій мережі ресурси (файли, програми) можуть бути розміщені на різних комп'ютерах мережі. Щоб забезпечити до них доступ певній групі користувачів, комп'ютери програмно об'єднують у робочі групи.



Робоча група — це група комп'ютерів локальної мережі, користувачі якої виконують однотипні завдання й регулярно обмінюються даними.

В одній мережі фізично об'єднаних комп'ютерів може бути декілька робочих груп, програмно розмежованих одна від одної. Робочим групам надають імена, наприклад WorkGroup.

Пригадайте, що таке обліковий запис користувача. Які типи облікових записів ви знаєте?



Користувачі однорангової мережі зазвичай мають різні права доступу до її ресурсів. Права доступу до папок комп'ютера визначає користувач, який має на цьому комп'ютері обліковий запис адміністратора. Одні папки доступні всім користувачам, інші — тільки окремим.

Користувачам клієнт-серверної мережі права доступу надає адміністратор сервера. За допомогою спеціального програмного забезпечення створюється база клієнтів (комп'ютерів, облікових записів), кожному із них призначаються права доступу до ресурсів.

У мережах із виділеним сервером комп'ютери об'єднують у домени (від англ. *domain* — володіння).

Домену надають ім'я, наприклад Classroom_inf. У великій локальній мережі може бути декілька доменів.



Домен — це група комп'ютерів, що обслуговуються спільним сервером, який керує розподілом прав доступу користувачів до ресурсів мережі. Такий сервер називають контролером домену.

Домен — це логічне об'єднання комп'ютерів, облікових записів і ресурсів клієнт-серверної мережі під одним іменем. У клієнт-серверній мережі ресурси загального доступу, такі, як папки з файлами, мережеві програми-сервіси, мережеві принтери, зосереджуються на одному або декількох серверах.

У клієнт-серверних мережах облікові записи користувачів, а також інформація про інші об'єкти мережі зберігаються централізовано на сервері. Це дає змогу користувачам працювати з мережевими ресурсами на різних комп'ютерах під одним обліковим записом.

У мережі комп'ютери обмінюються даними, тобто передбачається, що дані будуть надходити від передавача до приймача. Це вимагає ідентифікації всіх комп'ютерів, тому кожен комп'ютер у мережі має унікальне ім'я.



На локальних комп'ютерах користувачі можуть мати окремі облікові записи. Приклад входу користувача в систему в клієнт-серверній мережі SCHOOL — ім'я сервера, на якому зареєстрований користувач, Pupil — ім'я користувача.

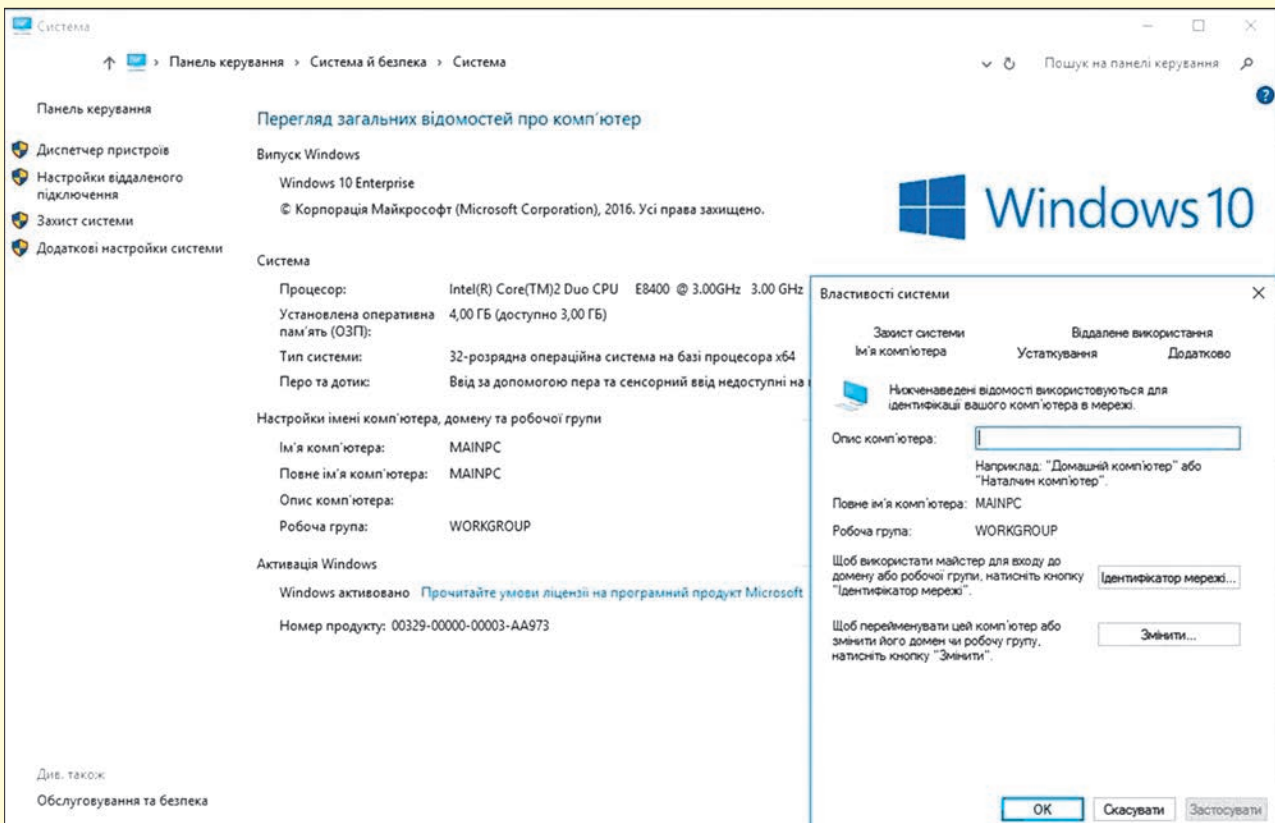
Приватна IP-адреса:

- 192.168.0.0/16 — цей блок надає IP-адреси в діапазоні від 192.168.0.1 до 192.168.255.254
- 172.16.0.0/12 — цей блок надає IP-адреси в діапазоні від 172.16.0.1 до 172.31.255.254
- 10.0.0.0/8 — цей блок надає IP-адреси в діапазоні від 10.0.0.1 до 10.255.255.254

Щоб переглянути імена, надані комп'ютеру та робочій групі або домену, необхідно відкрити контекстне меню об'єкта Мій комп'ютер та вибрати команду Властивості (див. [рисунок](#)).

Налаштування мережі здійснює адміністратор мережі — людина, яка відповідає за розподіл ресурсів та правильне функціонування мережі.

Комп'ютери локальної мережі можуть бути не під'єднані до Інтернету. В такому випадку використовуються так звані приватні IP-адреси, які вибираються із спеціально зарезервованих блоків адрес (192.168.0.0/16, 172.16.0.0/12 або 10.0.0.0/7). Більш детально це буде розглянуто в розділі 5.10.



Вкладка Ім'я комп'ютера вікна Властивості системи



Запитання для перевірки знань

- 1 Що називають робочою групою мережі?
- 2 Що називають доменом?
- 3 Як переглянути імена комп'ютера та робочої групи?
- 4 У чому полягає відмінність між робочою групою та доменом?
- 5 Наведіть переваги об'єднання комп'ютерів у робочу групу; домен.
- 6 Як відобразиться на роботі комп'ютерів локальної мережі вихід із ладу одного з комп'ютерів робочої групи?

5.6. Мережеві протоколи

Пригадайте, що основною задачею функціонування комп'ютерної мережі є обмін даними між її вузлами.

Розглянемо, яким чином здійснюється передавання даних мережею (рис. 1). Перед початком передавання всі дані поділяються на невеликі порції, до яких додають керуючу інформацію, а саме: адреси відправника й одержувача, порядковий номер порції. Така порція даних разом із керуючою інформацією називається **пакетом**.

Кожен пакет даних передається мережею на комп'ютер-одержувач, причому передавання може здійснюватися різними маршрутами. Якщо якийсь пакет не потрапив до адресата, його буде відправлено повторно. На комп'ютері-одержувачі відбувається процес з'єднання пакетів. Якщо інформація в якомусь із пакетів спотворилася, цей пакет також буде відправлено повторно.

У процесі обміну даними беруть участь і програмні, й апаратні засоби: комп'ютери, комунікаційні пристрої, мережеве обладнання та спеціальне програмне забезпечення. Здебільшого комп'ютери мають різну архітектуру, конфігурацію, операційні системи, тому найскладнішою частиною забезпечення надійного обміну даних є сумісність програмних та апаратних мережевих засобів.

Необхідною умовою для узгодженої взаємодії всіх учасників процесу обміну даними є дотримання певних правил, які називаються мережевими (комунікаційними) протоколами.

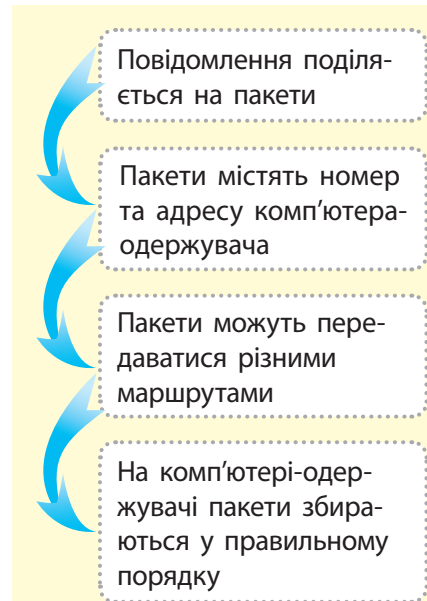


Рис. 1. Схема передавання даних мережею



Мережеві (комунікаційні) протоколи — це набір правил, за якими здійснюється обмін даними між пристроями комп'ютерних мереж.

У мережевих протоколах, зокрема, визначено: як дані поділяють на пакети; якими є сигнали підтвердження передавання; як будуть виявляти й опрацьовувати помилки передавання.



У переважній більшості комп'ютерних мереж обмін даними здійснюється на основі сімейства (стеку) протоколів TCP/IP (англ. *Transmission Control Protocol* — протокол керування передаванням; IP — *Internet Protocol* — міжмережевий протокол).

Протокол TCP забезпечує зв'язок між комп'ютерами та керує передаванням даних, визначає правила розбиття даних на пакети, доставку адресату та об'єднання пакетів в єдине ціле.

На основі **протоколу IP** здійснюється пересилання пакетів між комп'ютерами. Всі протоколи будуються за багаторівневим принципом. Протокол певного рівня визначає одне з технічних правил зв'язку.



1 січня 1983 року — офіційна дата народження Інтернету. Мережа ARPANET перейшла на новий протокол TCP/IP.



Вінтон Серф (ліворуч) та Роберт Кан — розробники стеку протоколів TCP/IP — вважаються батьками Інтернету. У 2004 році вони були нагороджені премією Т'юрінга.



Рис. 2. Рівні сімейства протоколів TCP/IP

У сімействі протоколів TCP/IP є чотири рівні (рис. 2):

- **прикладний** (верхній) — забезпечує виконання конкретних прикладних завдань;
- **транспортний** — забезпечує з'єднання між комп'ютером-відправником і комп'ютером-одержувачем та надійне передавання пакетів;
- **мережевий** — забезпечує взаємодію мереж різної архітектури;
- **канальний** (рівень доступу до середовища передавання) — описує спосіб кодування даних для передавання каналами зв'язку; перетворює IP-адреси у фізичні адреси мережі (MAC-адреси).

Розглянемо найпоширеніші протоколи *прикладного* рівня.

HTTP (*Hyper Text Transfer Protocol*) — це протокол передавання гіпертексту; його використовують під час пересилання веб-сторінок між комп'ютерами, під'єднаними до однієї мережі.

HTTPS (*Hypertext Transport Protocol Secure*) — це протокол, який забезпечує конфіденційність обміну даними між сайтом і комп'ютером користувача. Його використовують на сайтах, де вводиться й передається конфіденційна інформація (особисті дані користувачів, деталі доступу, реквізити платіжних карт), на сайтах з авторизацією, взаємодією з платіжними системами, поштовими сервісами. Захист інформації забезпечується за рахунок використання спеціальних криптографічних протоколів

FTP (*File Transfer Protocol*) — це протокол передавання файлів зі спеціального файлового сервера на комп'ютер користувача. Він надає можливість обмінюватися бінарними і текстовими файлами з будь-яким комп'ютером мережі. Встановивши зв'язок із віддаленим комп'ютером, користувач може скопіювати файл із віддаленого комп'ютера на свій або скопіювати файл зі свого комп'ютера на віддалений.

POP3 (*Post Office Protocol*) — це стандартний протокол поштового з'єднання. Сервери POP опрацьовують вхідну пошту, а протокол POP призначений для опрацювання запитів на отримання пошти від клієнтських поштових програм. При цьому всі електронні листи завантажуються користувачеві на комп'ютер. Саме на комп'ютері будуть проводитися усі подальші дії з листами.

Протокол POP3 унеможлиблює маніпулювання папками на сервері, вибірково отриманням частин повідомлення чи можливістю завантаження заголовків листів, тому було запропоновано протокол IMAP.

SMTP (*Simple Mail Transfer Protocol*) — це протокол, який задає набір правил для передавання пошти. Сервер SMTP повертає або підтвердження про прийом, або повідомлення про помилку, або вимогу додаткової інформації (рис. 3).

IMAP (англ. *Internet Message Access Protocol*) — це протокол прикладного рівня для доступу до електронної пошти. Він надає користувачеві великі можливості для роботи

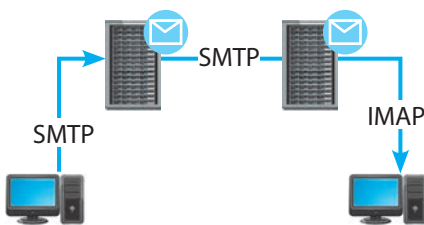


Рис. 3. Схема роботи протоколів електронної пошти IMAP і SMTP

з поштовими скриньками, що знаходяться на центральному сервері. Протокол підтримує роботу одночасно з декількома поштовими скриньками, надає доступ до однієї скриньки одночасно декільком клієнтам. Електронними листами можна маніпулювати з комп'ютера користувача (клієнта), наприклад, здійснювати пошук за ключовим словом без збереження пошти в локальній пам'яті.

TELNET — це протокол віддаленого доступу. Він надає можливість абонентові працювати з будь-яким комп'ютером, який знаходиться в мережі, як зі своїм власним, тобто запускати програми, змінювати режим роботи тощо. На практиці можливості обмежуються тим рівнем доступу, який заданий адміністратором віддаленої машини.

Далі розглянемо найпоширеніший стандарт *канального рівня*, про який ми вже згадували.

Ethernet — це сімейство технологій пакетного передавання даних для комп'ютерних мереж. Стандарти Ethernet визначають з'єднання й електричні сигнали на фізичному рівні. Оскільки наразі під'єднання відбувається через комутатори, то дані, які відправляються одним вузлом, доходять лише до адресата, внаслідок цього підвищується швидкість роботи та посилюється безпека мережі.

У 1978 році була розроблена модель OSI (англ. *Open Systems Interconnection Basic Reference Model*) (рис. 4), яка взаємодіє з сімейством протоколів TCP/IP.

Останнім часом зростає потреба в бездротовому з'єднанні. Існуючі протоколи передавання даних не розраховані на використання мобільного зв'язку і не враховують його особливостей. Тому було розроблено відкритий протокол WAP, упровадження якого дозволяє використовувати стільникові телефони для доступу в Інтернет.

WAP (*Wireless Application Protocol* — протокол бездротового доступу) ще називають *Wireless Access Protocol*. Остання назва краще відображає сутність WAP-технології, тому що її призначення полягає в забезпеченні доступу до Інтернету тільки за допомогою мобільного телефону.

Для того щоб дані, що містяться на веб-серверах, стали доступними, їх треба спеціально підготувати — переформатувати й подати у вигляді WAP-сторінок. Таким чином, під час застосування WAP-технологій інформація на серверах Інтернет буде подана у двох варіантах: у вигляді звичайних веб-сторінок та WAP-сторінок.



У 1984 р. Міжнародна організація зі стандартизації (ISO) представила індустріальний стандарт — модель взаємодії відкритих систем (*Open System Interconnection Reference Model* — OSI/RM). OSI є семирівневою логічною моделлю роботи мережі (рис. 4).



Рис. 4. Рівні моделі OSI



Запитання для перевірки знань

- 1 Опишіть призначення мережевих протоколів.
- 2 На основі якого набору протоколів здійснюється передавання даних у сучасних мережах?
- 3 Який протокол використовується під час передавання файлів?
- 4 Опишіть роботу протоколів електронної пошти.
- 5 Чим відрізняється стандарт Ethernet від WAP?
- 6 Знайдіть відомості про протокол UDP.
- 7 Знайдіть в Інтернеті відомості про модель OSI та зробіть повідомлення.

5.7. Призначення і структура Інтернету



Пригадайте, що вам відомо про Інтернет. Чому цю мережу називають глобальною?



Інтернет — всесвітня система взаємопов'язаних комп'ютерних мереж, що базуються на протоколах сімейства TCP/IP.



Інтернет (від англ. *Inter* — між, *network* — мережа, об'єднана мережа) називають також мережею мереж.

Кожні 10 років на сайті «The Opte Project» (<http://www.opte.org>) публікується мапа мережі Інтернет (рис. 2). Проект було створено для візуалізації всіх зв'язків усіх мереж в Інтернеті.

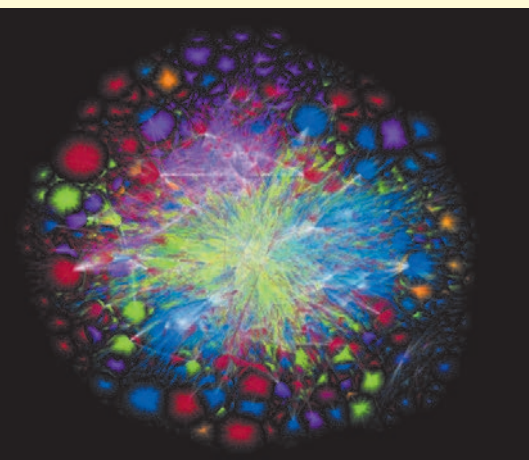


Рис. 2. На мапі мережі Інтернет (станом на 15 липня 2015 року) бекбони позначено білим кольором; лінії зв'язку: синім — у Північній Америці, зеленим — у Європі, червоним — в Азії, помаранчевим — в Африці

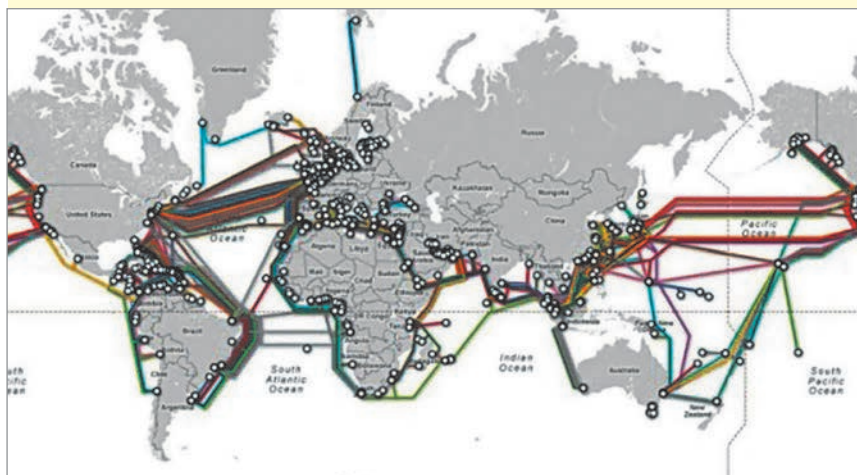


Рис. 1. Схема бекбонів Інтернету

Інтернет складається з мільйонів локальних і глобальних приватних, публічних, наукових, ділових і урядових мереж, пов'язаних між собою завдяки використанню різноманітних технологій.

Інтернет становить фізичну основу для розміщення величезної кількості інформаційних ресурсів і послуг, таких як взаємопов'язані гіпертекстові документи Всесвітньої павутини (*World Wide Web* — *WWW*), електронна пошта та ін.

Єдиного власника та центру управління мережі Інтернет не існує, а його ресурси мають розподілений характер. Це дозволяє забезпечити високу надійність мережі: при виході з ладу частини комп'ютерів і ліній зв'язку вона функціонуватиме й передаватиме повідомлення іншими лініями зв'язку.

Комп'ютери, під'єднані до мережі Інтернет, можуть мати будь-які апаратні й програмні платформи, але при цьому повинні підтримувати сімейство (стек) протоколів зв'язку TCP/IP.

Кожен користувач мережі Інтернет має доступ до інформаційних ресурсів незалежно від місця їх розташування, адже Інтернет — це глобальна інформаційна система.

Основою апаратної складової Інтернету є **бекбони** (від англ. *backbone* — хребет) — надшвидкісні магістральні канали передавання даних (рис. 1). Ці магістралі передавання даних контролюються комерційними, державними, науковими та іншими високопродуктивними центрами.

Власниками бекбонів також є транснаціональні телекомунікаційні корпорації, такі як AT&T Inc., MCI, Sprint, CenturyLink. Бекбони з'єднують не тільки великі міста, але й континенти завдяки глибоководним кабелям.

Доступ до бекбонів мають первинні провайдери (провайдери першого рівня або Tier-1-оператори), які не платять за Інтернет та обмінюються трафіком між собою за допомогою пірінгових угод.

Пірінг (від англ. *peering* — сусідство) — угода інтернет-операторів про обмін трафіком між своїми мережами, а також технічну взаємодію, що реалізує зазначену угоду: з'єднання мереж і обмін інформацією по мережевих маршрутах.

Передаванням даних бекбонами керують суперкомп'ютери. Інформація зберігається на серверах дата-центрів.

Дата-центр (від англ. *data center*) — спеціалізований технічний майданчик для розміщення інформації в мережі Інтернет, під'єднаний до неї як автономна система (або мережа в її складі) великою кількістю каналів зв'язку.

Завдяки високотехнологічній інфраструктурі дата-центри забезпечують безперебійну роботу серверного та комунікаційного обладнання різних компаній. Вони максимально убезпечують дані своїх клієнтів від ризику втрати, організують резервне копіювання даних, захист від всіляких атак та ін.

Компанія Intel має близько 100 000 серверів в своєму арсеналі, Facebook, AT&T і Time Warner Cable — від 20 000 до 30 000, Microsoft, Amazon і eBay — по 50 000 серверів кожний. Google має 1 млн виділених серверів, що перевищує частку в 2% від усіх серверів у світі (рис. 3).

Вторинними провайдерами (провайдерами другого рівня) — Tier-2 — є насамперед «національні» оператори рівня країни чи групи країн регіону. В Україні такими провайдерами є Укртелеком, Київстар, Воля, Датагруп.

Провайдери другого рівня купують трафік у провайдерів першого рівня та забезпечують доступ до каналів Інтернету своїм клієнтам — окремим користувачам, мережам і провайдерам нижчого рівня: Tier-3 і безтірним, які і становлять основну масу провайдерів.



За адресою <http://www.submarinecablemap.com/> можна знайти повний перелік усіх існуючих глибоководних кабелів. На початок 2017 року їхня кількість складала 428, загальна довжина — 1,1 млн км.

Найпотужніша серверна мережа належить флагману ІТ — Google (рис. 3). У місті Омаха (штат Айова, США) на площі 10 000 м² знаходяться сервери, що опрацьовують пошукові запити. На них зберігаються й відеоролики YouTube.

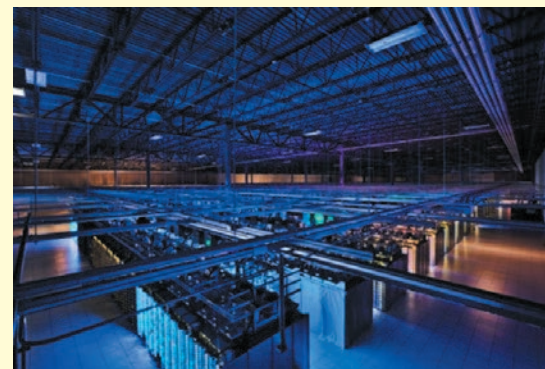


Рис. 3. Дата-центр Google



Запитання для перевірки знань

- 1 Із чого складається мережа Інтернет?
- 2 Що таке бекбони? Які організації мають доступ до бекбонів?
- 3 Що таке Tier-1, Tier-2, Tier-3? Чим вони відрізняються?
- 4 Яке призначення дата-центру?
- 5 Підготуйте повідомлення про використання мережі Інтернет у повсякденному житті.
- 6 З'ясуйте, які дата-центри називають «зеленими». Знайдіть інформацію про один із «зелених» дата-центрів.
- 7 Знайдіть інформацію про перший глибоководний трансатлантичний кабель.
- 8 Знайдіть в Інтернеті відомості про вторинних провайдерів (провайдерів другого рівня) в Україні. Зробіть повідомлення.

5.8. Способи під'єднання до мережі. Функції провайдера



Як ви вже знаєте, Інтернет є об'єднанням мереж та комп'ютерів незалежно від місця їх розташування. Які є способи під'єднання до Інтернету?

Технології DSL дозволяють передавати дані зі швидкістю, що значно перевищує доступну навіть найкращим аналоговим та цифровим модемам. Ці технології підтримують передачу голосу, високошвидкісну передачу даних і відеосигналів. Існуючі типи технології DSL відрізняються методом модуляції, що використовується для кодування даних, та швидкістю передавання даних.

Для під'єднання до мережі можна скористатись телефонними, кабельними каналами або каналами радіозв'язку. Розглянемо найпоширеніші способи (рис. 1).



Рис. 1. Способи під'єднання до Інтернету

Перевагою мережі ISDN є реалізація принципу єдиної розподіленої телефонної станції. Це дозволяє оптимізувати навантаження на канали зв'язку та практично миттєво встановлювати з'єднання. Крім того, вона здатна здійснювати автоматичну маршрутизацію з'єднань, що особливо важливо у випадках, коли між станціями є декілька альтернативних шляхів з'єднання і треба вибрати найбільш оптимальний.

Доступ по виділених лініях (аналогових та цифрових, по локальній мережі, з використанням каналів кабельної телевізійної мережі) — комп'ютер користувача з'єднаний із сервером за допомогою кабелю, причому це з'єднання є постійним (а не комутованим, як у випадку телефонного зв'язку).

xDSL (англ. *Digital subscriber line*) — сімейство технологій (ADSL, VDSL, ISDL, SDSL, SHDSL та ін.), які дозволяють значно розширити пропускну здатність абонентської лінії місцевої телефонної мережі шляхом використання ефективних лінійних кодів на базі сучасних методів цифрового опрацювання сигналу.

ISDN (англ. *Integrated Services Digital Network*) — цифрова мережа з інтегрованими службами (послугами). Це загальнодоступна телефонна мережа, що використовує цифрову технологію передавання сигналу та надає можливість передати кінцевому користувачеві по наявних телефонних проводах текст, графічні зображення, музику, відеосигнали тощо. Основне призначення ISDN — передавання даних зі швидкістю до 64 Кбіт/с дровою лінією й забезпечення інтегрованих телекомунікаційних послуг (телефон, факс, тощо).

SDN-станції забезпечують комутацію цифрових потоків. Перетворення аналогових сигналів у цифрові відбувається на

рівні ISDN-терміналів (тобто на обладнанні кінцевих користувачів). У зв'язку з цим ISDN-станція має можливість комувати однорідні цифрові потоки, «не знаючи», що саме передається по каналу.

До бездротових технологій належать WiFi, WiMAX, RadioEthernet, MMDS, LMDS. Розглянемо деякі з них окремо.

RadioEthernet — технологія широкосмугового доступу до Інтернету, яка ділиться між усіма активними користувачами й забезпечує швидкість передавання даних від 1 до 11 Мбіт/с. Для роботи RadioEthernet-каналу необхідна пряма видимість між антенами абонентських точок. Радіус дії становить до 30 км.

MMDS (від англ. *Multichannel Multipoint Distribution System* — багатоканальна багатоадресна розподільна служба) та **LMDS** (від англ. *Local Multipoint Distribution System* — локальна багатоадресна розподільна служба) — це стандарт стільникових мереж безпроводного передавання даних для фіксованих абонентів.

Мобільний GPRS-Інтернет. Для користування послугою «Мобільний Інтернет» за допомогою технології GPRS необхідно мати телефон із вбудованим GPRS-модемом та комп'ютер. Технологія GPRS забезпечує швидкість передавання даних до 114 Кбіт/с. Під час використання технології GPRS тарифікується не час з'єднання з Інтернетом, а загальний обсяг переданих та отриманих даних.

Мобільний CDMA-Інтернет. Мережа стандарту CDMA — це стаціонарний та мобільний зв'язок, а також швидкісний мобільний Інтернет. Для користування послугою «Мобільний Інтернет» за допомогою технології CDMA необхідно мати телефон із вбудованим CDMA-модемом або CDMA-модем та комп'ютер. Технологія CDMA забезпечує швидкість передавання даних до 153 Кбіт/с.

Супутниковий Інтернет — спосіб забезпечення доступу до Інтернету з використанням технологій супутникового зв'язку (рис. 2).

Доступ до інформаційних ресурсів надають спеціальні організації — постачальники послуг Інтернету, інтернет-провайдери (ISP, англ. *Internet Service Provider*).



Провайдер послуг Інтернет, або інтернет-провайдер — організація, яка надає послуги доступу та передавання даних інформаційними каналами.

Провайдери часто надають комплекс послуг із доступу та користування Інтернетом.

? Запитання для перевірки знань

- 1 Які існують способи під'єднання до Інтернету?
- 2 Які технології базуються на комутації з телефонними каналами?
- 3 Що таке бездротові технології?
- 4 Які послуги надають провайдери?
- 5 Опишіть особливості мобільного GPRS-Інтернету?

Послуги інтернет-провайдера:

- доступ до Інтернету по комутованих лініях надійного зв'язку
- надання дискового простору для зберігання та забезпечення роботи сайтів (хостінг)
- підтримка сервісів електронних поштових скриньок чи віртуального поштового сервера
- розташування обладнання клієнтів на власних технічних потужностях та лініях зв'язку (колокація)
- оренда відокремлених та віртуальних серверів, резервування даних тощо

Обмін даними через супутник

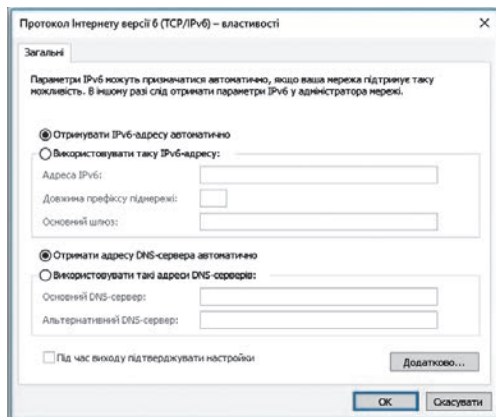
- Односторонній спосіб (one-way), або «асиметричний», — для приймання даних використовується супутниковий канал, а для передавання — доступні наземні канали
- Двосторонній спосіб (two-way), або «симетричний», — і для приймання, і для передавання використовуються супутникові канали

Рис. 2. Способи обміну даними через супутник

5.9. Адресація в Інтернеті. Поняття IP-адреси, доменного імені та URL-адреси



Пригадайте, яке максимальне десяткове число можна записати в 1 байт, 2 байти.



Налаштування IP-адрес у комп'ютері з ОС Windows 10

IP-адреса стандарту IPv4 складається з чотирьох однобайтових чисел, розділених крапками (32-бітового двійкового числа). Кожне число може набувати значення від 0 до 255 включно. Це означає, що в цьому стандарті можна використовувати 256^4 (4 млрд 295 млн) унікальних адрес. Наприклад: 198.162.5.7.

IP-адреса стандарту IPv6 складається з восьми двобайтових чисел, розділених двокрапками (128-бітового двійкового числа). Кожне число може набувати 65535 значень. Загальна кількість адрес у цьому стандарті сягає 340 трильйонів. Наприклад: 2001:0d68:11a3:09d7:1f34:8a2e:078a0:765d

Як відомо, в разі об'єднання пристроїв в мережу суттєвого значення набуває їхня швидка й однозначна ідентифікація. Як ідентифікатор виступає так звана IP-адреса (від англ. *Internet Protocol Address*) (див. [рисунок](#)).



IP-адреса — унікальний ідентифікатор пристрою, підключеного до локальної мережі або Інтернету.

Найпоширенішою є IP-адреса стандарту **IPv4**. Швидкий розвиток мереж Інтернету приводить до вичерпання вільних адрес, тому з 2008 року введено новий стандарт IPv6.

IP-адреси є основним типом адрес, на підставі яких мережевий рівень протоколу IP передає пакети між мережами. IP-адреса призначається адміністратором під час конфігурування комп'ютерів і маршрутизаторів.

IP-адреса складається з двох частин: номера мережі й номера вузла.

Коли мережа працює як складова частина Інтернету, то адреса мережі видається провайдером або регіональним інтернет-реєстратором (*Regional Internet Registry*, скорочено RIR).

Номер вузла в протоколі IP призначається незалежно від локальної адреси вузла. Маршрутизатор по визначенню входить відразу в кілька мереж, тому кожен порт маршрутизатора має власну IP-адресу. Кінцевий вузол також може входити в кілька IP-мереж. У цьому випадку комп'ютер повинен мати кілька IP-адрес, за числом мережевих зв'язків.

Таким чином, IP-адреса характеризує не окремий комп'ютер або маршрутизатор, а одне мережеве з'єднання.

Кількість адрес у мережі визначається тільки й виключно маскою підмережі.



Маска підмережі — бітова маска, яка визначає, яка частина IP-адреси вузла мережі належить до адреси мережі, а яка — до адреси самого вузла в цій мережі.

За допомогою маски підмережі можна визначити, що один діапазон IP-адрес буде в одній підмережі, а інший діапазон, відповідно, — в іншій підмережі.

Наприклад, вузол з IP-адресою 12.34.56.78 і маскою підмережі 255.255.255.0 знаходиться в мережі 12.34.56.0.

Щоб отримати адресу мережі, знаючи IP-адресу і маску підмережі, необхідно застосувати до них операцію кон'юнкції (логічне І) по кожному розряду. Розглянемо приклад.

Приклад. IP-адреса: (192.168.1.2).
Маска підмережі: (255.255.254.0).

1	1	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Тоді адреса мережі: (192.168.0.0).

Числові адреси вузлів дуже зручні для опрацювання обчислювальними пристроями. Але уявіть собі, що ви маєте запам'ятовувати десять-двадцять 32- або 128-бітових послідовностей. Так, значно зручніше використовувати адреси, які записані словами або скороченнями. Наприклад, на відміну від адреси 209.185.108.134 ми легко набираємо google.com.ua.



Доменне ім'я — це адреса ресурсу в мережі, яка записана з використанням слів або їх скорочень, розділених крапкою.

Для зручного користування інформаційними ресурсами їм надають унікальні адреси — URL (від англ. *Uniform Resource Locator* — уніфікований вказівник місцезнаходження ресурсу).



URL-адреса — стандартизована адреса інформаційного ресурсу в мережі, що складається з назви протоколу, за допомогою якого буде передаватися файл, шляху до файла на комп'ютері та імені самого файла.

URL-адреса має таку структуру:
протокол://доменне ім'я/шлях до файла/ім'я файла.

Наприклад: http://interactive.ranok.com.ua/upload/file/Ranok_Informatika_9%20klas_pogliblena.pdf
протокол: http
доменне ім'я: interactive.ranok.com.ua
шлях до файла: /upload/file
ім'я файла: Ranok_Informatika_9%20klas_pogliblena.pdf



Регіональні реєстратори (RIR) отримують номери автономних систем і великі блоки адрес в інтернет-корпорації із присвоєння імен та номерів (англ. *Internet Corporation for Assigned Names and Numbers, ICANN*), а потім видають номери автономних систем і блоки адрес меншого розміру локальним інтернет-реєстраторам (англ. *Local Internet Registries, LIR*).

Назва RIR	Регіон
ARIN	Північна Америка
APNIC	Південно-Східна Азія
AfriNIC	Африка



Запитання для перевірки знань

- 1 Що таке IP-адреса?
- 2 Які стандарти IP-адрес ви знаєте?
- 3 Що таке маска підмережі?
- 4 Поясніть необхідність появи доменних імен.
- 5 Що таке URL-адреса? Яке її призначення?
- 6 Знайдіть відомості про особливі IP-адреси. Зробіть повідомлення.
- 7 Знайдіть відомості про URL-адресу. Хто запропонував використовувати URL-адреси?



Завдання для самостійного виконання

- 1 Вирахуйте адресу мережі, якщо IP-адреса вузла 101.135.2.17, а маска підмережі 255.255.243.0.
- 2 Вирахуйте адресу мережі, якщо IP-адреса вузла 12.34.56.78, а маска підмережі 255.255.255.0.

5.10. Адміністрування доменних імен



Пригадайте, що таке доменне ім'я.



Система доменних імен вперше з'явилася в січні 1985 року:

- .com — для комерційних сайтів
- .edu — для освітніх сайтів
- .gov — для сайтів державних організацій
- .net — для сайтів, чия діяльність пов'язана з мережею
- .org — для некомерційних організацій
- .mil — для військових організацій США
- .int — для міжнародних організацій

Розглянемо правила формування доменних імен, як і ким вони надаються. Ми вже говорили про ICANN, інтернет-корпорацію з управління доменними іменами і IP-адресами. Це некомерційна організація, що відповідає за глобальну координацію системи унікальних елементів Інтернету, її стабільну роботу й безпечну організацію.



Саме ICANN надає імена доменам верхнього рівня.

Доменні верхнього (першого) рівня діляться на два типи: загального (generic) користування та національні (географічні). *Доменні загального користування* призначені для використання усією спільнотою Інтернету (com, net і т. д.). Створення *національних доменів* базується на міжнародному стандарті ISO 3166 та визначає кодові позначення держав (для присвоєння національних доменних імен верхнього рівня ICANN використовує двобуквенні коди ISO 3166-1 alpha-2). Ці типи доменів є початковими точками відліку, від яких почали з'являтися імена другого, третього й нижчих рівнів. Остання частина будь-якого доменного імені означає доменну зону.

Рівні доменного імені (рис. 1) відокремлюються крапками.

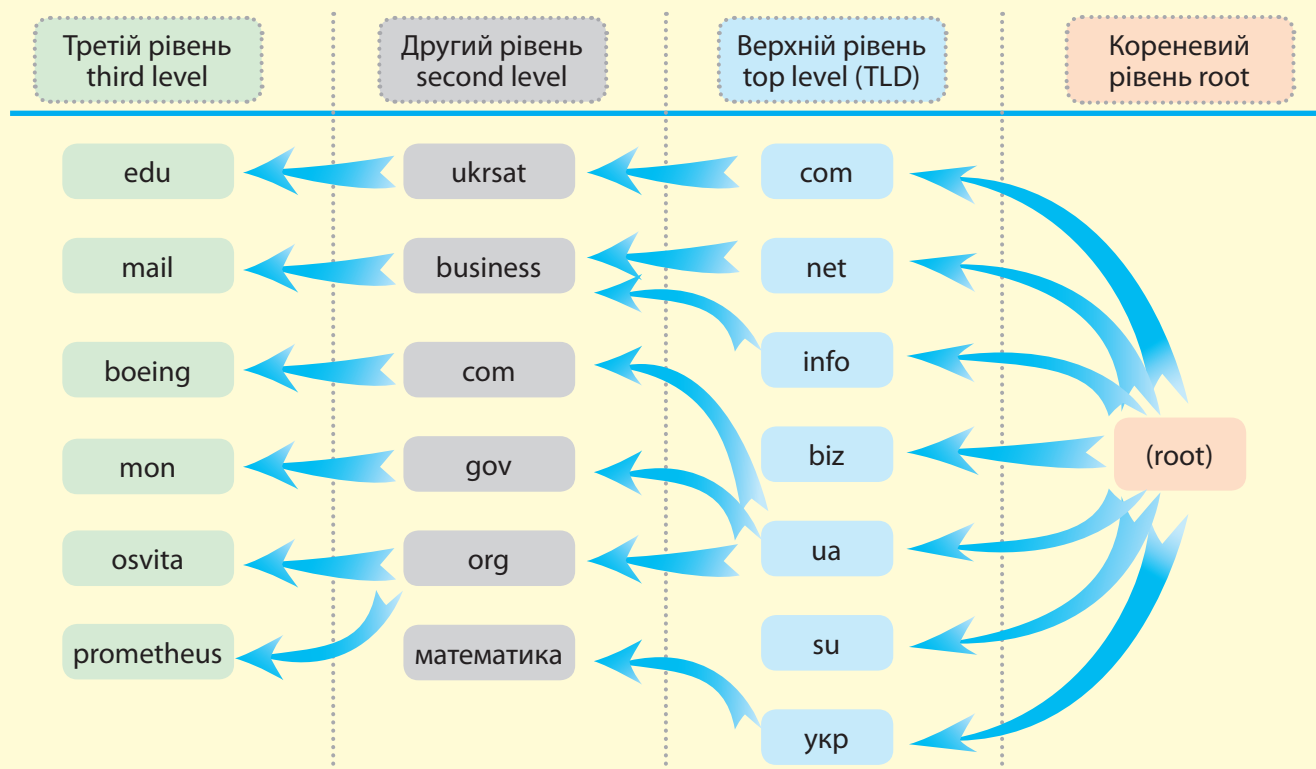


Рис. 2. Приклади рівнів доменних імен

Кореневий домен (англ. *Root domain*) — домен найвищого рівня; позначається порожнім іменем (не містить жодних символів). Оскільки кожен рівень доменного імені відокремлюється крапкою, в кінці імені може бути крапка, що відокремлює порожнє ім'я, яке відповідає кореневому домену. Практично всі браузерери автоматично ставлять крапку в кінці доменного імені.

✓ Доменні зони національних доменів відповідають двом літерам міжнародного коду країни. В Україні це .ua, у США — .us. Усього у світі існує близько 300 національних доменів верхнього рівня.

Піддомен (англ. *subdomain*) — підлеглий домен. Теоретично піддомен може досягати глибини в 127 рівнів, а кожна мітка може містити до 63 символів, поки загальна довжина разом із крапками не досягне 254 символів. На практиці обмеження суворіші.

За порядок у доменах зазвичай відповідає певний комп'ютер, користувачі-адміністратори якого слідкують за тим, щоб, наприклад, не було різних машин з однаковими IP-адресами. Технічні особливості системи доменних імен виключають можливість існування у цій системі двох тотожних доменних імен.

Домен верхнього рівня .ua є українською частиною Всесвітньої мережі й існує з грудня 1992 року. Станом на 2017 рік налічує понад півмільйона доменних імен. У домені здійснюється реєстрація доменних імен другого та третього рівнів. Реєстрація імен третього рівня доступна в 63 публічних доменах (рис. 2), у тому числі 57 географічних. Більш детально з інформацією про домен .ua можна ознайомитись на сайті компанії «Хостмайстер» — реєстратора домену (<https://hostmaster.ua>).

На сьогодні існують дві українські доменні зони: ua і укр.

! **Доменна система імен** (англ. *Domain Name System, DNS*) — ієрархічна розподілена система перетворення імені хоста (комп'ютера або іншого мережевого пристрою) в IP-адресу.

Служба імен DNS — це розподілена база даних. Можна вважати, що це кілька таблиць, у яких записано: яку IP-адресу має комп'ютер з певним іменем; яке ім'я має комп'ютер з визначеною адресою; що це за комп'ютер і яка ОС встановлена на ньому.

✓ Ім'я та IP-адреси не тотожні. Одна IP-адреса може мати безліч імен, що дозволяє підтримувати на одному комп'ютері безліч веб-сайтів (віртуальний хостинг). Зворотне теж справедливо — одному імені може бути зіставлено безліч IP-адрес. Наприклад, у Google.com близько 80 адрес.

? Запитання для перевірки знань

- 1 Яка організація надає імена доменам?
- 2 Що таке DNS?
- 3 Назвіть типи доменів верхнього рівня.
- 4 Яка інформація зберігається на DNS-сервері?
- 5 Запропонуйте власне доменне ім'я, використовуючи не менше ніж п'ять рівнів.

Список імен доменів верхнього рівня можна переглянути на сайті IANA (<https://www.iana.org/domains/root/db>).

Наразі список перевищує 4 тис.



Існують онлайн-ресурси, які дозволяють за доменним ім'ям дізнатися місцезнаходження IP-адреси або сайта, отримати інформацію про домен, визначити хостинг-провайдера сайта. Один із прикладів такого ресурсу: <https://2ip.ua/ua/>



Рис. 2. Приклад публічних доменів України

Корпорація ICANN послідовно розширювала список доменів загального користування. Зараз їх уже 21. Із 2001 року корпорація впровадила доменні зони .info, .biz, .name, .coop, .museum, .aero, .pro, .travel, .jobs, .cat, .asia, .eu, .mobi, .tel. У ICANN планують і надалі розширювати адресний простір.

5.11. Сучасні сервіси Інтернету



Пригадайте, що таке електронна пошта? Які служби Інтернету ви знаєте?



У 1989 році Тім Бернерс-Лі, працівник Європейської лабораторії з ядерних досліджень ЦЕРН (Швейцарія), запропонував проект, відомий під назвою «Всесвітня павутина». Проект мав на меті публікацію гіпертекстових документів, пов'язаних гіперпосиланнями для полегшення пошуку інформації у внутрішній мережі.



Документальний чотирисерійний фільм BBC «Віртуальна революція» (The Virtual Revolution) розповідає історію створення Інтернету, містить інтерв'ю з відомими засновниками Microsoft, Facebook, Twitter, Google, YouTube, Amazon.

Соціальними мережами користується чверть населення земної кулі. Це означає, що 1 млрд 730 млн людей публікують пости, додає цікаві матеріали в Pinterest, пише в Twitter, обмінюється короткими відео через Vine і викладає фотографії в Instagram. Кожні 60 с у Tumblr з'являється 4,7 млн записів, у Snapchat завантажується 277 000 мультимедійних файлів, а на YouTube переглядається понад 5 млн відео.

У попередніх класах ви ознайомилися зі службами Інтернету: електронною поштою, Всесвітньою павутиною.

Комп'ютерна мережа (Інтернет), мережа гіпертекстових документів (Всесвітня павутина) і програмне забезпечення (соціальні сервіси) пов'язують між собою не тільки комп'ютери та документи, а й користувачів. Завдяки мережевим зв'язкам формуються нові соціальні об'єднання.

Із розвитком комп'ютерних технологій у спільнот обміну знаннями з'являються нові форми для зберігання знань і нові програмні сервіси (рис. 1).

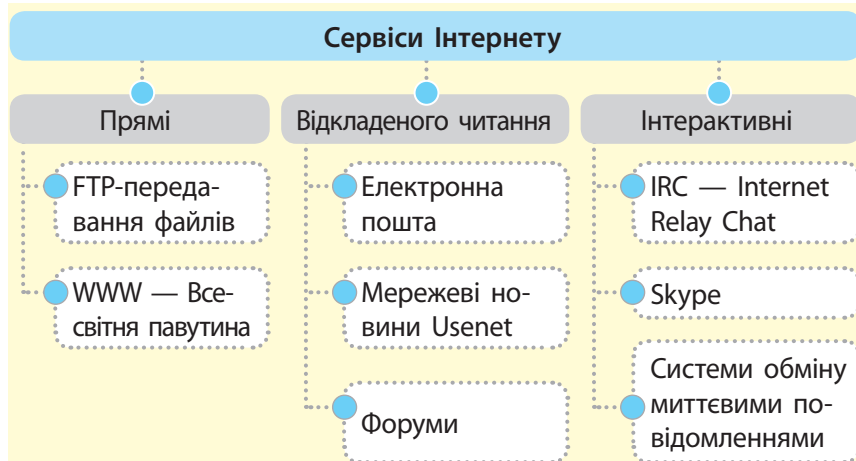


Рис. 1. Основні сервіси Інтернету



Мережева спільнота — це група людей, що підтримують спілкування й ведуть спільну діяльність за допомогою комп'ютерних мережеских засобів.

Комунікаційні служби надають можливість користувачам мережі обмінюватися новинами, обговорювати проблеми тощо.



Обговорення з певної теми групою співрозмовників, які перебувають на значній відстані один від одного, називають **телеконференцією**.

Телеконференції поділяють на відеоконференції та аудіоконференції. **Відеоконференція** передбачає використання засобів передавання відеозображень, **аудіоконференція** — засобів голосового спілкування.



Форум — інтернет-ресурс, популярний різновид спілкування в Інтернеті. На форумі створюються теми для спілкування.



Чат (англ. *chat* — «балачка») — мережевий засіб для швидкого обміну текстовими повідомленнями між користувачами Інтернету в режимі реального часу.

Зараз чати стають менш популярними, їх заміщують соціальні мережі. Однак чати підтримують свою популярність завдяки новим можливостям (приватне спілкування, відео-спілкування та ін.)

Із розвитком мобільного Інтернету дедалі більшої популярності набувають так звані інстант-месенджери.



Інстант-месенджер — система обміну миттєвими повідомленнями (англ. *Instant messaging*) — для обміну повідомленнями в реальному часі через Інтернет. Можуть передавати текстові повідомлення, звукові сигнали, зображення, відео.

Розглянемо інстант-месенджери детальніше.

Skype — один із найстаріших месенджерів, що зберігає популярність донині (рис. 2). Наразі його аудиторія становить 560 млн користувачів по всьому світу.

Skype підтримує можливість створення дуже великих групових чатів. Так, одночасно можуть листуватися до 300 осіб. У групових відеодзвінках можуть одночасно брати участь до 9 користувачів. Ще одна корисна функція — можливість працювати як файлообмінник. Месенджер дозволяє користувачам обмінюватися файлами будь-якого формату.

WhatsApp — найпопулярніший месенджер світу. Нині його аудиторія становить близько 800 млн користувачів. Безперечною сильною стороною месенджера є його простота. Він дозволяє здійснювати безкоштовні голосові дзвінки й відправляти безкоштовні SMS іншим користувачам WhatsApp.

Viber — другий за популярністю месенджер світу. Нині його аудиторія перевищує 600 млн користувачів. Месенджер надає можливість здійснення відеодзвінків, проте одночасно переписуватися в чаті можуть лише 15 осіб. Користувачі мають змогу обмінюватися файлами певних форматів, хоча месенджер дозволяє надсилати співрозмовникам фото і відео.

Telegram — найбезпечніший з усіх існуючих сьогодні месенджерів. Використовується тільки для онлайн-листування, проте в груповому може одночасно листуватися до 200 осіб. Він синхронізується з номером телефону й не вимагає реєстрацій, не дозволяє обмінюватися файлами, але дає змогу надсилати співрозмовникам фото і відео.

Сфера застосування чатів:

- all2all — групова комунікація
- p2p — персональні комунікації — для особистого спілкування
- b2b — ділові (робота в групах)
- b2c — споживацькі (підтримка клієнтів компанії на корпоративному сайті)



У 109 країнах найбільш популярним є WhatsApp, який випереджає аналоги з величезним відривом. У 49 країнах лідирує Facebook Messenger.

У Східній Європі (Білорусь, Молдова, Україна) більше популярний Viber. В Україні він встановлений на 65% пристроїв.

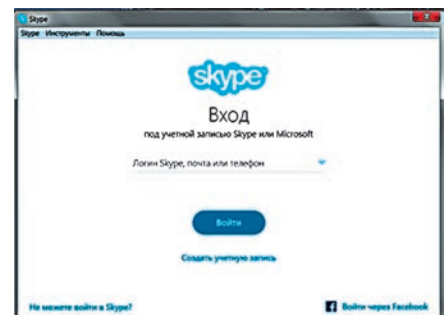


Рис. 2. Вікно месенджера Skype



Запитання для перевірки знань

- 1 Що таке мережева спільнота?
- 2 Що таке телеконференція?
- 3 Що таке форум?
- 4 У чому полягає різниця між форумом і чатом?
- 5 Назвіть сфери застосування чатів.
- 6 Які інстант-месенджери ви знаєте?

5.12. Сервіси Веб 2.0



Поясніть, що таке Всесвітня павутина. Назвіть відомі вам сервіси Інтернету.



Творцем терміна «Веб 2.0» є Тім О'Рейлі. 30 вересня 2005 року він написав статтю: <http://www.oreilly.com/pub/a/web2/archive/what-is-web-20.html>), в якій розповів про нову концепцію вебу.

Існує багато спеціальних онлайн-сервісів для створення власної хмари тегів. Найпопулярнішими є Word it Out (<https://worditout.com/>), Tagxedo (<http://www.tagxedo.com/>), Tagul (<https://tagul.com/>), Word Cloud (<https://www.jasondavies.com/wordcloud/>)



Рис. 1. Приклад хмари тегів

Веб — система доступу до документів, які пов'язані між собою та розміщені на комп'ютерах, під'єднаних до Інтернету.

Розрізняють кілька етапів розвитку веб-сервісів. Перший етап — Веб 1.0. Для них характерними є статичні веб-сторінки, а користувачі можуть спілкуватися лише на форумах та в чатах. Другий етап — Веб 2.0. Це технології, які дозволяють користувачам створювати й поширювати власний контент у Всесвітній павутині.

Реалізація Веб 2.0 стала можливою завдяки таким технологіям.

AJAX (англ. *Asynchronous JavaScript And XML*), асинхронний JavaScript (мова програмування) і XML (мова розмітки веб-сторінок) — дозволяє створювати зручні веб-інтерфейси користувача на сторінках сайтів, де потрібна активна взаємодія. Можна переглядати контент сайта, поки сервер все ще опрацьовує запит. Браузер лише довантажує потрібні дані.

RSS (англ. *Really Simple Syndication* — дійсно просте об'єднання) — дозволяє експортувати гіпертекст з одних ресурсів на інші. Стрічки RSS дозволяють швидко ознайомитися зі змінами змісту сайта, безпосередньо не заходячи на нього через браузер. Причому один користувач може зробити підписку на безліч стрічок різних сайтів.

FOAF (англ. *Friend of a Friend* — один одного) — забезпечує користувачам можливість оформляти підписку на отримання нових матеріалів та списків записів із журналів «друзів» (friends).

Теги — дозволяє більш зручно ідентифікувати й тематично сортувати контент завдяки використанню спеціальних міток (тегів). У Веб 2.0 мітки для навігації подано у вигляді так званої «хмари тегів» (рис. 1); найбільш популярні теги містяться в центрі хмари, найменш популярні — на «периферії».

На рис. 2 наведено сервіси Веб 2.0. Розглянемо основні можливості Веб 2.0.

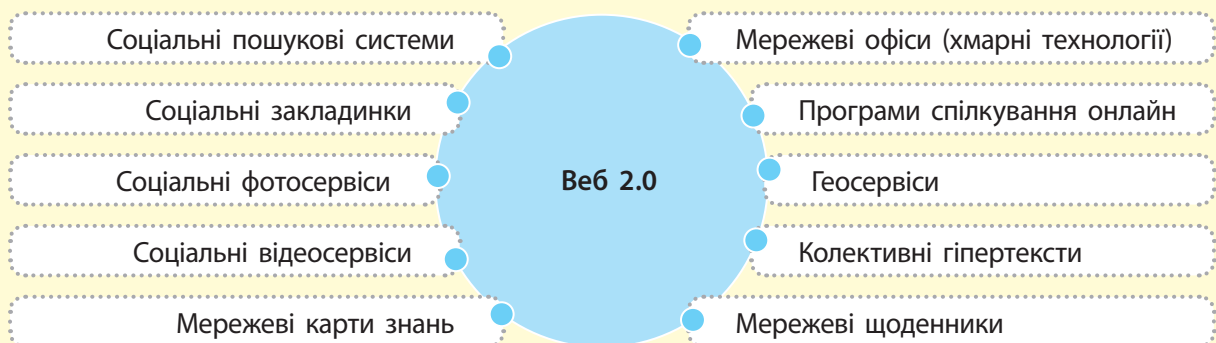


Рис. 2. Сервіси Веб 2.0

Веб-синдикація — дозволяє одночасно поширювати аудіо- та відеоінформації на різноманітних веб-сторінках або сайтах. Принцип полягає в поширенні заголовків матеріалів і посилань на них (наприклад, останні повідомлення форумів тощо). Ця технологія використовувалася на новинних ресурсах і в блогах, але поступово сфера застосування розширилася.

«**Розумні (динамічні) помічники**» — використовують технології, що дозволяють під час користування сторінкою надавати користувачеві потрібну інформацію, враховуючи його (або загальний) досвід користування даним сайтом: спливаючі підказки, автодоповнення, «розумний пошук».

Доступ до веб-сервісів здійснюється за допомогою звичайного інтернет-браузера, без установлення жодних додаткових програм. Користувачам не треба піклуватися про постійні оновлення версій. Виконання будь-яких обчислень веб-сервісами не вимагає від комп'ютера користувача жодних витрат ресурсів — усі операції виконує сервер.

Mash-up (англ. змішування) — надає можливість створення нового веб-сервісу шляхом інтегрування програмних можливостей декількох інших веб-сервісів. Один із принципів Веб 2.0 — можливість збирати свої додатки з чужих компонентів так само, як збирається комп'ютер із окремих комплектуючих.

Трекбек (англ. *track back* — шлях назад) — це зворотні посилання, що розміщуються зазвичай під коментарями до статті, на той блог або ресурс, який посилається на цю статтю. Тобто під час розміщення прямого відкритого посилання на статтю іншого ресурсу автоматично відсилається пінг до джерела, яке, у свою чергу, розміщує як подяку посилання на той блог, де його згадують із назвою статті.

Wiki-сайти — дозволяють своїм користувачам власноруч редагувати, додавати або видаляти інформацію на сайті, створювати нові сторінки (рис. 3). У такий спосіб користувачі більш активно беруть участь у наповненні сайтів інформацією, потрібною саме їм. Найяскравішим прикладом є Wikipedia.

Використання сервісів Веб 2.0 сприяє утворенню соціальних спільнот. Ведення мережеских щоденників, блогів — показовий приклад **соціалізації**. За допомогою блога кожен користувач може персоналізувати певну зону сайту: додати особисті аудіо- та відеофайли, зображення, ділитися новинами та ін.

Все наведене дозволяє користувачам діяти спільно: обмінюватися інформацією, зберігати посилання та мультимедійні документи, створювати та редагувати публікації, таким чином розвивається соціальна взаємодія. Тому технології Веб 2.0. називають соціальними сервісами Інтернету.

Карти знань (карта розуму, або карта пам'яті, думок, інтелект-карта тощо) — це зручна й ефективна техніка візуалізації мислення та альтернативного запису. Онлайн-сервіси для створення ментальних карт: XMind (<http://www.xmind.net/>), Mindmeister (<https://www.mindmeister.com/>), FreeMind (<http://freemind.sourceforge.net/>).

Соціальні мережескі фотосервіси — засоби Інтернету, які дозволяють зберігати, класифікувати, обмінюватися цифровими фотографіями і організувати їх обговорення. Найбільш популярними є Флікр (<http://Flickr.com>), Панораміо (<http://www.panoramio.com>), Пикаса (<http://picasaweb.google.com>). Серед соціальних мережеских відеосервісів лідером є YouTube.



Рис. 3. Приклади wiki-сайтів

? Запитання для перевірки знань

- 1 Чим відрізняється Веб 2.0 від класичного Веб 1.0?
- 2 Які технології використовуються у Веб 2.0?
- 3 Перелічіть компоненти Веб 2.0.
- 4 Які сервіси Веб 2.0 ви використовуєте?

5.13. Геосервіси



WikiМарія — проект, який об'єднує інформацію Google Maps із технологією ВікіВікі, заснований у 2006 році. Його метою є опис всієї Землі. Переглядаючи карту WikiМарія, відвідувач сайта може бачити об'єкти, обмежені рамками, отримувати текстовий опис кожного, редагувати тексти та виділяти нові ділянки карти. Об'єкти також позначаються тегами (у режимі пошуку відображаються тільки ті з них, в описі яких присутній даний тег). Додаток із позначками WikiМарія можна підключити до програми Google Earth.

Що ви робитимете, коли слід з'ясувати, де знаходиться потрібна вулиця?

Широке використання смартфонів з мобільним Інтернетом та GPS сприяло бурхливому розвитку геосервісів. Сьогодні, коли необхідно з'ясувати, як потрапити до певного місця, ми звертаємось саме до них.



Геосервіси — інформаційні ресурси, які надають користувачу інструменти для роботи з геоданими та існують у вигляді самостійного продукту або підключаються до сервісу.

Геосервіси розробляються на базі готових геоінформаційних (ГІС) і веб-геоінформаційних (Веб-ГІС) систем, забезпечують певні сценарії використання даних і надають можливості для приймання, зберігання, опрацювання й аналізу. Геосервіси досить часто вирішують завдання моніторингу, моделювання, планування й проведення розрахунків на основі просторових даних. Розглянемо найвідоміші геосервіси.

Google Maps — загальна назва для додатків (рис. 1), побудованих на основі безкоштовного картографічного сервісу й технології, що надаються компанією Google за адресою <http://maps.google.com/>. У відкритому доступі — мапа та супутникові знімки всього світу (а також Місяця й Марса).

Все більшої популярності набуває геокешинг (*geocaching*, від грец. *γεο* — земля та англ. *cache* — схов) — туристична гра із застосуванням GPS. Вона полягає у знаходженні сховів, створених іншими учасниками гри. Одні гравці готують схов, за допомогою GPS визначають його географічні координати та повідомляють про них в Інтернеті. Інші гравці використовують ці координати та свої GPS-приймачі для пошуку схову.

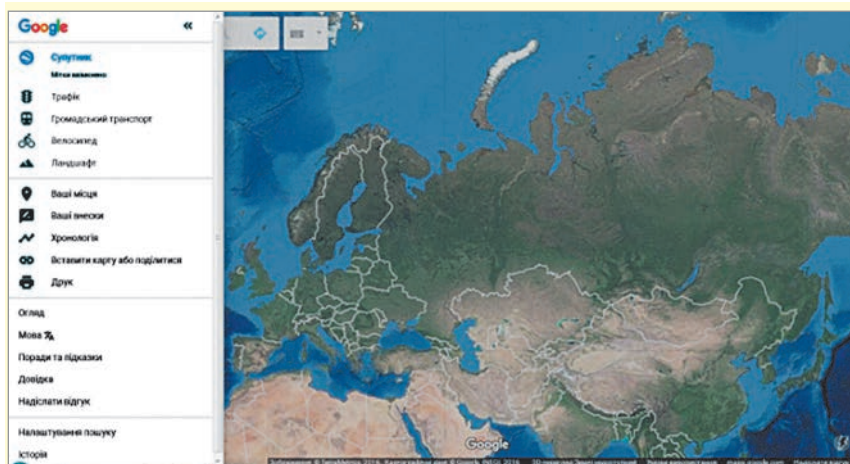


Рис. 1. Скріншот веб-сторінки сайта Google Maps

Залежно від місцевості у службі Google Maps доступні різні режими відображення (подання). Перехід між режимами можна виконати в лівому верхньому куті мапи:

- **мапа** — відображення мапи з традиційним зображенням доріг, парків, кордонів, водойм тощо;

- **супутник** — відображення аерофотознімків. Для відображення назв вулиць та інших відомостей необхідно встановити прапорець Показувати мітки;
- **ландшафт** — відображення високих об'єктів як тривимірного рельєфу. Режим також дозволяє переглядати назви вулиць та інші відомості.

Google Earth — окрема програма, яку можна безкоштовно завантажити із сайта (<http://earth.google.com/>) і на локальному комп'ютері використовувати тривимірну модель земної кулі (рис. 2). Якщо комп'ютер під'єднано до Інтернету, програма дає змогу переглядати знімки земної поверхні, змінювати масштаб і налаштовувати маршрути пересування.

Перевагою програми є тривимірне відображення земної поверхні (з урахуванням рельєфу), можливість спостереження під довільним кутом, поступове уточнення зображення під час завантаження більш детальних фотознімків, можливість плавної зміни масштабу. Глобальна інтерактивна геологічна карта-глобус за адресою <http://www.onegeology.org/> відображає всі відомі наразі родовища корисних копалин.

У розробці сервісу брали участь учені з різних країн світу. Роздільна здатність карти становить 1 сантиметр на 10 кілометрів, завдяки чому всі охочі можуть як переглядати її, так і використовувати у своїх дослідженнях. Карта постійно оновлюється.

Геотегінг (від грец. «гео» — земля, англ. *tag* — ярлик, етикетка, бирка; мітити) — процес приєднання географічних метаданих до різних інформаційних ресурсів, таких як веб-сайти, RSS, фотографії, за допомогою стандарту EXIF. Ця інформація зазвичай складається з координат широти й довготи, хоча може включати також висоту, відстань і напрям до населених пунктів. Геотегінг може допомогти в пошуках різної інформації, прив'язаної до певної місцевості.

EXIF (англ. *Exchangeable Image File Format*) — стандарт, який дозволяє додавати до зображень та інших медіафайлів додаткову інформацію (метадані), коментувати файл, описувати умови і способи його отримання, авторство тощо. EXIF набув поширення у зв'язку з появою цифрових фотокамер.

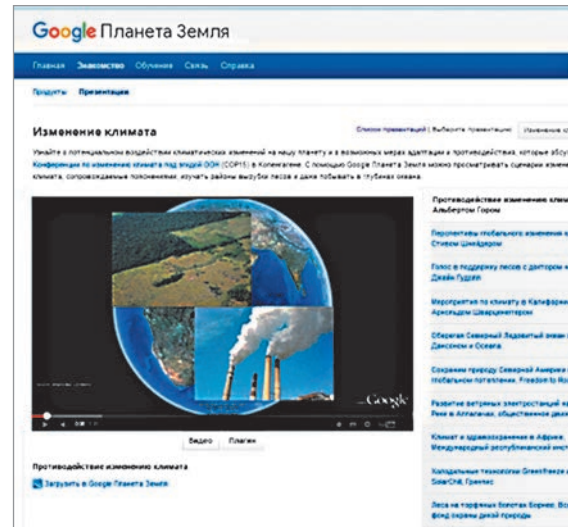


Рис. 2. Приклад презентації на сайті Google Earth

Можливі сервіси Google Maps

- **Мої чудові місця:** створення власних об'єктів — точок, кривих, полігонів — засобами сервісу maps.google.com (опція Мої карти); збереження власного контенту.
- **Сам собі провідник:** GPS-навігаційні пристрої; визначення координат, запис точок і треків; перенесення даних у Гугл «Планета Земля»; планування маршрутів і навігація.
- **Вихід у світ:** створення геотегів — гіперактивних переходів із html-сторінок на карту Гугл із відображенням заданої позиції; геотегінг фотографій, ВікіВікі-сервісів і блогів.



Запитання для перевірки знань

- 1 Дайте означення геосервісів.
- 2 Наведіть приклади геосервісів.
- 3 Що таке геотегінг?
- 4 Порівняйте сучасні геосервіси, назвіть їх переваги та недоліки.
- 5 Наведіть міркування щодо безпечності геотегінгу фотографій. Які, на вашу думку, переваги й недоліки використання геотегінгу?
- 6 Знайдіть в Інтернеті інформацію про туристичну гру геокешинг.

5.14. Спеціальні інструменти пошуку



Пригадайте, як здійснюється пошук в Інтернеті.

Можливості пошукових інструментів

- аналіз веб-сторінок і занесення результатів аналізу на той чи інший рівень бази даних пошукового сервера
- пошук інформації за запитом користувача
- забезпечення зручного інтерфейсу для пошуку інформації та перегляду результату пошуку користувачем

Рис. 1. Призначення пошукових інструментів



Однією з переваг тематичних каталогів є те, що пояснення до посилань дають автори каталогу й повністю відображають його зміст. Таким чином користувач отримує можливість точніше визначити, наскільки відповідає зміст тематичного каталогу меті його пошуку.



Інформаційний пошук (*information retrieval, data search*) — процес знаходження, відбору й видачі певної інформації із задалегідь заданими ознаками, в тому числі документів, їх частин та/або даних, із масивів і записів будь-якого виду.

Пошук даних, пошук документів, інформаційний пошук і текстовий пошук є самостійними напрямками інформатики. Вони мають власні методики та практичне застосування.



Пошукові інструменти — спеціальне програмне забезпечення, основна мета якого — забезпечити найбільш оптимальний і якісний пошук інформації для користувачів Інтернету.

Залежно від призначення пошукові інструменти розміщуються на спеціальних серверах (рис. 1). Зазвичай інтерфейс цих інструментів подано у вигляді сторінки з гіперпосиланнями, рядком подання запиту (рядком пошуку) й інструментами активізації запиту — ключового слова або фрази, яка вводиться користувачем у рядок пошуку. Для формування різних запитів використовують спеціальні («», ~) і математичні (*, +, ?) символи.

Найпоширенішими пошуковими інструментами є каталоги інтернет-ресурсів та пошукові системи.



Каталог інтернет-ресурсів (веб-каталог) — це ієрархічний каталог, який постійно оновлюється, поповнюється і містить безліч категорій та окремих веб-серверів зі стислим описом їх вмісту (рис. 2).

Спосіб пошуку за каталогом базується на принципі «рух сходами вниз», тобто рух від загальних категорій до більш конкретних.

Базу даних каталогу заповнюють працівники, зазвичай вручну. Вони переглядають веб-сторінки та обирають ті, що, на їхню думку, можуть бути цікавими, заносять URL-адреси сторінок до відповідної рубрики каталогу, причому одна й та сама сторінка може бути занесена до різних рубрик.

Найбільшого поширення набули спеціальні пошукові системи (машини веб-пошуку).



Машини веб-пошуку — це сервери з величезною базою даних URL-адрес, які автоматично звертаються до сторінок Всесвітньої павутини за цими адресами, формують і прописують ключові слова зі сторінок у свою базу даних (індексують сторінки). Таким чином, пошук здійснюється вже в цих базах даних.

До найпопулярніших пошукових систем належать Google, Yahoo! і MSN — їм надають перевагу 80 % користувачів (рис. 3).

Пошукові машини пропонують два види пошуку в Інтернеті: *простий* та *розширений*.

Із простим пошуком ви ознайомились у 6 класі. Згадаємо, що при цьому користувач сам формулює запит. Разом із тим він може отримати величезний обсяг інформації, часто зайвої.

Якщо користувач невдало сформулював запит, він може взагалі не отримати потрібний результат — той просто потоне в океані відповідей. Подолати цю проблему можна за допомогою використання операторів математичної логіки, а саме: AND (логічне множення, &), R (логічне додавання, |) і NOT (заперечення).

Каталог сайтів РЕЄСТРАЦІЯ У КАТАЛОЗІ МОЇ САЙТИ ІНФОРМЕРИ

- Hi-Tech**
Навігація, Безпека, Інтерфейс, Мережі і зв'язок, Мобільний зв'язок, Програми, ...
- Авто**
Автомобільні видання, Автопортали, Запчастини та аксесуари, Мотоцикли, Продаж автомобілів, Технічне обслуговування, ...
- Бізнес та фінанси**
Будівництво, Виробництво і постачання, Все для офісу, Нерухомість, Послуги, Реклама, Фінанси, ...
- Відпочинок**
Де розважитися, Охота і рибалка, Подорожі і туризм, Хобі і захоплення, ...
- Довідки**
Адреси і телефони, Афша, Мапи, Погода, Пошук людей, Словники і перекладачі, Транспорт, ...
- Дом**
Все для свята, Домашні тварини, Квартира та дача, Кулінарія, Мода та краса, Сім'я, Товари для будинку, ...
- Здоров'я**
Здоровий спосіб життя, Ліки, Лікувальні установи, Медичні видання, ...
- Інтернет дітям**
Дитяча література, Дитячі портали, Інше, Понавалні сайти, ...
- Культура**
Кіно, Література, Музеї, Музика, Образотворче мистецтво, Танці, Театри, Фотографія, ...
- Навчання**
Курси, Науки, Учебні матеріали і реферати, ...
- Робота**
Вакансії і резюме, Заробток в мережі, Кадрові агентства, Робота за кордоном, ...
- Розваги**
Ігри, Непівняне, Психологічні тести, Розважальні портали, Чати і форуми, Юмор, ...
- СМІ**
Інтернет-видання, Інформаційні агентства, Періодика, Радіо, Телебачення, ...
- Спорт**
Автоспорт, Баскетбол, Екстремальний спорт, Лпні види спорту, Теніс, Товари для спорту, Футбол, Хокей, ...
- Суспільство**

Рис. 2. Сторінка каталогу сайтів

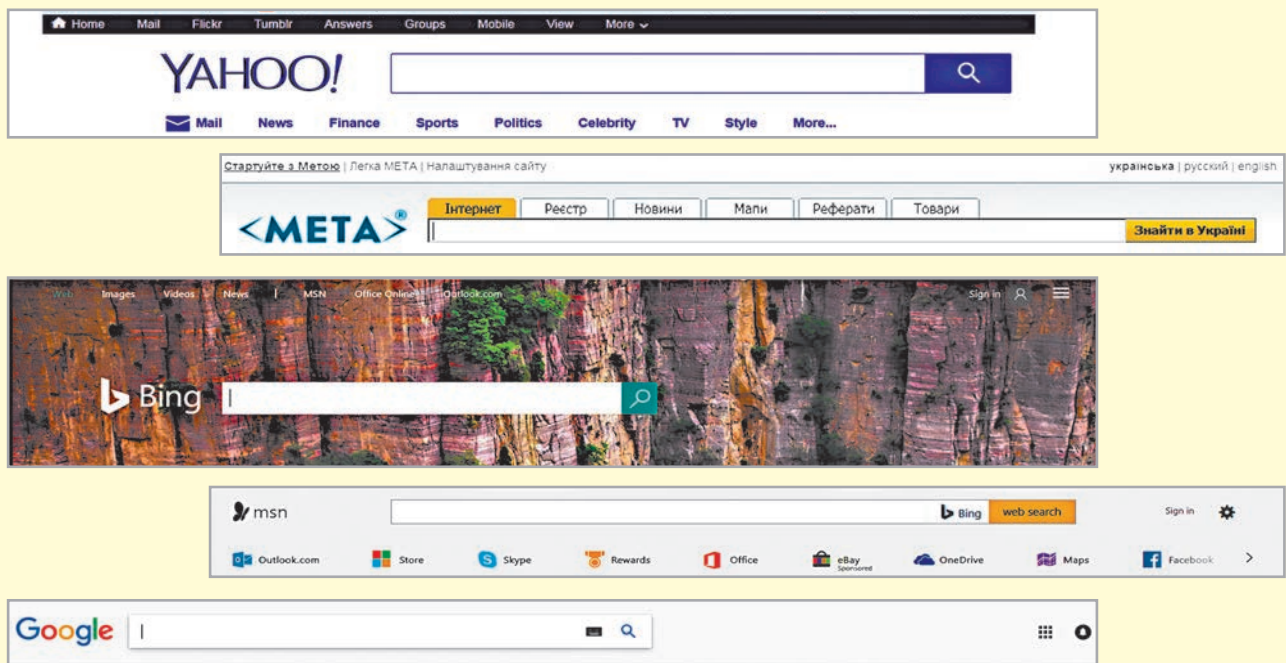


Рис. 3. Популярні пошукові системи

У пошуковій системі Google існує потужний інструментарій для роботи із зображеннями — Google Images (Зображення). Він дозволяє шукати по зображенню, завантажувати його в рядок запиту або ж вказувати його URL-адресу. Крім того, можна налаштувати пошук зображень, використовуючи команди меню Інструменти (рис. 4).

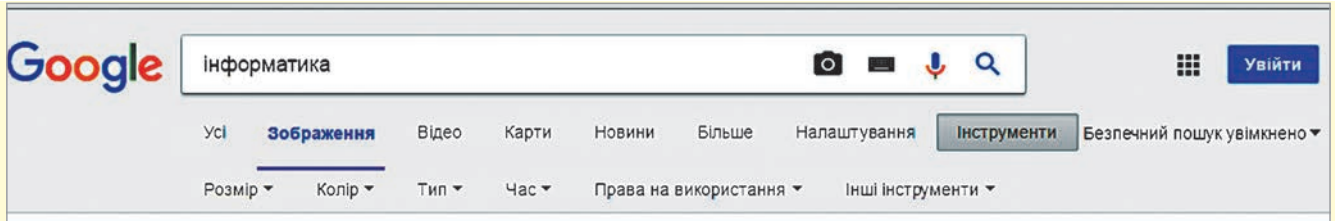


Рис. 4. Інструменти пошуку зображення в пошуковій системі Google

Деякі сучасні сервіси Google

- Google Alerts — дозволяє здійснювати моніторинг оновлення інформації (за темами або ключовим словом) й отримувати повідомлення на акаунт Gmail.
- Google Trends — показує тенденції пошуку в Інтернеті з певної теми в певний час і в певному місці; допоможе стежити за темою і бачити, який ефект справила публікація онлайн-матеріалу.
- Google Correlate — показує зв'язок питань, поставлених у мережі, з трендами й подіями реального світу.
- Google Fusion Tables — дозволяє створювати карти, діаграми й графіки на основі ваших даних, забезпечені прив'язкою до певного місця.

Розширений пошук використовують для пошуку даних за їхніми властивостями шляхом здійснення серії послідовних запитів, які становлять набір правил для пошуку.

Сучасні пошукові системи за посиланням Розширений пошук пропонують користувачам спеціальні зручні форми, призначені для здійснення команд розширеного пошуку.

Команди розширеного пошуку призначені не для пошуку документів за їх змістом, а, наприклад, для пошуку веб-вузлів за їх назвами, за фрагментами їх адрес, за адресами посилань, які зустрічаються на їх веб-сторінках, тощо (рис. 5).

Команди розширеного пошуку дають змогу знаходити веб-сторінки за їх заголовками, доменними іменами серверів, датами створення, наприклад — документи, опубліковані в певний день або після певної дати.

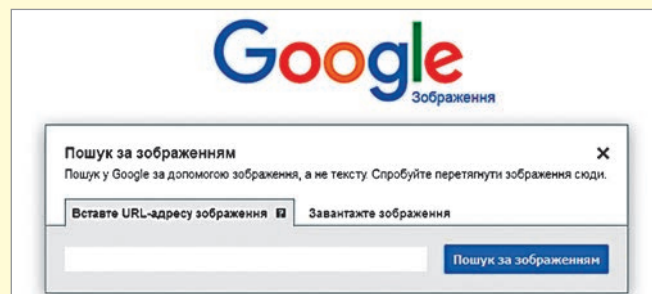


Рис. 5. Пошук за зображенням пошуковою системою Google



Запитання для перевірки знань

- 1 Що є результатом пошуку інформації з використанням пошукової системи?
- 2 Назвіть види веб-каталогів.
- 3 Поясніть, у чому полягають і з чим пов'язані переваги та недоліки веб-каталогів.
- 4 Які послуги можуть надавати користувачам пошукові системи?
- 5 Знайдіть інформацію про правила безпечної роботи в Інтернеті та підготуйте повідомлення.
- 6 Порівняйте особливості пошуку даних із використанням веб-каталогів і пошукових систем. Визначте переваги та недоліки кожного типу пошукових систем.

5.15. Інтернет-ресурси для вивчення навчальних предметів

Назвіть основні види освітніх електронних ресурсів. За якими ознаками їх розрізняють? Чому в інформаційному суспільстві значну роль відіграє освіта?



Нині дедалі більшого поширення в мережі Інтернет набувають освітні та навчальні ресурси. Умовно їх можна розподілити на портали, де збирається інформація про навчальні заклади основної та додаткової освіти, позашкільні виховні установи, різні форми безперервної або дистанційної освіти, та ресурси, які надають допоміжні можливості освіти й самоосвіти в конкретних галузях знання. Їх основна аудиторія — учні й абітурієнти.



Освітні інтернет-ресурси — це ресурси освітнього характеру, розташовані у веб-просторі мережі Інтернет.

До таких ресурсів належать не тільки сайти та портали, а й інші електронні ресурси, які зберігають на веб-серверах у різних форматах (текстових, графічних, архівах, аудіо, відео тощо). Ознайомимося з деякими з них.

Найповнішим каталогом усіх існуючих освітніх сайтів та блогів є портал Освітні ресурси Інтернету (<https://sites.google.com/site/osvitnires/>). Каталог постійно оновлюється.

На сервері Educational Network Ukraine (<http://www.ednu.kiev.ua>) представлені всі українські інформаційні ресурси, пов'язані з освітою: організації, вищі навчальні заклади, докладна інформація про системи освіти найбільш розвинених країн, огляд преси, виставки, конференції тощо. Крім того, існують веб-сторінки, які можуть стати в пригоді вчителям, студентам, учням тощо.

Особливу увагу слід звернути на сайт електронної бібліотеки Національної бібліотеки України імені В. І. Вернадського (<http://www.nbuv.gov.ua>). Вона є унікальним зібранням джерел інформації, яке містить книги, журнали, наукові праці, автореферати дисертаційних досліджень, образотворчі матеріали, рукописи, стародруки, газети тощо.

На Українському освітньому порталі (<http://www.osvita.org.ua/>) можна знайти рейтинг освітніх сайтів України, перелік освітніх ресурсів: реферати, колекції, музеї, словники, перекладачі, електронні бібліотеки тощо.

Наукова світлиця (<http://lecbank.jimdo.com>) — україномовний сайт із невеликою базою лекцій з математики, фізики, хімії та української мови (рис. 1).

Термін «дистанційна освіта» використовують щодо широкого спектра освітніх програм і курсів. Задля забезпечення

Освітні інтернет-ресурси

- Каталоги освітніх ресурсів
- Платформи для онлайн-курсів
- Сайти підтримки вивчення навчального предмета
- Електронні підручники
- Веб-енциклопедії
- Онлайн-словники
- Освітні відеоканали
- Сервіси для створення презентацій
- Сервіси для створення інфографіки
- Сервіси для перевірки знань з навчального предмета

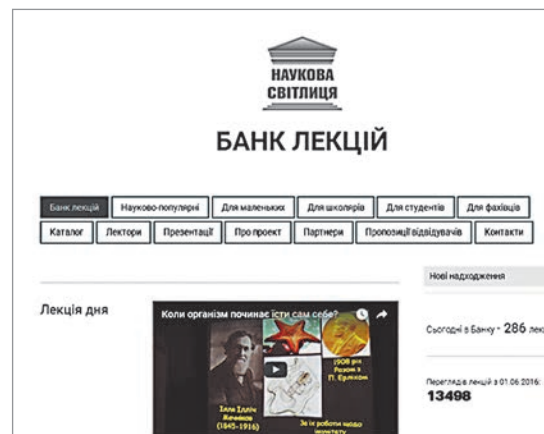


Рис. 1. Скріншот веб-сторінки сайту Наукова світлиця



Сайт Дистанційне навчання інформатики (<https://dystosvita.gnomio.com>) пропонує майже 30 курсів, які охоплюють всі теми шкільної інформатики, як з ІТ (веб-дизайн, робота з табличним процесором, базами даних тощо), так і програмування (C++, Python).

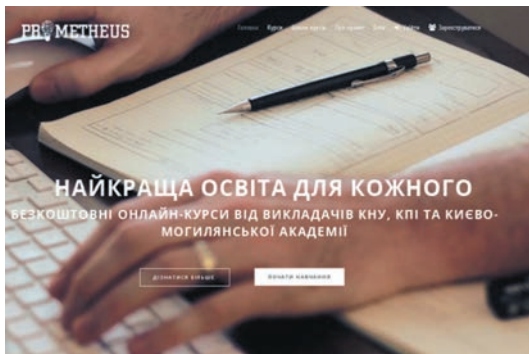


Рис. 2. Головна сторінка сайта Prometheus

ефективної взаємодії під час дистанційного навчання використовують цілий набір інструментів, включаючи інтерактивні комп'ютерні програми, електронну пошту та ін.

Безумовною перевагою дистанційної освіти є можливість незалежно від місцезнаходження слухача отримати доступ до лекцій викладачів найкращих університетів світу. Єдиною вимогою є наявність Інтернету. Все більшої популярності набувають відкриті онлайн-курси на платформах Prometheus, EdEra, EdX, Coursera.

Prometheus — це перший україномовний громадський проект (рис. 2) масових відкритих онлайн-курсів. Головною метою проекту є безкоштовне надання доступу до онлайн-курсів університетського рівня всім бажаючим, а також надання можливості публікувати та розповсюджувати такі курси провідним викладачам, університетам та компаніям. Крім того, Prometheus надає доступ до онлайн-курсів підготовки до ЗНО.

EdEra (<https://www.ed-era.com>) — ще один україномовний освітній ресурс. EdEra ставить на меті зробити освіту якісною та доступною і, на відміну від інших платформ, орієнтований саме на базову середню освіту. Автори створюють онлайн-курси повного циклу — від лекцій до книжок. Це проект незамінний для бажаючих добре підготуватися до складання ЗНО.

Сьогодні численні школи, некомерційні організації, корпорації та міжнародні організації пропонують або планують запровадити курси на ресурсі **EdX**. Станом на 2016 рік EdX має понад 3,5 млн користувачів, що є слухачами більше ніж 250 курсів в Інтернеті.

Coursera — проект у галузі масової онлайн-освіти, заснований професорами інформатики Стенфордського університету Ендрю Іном і Дафною Коллер. У його рамках існує проект із публікації освітніх матеріалів в Інтернеті у вигляді набору безкоштовних онлайн-курсів.

? Запитання для перевірки знань

- 1 Дайте означення освітнім інтернет-ресурсам.
- 2 Які платформи для онлайн-курсів ви знаєте?
- 3 Назвіть переваги дистанційної освіти.
- 4 Перегляньте перелік курсів на платформах Prometheus та Coursera. Які курси, на вашу думку, користуються найбільшим попитом? Обґрунтуйте ваш висновок.

Практична робота № 11

Тема. Сучасні сервіси Інтернету.

Завдання: розробити маршрут прогулянки за допомогою програми Google My Maps.

Обладнання: комп'ютер із виходом у мережу Інтернет.

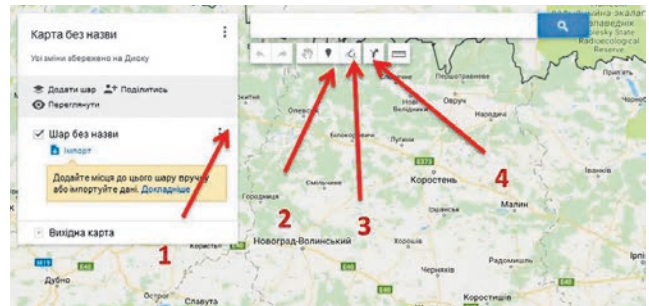
Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Наберіть у пошуковому рядку браузера Google My Maps. Викличте програму.
- 2 У вікні, що відкрилося, клацніть кнопку Створити карту.

За замовчуванням з'являється мапа з першим шаром Шар без назви, у правому кутку шару кнопка Параметри шару, яка дає змогу редагувати назву та видаляти шар (див. рисунок).

- 3 Відредагуйте назву шару «Моя улюблена прогулянка».
- 4 Виберіть на карті місце, де ви буваєте під час прогулянки. Виберіть із рядка інструментів Додати маркер, установіть маркер на потрібному місці. У вікні, що з'явилося, введіть назву вибраного місця та додайте короткий опис.
- 5 Додайте фотографію, натиснувши на значок фотоапарату. Можна або завантажити, або вказати URL-адресу вибраної в Інтернеті фотографії.
- 6 Виберіть колір та вигляд маркера. Для цього клацніть кнопку Стилі створеного маркера, яка розташована в правому кутку рядка маркера.
- 7 Повторіть пп. 5, 6 для наступних місць.
- 8 Виберіть на рядку інструментів кнопку Накреслити лінію.



Карта з новим шаром Шар без назви і рядком інструментів: 1 — параметри шару, 2 — додати маркер, 3 — накреслити лінію, 4 — додати маршрут

- 9 Сполучіть вибрані місця лінією.
 - 10 Аналогічно опису в п. 6 виберіть колір та товщину лінії.
 - 11 Збережіть створену мапу.
- **Зробіть висновки:** як можна використовувати власні створені карти; у яких галузях людської діяльності, на вашу думку, використовуються подібні карти; чи плануєте ви в подальшому створювати такі карти.

Практична робота № 12

Тема. Інтернет-ресурси для навчання.

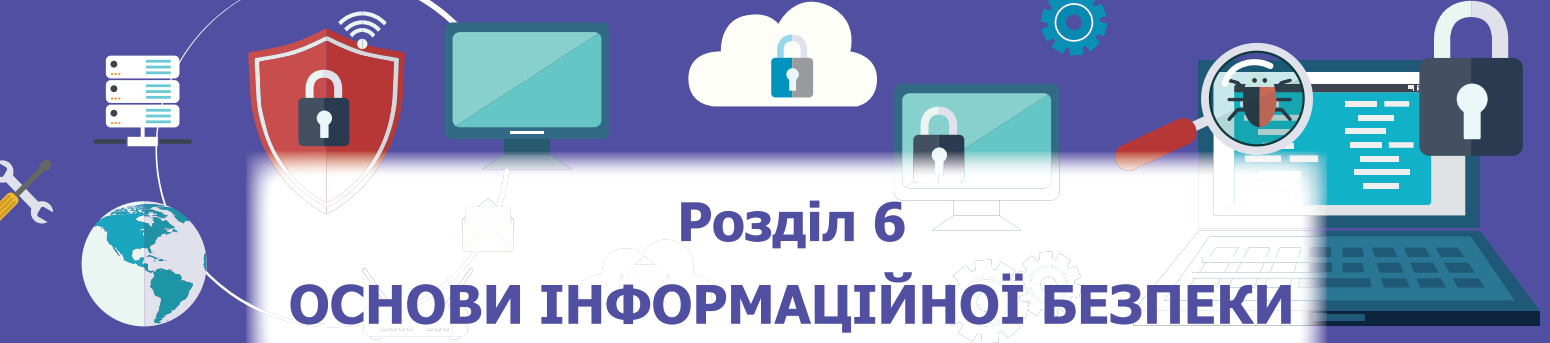
Завдання: навчитися користуватися інтернет-ресурсами для навчання.

Обладнання: комп'ютер із виходом у мережу Інтернет.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Зареєструйтеся на сайті «Дистанційне навчання інформатики» (<http://dystosvita.mdl2.com>).
 - 2 Зареєструйтеся на курс Старша школа Інформатика 10 клас.
 - 3 Пройдіть тематичне тестування «Служби Інтернету».
 - 4 Повідомте оцінку вчителю.
 - 5 Перейдіть на сторінку «Робота з мультимедіа».
 - 6 Пройдіть тестування до розділу «Основи роботи з мультимедіа».
 - 7 Повідомте оцінку вчителю.
 - 8 Перегляньте існуючі курси на сайті «Дистанційне навчання інформатики».
 - 9 У середовищі Microsoft Excel створіть таблицю з назвами курсів, які пропонуються на сайті, поставте позначки, які курси ви хотіли б вивчати.
 - 10 Збережіть таблицю та надішліть її на вказану вчителем адресу.
- **Зробіть висновки:** яка ваша думка щодо ідеї дистанційного навчання; які курси, на вашу думку, користуватимуться найбільшим попитом; які навички стануть у пригоді в майбутньому.



Розділ 6

ОСНОВИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

6.1. Класифікація загроз інформаційній безпеці. Морально-етичні і правові засоби захисту даних



Розвиток інформаційних технологій суттєво впливає на всі сторони життя сучасної людини. Які ви знаєте способи для захисту даних у комп'ютері?



Рис. 1. Питання захисту інформації є дуже актуальним

Коли користувач вмикає комп'ютер і мандрує сторінками сайтів, здійснює покупки за електронні гроші чи використовує прикладні програми, то хоче бути впевненим, що доступ до його даних знаходиться під його контролем (рис. 1), тобто що інформація захищена.



Інформаційна безпека — це стан захищеності систем опрацювання й зберігання даних, при якому забезпечено конфіденційність, доступність і цілісність даних, або комплекс заходів, спрямованих на забезпечення захищеності даних від несанкціонованого доступу, використання, оприлюднення, руйнування, внесення змін, ознайомлення, перевірки, запису чи знищення.

Розглянемо принципи, на яких базується інформаційна безпека (рис. 2).

Цілісність означає, що в ході передавання й зберігання інформація зберігає зміст і структуру; створювати, знищувати або змінювати дані має право лише власник.

Конфіденційність означає, що доступ до конкретної інформації мають лише особи, коло яких узгоджено з власником.

Доступність означає здатність забезпечувати своєчасний і безперешкодний доступ повноправних користувачів до необхідної інформації.

Достовірність означає неможливість викривлення чи спотворення інформації.

Як ми вже з'ясували, під час користування Інтернетом часто виникають загрози інформаційній безпеці. Загрози можуть мати як природний, так і штучний характер. А вони, у свою чергу, можуть бути або випадковими, або навмисними. Загрози пошкодження даних можна класифікувати за чинниками, що їх спричиняють. Слід зазначити, що серед усіх загроз найбільший відсоток за масштабом і ступенем пошкодження даних припадає на людський чинник. Тож насамперед зазвичай розглядають захист від дій, які може спричинити людина.

Інформаційна безпека

Цілісність

Конфіденційність

Доступність

Достовірність

Рис. 2. Принципи інформаційної безпеки

Ознайомимося з видами загроз інформаційній безпеці (рис. 3) та розглянемо їхні особливості.

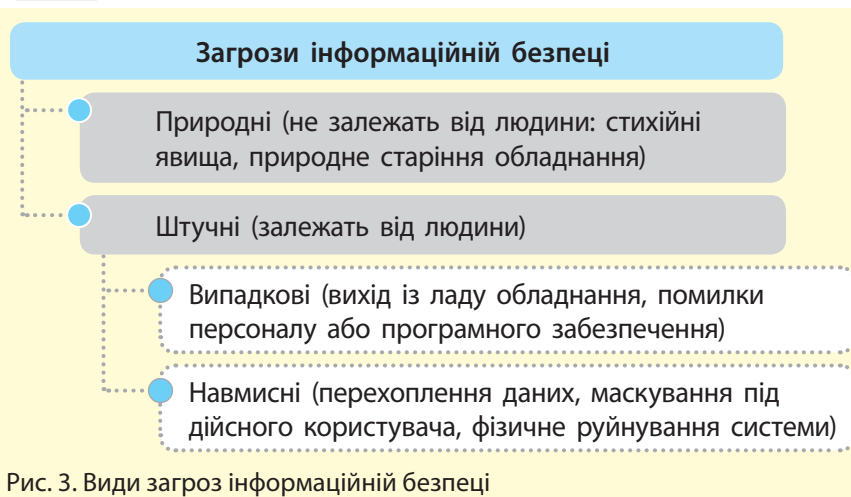


Рис. 3. Види загроз інформаційній безпеці

У сучасному суспільстві кожна людина має вільний доступ до інформації. Разом із тим важливо, щоб із розвитком інформаційних технологій забезпечувалося дотримання прав людини стосовно захисту авторських прав, інтелектуальної власності та безпеки.

Інформаційна етика розглядає проблеми власності, доступу, безпеки й спільності інформації. У світі складаються певні морально-етичні норми поведінки користувачів, наприклад, *не втручатися в роботу інших користувачів мереж; не використовувати файли, не призначені для вільного використання; не використовувати комп'ютер для розповсюдження неправдивої інформації* та ін. Ці норми здебільшого не є обов'язковими і не затверджені в законодавчому порядку. Але їх невиконання часто призводить до падіння авторитету та престижу людини, групи осіб, організації або країни.



Правові засоби захисту — чинні закони, укази та інші нормативні акти, які регламентують правила користування інформацією і відповідальність за їх порушення, захищають авторські права програмістів та регулюють інші питання використання інформаційних технологій.

Сьогодні в усіх розвинених країнах світу та багатьох міжнародних об'єднаннях ухвалено спеціальні закони щодо комп'ютерних злочинів, і вони постійно доповнюються. Загальною тенденцією є підвищення жорсткості кримінальних законів щодо осіб, які скоїли комп'ютерні злочини.

? Запитання для перевірки знань

- 1 Яку інформацію називають конфіденційною?
- 2 Що таке інформаційна безпека?
- 3 Як шкідливі програми можуть потрапляти в комп'ютер?
- 4 Які основні складові має інформаційна безпека?
- 5 Назвіть основні джерела загроз інформаційній безпеці.
- 6 Назвіть нормативно-правові документи у сфері інформаційної безпеки.



Існують як неписані морально-етичні норми, так і оформлені в деякий статут, як, наприклад, Кодекс професійної поведінки членів Асоціації користувачів комп'ютерів США.



В Україні створено Державну службу спеціального зв'язку та захисту інформації України — державний орган спеціального призначення, який опікується питаннями забезпечення формування і реалізації державної політики у сферах захисту державних інформаційно-телекомунікаційних систем, криптографічного й технічного захисту інформації, використання і захисту державних електронних інформаційних ресурсів.



В Україні захист інформації забезпечується низкою законів: «Про інформацію», «Про захист інформації в інформаційно-телекомунікаційних системах», «Про державну таємницю», «Про захист персональних даних», «Про основні засади розвитку інформаційного суспільства в Україні на 2007–2015 роки».

6.2. Захист даних. Шкідливі програми, їх типи, принципи дії і боротьба з ними



Пригадайте, які інформаційні процеси ви знаєте.



Вважають, що ідею створення комп'ютерних вірусів «підкинув» письменник-фантаст Т. Райн у 1977 році в США, він описав епідемію, яка за короткий час вразила 7000 комп'ютерів. А спричинив її вірус. Передаючись від одного комп'ютера до іншого, він убудовувався в ОС і виводив комп'ютери з-під контролю людини.

Шляхи потрапляння шкідливих програм у комп'ютер:

- під виглядом ігрових та інших програм, завантажених з мережі Інтернет
- під час перегляду веб-сторінок;
- електронною поштою
- у процесі копіювання файлів з носіїв, які до цього використовувалися на заражених комп'ютерах

Захист даних у комп'ютері є однією з найважливіших проблем сьогодення. Він передбачає як заходи щодо закриття даних від доступу до них сторонніх осіб (несанкціонованого доступу), так і збереження цілісності (структури і вмісту) даних у комп'ютері.



Захист даних (англ. *Data protection*) — сукупність методів і засобів, які забезпечують цілісність, конфіденційність і доступність даних в умовах впливу загроз природного або штучного характеру, реалізація яких може призвести до завдання шкоди власникам і користувачам інформації.

Пошкодження або руйнування даних може статися в результаті неправильної роботи апаратного чи програмного забезпечення. Так, фізичне пошкодження жорсткого диска може призвести до повної втрати даних або їх частини.

Порушення цілісності даних (зміна або вилучення окремих файлів, зміна структури файлів на диску тощо) часто спричиняється роботою некоректно написаних програм. Та найчастіше справжньою причиною збою в роботі комп'ютера є програми, спеціально написані для заподіяння шкоди даним.



Шкідлива програма — це спеціально розроблена програма для пошкодження і знищення файлів даних і програмного забезпечення, викрадення особистих даних і коштів користувача.

Розрізняють три типи шкідливих програм: *комп'ютерні віруси*, *шпигунські програми*, *хакерські утиліти* (див. [таблицю](#)).

Таблиця. ТИПИ ШКІДЛИВИХ ПРОГРАМ

Шкідливі програми	Опис дії
Комп'ютерні віруси	Шкідливі програми, що здатні до самокопіювання та вбудовування в код інших програм, системні ділянки пам'яті чи завантажувальні сектори, а також до поширення своїх копій різними каналами зв'язку.
Шпигунські програми	Потрапляють зазвичай через мережу. Не самовідтворюються, тому їх не можна назвати вірусами. Їхнє призначення — слідкувати за діями, що виконуються на комп'ютері: відправляти на сайт хакерів установлені коди, паролі входження у платіжні системи, розсилати рекламні листи за рахунок користувача. Ці програми також можуть порушувати нормальну роботу комп'ютера. До них належать троянські коні (трояни) і хробаки
Хакерські утиліти	Завдають шкоди з віддалених комп'ютерів. Не є ані троянськими платформами, ані вірусами; не можуть заподіяти шкоди локальним пристроям, на які встановлені. Можуть слугувати програмами для автоматизації формування вірусів, троянських програм і хробаків. Це програмні бібліотеки, утиліти, які приховують код «хворих» файлів від перевірки антивірусом (шифрувальники файлів)

Розглянемо особливості деяких відомих комп'ютерних вірусів.

Файлові віруси — під час поширення своїх копій обов'язково змінюють вміст виконуваних файлів. У більшості випадків вміст файлів, атакованих вірусом, повністю або частково втрачається.

Скриптові віруси — вимагають наявності однієї зі скриптових мов (Javascript, VBScript) для самостійного проникнення в неінфіковані скрипти. Віруси цього типу вражають або вихідний код програми, або її компоненти (OBJ-, LIB-, DCU-файли), а також VCL- і ActiveX-компоненти.

Стелс-віруси — повністю або частково приховують присутність у системі шляхом перехоплення запитів операційної системи на читання/записування додаткової інформації про заражені об'єкти (завантажувальні сектори, елементи файлової системи, пам'ять тощо).

Завантажувальні віруси — розміщуються у виконуваних файлах (exe, com, sys, dll) операційної системи й активізуються під час завантаження. Вони часто розміщуються у файлах документів, у яких застосовуються макроси. Тому руйнують не лише додатки, але й файли.

Бекдор, backdoor (від англ. *Back door* — чорний хід, задні двері) — дефект алгоритму, який навмисно вбудовується в нього розробником і дозволяє отримати таємний доступ до даних або віддаленого управління комп'ютером.

Дропер (англ. *Dropper* — бомбоскидач) — сімейство шкідливих програм (зазвичай трояни), скерованих на несанкціоноване та приховане від користувача встановлення на комп'ютер інших шкідливих програм, які містяться в самому тілі дропера або завантажуються мережею.

Закладка (програмна закладка) — приховано впроваджена в захищену систему програма або навмисно змінений фрагмент програми, що дозволяє здійснити несанкціонований доступ до ресурсів системи на основі зміни властивостей системи захисту. Часто виконує роль перехоплювача паролів, трафіку, а також провідника для комп'ютерних вірусів. Закладку неможливо виявити за допомогою стандартних антивірусних засобів — тільки спеціальними тестовими програмами.



У світі розроблено велику кількість антивірусних програм, і надати перевагу тій чи іншій досить важко. Проте для ОС Windows 10 існує безкоштовний убудований антивірус **Windows Defender (Захисник Windows)** компанії Microsoft.

Програма Захисник Windows встановлюється за замовчуванням. Вікно цієї програми на відкритій вкладці На початок зображено на рис. 1. Якщо на комп'ютері не було встановлено антивірусної програми, то під час встановлення програми Захисник Windows з'явиться повідомлення про потребу в перевірці. На вкладці Оновити надається можливість оновлення антивірусних баз програми. Вкладка Журнал дозволяє переглянути потенційно небезпечні об'єкти і взаємодіяти з ними. За допомогою кнопки Налаштування можна вмикати і вимикати деякі можливості програми Windows Defender.



Троянська програма SURPRISE («Сюрприз») була написана мовою Бейсик і видаляла вміст поточного диска, а потім виводила на екран повідомлення «Surprise!».

Унаслідок дії троянської програми, написаної в 1989 році американським студентом Робертом Морісом і запущеної ним в Інтернет, були заражені і виведені з ладу тисячі комп'ютерів, у тому числі ті, що належать Міністерству оборони США. Автор програми був засуджений до трьох місяців в'язниці і штрафу в 270 тис. доларів.

Правила профілактики зараження комп'ютера шкідливими програмами і збереження даних

- Користуйтеся паролями.
- Не користуйтеся піратськими копіями програмного забезпечення.
- Регулярно оновлюйте програмне забезпечення.
- Не запускайте на виконання невідомі програми.
- Не відкривайте файли, не перевірені на наявність вірусів.
- Встановлюйте програму захисту від вірусів, що постійно оновлює свою базу знань.



До популярних комерційних антивірусних програм належать ESET NOD 32, BitDefender, Kaspersky Internet Security тощо. Майже не поступаються їм за ефективністю безкоштовні програми: 360 Total Security, Panda Free Antivirus, Avast Free Antivirus, Avira Free Antivirus та ін.

Щоб перевірити окрему папку або файл за допомогою програми Захисник Windows, потрібно у файловому провіднику відкрити контекстне меню папки (файла), виконати команду Перевірка з використанням Windows Defender.

За допомогою програми можна здійснити як повну перевірку програм і даних, розташованих у комп'ютері, так і перевірку окремих частин, наприклад, диска, папки тощо (експрес-сканування).

Після виконання перевірки зазначеної папки відкриється вікно з результатами перевірки, можливий варіант якого зображено на рис. 2.

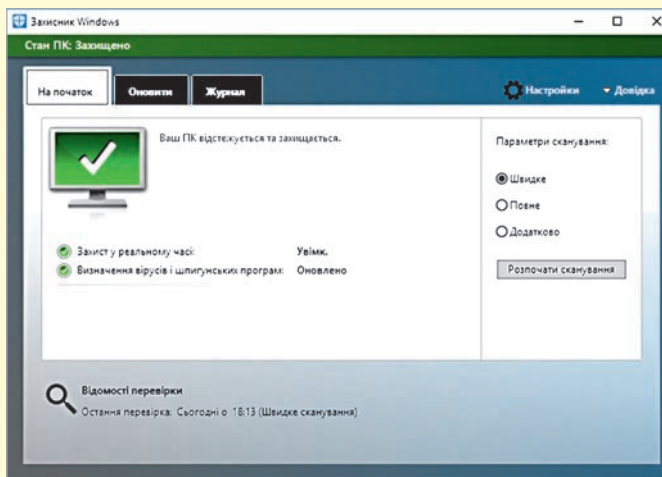


Рис. 1. Вікно програми Захисник Windows на вкладці На початок

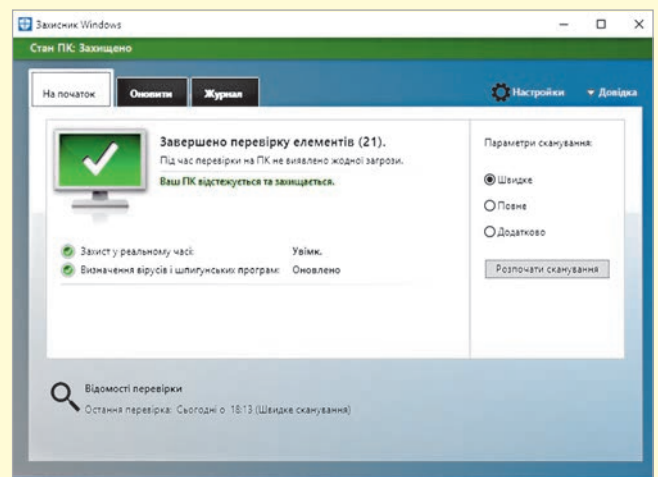


Рис. 2. Вікно програми Захисник Windows із результатами перевірки папки

Щоб **вимкнути** або **увімкнути** програму Захисник Windows потрібно виконати команду Пуск → Налаштування → Оновлення та захист, вибрати пункт Захисник Windows і увімкнути або вимкнути параметр Захист у реальному часі.



Запитання для перевірки знань

- 1 Що розуміють під захистом даних?
- 2 Які є типи шкідливих програм?
- 3 Наведіть приклади комп'ютерних вірусів.
- 4 Чим відрізняються віруси від шпигунських програм?
- 5 Яких правил слід дотримуватися для забезпечення комп'ютерної безпеки?
- 6 Чим відрізняється експрес-сканування від повного сканування?



Завдання для самостійного виконання

- 1 Виконайте за допомогою програми **Захисник Windows** сканування двох будь-яких непорожніх папок або файлів, розташованих на вашому комп'ютері. Проаналізуйте отримані результати.
- 2 Виконайте за допомогою антивірусної програми експрес-сканування жорсткого диска до повного його завершення. Скільки часу виконувалося сканування? Проаналізуйте його результати.

6.3. Загрози при роботі в Інтернеті. Засоби браузера, призначені для гарантування безпеки

Чи може зараження комп'ютера шкідливими програмами статися під час роботи в Інтернеті?



Активування більшості шкідливих програм вимагає безпосередньої участі користувача. Загальновідомо, що коли на незнайомому сайті пропонують щось завантажити, то слід негайно відмовитися. Деякі зловмисники цілеспрямовано полюють на приватну інформацію користувача (банківські реквізити, електронне листування та ін.), деякі намагаються отримати віддалений доступ до персональних пристроїв — комп'ютерів, планшетів або мобільних телефонів.

Розглянемо засоби браузерів, призначені для захисту інформації користувача, на прикладі найбільш потужного і популярного браузера Google Chrome.

У Google Chrome передбачено захист акаунта Gmail від потенційно небезпечних листів, які можуть містити шкідливі програми та попередження в разі спроби завантажити файл, у якому під виглядом рінгтона або PDF-документа поширюється небезпечний код.

У Google Chrome використовуються додаткові модулі (плагіни), які дозволяють користувачу переглядати мультимедійні файли або відео. Встановлені плагіни також можуть використовуватися зловмисниками для злому комп'ютера. Тому Google Chrome, виявивши застарілий плагін із вразливою системою безпеки, пропонує користувачу виконати оновлення. І блокує плагін, поки користувач цього не зробить. Якщо користувач спробує відкрити небезпечний сайт, Google Chrome попередить про це.

Небезпечними вважаються сайти, підозрювані у фішингу або поширенні спаму, що є окремими видами комп'ютерних загроз.

Розглянемо особливості фішингу і спаму та засоби захисту від них.

Спам — рекламні та інші повідомлення (наприклад, про нові послуги, варіанти відпочинку, дешеві товари тощо), які надходять на поштову скриньку осіб, які не давали дозвіл на їх отримання. Окремі спам-повідомлення можуть нести комп'ютерні віруси. Ніколи не слід відповідати на спам-листи та переходити за посиланнями, які в них містяться.

Фішинг — масове розсилання електронних листів від імені популярних брендів із метою отримання доступу до логіна і пароля користувача, банківських рахунків та інших персональних даних.

У листах може міститися посилання на сайт, який зовнішньо майже не відрізняється від справжнього. Після відкриття проблемної веб-сторінки шахраї намагаються в різний спосіб примусити користувача увести логін і пароль.

З метою підвищення надійності захисту від фішингу виробники основних браузерів домовилися про застосування



Через інтерфейс браузера шкідливі програми можуть потрапляти до комп'ютера з різних плагінів типу Flash, Java і QuickTime. Тому необхідно оновлювати не тільки сам браузер, а й усі зазначені компоненти.

Вбудовані засоби захисту від загроз мають Mozilla Firefox, Google Chrome, Safari, Opera.



Google виявляє веб-сторінки, які можуть завдати шкоди. Щодня знаходить і позначає понад 10 тис. небезпечних сайтів, виводить попередження для 14 млн результатів пошуку Google і 300 тис. завантажень.



Захищеним є сайт, який використовує для обміну даними протоколи захищеного зв'язку. Підтримка захищених каналів зв'язку потрібна насамперед сайтам, які працюють із персональною інформацією користувачів (сайти банків, урядових установ тощо).



Позначка захищеного сайта

Технологія SSL забезпечує кращий захист і збереження конфіденційності інформації, на відміну від незашифрованого веб-з'єднання.

Протокол забезпечує конфіденційність обміну даними між клієнтом і сервером, які використовують TCP/IP.

однакових способів інформування користувачів про перехід на підозрілий сайт. Ознайомимося з поділом підозрілих сайтів.

- **Підроблені сайти** — метою таких сайтів є фішинг та інші види соціальної інженерії. Вони обманом намагаються отримати доступ до особистих даних і паролів користувачів.
- **Небезпечні сайти** — мета таких сайтів полягає в установленні на пристрій користувача шкідливого або небажаного програмного забезпечення, яке може порушити роботу системи і викликати неполадки в мережі.

Зараз у мережі Інтернет існує немало підозрілих сайтів. Посилання на такі ресурси можуть поширюватися в різний спосіб, наприклад електронною поштою.

Щоб **забезпечити безпечний перегляд**, браузері складають список сайтів, які підозрюються у фішингу та поширенні шкідливих програм. Вони аналізують усі веб-сторінки і попереджають користувача, якщо сайт становить загрозу.

Зверніть увагу на адресу певного сайта. Для передавання гіпертекстових даних мережею Інтернет часто використовується протокол HTTP (*Hyper Text Transfer Protocol*). Але з'єднанням за цим протоколом не є захищеним.

Існує також HTTPS (*Hypertext Transport Protocol Secure*) — протокол зашифрованого під'єднання, який забезпечує більш ефективний захист даних. У деяких браузерах поряд із позначенням протоколу `https://` відображається *значок замка* (див. рисунок). Це означає, що з'єднання є захищеним і більш безпечним.

Однак досвід користування деякими ресурсами свідчить, що це не завжди так. Зображення замка, який є на сторінках таких ресурсів, можна вважати простою картинкою. Це означає, що в сайта відсутній сертифікат справжності, не підтримується і SSL-з'єднання.

SSL (англ. *Secure Sockets Layer* — рівень захищених сокетів) — криптографічний протокол, який гарантує встановлення безпечного з'єднання між клієнтом і сервером.

SSL-сертифікат — це сертифікат безпеки; цифровий спосіб ідентифікації, який підтверджено Центром сертифікації. У сертифікаті міститься ідентифікаційна інформація, період дії, публічний ключ, серійний номер і цифрові підписи.

Сучасні браузери мають вбудовану систему безпеки, підтверджену сертифікатами, отриманими в спеціальних центрах. Система браузера вважатиме сертифікат справжнім, якщо центр сертифікації, який його видав, є в переліку надійних закладів.

Якщо з'єднання, яке потребує введення конфіденційної інформації, є незахищеним, то слід скористатися іншим ресурсом.



Запитання для перевірки знань

- 1 Які існують типи шкідливих програм?
- 2 Чим відрізняються віруси від шпигунських програм?
- 3 Які сайти вважаються підозрілими? Опишіть їх.
- 4 Що таке протокол зашифрованого під'єднання? Чому і де він використовується?

Практична робота № 13

Тема. Налаштовування параметрів безпеки в середовищі браузера.

Завдання: налаштувати параметри безпеки в середовищі браузера Google Chrome.

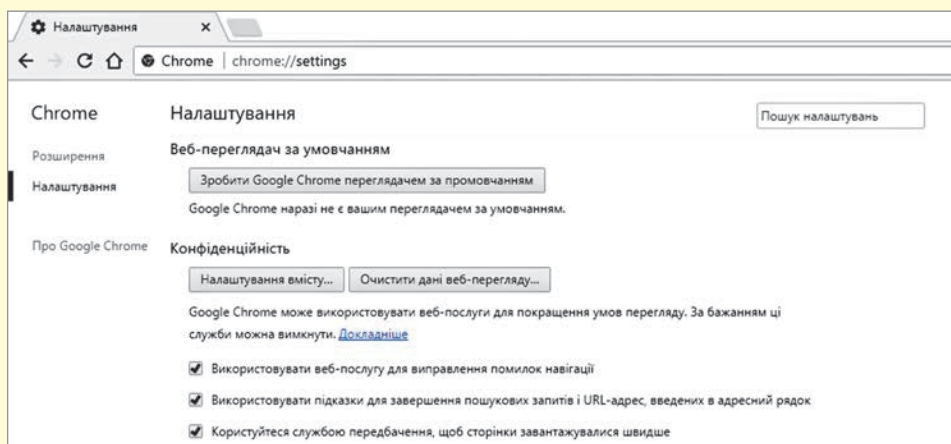
Обладнання: комп'ютер зі встановленим браузером Google Chrome.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Відкрийте браузер Google Chrome.
 - 2 Відкрийте меню Налаштування.
 - 3 Виконайте команду Показати розширені налаштування.
 - 4 Налаштуйте послугу Захистіть себе та свій пристрій від небезпечних сайтів.
 - 5 Виберіть команду Від'єднатися від облікового запису Google, оскільки синхронізація акаунта з браузером означає, що особисті дані користувача зберігаються на серверах Google.
 - 6 Натиснувши кнопку Розширені налаштування синхронізації, виберіть команду Шифрувати синхронізовані дані за допомогою власної пароліної фрази.
 - 7 Налаштуйте вміст. Для цього під рядком Конфіденційність (див. [рисунок](#)) натисніть кнопку Налаштування вмісту та виберіть команду Блокувати файли cookie. Це забезпечить видалення ваших cookie після закриття браузера, і рекламодавці не зможуть відстежувати вас, використовуючи cookie третіх сторонніх сайтів.

Примітка. **Куки** (Cookie— печиво) — невеликий текстовий файл (набір даних), який сервер відвідуваного сайту записує в браузер користувача. У файлі зберігається інформація про ім'я та пароль, індивідуальні налаштування та вподобання користувача, статистику відвідування. Щоразу при відвідуванні сайту браузер надсилає серверу такий файл для ідентифікації користувача. Саме завдяки кукам користувач на власному комп'ютері відразу потрапляє у свій обліковий запис, авторизація відбувається автоматично. З точки зору безпеки є деякі недоліки: стороння людина може відвідувати форуми та сайти під іменем власника комп'ютера, за куками можна відслідковувати, які сайти і коли було відвідано.
 - 8 Виберіть опцію Заборонити всім сайтам запускати JavaScript для захисту.
 - 9 Виберіть опцію Заборонити всім сайтам відображати спливаючі вікна.
 - 10 Виберіть опцію Заборонити всім сайтам відстежувати ваше фізичне місцезнаходження.
 - 11 Налаштуйте автозаповнення. Для цього зніміть прапорець із команди Увімкнути автозаповнення. Це допоможе збереженню браузером ваших логінів, паролів та іншої конфіденційної інформації, введених у форми веб-сайтів.
 - 12 Збережіть налаштування Google.
- Зробіть висновки:** як налаштовувати параметри безпеки в середовищі браузера Google Chrome.



6.4. Безпечне зберігання та видалення даних. Резервне копіювання та відновлення даних



Пригадайте, що таке файлова система.

Правила створення кількох облікових записів

- Лише один з облікових записів повинен мати права адміністратора — права встановлювати та видаляти програмне забезпечення.
- Теки, що містять програмне забезпечення, мають бути доступними для читання усіх користувачів, окрім адміністратора, з можливим виключенням для папок із налаштуваннями програми.
- Кожний обліковий запис повинен мати папку з повним доступом лише для цього облікового запису та адміністратора.

Працюючи за комп'ютером, користувач впевнений, що всі опрацьовувані дані надійно зберігаються і ніщо не завадить продовжувати з ними працювати. Проте досить часто виникають проблеми з доступом до даних. Наприклад, після оновлення програм або встановлення нових пристроїв не завантажуються ОС, з'являються сторонні повідомлення, зникають файли тощо. Щоб уникнути подібного, слід дотримуватися *правил безпечного зберігання даних*.

Насамперед слід розділити жорсткий диск на декілька розділів (логічних дисків) з тим, щоб на одному (системному) зберігати програмне забезпечення й системні дані, на інших — решту. Навіть на власному комп'ютері краще створити кілька облікових записів, дотримуючись певних правил.

Виходячи з того, що однією з основних причин втрати даних (після ненавмисного видалення) є наслідки дій шкідливих програм, дуже важливо використовувати антивірус.

Ще одним важливим правилом безпечного зберігання даних є створення резервних копій (див. [рисунок](#)). Відмова апаратної частини (жорсткий диск), вірусна атака чи неухважність самого користувача (випадкове видалення інформації) можуть призвести до втрати важливих даних.



Резервне копіювання даних — це створення копій важливої інформації, що є на комп'ютері, для її збереження в інших сховищах даних (флеш-накопичувач, жорсткий диск, DVD-диск, хмарний сервіс тощо).

Типи резервних копій

- резервна копія ОС (установити ОС, драйвери й потрібні програми, зробити резервну копію ОС)
- резервна копія логічного диска (розділу)
- резервна копія окремих файлів і папок — найпоширеніший спосіб резервного копіювання.

Рис. 2. Резервне копіювання даних

Резервне копіювання можна виконувати за допомогою спеціальних утиліт, які забезпечують створення компактних архівів. Наприклад, одною з таких утиліт є Microsoft Backup, що входить до комплекту Windows. Щоб скористатися нею, на Панелі керування потрібно вибрати відповідне посилання.

Однією з альтернатив резервного копіювання, наприклад, під час роботи з великими проектами (науковими роботами, масштабними кресленнями тощо), може служити створення контрольних точок відновлення системи.



Точка відновлення — це подання збереженого стану системних файлів комп'ютера.

У разі непередбачуваної ситуації, наприклад дій шкідливого програмного забезпечення, можна «відкотитися» до стану системи з усіма параметрами й файлами на період збереження. Для цього не обов'язково встановлювати додаткове програмне



Acronis True Image — лідер серед програм для резервного копіювання. Вона дозволяє зберігати файли і папки, розділи диска повністю або робити образ розділу з ОС, програмами і конфігурацією. Ці дії можна налаштувати на автоматичне і періодичне виконання. Якщо ОС пошкоджено, можна завантажити програму з флеш-накопичувача (або іншого носія) й відновити резервну копію.



Під час інсталювання дані про програму додаються до системних файлів, реєстру ОС. Окремі компоненти програм можуть копіюватися до інших системних папок. Тому щоб видалити інсталювані програми, недостатньо просто видалити файли програми.

забезпечення. Сучасні операційні системи вже мають таку функцію як вбудовану, що дозволить не втратити повністю всю інформацію. Однак все, що було змінено, навіть за умови збереження після створення контрольної точки, буде втрачено. Це найголовніший недолік такого способу страхування. Проте неповне відтворення краще, ніж повна втрата даних.

Для роботи з утилітою Відновлення системи необхідно зберегти відкриті файли та закрити всі програми; клацнути кнопку Пуск, у головному меню вибрати Програми → Стандартні → Системні інструменти (Службові) → Відновлення системи або вибрати відповідну дію на Панелі керування.

Програми, що стали причиною збоїв, слід видалити за допомогою утиліти Програми та засоби, яку запускають із Панелі керування.

Пригадайте, як встановити програму на комп'ютер.

У разі видалення дані фізично залишаються на накопичувачі, проте у файлової системі більше не відображаються. Місце на носії, де містяться ці дані, позначається як вільне для запису нових даних. Якщо в даний розділ або логічний диск записати дані, може статися часткове або повне заміщення даних, позначених як видалені. Це стосується текстових, графічних та мультимедійних даних.



Просте видалення папки з назвою програми з системного диска є некоректним і може призвести до збоїв у роботі операційної системи та накопичення непотрібних файлів.

Вибір режиму видалення (деінсталяції) запускає спеціальну програму, яка коректно видаляє всі компоненти програми.

Розглянемо *алгоритм видалення встановлених програм*.

Крок 1	Виконайте команду Пуск → Параметри → Система → Застосунки та можливості.
Крок 2	У вікні, що з'явилося, виберіть зі списку інсталюваних програм потрібну.
Крок 3	Натисніть на кнопку Видалити .
Крок 4	Підтвердьте бажання видалити програму вибором кнопки Продовжити .
Крок 5	Дотримуйтесь вказівок програми деінсталяції.



Запитання для перевірки знань

- 1 Назвіть правила безпечного зберігання даних.
- 2 Коли операційна система потребує відновлення?
- 3 Що таке точка відновлення?
- 4 Яка мета резервного копіювання?
- 5 Як коректно видалити програму, встановлену на комп'ютері?
- 6 Чи можна відновити дані, переміщені в папку **Кошик**, після очищення цієї папки?



Розділ 7. СТВОРЕННЯ ПЕРСОНАЛЬНОГО НАВЧАЛЬНОГО СЕРЕДОВИЩА

7.1. Поняття персонального навчального середовища. Використання інтернет-середовищ для створення та публікації документів



Пригадайте, які послуги надає мережа Інтернет. Як зберегти файл, створений удома, для перегляду та редагування в класі?



LMS — система управління навчальною діяльністю, яка використовується для розробки, управління та поширення навчальних онлайн-матеріалів із забезпеченням спільного доступу. Серед її можливостей: виконання індивідуальних завдань, проектів для організації сумісної роботи в малих групах тощо.

Завдяки розвитку інформаційних технологій мережа Інтернет і сучасні мобільні пристрої зв'язку стають усе більш доступними.



Інтернет містить сервіси для створення, спільного редагування та обміну документами, створення віртуальних навчальних груп, має засоби для забезпечення аудіовізуальної комунікації, наприклад для проведення інтернет-конференцій і мережевих семінарів (вебінарів).

Спільна навчальна діяльність за допомогою Інтернету здійснюється за двома напрямками: з використанням систем управління навчанням (LMS — Learning Management System) та в персональному навчальному середовищі (PLE — Personal Learning Environments) (ПНС).

Якщо дистанційні курси предметно-орієнтовані (зорієнтовані на вивчення одного курсу, циклу курсів), то персональне навчальне середовище (ПНС) зорієнтоване на особу, яка навчається.



Персональне навчальне середовище — це інформаційне середовище, призначене для індивідуального використання особою, яка самостійно керує своїм навчанням для досягнення визначеної навчальної мети.

Прикладом LMS є дистанційні курси, розроблені з використанням таких ресурсів, як MOODLE, Black Board, Learning Space тощо.

Одним із перших великих центрів дистанційного навчання стала Хан-Академія (<https://buki.com.ua/videos/>). Поштовхом до створення стали відеоуроки її автора Салмана Хана, американського викладача й підприємця, які у YouTube переглянули мільйони людей.

Типове ПНС охоплює весь набір ресурсів, за допомогою яких можна забезпечити зміст навчання та зробити його процес більш наочним, створювати зворотний зв'язок із колегами та друзями. ПНС має великі можливості для самоосвіти, дозволяє вчитися дистанційно, контролювати свої навчальні ресурси (змінювати їхню структуру, інтегрувати навчальні ресурси з різних установ і джерел тощо).

ПНС складають сукупність апаратних засобів, сервісів, які підтримують роботу над спільним проектом, і програмних засобів для навчання. Вважають, що ПНС є результатом розвитку Веб 2.0 і його впливу на навчальний процес.

Пригадайте, які сервіси має Веб 2.0. Які месенджери ви знаєте?



Особливе місце у створенні ПНС займають сервіси Веб 2.0. Такі сервіси Веб 2.0, як блог, вікі, Delicious, YouTube, зручні у використанні та мають властивості (доступність, дружність, безкоштовність тощо), які дозволяють ефективно застосовувати їх у навчальному процесі.

Блог (від англ. *blog, web log*) — інтернет-журнал подій, інтернет-щоденник, у який регулярно додаються записи у вигляді тексту, зображень або мультимедіа. Від особистого щоденника відрізняється тим, що до його перегляду мають доступ читачі, які можуть залишати коментарі. У блозі передбачена можливість показувати або приховувати певний матеріал за бажанням власника блогу.

Створення блогу, його ведення (оновлення) відбувається за допомогою спеціальних мережесервісів, які називають **блог-платформою**. Це повністю автоматизований інтерактивний сайт, на якому користувач створює й веде власний сайт.

Створення блогу на блог-платформі є простим і доступним кожному користувачу процесом, який не передбачає знання програмування і вміння створювати сайти. Популярними безкоштовними блог-платформами є Google Blogger (blogger.com), [livejournal](http://livejournal.com) (livejournal.com) та ін.

Вікі (WikiWiki) — веб-сайт, структуру і зміст якого користувачі можуть спільно змінювати за допомогою інструментів, які й надаються сайтом. Найбільшим відомим і поширеним вікі-сайтом є Вікіпедія (wikipedia.com).

Prezi.com — сервіс для створення презентацій, головною особливістю яких є будова «хмаринками». Сервіс дуже зручний у використанні, принципи роботи в його середовищі такі самі, як і в середовищі офісних систем презентацій.

Delicious (від англ. *delicious* — чудовий) — сайт, який безкоштовно надає зареєстрованим користувачам послуги зберігання й публікації закладок на дібрані з певною метою сторінки Всесвітньої мережі. (Про закладки та використання цього сайту для створення закладок ви дізнаєтесь у § 7.3.)

Для створення блогу на блог-платформі Google Blogger необхідно мати обліковий запис у Google. Якщо ви маєте доступ (обліковий запис) до одного із сервісів Google, то можете користуватися можливостями всіх інших сервісів Google. Згадаємо, що *обліковий запис* — це об'єднана система входу, яка дозволяє отримати доступ до різноманітних послуг сервісу.

Розглянемо процес створення блоку на сервісі Blogger.com.

Для знайомства з можливостями створення блогу виконаємо завдання (с. 123).



Пост — повідомлення блогу (містить заголовок, дату публікації, зміст). Користувачів, які ведуть блоги, називають блогерами.

Подкаст (Podcast) — інноваційний спосіб проведення занять у формі записаних аудіо- та відеофрагментів.



Крім Prezi, для створення онлайн-презентацій і відео використовують такі сервіси, як Google Slides (є складовою Google Docs), PowerPoint Online, YouTube Video Editor, SlideShare (YouTube для презентацій — частина LinkedIn), Adobe Voice, Office Sway (онлайн-додаток у складі Microsoft Office).

Професійними платформами є WordPress, Joomla (за хостинг блогу необхідно вносити оплату) тощо. Перевагою Wordpress.com є взаємозв'язок із програмами соціальних медіасередовищ: YouTube.com, SlideShare.net, Vimeo.com.

На сайтах Ukr.Media, Ukr.Net, Yandex.ua дозволяється безкоштовно створити власний блог, проте його функціональність буде зменшеною.

Для розміщення блогу на серверах є українські безкоштовні хостинги, які спеціалізуються на блогах (TopUa — сервери хостингу знаходяться у Німеччині; Ho.ua, InfiumHost — сервери хостингів знаходяться в Україні).



Гаджет — додатковий функціональний засіб, який власник може розмістити на своєму блозі (практично, це складова блогу з визначеними функціями) — сторінки, список посилань, мітки, канал, логотип тощо.

Для проектування вигляду блогу користуються пунктом меню Макет, у якому вибирають команду Усі стовпці → Додати гаджет (рис. 1).

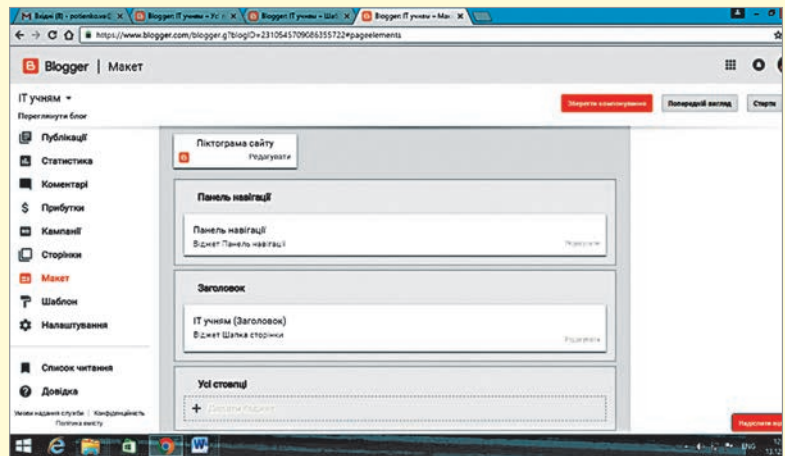


Рис. 1. Команди меню групи Макет

Blogger.com надає також можливість створення онлайн-опитування для з'ясування ставлення користувачів до різних питань. Вікно для створення форм опитування має інтуїтивно зрозумілий інтерфейс. Розглянемо, які є поля для заповнення у вікні Налаштування опитування (рис. 2).

Для введення питання призначено поле **Запитання**

Для введення варіантів відповідей призначені поля **Відповіді**

Для розміщення додаткового варіанта відповіді користуються командою **Додати іншу відповідь**

Якщо питання містить кілька правильних відповідей, необхідно поставити «галочку» біля команди **Дозволити відвідувачам вибирати кілька відповідей**

Форма містить поля для фіксування дати та часу заповнення форми відвідувачем

Рис. 2. Вікно форми опитування

? Запитання для перевірки знань

- 1 Які види освітніх інтернет-середовищ ви знаєте?
- 2 Що означає поняття «персональне освітнє середовище»?
- 3 Які інтернет-сервіси використовують в освіті?
- 4 Які особливості має блог?
- 5 Як створити форму опитування за допомогою інтернет-сервіса?
- 6 З'ясуйте, чим блог відрізняється від сайта.

Завдання для самостійного виконання

- 1 Перейдіть за посиланням <http://www.blogger.com>.
 - 2 Виконайте команду **Створити блог**. У вікні (рис. 3) вкажіть назву блогу, його майбутню адресу в Інтернеті (бажано, щоб адреса містила тематику блогу).
 - 3 Виберіть шаблон блогу. Натисніть кнопку **Створити блог**.
 - 4 Для наповнення сторінки інформацією в колонці меню виберіть **Публікації** та активуйте команду **Новий допис** — у результаті відкриється вікно редагування, в робочу зону якого можна ввести текст, графічні та відеозображення, створити посилання тощо.
- Вікно також містить меню з інструментами для роботи команд редагування символів, абзаців (рис. 4).
- 5 Додайте до блогу текст, підберіть і розмістіть графічне зображення. Виберіть команду **Переглянути**, і, якщо вас задовольняє вигляд блогу,— команди **Зберегти**, **Закрити**.
 - 6 Перейдіть у меню **Налаштування**, активуйте команду **Дописи**, коментарі та спільний доступ. Налаштуйте доступ до свого блогу. (У налаштуваннях автоматично створюються розділи «Архів блога», «Постійні читачі», «Про мене».) Збережіть блог.

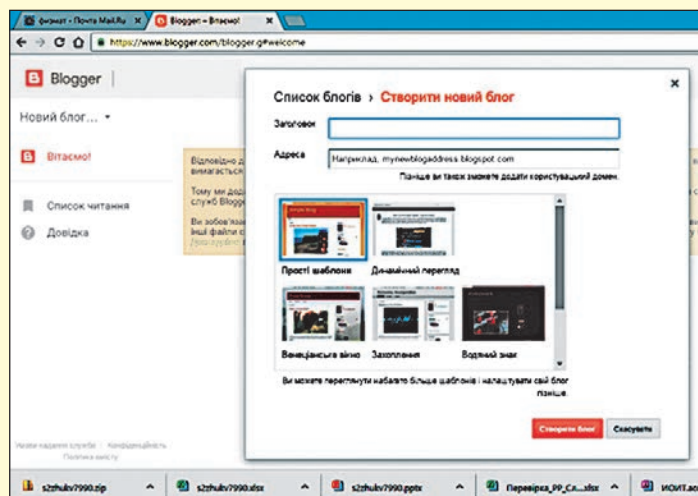


Рис. 3. Вікно створення блогу

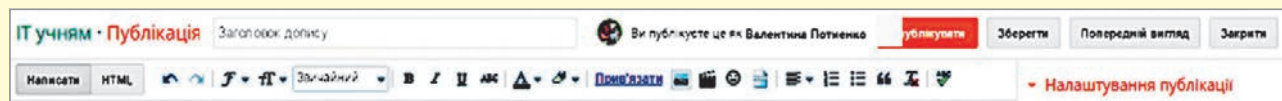


Рис. 4. Інструменти меню

7.2. Організація та планування колективної діяльності. Використання офісних веб-програм для створення спільних документів



Для спільного створення й редагування файлів існують онлайн-сервіси, наприклад Google Docs, Zoho Docs, Microsoft Office 365, Stixy, Dabbleboard. Вони дозволяють створювати текстові документи, електронні таблиці, презентації, обмінюватися ними, можуть використовуватися як онлайн-сховище.

Можливості онлайн-сервісів для роботи з документами:

- дозволяє працювати з документами відразу кільком особам на рівні доступу до документів
- підтримує велику кількість популярних форматів (наприклад, DOCX, PDF, ODT, RTF, TXT, HTML)
- зберігає історію редагування, що дозволяє повернутися на кілька кроків назад (до більш ранньої версії)
- дозволяє створювати й редагувати документи на Google iPhone, iPad і Android
- надає можливість працювати в режимі офлайн (без Інтернету); файли доступні в автономному режимі, після під'єднання до мережі автоматично оновлюються на сервері

Чи доводилося вам доопрацьовувати в класі документ, створений удома? Чи брали ви участь у колективній роботі зі створення документів у програмах, які входять до офісних пакетів? Опишіть особливості колективної роботи.

Ви вже знайомі з сервісами Інтернету, в середовищі яких можна створити документ для перегляду іншими особами з можливістю залишити в ньому коментарі. Такий документ створюється власником, це може бути одна особа, а може бути й кілька осіб.



Колективною діяльністю є діяльність групи з кількох осіб, спрямована на отримання результату. Якщо відбувається колективна діяльність зі створення та опрацювання документів, то такі документи називають **спільними**.

Оскільки файли зберігаються на спеціальних серверах, доступ до цих файлів надається користувачам зі спеціальними правами доступу (можна або лише передивлятися документи, або ще й редагувати їх, залишати коментарі, спілкуватися безпосередньо в документі).

Доступ до ресурсів сервера та спільне використання програмного забезпечення є важливим аспектом колективної роботи, адже на комп'ютерах може бути встановлено різні версії програм. Це може створити проблеми під час спільного опрацювання файлів.

Розглянемо можливості сервіса Google для колективної роботи з документами різного типу.



Google Документи — безкоштовний онлайн-офіс, який надає послуги створення й опрацювання текстових документів, таблиць (аналогів електронних таблиць) та презентацій, а також інтернет-сервіс хмарного зберігання файлів із функціями файлообміну.

Щоб увійти на веб-сторінку за адресою <https://docs.google.com>, необхідно ввести логін і пароль вашого облікового запису в Google. Якщо облікового запису Google немає, то Google пропонує зареєструватися, перейшовши за посиланням Створити обліковий запис, після чого у браузері відкривається вікно Google Документ з елементами, наведеними на рис. 1.

Зазвичай робоча зона містить документи, з якими користувач вже працював у сервісі Google. Якщо ж користувач вперше відкрив Google Документ, робоча зона буде порожньою.

Щоб створити новий документ, потрібно скористатися кнопкою створення нового документа (червоною кнопкою зі знаком + у правому нижньому куті робочої зони).

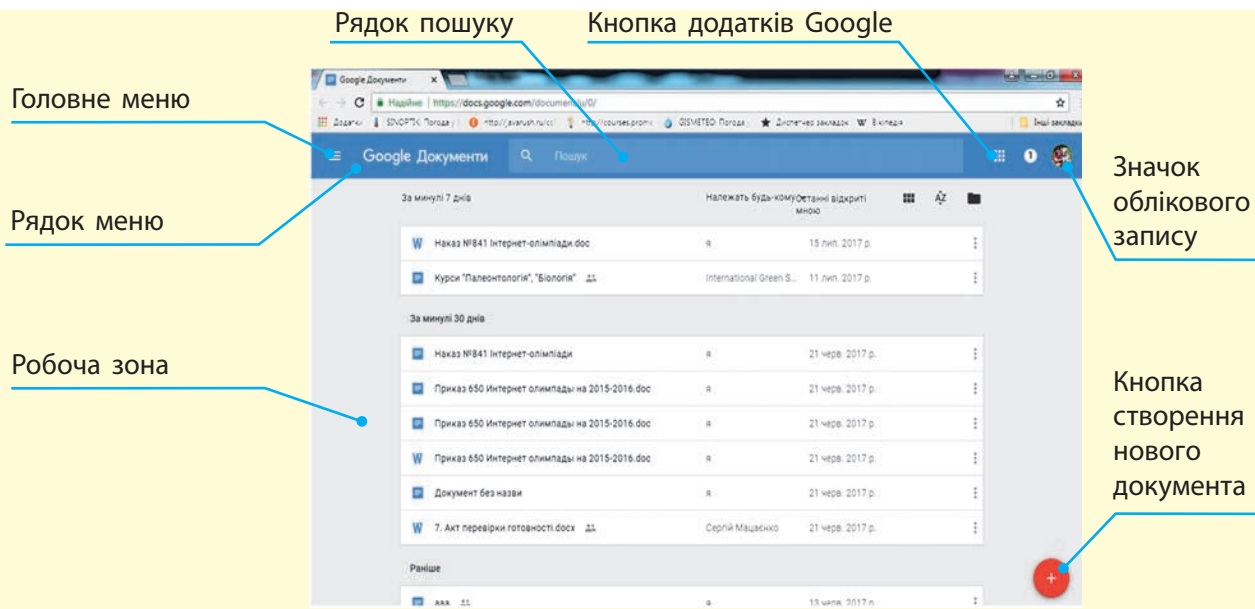


Рис. 1. Вікно Google Docs

Роботу з новим документом або редагування вже існуючого документа також можна розпочати за допомогою команд додатку Google-документів: Файл → Створити → створити новий файл або Файл → Відкрити → Імпортувати існуючий файл. Після цього слід вибрати команду Завантажити.

Сервіс Google Документ призначено для створення й опрацювання текстових документів, таблиць (аналог електронних таблиць) і презентацій. Щоб визначити тип документа, слід скористатися кнопкою головного меню з одноіменними командами. Головне меню також містить команди налаштування мови і режиму оновлення документа після його опрацювання в автономному режимі. Але для цього необхідно працювати у браузері Google Chrome.

На рис. 2 наведено вікна завантаження режиму роботи з документами, таблицями або презентаціями відповідно. Всі додатки містять рядок меню з командами опрацювання документа вибраного типу.

Сервісом передбачено як створення нового документа, так і завантаження існуючого — сервіс працює з такими типами форматів файлів, як DOC (DOCX), XLS (XLSX), ODT, ODS, RTF, CSV, PPT. Це веб-орієнтоване програмне забезпечення, тобто воно є програмою, яка працює в рамках веб-браузера без інсталяції на комп'ютер користувача. Такі програми називаються веб-програмами.

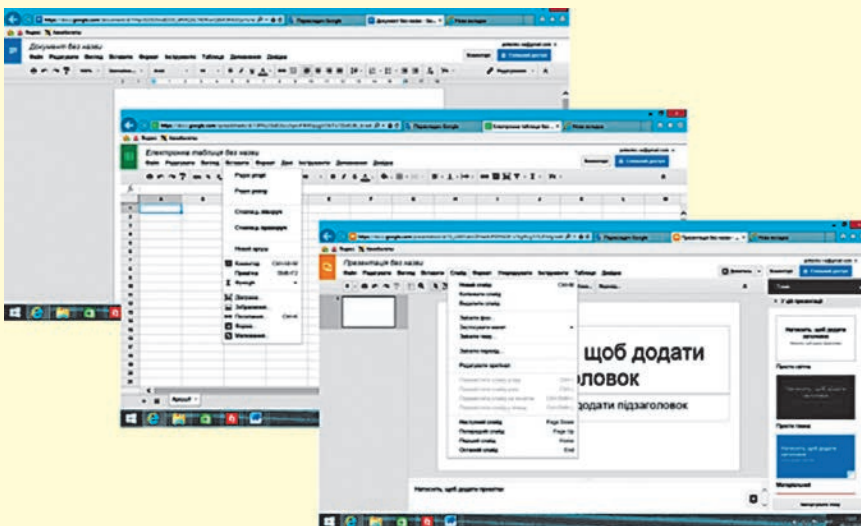


Рис. 2. Вікна документів різного типу

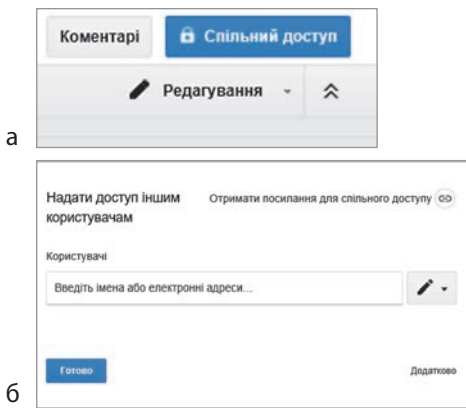


Рис. 3. Вікно налаштування доступу

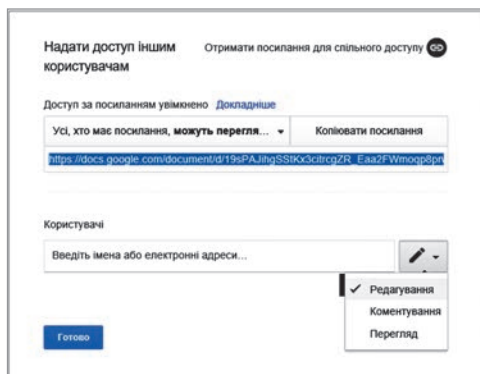


Рис. 4. Область вікна для налаштування прав доступу

Створений документ містить один аркуш. Для додавання нового аркуша вибирають «+» на нижній панелі.

Після створення нового документа (будемо називати документом і текстовий файл, і таблицю, і презентацію) Google запропонує встановити спільний доступ до файла, що опрацьовується. У правій частині рядка меню з'являється команда встановлення режиму доступу. При переході до цієї команди з'являється вікно Надати доступ іншим користувачам (рис. 3, а). У ньому визначається адреса посилання на документ (рис. 3, б) та налаштовуються права облікових записів користувачів для спільної роботи (рис. 4): перегляд, редагування тощо.



Зверніть увагу на те, що членам колективу, який працює над створенням документа, варто надати право редагування, а решті користувачів, які знатимуть адресу розміщення документа,— право перегляду.

Для публікації готових робіт в Інтернеті користуються командою Файл → Опублікувати в Інтернеті. Решта команд меню вам знайомі, оскільки такі самі команди містять офісні програми.

Сервіс Google Документи також дозволяє створювати форми. Google-форми призначені для створення форм опитування, які надсилають передплатникам розсилки або вбудовують у сторінки сайта. Сервісом передбачено створення статистики по відповідях у зручній формі.

Розглянемо порядок дій для створення форми за допомогою Google-сервісу.

Приступаючи до роботи, слід вибрати команду Таблиця, а потім команду Вставити → Форма — з'являється вікно створення форми (рис. 5). Якщо в адресному полі браузера набрати docs.google.com/forms, то відбудеться звернення одразу до сторінки створення форми. Далі потрібно ввести назву форми в рядок заголовка і почати роботу з питаннями та відповідями до них.

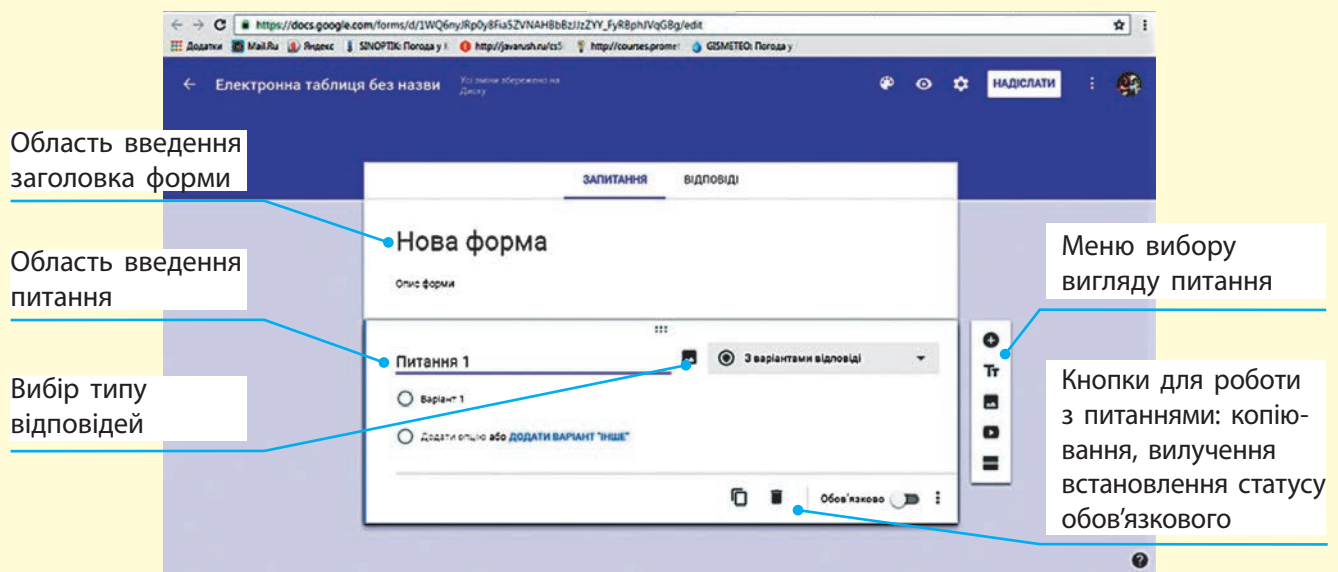


Рис. 5. Вікно створення форми сервісом Google

Праворуч від бланка форми міститься меню для вибору вигляду питання: текстове, у вигляді графічного або відео-зображення. Питання можуть бути розподілені за розділами (остання команда меню); для додавання нового питання призначена кнопка із знаком «+».

Після введення питання відкривають список. Для кожного питання вибирають тип відповідей, який визначає тип питання: множинний вибір, текст, текст абзацу, прапорці, шкала, сітка, дата, час (рис. 6).

Команди, що містяться у правому нижньому куті вікна, надають можливості копіювання, вилучення та опрацювання питань.

За потреби питанню можна встановити *статус обов'язкового* — після створення форми воно буде позначено червоною зірочкою. Якщо користувач не вводить відповідь до такого питання, заповнена форма не надсилається. Також можна перемішувати варіанти, якщо у формі створюються тести для широкого кола осіб, які одночасно дають відповіді на питання.

Створивши форму, її зберігають як таблицю або одразу розсилають користувачам: після запуску команди Надіслати з'являється вікно (рис. 7), у поля якого вводять дані.

Меню роботи з таблицями містить пункт Форми, в якому розміщені команди редагування поточної форми, надсилання або розміщення на сайті, перегляду відповідей.

Статистику відповідей, у тому числі у вигляді діаграми, знаходять у формі, а відповіді респондентів — в автоматично створеній таблиці Google, яку можна зберігати на комп'ютері як таблицю Excel.

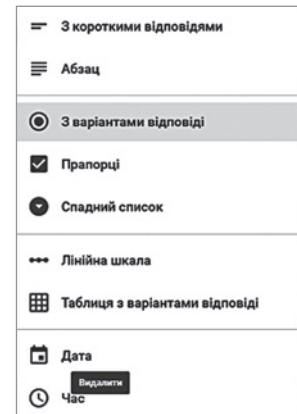


Рис. 6. Вибір типу відповідей за типом питання

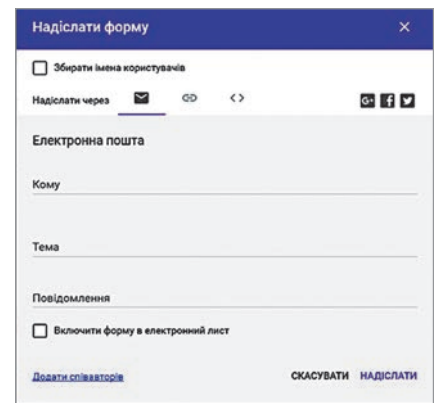


Рис. 7. Вікно надсилання форми

? Запитання для перевірки знань

- 1 Поясніть поняття спільного документа та сутність колективної роботи з ним.
- 2 Назвіть програми, які зручно використати для колективної роботи.
- 3 Які можливості має сервіс Google Документи для організації колективної роботи?
- 4 Файли яких типів можна створювати і розміщувати в Google-документи?
- 5 Як створити форму опитування за допомогою сервіса Google Документи?
- 6 З'ясуйте, чим відрізняється форма блогу від Google-форми.

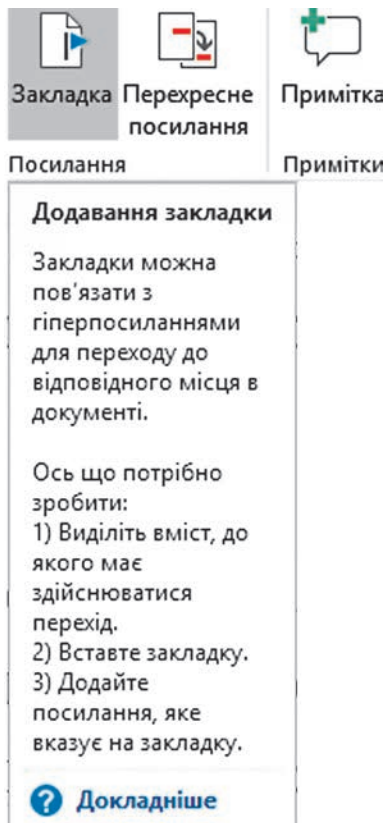
🖥️ Завдання для самостійного виконання

- 1 Перейдіть за посиланням <https://docs.google.com>.
- 2 Активізуйте кнопку **Створити** (має вигляд папки в меню налаштування вікна перегляду документів).
- 3 У вікні **Створити** в рядку меню виберіть команду **Завантажити**.
- 4 Скористайтеся командою **Виберіть файл з комп'ютера**, якщо ви працюєте з вікном у повноекранному режимі — відкриється звичайне вікно відкриття файлів.
- 5 Виберіть необхідний для завантаження файл. Він завантажиться і відкриється для перегляду та редагування у вікні документів.
- 6 Якщо не плануєте працювати із змістом файла, закрийте його. Файл буде закачано у сховище Google.

7.3. Створення і використання спільних електронних закладок



Пригадайте призначення команди **Закладка** (панель **Вставлення** → група **Посилання**) у текстовому процесорі *Microsoft Word*.



Додавання закладки в MS Word

Раніше учням під час підготовки, наприклад до іспиту чи написання твору, доводилося закладати потрібні сторінки книжок. Тому доводилося тримати на столі купу посібників.

Закладки, створені в електронному вигляді (див. [рисунок](#)), відіграють таку саму роль.

Персональне навчальне середовище призначене для навчання та розвитку особи, тож у ньому доцільно мати закладки на джерела з тематики середовища.



Закладкою (інколи називають **вибране**) є посилання на необхідне джерело, виконане як запис, який складається з адреси сайта, його назви, додаткових параметрів. Додатковими параметрами є короткий опис та мітки (ключові слова).

Закладки можна створювати в браузері для збереження посилань на часто відвідувані сайти чи веб-сторінки. Всі створені закладки розташовано у відповідному рядку браузера.

Для створення та використання закладок користувач має виконати мінімум операцій. Сервіси, які використовуються для ПНС, для колективної роботи з документами мають послуги створення та опрацювання (оновлення, вилучення) закладок.

Інколи виникає потреба створити посилання не на весь документ (він може бути досить об'ємним), а на його частину, або перейти з одного місця документа в інше без прокручування.

Розглянемо можливості сервіса Google щодо *створення та використання закладок* у текстовому документі (для таблиць і презентацій функція створення закладок відсутня).

Приклад.

1) Створення закладки

Створюємо або відкриваємо документ у середовищі Google (попередньо перейшовши за посиланням <https://docs.google.com>).

Встановлюємо вказівник миші в місце для закладки або виділяємо фрагмент тексту, який і буде закладкою.

Звертаємося до пункту меню **Вставити** та активізуємо команду **Закладки** — з'явиться вікно встановлення закладок.

Вибираємо в цьому вікні команду **Посилання** та клацаємо кнопку **ОК**.

Як результат, документ міститиме прапорець на полі.

Закладка має сенс, якщо користувач може переглянути вміст, позначений закладкою. Для створення посилання на закладку здійснюємо перехід встановленням посилання на неї з потрібного місця — у документі виділіть текст, який і буде посиланням. Із пункту меню **Вставити** вибираємо команду **Посилання**, а у вікні **Закладки**, що відкриється, — буде показано всі встановлені закладки (і не тільки з поточного документа). Далі вибираємо потрібну закладку і команду **Застосувати**.

2) Використання закладок

Цей крок є найпростішим, адже посилання на закладку в документі виокремлено іншим кольором. Якщо наведемо на посилання вказівник миші й клацнемо, відбудеться перехід до зазначеного місця тексту.

Якщо під час колективної роботи певний документ є спільним, то текст, позначений закладками, буде доступний іншим користувачам. Для закладок до окремих частин, наприклад власного блогу, можна привласнити мітки.



Міткою є окремі ключові слова або фрази, які вказують закладки, за якими здійснюється перехід по тексту документа.

Як ми вже говорили, у звичайній книжці закладкою є паперова смужка. Вона вкладається в певне місце книжки і полегшує пошук потрібної сторінки. Мітка в електронному документі обов'язково має назву, по суті є посиланням на місце, для якого створювалася. Сайт може містити кілька сторінок, тематично пов'язаних між собою, які доцільно позначити мітками з однаковою назвою. У разі звернення до мітки з певною назвою будуть відкриті всі публікації сайту з такою міткою.

Наприклад, блог містить публікації, серед міток яких є однакові. Під час переглядання публікації під її назвою будуть зазначені мітки. Після активування однієї з них на сторінці сайту з'являться всі публікації з такою міткою.

Для ознайомлення з алгоритмом створення й перегляду міток виконайте запропоноване завдання.

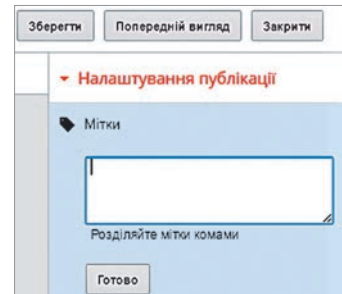


Рис. 1 до завдання

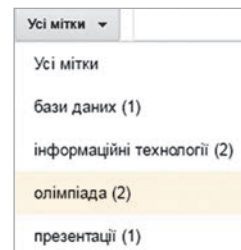


Рис. 2 до завдання



Завдання для самостійного виконання

- 1 Відкрийте створений вами блог. Створіть порожню публікацію: виконайте команду **Новий допис**, уведіть назву публікації **Публікація 1**.
- 2 У вікні Налаштування публікації виберіть команду **Мітки** (див. рис. 1 до завдання).
- 3 У полі для введення міток уведіть ключове слово, яке відбиває тему публікації, наприклад олімпіада, бази даних. На підтвердження операції створення міток виберіть команду **Готово**.
- 4 Створіть так само ще дві порожні публікації з назвами **Публікація 2** і **Публікація 3** з мітками: інформаційні технології, олімпіада та інформаційні технології, презентації відповідно. Поверніться у блог на сторінку **Публікації**.
- 5 Скористайтеся списком **Усі мітки**, щоб побачити, скільки публікацій відповідають темі тієї чи іншої мітки (див. рис. 2 до завдання).
- 6 Перейдіть за міткою **Олімпіади** — на сторінці **Публікації** буде показано тільки ті публікації, які й позначено міткою **Олімпіади**.



Запитання для перевірки знань

- 1 Поясніть поняття закладки.
- 2 Поясніть поняття мітки.
- 3 Назвіть сервіс Інтернету, створений для формування системи закладок.
- 4 Поясніть, для яких об'єктів можна створити закладку в Google-документі.
- 5 Як створити закладку за допомогою сервіса Google?
- 6 З'ясуйте, як створюють бібліотеку закладок.

7.4. Канали новин



Напевно, є сайти, які ви періодично відвідуєте, читаете статті, слідкуєте за новинами. Як ви дізнаєтесь про оновлення змісту сайтів?

Файл формату XML (англ. *eXtensible Markup Language* — це розширена мова розмітки) — звичайний текстовий файл, у якому за допомогою спеціальних кодів, які називаються тегами (іноді маркерами), визначається структура документа та його зміст. XML-файл можна редагувати будь-яким тестовим процесором і опрацьовувати стандартними XML-аналізаторами. Прикладами формату XML є формати XSLT, XAML, XUL, MathML, GraphML, XHTML, SVG, XML Schema.

Основою кожного сайту є його контент — статті, новини, коментарі, який він періодично оновлюється. Особам, які цікавляться тематикою певних сайтів, необхідно слідкувати за оновленням інформації на них. Щоб уникнути періодичного відвідування сайту з метою відслідковування новин, є сенс підписатись на канал новин.



Каналом новин є формат подання оновлених даних сайту (XML-формат даних) з визначеної тематики, за допомогою якого відслідковують оновлення інформації на сайті.

Канал новин складається із посилань на оновлену на певному сайті інформацію. Він містить заголовок, анонс на публікацію чи саму публікацію (залежить від послуг, які надаються спеціальними програмами), дані автора каналу або сайту.

Щоб автоматично отримувати, наприклад, оновлені новини улюбленого сайту, здійснюють підписання на його новини. Для цього потрібно знайти на цьому сайті посилання на RSS-канал, яке зазвичай подається у вигляді піктограми — це загальноприйняте візуальне позначення RSS (рис. 1).



Рис. 1. Піктограма каналу новин



RSS — це множина XML-форматів, яку використовують для публікації та постачання інформації, що часто оновлюється.

RSS (англ. *Rich Site Summary* — RSS 0.9x) у перекладі означає збагачене зведення сайту; Really Simple Syndication — дуже просте отримання інформації. Є й інші визначення: RDF Site Summary (RSS 0.9 и 1.0) — зведення сайту із застосуванням інфраструктури опису ресурсів; Really Simple Syndication (RSS 2.x) — дуже просте розповсюдження/поширення.

Документ у стандарті RSS називають **RSS-каналом новин** (RSS-каналом) або **RSS-стрічкою** (стрічкою, веб-стрічкою). Багато сучасних браузерів, поштових клієнтів і програм миттєвого обміну повідомленнями вміють працювати з RSS-каналами. Процес підписання на RSS-канал передбачає також отримання повідомлень про оновлені дані на поштову скриньку.

Канали новин опрацьовуються спеціальними програмами — **RSS-агрегаторами**. Агрегаторами можуть бути веб-сервіси — *онлайн-агрегатори* і програми на локальному комп'ютері — *локальні агрегатори*.

Локальні агрегатори вбудовані у браузер, поштовий сервіс або можуть бути окремим додатком. Вони мають відслідковувати оновлення на сайтах. Якщо інформація на сайті оновилась, агрегатор має подати анонс або повідомлення (залежно від налаштувань).

Контент-агрегатором є програма або сервіс, що виконує завдання накопичення інформації з різних сайтів, форумів, блогів в одне джерело.

Онлайн-агрегатори нагадують поштові сервіси: в одному із сервісів необхідно створити обліковий запис, і всі канали новин будуть надсилатися на поштову адресу облікового запису користувача. Отже, оновлення онлайн-матеріалу, який цікавить користувача, відбуватиметься незалежно від того, працює користувач за комп'ютером, під'єднаним до мережі, чи ні. Перевірити свою поштову скриньку можна в будь-який час на будь-якому пристрої, під'єднаному до мережі.

RSS-канал має універсальні характеристики для об'єднання матеріалу. В його основі може бути будь-який мережевий ресурс: сайт, блог, форум, прайс-лист мережевого магазину тощо. Кількість каналів на одному сайті не обмежується.

Переваги використання RSS-каналів:

- швидке ознайомлення зі змінами змісту сайта без відвідування самого сайта;
- достатність одного RSS-агрегатора для підписки на безліч каналів різних сайтів;
- миттєве розповсюдження інформації для підписувачів на RSS-канал без анонсів електронною поштою або в інший спосіб.

Є дуже популярні веб-агрегатори, що являють собою сайти для збирання та відображення RSS-каналів. Додати RSS-канал новин можна в середовищі каталогу каналів за посиланням <http://rssnews.org.ua/>.

Сервіс має такі можливості:

- компоновка найбільш популярних каналів новин (за популярністю, за певними рубриками тощо);
- додавання власного каналу в середовище сервісу;
- трансляція каналів — відображення каналів на своєму сайті, що дозволяє одразу відслідковувати оновлення.

RSS-канал можна налагодити в інтернет-браузері.

Щоб **підписатися на канал новин певного сайта**, необхідно знайти посилання на RSS-канал на цьому сайті — зазвичай воно має вигляд невеликої піктограми. Посилання на RSS-канал також може виглядати як звичайне текстове посилання (рис. 2). При переході за посиланням необхідно відслідковувати команди сайта, оскільки їх налаштування для окремих сайтів різні.



У загальному розумінні агрегатор (від лат. *Aggregatio* — накопичення) — система-посередник, яка об'єднує роботу з кількома об'єктами в одному каналі.



Atom — формат, заснований на XML, з'явився пізніше RSS і враховував багато недоліків згаданого формату.

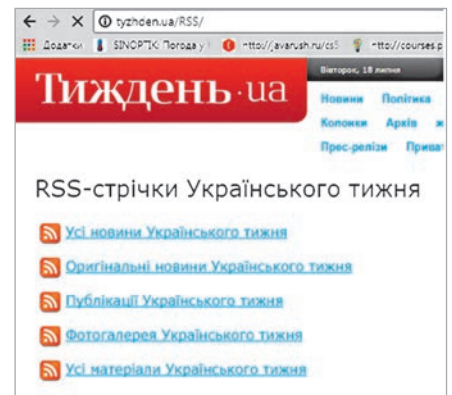


Рис. 2. Перелік RSS-стрічок сайта електронної версії газети Український тиждень



Запитання для перевірки знань

- 1 Поясніть поняття каналу новин.
- 2 Поясніть призначення каналу новин.
- 3 Як ще називають RSS-канали?
- 4 Поясніть різницю між локальним та онлайн-новим агрегаторами.
- 5 Які можливості має RSS-агрегатор?
- 6 Назвіть можливості каталогу каналів новин. Наведіть приклад.

7.5. Синхронізація даних



Назвіть засоби збереження даних. Поміркуйте над ситуацією: виконучи домашнє завдання, ви вдома створили файл і зберегли на електронному носії, а в школі його відредагували. Чи різним буде вміст файлів, збережених на електронному носії та в домашньому комп'ютері?



Термін «синхронізація» (від грец. *synchronos* — одночасний) використовується у двох значеннях: синхронізація процесів та/або синхронізація даних.

Синхронізація процесів є режимом взаємодії кількох процесів, при якому певні стадії різних процесів відбуваються у визначеному порядку або одночасно.

Приклади сайтів із програмами синхронізації:

<http://www.cordes-dev.com/>
(програма Backer 6.4)

<http://www.2brightsparks.com/> (різні версії програми SyncBack)

<http://www.smsync.com/>
(SmartSync Pro)

<http://www.websynchronizer.com/> (WebSynchronizer)

Багатьом користувачам сьогодні доводиться працювати на двох і навіть більше комп'ютерах, що спонукає до додаткової роботи з файлами. Для відстеження змін у файлах і папках необхідно копіювати їх з одного комп'ютера на інший. Щоб уникнути такої роботи, для основних папок створюють завдання синхронізації і запускають в автоматичному режимі, наприклад коли вмикають комп'ютер. Синхронізація можлива для комп'ютерів, з'єднаних мережею (локальною чи Інтернетом).



Синхронізація даних — процес перевірки та реалізації ідентичності вибраних даних на різних пристроях.

Наприклад, створений текстовий документ міститься у файлах на кількох комп'ютерах, під'єднаних до сервісу. Якщо на одному з комп'ютерів у документ увести зміни, то на інших комп'ютерах відбудеться оновлення змісту відповідних файлів. Також синхронізація допоможе зберегти ваші дані, якщо файли вашого комп'ютера пошкодить вірус.

Під'єднання комп'ютерів або інших пристроїв (планшетів, смартфонів тощо) до сервісу синхронізації означає використання на пристроях відповідного інтернет-сервісу, який передбачає встановлення на пристроях спеціальних програм. Програма створить папку, в якій і будуть зберігатись файли для синхронізації. Файли також будуть завантажені у сховище інтернет-сервісу, до папки з файлами буде надано доступ із будь-якого пристрою за умови під'єднання пристрою до Інтернету.

Завдання синхронізації:

- перевірка ідентичності даних на кількох комп'ютерах (робочому і домашньому, ноутбуку і стаціонарному тощо);
- перевірка ідентичності вмісту робочих папок на локальному комп'ютері з оновленням сайта;
- створення резервних копій даних — копіювання даних, які не змінювались, не відбувається;
- порівняння даних (програми синхронізації в процесі перевірки даних виявлять і повідомлять про файли, в яких не збігаються дані);
- обмін файлами з іншими особами за наданим посланням.

Як уже зазначалося, для здійснення синхронізації на своєму пристрої необхідно встановити спеціальні програми.

Всі додатки Google мають функцію автоматичної синхронізації для облікового запису користувача.

Розглянемо *алгоритм встановлення синхронізації*.

Крок 1	Запустіть сервіс Google. Перейдіть у вікно вашого сховища на Google.
Крок 2	Виберіть з меню налаштувань команду синхронізації.
Крок 3	Налаштуйте синхронізацію та підтвердьте вибір — дані будуть синхронізовані.

Для ознайомлення з порядком синхронізації даних на Google Диску виконайте завдання та розгляньте алгоритм скасування синхронізації даних з облікового запису.

Алгоритм скасування синхронізації даних

Крок 1	У новій вкладці в браузері Chrome відкрийте вікно Налаштування .
Крок 2	Перейдіть за командою Інформаційна панель Google .
Крок 3	У вікні, що відкрилося, натисніть Скинути синхронізацію .

Після цього синхронізовані дані будуть видалені з облікового запису, тобто на локальному комп'ютері вони залишаться, але перейти до них з інших пристроїв, під'єднаних до Інтернету, буде неможливо.

? Запитання для перевірки знань

- 1 Поясніть поняття синхронізації даних.
- 2 Навіщо використовують синхронізацію даних?
- 3 Поясніть, які завдання вирішуються синхронізацією даних.
- 4 Яку локалізацію мають дані, що синхронізують?
- 5 Як здійснити синхронізацію даних?
- 6 Як скасувати для даних їхню синхронізацію в обліковому записі?

📁 Завдання для самостійного виконання

- 1 Увійдіть у сервіс **Google Документи**. У меню сервіса зверніться до команди запуску власного диску — в робочій зоні будуть відображені ваші файли та папки, збережені в хмарному середовищі (або увійдіть у ваш диск на Google сервісі: <https://drive.google.com>).
- 2 Розкрийте меню браузера — після адресного рядка розташована кнопка меню команд.
- 3 Виберіть команду **Налаштування** — відкриється вікно налаштувань.
- 4 Виберіть команду **Параметри синхронізації** — відкриється вікно налаштування параметрів синхронізації.
- 5 Встановіть параметри синхронізації:
 - в разі потреби синхронізування всіх об'єктів, виберіть команду **Синхронізувати всі об'єкти**;
 - для синхронізування тільки вибраних об'єктів активізуйте відповідні команди (кляцанням по маркеру команди встановіть прапорець або перемістіть перемикач у положення **ВКЛЮЧЕНО** для різних версій Windows).
- 6 У розділі **Параметри шифрування** виберіть спосіб захисту даних.

7.6. Етапи створення веб-сторінок. Конструювання сайтів. Поняття мови розмітки гіпертексту



Який документ називають сайтом? Яка послуга Інтернету дозволяє переглядати сайти? Чи однакове значення мають поняття веб-сторінки та сайта? Поясніть поняття хостингу.

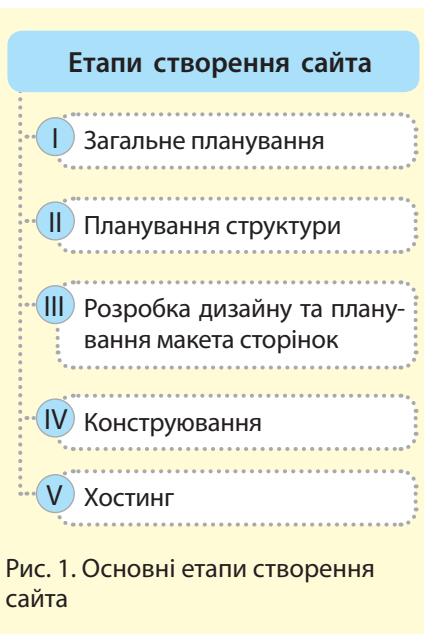


Рис. 1. Основні етапи створення сайта

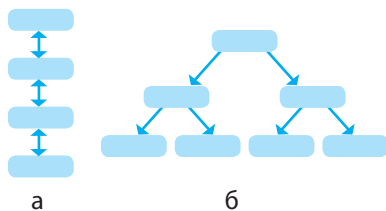


Рис. 2. Лінійна (а) та ієрархічна (б) структури сайтів

Приклад сайта з лінійною структурою — сайт навчання певного курсу; приклад сайта з ієрархічною структурою — сайт із каталогами джерел, впорядкованих за тематикою.

Створення сайта складається з кількох етапів (рис. 1) і починається з аналізу його тематики, аудиторії для перегляду й використання та планування його змісту та структури. Необхідно визначити, які розділи повинен містити сайт, на яких сторінках має бути той чи інший розділ, якою буде навігація по сайту. Визначають також кількість сторінок сайта та навігацію між ними.

Сайти поділяють на сайти з *лінійною* та *ієрархічною структурою* — на рис. 2 сторінки визначаються блоками, а навігація — стрілками між ними. Також є сайти з довільною структурою, на яких навігація здійснюється в довільному порядку. Сайтом з довільною структурою є, наприклад, Вікіпедія.

Дизайн сайта має відповідати його тематиці, складу аудиторії, на яку розраховано матеріал. Після визначення структури сайта та оформлення його сторінок необхідно визначитися з розміщенням матеріалу та приступити до *конструювання сайтів*, яке здійснюють за допомогою мови розмітки гіпертексту HTML або з використанням програм автоматизованого створення сайтів.



Мова розмітки гіпертексту HTML — набір вказівок (тегів), які браузер інтерпретує для відтворення сторінок сайта у своєму вікні.

Для написання тегів можна скористатися текстовим редактором, але зберегти файл із розширення .htm, щоб він відкривався браузером. Для створення інтерактивності сайта використовують такі мови програмування, як Java Script, PHP.

Створення сайтів із прямим указанням тегів HTML, кодів Java Script, PHP називають **конструюванням сайтів вручну**.

Існують спеціальні середовища для створення сайтів, які надають можливість створювати сайти звичайним користувачам і роблять їхню роботу зручною. Адже автоматизоване створення сайтів не вимагає досконалих знань тегів (приклад програм Microsoft FrontPage, DreamWeaver (Macromedia)).

Створення сайтів за допомогою спеціальних програм називають **автоматизованим конструюванням сайтів**.

Програми для створення сайтів містять готові шаблони, складові яких розташовують на сторінках сайта, їх наповнюють потрібною інформацією. Такі програми встановлюють на локальному комп'ютері — для конструювання сайта Інтернет не потрібен. Існують інтернет-сервіси, які дозволяють

організувати онлайн-роботу зі створення сайта, вони організують створення сайта в середовищі — системі управління веб-контентом, наданому платними або безкоштовними сервісними службами.

Онлайн-сервіси надають послуги створення сайта і супроводження їх у подальшому.

Перевагами онлайн-сервісів є:

- конструювання сайта не потребує досконалого знання мови тегів і додаткових мов програмування;
- наявність готових шаблонів;
- оперативна публікація сторінок;
- можливість редагування в режимі онлайн;
- зручний інтерфейс.

Ознайомимося з *алгоритмом створення сайта за допомогою інтернет-сервісів*.

Крок 1	Увійдіть у свій обліковий запис на сервісі або зареєструйтесь, якщо обліковий запис відсутній.
Крок 2	Виберіть ім'я сайта та визначте шаблон його оформлення.
Крок 3	Створіть сторінки сайта та систему навігації.
Крок 4	Заповніть сторінки сайта інформаційними матеріалами.

Процес **створення сайта в режимі онлайн** розглянемо на прикладі сервіса Google.

Приклад. Переходимо за посиланням <https://sites.google.com/>; вводим логін і пароль вашого облікового запису, після чого відкриється вікно створення сайтів. Далі після запуску команди Створити вибираємо режим створення сайтів, наприклад Створити в новій версії (а після вибору команди Створити в класичній версії буде запропоновано вибрати шаблон сайта).

Відкриється вікно, яке містить зону складових сайта, які необхідно заповнити інформацією, та рядок меню з командами: Відмови та повернення операцій, Попередній перегляд, Додати редакторів (вказані команди подані піктограмами), Опублікувати, Додати, Сторінки, Теми.

Назви команд вказують на їх функціональне призначення. Командою Додати редакторів можна встановити колективний режим створення сайта. Команда Додати призначена для додавання на сторінку сайта тексту, графічного зображення, матеріалу, розміщеного за посиланням із зазначенням адреси URL або в хмарному сховищі.

На початку роботи над сайтом йому необхідно дати назву. Для цього у вікні створення сайтів є спеціальне поле. Одночасно в автоматичному режимі в рядку адреси сайта буде запропонована URL-адреса головної сторінки сайта: <https://sites.google.com/site/...> (тут ... — назва сайта). Якщо ви надали назву сайта кирилицею, то зробіть зміни в адресі сайта — вона складається з літер латиниці та цифр.

Заповнення необхідних полів створення сайта підтвердьте командою Створити. У вікні браузера відкриється домашня сторінка сайта з назвою, панеллю навігації сайта із заголовком «Домашня сторінка». Панель навігації складається з двох пунктів (Домашня сторінка та Карта сайта, яка містить посилання на Домашню сторінку).

Для створення нових сторінок сайта та наповнення їх інформацією користуються командами рядка меню. У процесі створення сайта варто використовувати команду перегляду, щоб знати, як буде виглядати сторінка.

HTML (HyperText Markup Language) мова розмітки гіпертексту.

Для перегляду тегів будь-якої завантаженої в браузер сторінки виходять у меню браузера і використовують команду Інші інструменти — Інструменти розробника. У результаті зона показу сайта буде розділена на дві: в одній сайт, а в другій теги створення цього сайта.



Завдання для самостійного виконання

- 1 Назвіть етапи створення сайтів.
- 2 За допомогою якої мови створюють сайти?
- 3 Як переглянути HTML-теги, за допомогою яких створено сайт?
- 4 Назвіть переваги використання онлайн-сервісів для створення сайтів.
- 5 Порівняйте поняття веб-сторінки та сайта, поясніть їх різницю.
- 6 Поясніть сутність поняття хостингу. Чи потрібен хостинг для сайтів, створених у середовищі будь-якого онлайн-сервіса?

Практична робота № 14

Тема. Офісні веб-програми для створення спільних документів. Опитування з використанням онлайн-форм.

Завдання: створити форму для проведення онлайн-опитування на тему «Історія розвитку обчислювальної техніки».

Обладнання: комп'ютер, під'єднаний до мережі Інтернет, обліковий запис у сервісі Google, файл із фотографією Глушкова В. М.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Перейдіть за посиланням docs.google.com/forms — відкриється вікно форм сервісу. (Якщо ви створюєте форму не вперше, в робочій зоні будуть створені раніше форми.)
 - 2 Запустіть команду створення нової форми (кнопка із знаком «+»).
 - 3 У вікні, що відкрилось, у поле Форма без назви введіть назву форми «Історія розвитку обчислювальної техніки». У поле Запитання без назви введіть прізвище, клас учня.
 - 4 Виберіть тип відповіді — Абзац. У поле Запитання без назви введіть питання *Який рік вважають початком механічного періоду розвитку обчислювальної техніки.*
 - 5 Виберіть тип відповіді — 3 короткими відповідями.
 - 6 Додайте питання — використайте знак + у меню праворуч від бланка форми. Виберіть прізвище автора аналітичної машини. Виберіть тип питання: 3 варіантами відповіді та вставте чотири варіанти відповідей (один із них — Беббідж).
 - 7 Уведіть нове питання *Хто зображений на фото* і вставте з файла фотографію Глушкова В. М. Виберіть тип відповіді Прапорці та вставте 6 варіантів відповідей — обов'язково мають бути Глушков В. М.; Піонер комп'ютерної техніки; Засновник інформатики в Україні.
 - 8 Додайте ще три питання з різними варіантами відповідей на власний розсуд.
 - 9 Додайте питання *Дата опитування*, виберіть тип відповіді *Дата*.
 - 10 Перейдіть до налаштування форми — використайте іконку налаштування в рядку меню і виберіть команду Презентація; поставте прапорець біля команди Перемішати запитання. Виберіть команду Зберегти.
 - 11 Поверніться до вікна форм за допомогою стрілочки в лівому верхньому куті вікна. *Примітка.* У процесі створення форми в нижньому правому куті вікна є знак «?» для переходу до довідкової системи. Скористайтесь довідкою, щоб дізнатись, як вставляти відеофрагмент у форму.
 - 12 Запустіть форму декілька разів, перевірте її роботу. Покажіть роботу вчителів.
- Зробіть висновки:** чому питання подаються щоразу в різному порядку; у яких випадках недоцільно змінювати порядок показу запитань, наведіть приклади.

Практична робота № 15

Тема. Конструювання сайтів з використанням онлайн-систем.

Завдання: створити сайт на задану тему (наприклад, про життя вашого класу).

Обладнання: комп'ютер, під'єднаний до мережі Інтернет, обліковий запис у сервісі Google, файли з фотографіями зі шкільного життя.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Перейдіть за посиланням <https://sites.google.com> — відкриється вікно перегляду та створення сайтів у сервісі Google.
 - 2 Увійдіть у свій обліковий запис на сервісі Google — відкривається вікно сайтів на Google.
 - 3 Перейдіть до режиму створення сайта — натисніть кнопку Створити й у списку виберіть команду У класичній версії.
 - 4 У вікні, що відкрилось:
 - 1) виберіть шаблон сайта на власний розсуд;
 - 2) уведіть назву сайта (Життя нашого класу);
 - 3) уведіть адресу сайта;
 - 4) виберіть тему оформлення сайта.
 - 5 Підтвердьте, що ви не робот, надайте відповіді у вікні, що відкриється; виберіть команду Створити.
 - 6 Виберіть команду Створити сайт (одна сторінка сайта створена — зазвичай вона називається головною або домашньою).
 - 7 За допомогою головного меню перейдіть до свого сайта. Виконайте команду Сторінки для створення нової сторінки сайта.
 - 8 Уведіть назву сторінки у відповідне поле (наприклад, Клас і події у школі), підтвердьте введення за допомогою команди Готово.
 - 9 Перейдіть до перегляду сторінки — вона порожня, містить тільки назву та панель навігації. Перевірте, як працює навігація. Закрийте режим перегляду. У правій частині показано перелік сторінок.
 - 10 Перейдіть до домашньої сторінки і за допомогою команд меню Додати, заповніть сторінку описом матеріалів, які буде розміщено на сторінках сайта.
 - 11 Заповніть сторінку, присвячену подіям у вашому закладі (Клас і події у школі), в яких брав участь ваш клас. Виберіть шаблон сторінки на власний розсуд. Розмістіть фотографії на сторінці за тематикою подій.
 - 12 Створіть сторінку, присвячену подорожам вашого класу або окремих осіб. Додайте фотографії з подорожей, якщо вони існують, а також знайдіть в Інтернеті матеріал про місця подорожей та додайте на сторінку.
 - 13 Повідомте учителю URL-адресу вашого сайта.
- **Зробіть висновки:** чи була б у вас можливість за час, витрачений на створення сайта, створити сайт з використанням мови розмітки HTML. Відповідь обґрунтуйте.

Розділ 8. МОДЕЛЮВАННЯ

8.1. Поняття моделі та моделювання



Пригадайте, якими моделями ви користувалися в кабінетах фізики і географії. Чи доводилося вам самостійно розробляти моделі? Які основні дії при цьому ви виконували?



Модель (від латин. *modulus* — міра, зразок) — це штучно створений об'єкт для об'єкта-оригіналу, за допомогою якого можна вивчати і досліджувати властивості оригіналу.

Модель може бути створена у вигляді фізичного пристрою, схеми, фотографії, словесного опису, математичних формул тощо. Наприклад, для визначення маршруту поїздки в метро достатньо мати схему розташування станцій метрополітену, яка є його моделлю. Для початкового ознайомлення з архітектурою Києво-Печерської лаври не обов'язково їхати до самої лаври, це можна зробити за допомогою фотографій, схем, макетів тощо, які також є моделями лаври. Площу земельної ділянки фермерського господарства не обов'язково вимірювати за допомогою, наприклад, рулетки безпосередньо на полі, її можна обчислити за допомогою відповідних математичних формул. Людина, яка знає музичні ноти, може мати уявлення про симфонію № 9 Людвіга ван Бетховена, не чуючи саму симфонію, а шляхом ознайомлення з відповідними нотами. Учень отримує перше уявлення про земну кулю за допомогою глобуса (рис. 1, а-в).



Моделі особливо корисні, коли досліджуваний об'єкт має великі (Сонячна система) або малі розміри (атом), коли процес протікає досить швидко (ракетний двигун) або повільно (плавлення металу), коли оригінал коштує досить дорого (запуск ракети) або в реальному часі реальний об'єкт не існує (певна модель автомобіля).



Початковий варіант моделі для будь-якого об'єкта завжди виникає у свідомості людини. Потім модель може поступово змінюватися, удосконалюватися і набувати різних форм. Наприклад, архітектор спочатку створює модель будинку у своїй свідомості, потім він робить ескіз будинку на папері. Далі може бути створений макет і проект будинку. При цьому на всіх етапах один вид моделі доповнює інші й впливає на їх удосконалення.



а



б



в

Рис. 1. Моделі земної кулі (а), космічного корабля (б) і Києво-Печерської лаври (в)

Часто ми навіть не уявляємо, що наше життя пов'язане з розробкою, вивченням, дослідженням та аналізом моделей. У повсякденні ми не згадуємо, що наша діяльність часто здійснюється завдяки створеній у нашій свідомості моделі. У кращому випадку ми говоримо, що діємо згідно з розробленим планом, який фактично і є одним із видів моделі. Наприклад, виконуючи кидок баскетбольного м'яча в кошик із деякої точки майданчика, ми свідомо обираємо кут, напрям і силу його кидка саме тому, що попередньо створили модель цієї дії.

Будь-який об'єкт може мати модель. Об'єктом є цілісна частина навколишнього світу. Об'єктами є не лише предмети та явища матеріального світу, але й поняття, які створюють і використовують люди під час спілкування, наприклад суспільство, освіта, економіка.

Модель завжди розробляють із конкретною метою. Наприклад, перш ніж побудувати літак, конструктор розробляє декілька його моделей, кожну з яких досліджують у конкретних умовах. Ці результати враховують у процесі конструювання реального літака. На початку забудови, наприклад, мікрорайону міста архітектори розробляють його макет, тобто модель, до якої після цього вносяться необхідні зміни. Отже, завдяки створеним моделям вивчають і досліджують реальні об'єкти, а на основі даних, отриманих у процесі дослідження моделі, можна детальніше пізнати сутність *оригіналу*, тобто реально існуючого об'єкта або того, що розробляється. Основні цілі створення моделі наведено на рис. 2.

Кожен об'єкт має ім'я (назву) і характеризується певними параметрами. Наприклад, море має максимальну глибину, площу, температуру води тощо. Футболіст має певний зріст, масу, йому притаманні певні професійні якості.

Серед багатьох параметрів об'єкта є головні, які визначають його найсуттєвіші властивості. Наприклад, для комп'ютера такими параметрами є тактова частота процесора, ємність оперативної пам'яті, довжина діагоналі та роздільна здатність монітора, ємність вінчестера. Кількість головних параметрів міститься у межах від одиниць до кількох десятків. Чим більше параметрів об'єкта відомо, тим точніше його характеристика. Але не завжди відомі всі параметри об'єкта. Для визначення деяких із них потрібен тривалий час і певні умови. Наприклад, багато параметрів об'єкта Місяць були визначені лише після спеціальних досліджень.

Кожний об'єкт існує в певних умовах. Наприклад, метал у твердому стані перебуває лише за певної температури; риба не може існувати без води, людина — без повітря.

Над об'єктами можуть виконуватися певні дії, або сам об'єкт може виконувати їх. Наприклад, людина може спати, грати в баскетбол, писати, читати, керувати автомобілем тощо. Літак може стояти на землі, летіти горизонтально, йти на посадку тощо. Отже, **об'єкт — це частина довкілля, що може розглядатися як єдине ціле.** Об'єкт характеризується назвою, параметрами і діями, які над ним можна виконати.

Основні цілі створення моделі

показати, який вигляд матиме об'єкт, якого реально ще не існує, й обговорити його у певному колективі

дослідити на моделі поведінку об'єкта, якщо дослідження оригіналу взагалі неможливе, небезпечно або дорого коштує

навчитися керувати реальним об'єктом

Рис. 2. Цілі створення моделі

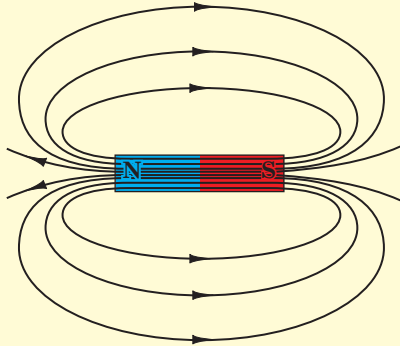


Рис. 3. Модель ліній магнітного поля



Рис. 4. Модель цунамі



Модель може розроблятися не тільки для матеріальних об'єктів, але й для **процесів і явищ**.

Моделі **процесів** можуть розроблятися для матеріальних об'єктів, наприклад модель плавлення металу в доменній печі, модель ковзання вантажу з похилої поверхні, модель розповсюдження морської хвилі, а також моделі для суспільних, економічних та інших процесів. Приміром, існують моделі економічного розвитку підприємства, галузі, держави, модель розвитку суспільства протягом певного періоду тощо. На рис. 3 зображена модель ліній магнітного поля.

Приклади моделей **явищ**: модель землетрусу, яка дозволяє оцінити його наслідки і визначити технічні вимоги до будівництва споруд; модель руху циклонів, яка дозволяє зробити прогноз погоди на певний період; модель цунамі (рис. 4), що дозволить попередити населення про загрозу.

Кожен об'єкт, процес або явище можуть мати декілька різних моделей. Конкретний тип моделі визначається не тільки метою її створення, але й засобами, які для цього використовуються. Наприклад, модель земної кулі може бути виготовлена у вигляді глобуса, створеного за допомогою картону, клею і фарби. Вона може також бути створена на екрані монітора комп'ютера за допомогою графічного редактора.

З описаного слідує, що поняття моделі досить велике за обсягом. Воно охоплює майже всі сторони людської діяльності, тому однозначного тлумачення цього терміна не існує. Далі будемо дотримуватися такого визначення моделі.



Модель — це сукупність найважливіших властивостей об'єкта, процесу, явища, яка визначається метою її створення і може мати різні форми вираження. Процес створення моделі називають **моделюванням**. *Основне завдання моделювання* — вибір найбільш адекватної до оригіналу моделі і перенесення результатів його дослідження на оригінал.

Відзначимо, що оригінал завжди складніший за будь-яку свою модель. Модель розробляється з конкретною метою і визначає не всі, а лише ті властивості оригіналу, які відповідають меті створення моделі. Разом із тим модель відображає найсуттєвіші сторони оригіналу з точки зору мети моделювання. Дослідження моделі слугує засобом для отримання знань про оригінал.



Запитання для перевірки знань

- 1 Наведіть приклади моделей.
- 2 Які моделі вам доводилося розробляти?
- 3 З якою метою розробляють моделі?
- 4 Що називають моделюванням?
- 5 Чи завжди модель відображує реальний стан об'єкта?
- 6 Поясніть сутність терміна «модель».

8.2. Класифікація моделей

Відомо, що модель може бути описана словесно, подана у вигляді схеми, приладу тощо. Поміркуйте, за якими ознаками можна класифікувати моделі.



Моделі класифікують за багатьма ознаками (рис. 1).



Рис. 1. Спрощена схема класифікації моделей

За способом подання моделі поділяють на дві основні групи: матеріальні (фізичні) і нематеріальні (інформаційні). *Матеріальні* — це реально існуючі моделі, тобто ті, які можна побачити й доторкнутися до них. Приклади матеріальних моделей: іграшка автомобіля, шкільний фізичний прилад для демонстрації взаємодії електромагнітних полів, макет гідроелектростанції, лялька, макет космічного корабля. Матеріальні моделі можуть бути носіями хімічних, геометричних, фізичних та інших властивостей реального об'єкта, процесу, явища.



Рис. 2. Карта Сумської області

Інформаційна модель — це вербальний, графічний, математичний, табличний тощо опис об'єкта.

Вербальною моделлю об'єкта називають його опис природною мовою. Приклади вербальної моделі: правила дорожнього руху, алгоритм пошуку мінімального елемента в одновимірному масиві, рецепт виготовлення соку.

Графічні моделі призначені для наочного подання об'єктів, процесів, явищ. Приклади графічних моделей: карта місцевості (рис. 2), схема електричного кола, креслення геометричних фігур (рис. 3).

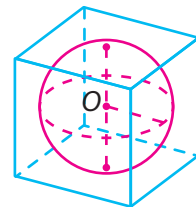


Рис. 3. Куля, вписана в куб

Математична модель — це система математичних рівнянь, формул (рис. 4), числових множин, які описують деякі властивості реального об'єкта, процесу або явища. Багато задач із фізики, біології, хімії розв'язують за допомогою рівнянь і нерівностей.

$$I = \frac{U}{R}$$

$$U = I \cdot R$$

$$R = \frac{U}{I}$$

Рис. 4. Формули закону Ома та наслідків із нього



Відомо, що Миколай Коперник (1473–1543) побудував геліоцентричну модель Всесвіту, відповідно до якої планети рухаються навколо Сонця певними орбітами. Перед появою цієї моделі існувала інша, у якій припускалося, що планети і Сонце рухаються навколо Землі. Зараз учні користуються більш точною моделлю Сонячної системи, яка дає змогу обчислювати маси і закони руху планет.



Рис. 5. Дорожні знаки



Рис. 6. Модель людини

Математичні моделі особливо корисні під час дослідження будівельних споруд, в аерокосмічній промисловості тощо.

За допомогою математичної моделі підприємства можна прогнозувати його фінансовий стан. Для побудови математичної моделі підприємства потрібно описати його найважливіші параметри, структуру та особливості поведінки.



Серед інформаційних моделей розрізняють також *знакові моделі*, які створюють за допомогою спеціальних знаків: дорожніх (рис. 5), музичних, мовних та ін. Вербальні й знакові моделі завжди доповнюють одна одну.

Вибір конкретного типу моделі суттєво впливає на те, який вигляд матиме об'єкт, що створюється, і якою буде його поведінка за реальних умов. Для всебічного оцінювання реального об'єкта й визначення його структури бажано водночас користуватися кількома моделями. Кращою є та модель, яка найточніше відображає реальний об'єкт.

За **призначенням** моделі поділяють на науково-дослідницькі, імітаційні, навчальні, ігрові та ін. *Науково-дослідницькі моделі* створюють з метою дослідження об'єкта, процесу, явища в певних умовах. Наприклад, існують моделі нервової клітини, дослідження яких допомагає створювати й удосконалювати теорію її збудження. *Імітаційні моделі* розробляють в основному для дослідницьких цілей. Наприклад, на комп'ютері може бути реалізована модель функціонування електронних схем окремих пристроїв, алгоритмів із метою визначення правильності їх розробки, надійності та якості.

До *навчальних моделей* відносять наочні прилади, плакати, схеми, тренажерні системи та ін. Особливо широко використовують ці моделі в медицині (рис. 6), космонавтиці (макети орбітальних станцій), біології (макети рослин).

До *ігрових моделей* належать спортивні, економічні та інші ігри, в ході яких «програються» різноманітні ситуації і визначається стратегія дій у певних умовах.

За **станом** розрізняють статичні й динамічні моделі. *Статична модель* фіксує стан об'єкта, процесу, явища на певний момент часу. Наприклад, стан дорожнього руху в годину пік, покази приладів літака в момент відриву від злітної смуги.

Динамічна модель відображає поведінку об'єкта протягом певного часу. Наприклад, основні покази приладів літака через кожну секунду під час зльоту. Динамічні моделі можуть досліджуватися в реальному масштабі часу, а також у скорочений чи збільшений період часу. Наприклад, процес плавлення металу, який реально здійснюється протягом декількох годин, можна змодельовати протягом десяти хвилин.



Запитання для перевірки знань

- 1 Які типи моделей застосовують на практиці?
- 2 Яку модель називають математичною?
- 3 За якою моделлю доцільно досліджувати об'єкт?
- 4 За якими ознаками класифікують моделі?

8.3. Комп'ютерне моделювання

Які моделі, на вашу думку, доцільно реалізувати на комп'ютері? Наведіть приклади моделей, які вам уже вдалося реалізувати на комп'ютері.



Комп'ютерне моделювання — це процес реалізації інформаційних моделей на комп'ютері. Для реалізації інформаційних моделей можуть застосовуватися системи програмування, електронні таблиці, математичні пакети, системи управління базами даних, графічні редактори та інші програмні засоби.

Комп'ютерне моделювання — це один з ефективних методів наукового пізнання і вивчення систем. Комп'ютерні моделі просто і зручно досліджувати. Вони забезпечують проведення обчислювальних експериментів, які у звичайних умовах реалізувати неможливо або досить складно. Основні переваги комп'ютерного моделювання подано на рис. 1.

Порядок реалізації інформаційної моделі на комп'ютері складається з таких основних етапів: постановка завдання, розробка інформаційної моделі, вибір інструментального програмного засобу для реалізації моделі, безпосередня реалізація моделі, аналіз отриманих результатів і прийняття відповідного рішення.

Системи комп'ютерного моделювання застосовують у різних сферах (рис. 2).

Переваги комп'ютерного моделювання

- Можливість багаторазового повторення тих самих дій
- Невелика вартість реалізації моделі
- Висока наочність візуалізації процесів
- Безпечність реалізації моделі
- Висока швидкість виконання дослідження моделі
- Отримання результатів моделювання у зручному вигляді для аналізу

Рис. 1. Переваги комп'ютерного моделювання



Рис. 2. Сфери застосування систем комп'ютерного моделювання

Комп'ютерне моделювання особливо інтенсивно застосовують для випадкових процесів. Випадкові процеси зустрічаються в нашому житті досить часто. Хоча не можна дати точну й однозначну відповідь на результат конкретної події або процесу, існують деякі їх закономірності, вивчення та дослідження яких допомагає людині прийняти правильне рішення в певних умовах або визначити стратегію поведінки. Дослідження закономірностей випадкових процесів у техніці сприяє підвищенню її надійності, поліпшенню параметрів апаратури, зменшенню матеріальних витрат тощо. Дослідження випадкових процесів на залізничному транспорті



Розповсюдженими нині є комп'ютерні симулятори, які імітують управління якимось процесом, апаратним або транспортним засобом. Існують комп'ютерно-механічні симулятори, у яких відтворено інтер'єри робочих місць певних фахівців: кабіни, у яких тренуються пілоти, водії, машиністи потягів та інші спеціалісти.



сприяє підвищенню безпеки і надійності руху, у медицині — попереджує розповсюдження епідемій, в іграх — підвищує ймовірність виграшу. Вивчення і дослідження випадкових процесів у сучасних умовах здійснюється, як правило, за допомогою комп'ютера.



Комп'ютерне моделювання широко застосовується в різних наукових галузях, зокрема для моделювання хімічних, фізичних, біологічних та інших процесів.



Приклад 1. Хлорид натрію розчиняється у воді. Необхідно визначити, яку масу хлориду натрію і води потрібно взяти для виготовлення необхідної кількості розчину певної концентрації.

Відомо, що хімічна формула хлорида натрію NaCl , а води — H_2O . Нехай m — маса необхідного розчину, m_1 — маса NaCl , m_2 — маса H_2O , p — відсоток концентрації розчину.

Будемо вважати, що маса розчину визначається в кілограмах (кг). З умови завдання маємо: в 100 кг розчину повинно міститися p (кг) NaCl , а в розчині масою m (кг) — m_1 (кг)

NaCl . Отримаємо співвідношення: $\frac{100}{p} = \frac{m}{m_1}$,

звідки $m_1 = \frac{p \cdot m}{100}$ (кг); $m_2 = m - m_1$.



В останні десятиріччя комп'ютерне моделювання ефективно використовується для розв'язування економічних задач.

Приклад 2. У банк поклали суму грошей s під k відсотків річних. Потрібно розробити модель розрахунку накопиченої суми (з округленням до другого знака після коми) за кожний із n років. Процес розв'язування цієї задачі вручну наведено в таблиці.

На основі аналізу дій, наведених у таблиці, можна розробити її математичну модель, яка подається системою формул:

$$p_i = s_{i-1} \cdot k, \quad s_i = s_{i-1} + p_i \quad \text{для } i = 1, 2, \dots, n,$$

$$y_0 = 0, \quad s_0 = s,$$

де:

p_i — прибуток за i -й рік;

s_{i-1} — накопичена сума за $(i-1)$ років;

s_{i-1} — накопичена сума за i років;

s_0 — початкова сума вкладу.

Таблиця. ПРОЦЕС ОБЧИСЛЕННЯ ПРИБУТКУ ЗА БАНКІВСЬКИМ ВКЛАДОМ

Рік	Прибуток за рік, грн	Загальний прибуток, грн	Накопичена сума, грн
1	$1000 \cdot 0,05 = 50$	50	1050
2	$1050 \cdot 0,05 = 52,50$	$50 + 52,50 = 102,50$	$1050 + 52,50 = 1102,50$
3	$1102,50 \cdot 0,05 = 55,13$	$102,50 + 55,13 = 157,63$	$1102,50 + 55,13 = 1157,63$
4	$1157,63 \cdot 0,05 = 57,88$	$157,63 + 57,88 = 215,51$	$1157,63 + 57,88 = 1215,51$



Завдання для самостійного виконання

- Кубик, на гранях якого містяться цифри від 1 до 6, кидають доти, доки сума цифр стане більше 50. Розробіть програму визначення кількості кидань кубика.
- У банк поклали 500 грн на 4 роки під 12,5 % річних. Розробіть програму визначення прибутку.

8.4. Використання програмних засобів для комп'ютерного моделювання

Якими програмними засобами, на вашу думку, доцільно користуватися для обчислення об'єму кулі; побудови піраміди; пошуку кількості учнів, які народилися у квітні?



Раніше вже зазначалося, що для комп'ютерного моделювання можуть використовуватися різноманітні програмні засоби. Найчастіше для цього застосовують системи програмування, графічні редактори, електронні таблиці й системи управління базами даних.

Розглянемо приклади використання деяких із цих засобів.



Приклад 1. Інвестор може вкласти в модернізацію виробництва різну суму гривень на десять років. Для поточної модернізації достатньо вкласти 1 млн грн. У цьому випадку прибуток складатиме щорічно 15 % від вкладеної суми через 3 роки. Для часткової модернізації потрібно вкласти 2 млн грн. Прибуток почне надходити щорічно через 4 роки у розмірі 18 % від вкладеної суми. Для капітальної модернізації виробництва потрібно вкласти 3 млн грн. У цьому випадку прибуток почне надходити щорічно через 7 років у розмірі 19 % від вкладеної суми. Потрібно розробити модель, за допомогою якої можна визначити найпривпливіший для інвестора варіант.

Позначимо змінною K_i кількість років, через які інвестор почне отримувати прибуток для i -го варіанта, де $i=1, 2, 3$, змінною S_i — суму інвестиції для i -го варіанта, N_i — кількість відсотків для i -го варіанта інвестиції,

змінною P_i — прибуток після 10 років. Тоді математичну модель цього процесу можна записати так: $P_i = S_i \cdot N_i \cdot (10 - K_i)$ для $i=1, 2, 3$.

Розмістимо на формі середовища Lazarus 4 компоненти TMemo, компонент TLabel і компонент TButton. У поле об'єкта Memo1 введемо можливі суми інвестицій, у поле об'єкта Memo2 — відсотки прибутків, а у поле об'єкта Memo3 — кількість років, через які почне повертатися прибуток. Об'єкт Memo4 використаємо для виведення сумарного прибутку за кожним варіантом через 10 років. В об'єкт Label1 введемо повідомлення Результати варіантів інвестиції. Для об'єкта Button1 встановимо подію OnClick. Обробник події зображено на рис. 1.

Результат виконання програми наведено на рис. 2. Як бачимо, найкращий результат через 10 років досягається в процесі часткової модернізації виробництва.

```
{ TForm1 }
//Інвестиції у виробництво протягом 10 років
procedure TForm1.Button1Click(Sender: TObject);
var i:integer; p:real;
begin
  for i:=1 to 3 do
  begin
    //обчислення результатів інвестицій
    p:=StrToInt(Memo1.Lines[i-1])*
    StrToInt(Memo2.Lines[i-1])*(10-StrToInt(Memo3.Lines[i-1]));
    //виведення варіантів результатів інвестиції
    Memo4.Lines.Add(FloatToStr(p));
  end;
end;
end.
```

Рис. 1. Програмний код визначення результатів від інвестиції

Рис. 2. Результати варіантів інвестиції

Приклад 2. Відомі харчові властивості п'ятих видів продуктів: вівсяна каша на молоці, картопля смажена, телятина тушкована, суп овочевий і вінегрет. Потрібно розробити комп'ютерну модель, за допомогою якої визначається загальний вміст білків, жирів, вуглеводів і калорійність для цих видів продуктів. Систематизуємо ці дані у таблиці (у розрахунку на 100 г продукту).

Таблиця. ВМІСТ І КАЛОРІЙНІСТЬ ХАРЧОВИХ ПРОДУКТІВ

Продукти	Білки, г	Жири, г	Вуглеводи, г	Калорії, ккал
Каша вівсяна на молоці	3,2	4,1	14,2	102
Картопля смажена	3,7	10,6	24,8	203,3
Телятина тушкована	16,8	18,3	0	232
Суп овочевий	1,7	1,8	6,2	43
Вінегрет	1,7	10,3	8,2	130,1

Модель визначення загального вмісту основних компонентів харчових продуктів і калорійності найдоцільніше реалізувати за допомогою електронної таблиці MS Excel. На рис. 3 зображено вміст розробленої електронної таблиці й обчислено загальний вміст білків, жирів, вуглеводів і калорій у цих видах продуктів.

	A	B	C	D	E	F	G
1	Вміст і калорійність харчових продуктів						
2	Продукти		Білки, г	Жири, г	Вуглеводи, г	Калорії, ккал	
3	Каша вівсяна на молоці		3,2	4,1	14,2	102	
4	Картопля смажена		3,7	10,6	24,8	203,3	
5	Телятина тушкована		16,8	18,3	0	232	
6	Суп овочевий		1,7	1,8	6,2	43	
7	Вінегрет		1,7	10,3	8,2	130,1	
8	Сума		27,1	45,1	53,4	710,4	

Рис. 3. Реалізація таблиці у середовищі MS Excel

Приклад 3. Протягом двох днів постачальники Костирко, Липко і Личко доставили у магазин необхідну кількість фруктів (у кг) за домовленою ціною, які магазин реалізує на 35 % дорожче. Необхідно розробити модель, за допомогою якої визначаються: а) дані про фрукти, ціна яких дорівнює або більша ніж 14 грн за кг; б) сума (у грн), отримана від реалізації фруктів, доставлених постачальником Липко.

Для розробки моделі доцільно скористатися системою управління базами даних, наприклад системою Access 2016. Створимо таблицю з іменем Фрукти, вміст якої зображено на рис. 4.

Для визначення даних про фрукти, ціна яких дорівнює або більше ніж 14 грн за кг,

упорядкуємо вміст таблиці у порядку зростання цін, після цього встановимо курсор у стовпці Маса на значенні 14 і виконаємо команду Виділення у групі Сортування і фільтр. Отримаємо результат, зображений на рис. 5.

Для отримання суми від реалізації фруктів, доставлених постачальником Липко, розробимо запит з іменем Запит2 з полями, що обчислюються. У запиті залишимо поля Назва і Постачальник, а також введемо поле Сума такого вмісту:

$$\text{Сума: [Маса]*([Ціна]+[Ціна]*0,35)}.$$

У результаті виконання створеного запиту отримаємо результат, зображений на рис. 6.

Назва	Дата	Постачальник	Маса	Ціна	Примітка
Слива	17.08.2016	Костирко	25	9,00	
Яблуко	15.08.2016	Липко	30	10,00	
Персик	17.08.2016	Костирко	20	14,00	маса в кг
Виноград	15.08.2016	Липко	27	15,00	ціна в грн за кг
Абрикос	15.08.2016	Личко	28	16,00	

Рис. 4. Таблиця про дані фруктів, поставлених до магазину

Назва	Дата	Постачальник	Маса	Ціна	Примітка
Персик	17.08.2016	Костирко	20	14,00	маса в кг
Виноград	15.08.2016	Липко	27	15,00	ціна в грн за кг
Абрикос	15.08.2016	Личко	28	16,00	

Рис. 5. Вміст таблиці після фільтрації записів

Назва	Постачальник	Сума
Яблуко	Липко	405
Виноград	Липко	546,75

Рис. 6. Результат виконання запиту

**Завдання для самостійного виконання**

- 1 На похилій площині, розташованій під кутом α до горизонталі, розміщено брусок масою m (рис. 7). Розробіть комп'ютерну модель визначення умови, за якої брусок буде нерухомий, якщо сила тертя відсутня.
- 2 Розробіть і дослідіть комп'ютерну модель рівноприскореного руху об'єкта.
- 3 Обґрунтуйте програмний засіб для обчислення кількості і вартості шпалер для обклеювання кімнати розмірами $4 \times 3,5 \times 2,5$ м. Ширина рулонів 55 см, довжина — 10 м.
- 4 У графічному редакторі створіть схему вашого кабінету інформатики.

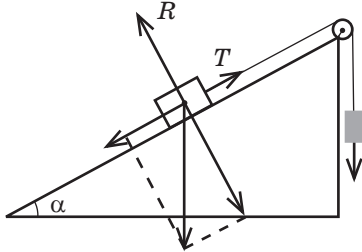


Рис. 7. Брусок на похилій площині

Практична робота № 16

Тема. Створення, опрацювання і дослідження інформаційних моделей.

Завдання: побудувати інформаційну модель для аналізу ВВП і земельних ресурсів на одну людину в п'яти країнах світу за даними таблиці.

Таблиця. ОСНОВНІ ПОКАЗНИКИ КРАЇН СВІТУ

Країна	Дата	Населення, млн	Площа, км кв.	ВВП, млрд доларів	Частина світу
США	01.12.2015	298	9 363 000	17 047	Америка
Німеччина	01.12.2015	81	357 021	3841	Європа
Японія	01.12.2015	127	337 835	4830	Азія
Китай	01.12.2015	1313	9 596 960	19 392	Азія
Велика Британія	01.12.2015	63	242 514	2679	Європа

Обладнання: комп'ютер із встановленими програмами: табличним процесором, середовищем програмування.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Виберіть програмний засіб для визначення ВВП, що припадає на одну людину в кожній країні.
 - 2 Розробіть модель для визначення ВВП, що припадає на кожну людину. Результат бажано отримати у вигляді діаграми.
 - 3 Виберіть програмний засіб для визначення кількості гектарів, які припадають на кожну людину в цих країнах.
 - 4 Розробіть модель на основі вибраного програмного засобу для визначення кількості гектарів. Результат бажано отримати у вигляді таблиці.
 - 5 Дізнайтеся про середній вік життя людини у цих країнах. Розробіть модель, за допомогою якої результат подається у вигляді стовпчастої діаграми.
 - 6 Для даних, наведених у таблиці, самостійно сформулюйте завдання для розробки моделі та реалізуйте її за допомогою вибраного програмного засобу.
- Зробіть висновок:** від чого залежить вибір конкретного програмного засобу для моделювання.

Розділ 9. ОДНОВИМІРНІ МАСИВИ

9.1. Основи роботи у середовищі Lazarus (повторення)



Пригадайте, що називають середовищем програмування; які основні функції воно повинно виконувати? Яке середовище програмування ви використовували у 5–7 класах; у 8 класі? Які недоліки, на вашу думку, мають ці середовища?

Загальні відомості про середовище Lazarus

ООП — це методика розробки програм, в основі якої є поняття об'єкт, метод і клас. Основна ідея ООП полягає в об'єднанні даних, з якими працює програмний код, і процедур їх опрацювання в єдине ціле.

Lazarus — це вільне інтегроване середовище візуально-го програмування мовою Free Pascal. У середовищі Lazarus реалізовані елементи об'єктно-орієнтованого програмування (ООП). Середовище функціонує під управлінням ОС Windows і Linux. Воно забезпечує розробку програм як у консольному, так і у візуальному режимах. Методика розробки програм у **консольному режимі** принципово не відрізняється від методики створення програм у середовищі Free Pascal. Тому далі основна увага буде приділятися візуальному режиму.

Початковий вигляд вікон середовища Lazarus зображено на рис. 1.

Головне вікно

Вікно інспектора об'єктів

Вікно редактора тексту

Вікно форми

Вікно повідомлення

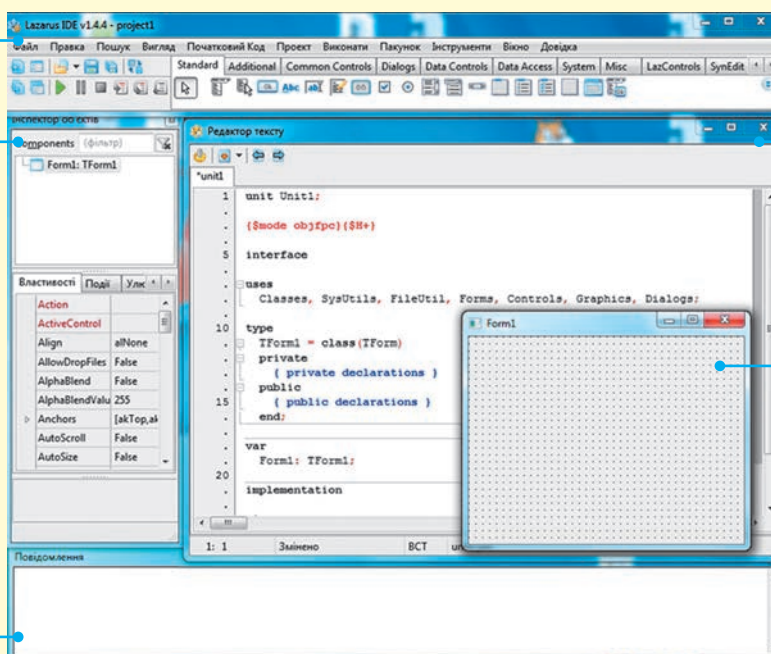


Рис. 1. Вікна середовища Lazarus

Головне вікно Lazarus (рис. 2) складається з трьох частин: головного меню (Файл, Правка та ін.), панелі інструментів (ліворуч під головним меню) і палітри компонентів на вкладках Standard, Additional та ін.

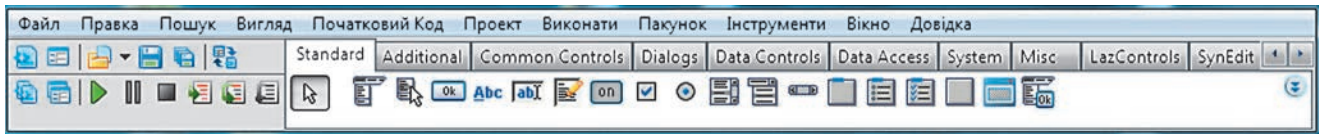










Рис. 2. Головне вікно середовища візуального програмування Lazarus


Головне вікно відкрите протягом усього періоду роботи середовища. Призначення окремих пунктів головного меню (Файл, Правка, Пошук, Вигляд та ін.) аналогічне призначенню однойменних пунктів інших прикладних програм, зокрема стандартного редактора тексту. На панелі інструментів розташовані кнопки команд, що часто використовуються. Вони дублюють деякі команди головного меню. Назви цих кнопок з'являються після встановлення на них вказівника миші. Тут розташовані, зокрема, такі кнопки: Нова форма , Зберегти , Виконати  та ін. Кнопки цієї панелі призначені в основному для зручності і прискорення роботи користувача.

На палітрі компонентів розташована значна кількість компонентів, які користувач може розміщувати на формі в процесі створення інтерфейсу майбутньої програми. Ці компоненти згруповані за функціональним призначенням. За замовчуванням відкриваються компоненти групи Standard (див. рис. 2). Найчастіше з цієї групи застосовують такі компоненти: TButton , TLabel , TEdit . Можна скористатися також компонентами групи Additional та ін. Для розкриття компонентів інших груп необхідно клацнути назву відповідної групи.

 **Компонент, перенесений на форму, є об'єктом. Кожний об'єкт має властивості і методи.** Властивості визначають зовнішній вигляд об'єкта (ім'я, розміри, колір та ін.). Метод визначає поведінку об'єкта, наприклад, як він буде реагувати на клацання кнопки миші або зміну його розміру. Для налаштування властивостей об'єктів використовують Інспектор об'єктів. Значення деяких властивостей можуть змінюватися також програмно.

Вікно інспектора об'єктів призначене для надання компонентам, розташованим на формі, необхідних характеристик (параметрів), наприклад кольору, розміру та ін. Вікно містить вкладки Властивості і Події. У лівому стовпці розташовані назви властивостей або подій, а в правому — значення властивостей або імена обробників подій. Для вибору властивості або події необхідно клацнути кнопкою миші на необхідному рядку, після чого можна вибрати або ввести її значення.


«Найкращий спосіб в чомусь розібратися до кінця — це спробувати навчити цьому комп'ютер».
Дональд Ервін Кнут

Перенесення компонента на форму виконується клацанням кнопки миші цього компонента, потім клацанням цієї кнопки у тому місці форми, де передбачається його розмістити. Для відмовлення від перенесення на форму вибраного компонента слід клацнути кнопку зі стрілкою , розташовану на вкладці ліворуч.

Кожний об'єкт має ім'я і властивості. Поведінка об'єкта і його взаємодія з іншими об'єктами описується за допомогою **методів**, які фактично є аналогами підпрограм процедурних мов програмування. **Клас** — це сукупність змінних і методів їх опрацювання. Часто говорять, що **клас** — це **шаблон об'єкта**, а **об'єкт** — це **екземпляр класу**.



Швейцарський інженер і дослідник світу програмування. Автор і один з розробників мови програмування Паскаль. Ніклаус Вірт є одним з найавторитетніших у світі вчених в області комп'ютерних наук, його книга «Алгоритми + структури даних = програми» вважається одним із класичних підручників зі структурного програмування.

У першому стовпці вкладки **Події** відображається ім'я події, а в другому — ім'я підпрограми для її опрацювання (ім'я опрацювача події).

Розробка програми у середовищі Lazarus

- створення інтерфейсу програми
- розробка програмного коду
- компіляція, зберігання і виконання проекту

Рис 3. Етапи розробки програми у середовищі візуального програмування Lazarus

Властивості поділяються на *прості* та *складні*. Прості властивості мають лише одне значення, а складні — декілька значень. Наприклад, властивість **Caption** (Заголовок) визначається рядом символів, а властивість **Font** (Шрифт) — це набір шрифтів, із яких можна вибрати один конкретний шрифт. Ліворуч від назви складних властивостей розташований трикутник. Якщо клацнути світлий трикутник (після цього він набуде темного кольору), то відкриється список його значень. Клацання на темному трикутнику приведе до згортання списку. Біля простих властивостей трикутника немає.

Для активації будь-якої властивості необхідно клацнути на ній кнопкою миші. Після активації властивості в кінці рядка може з'явитися кнопка з трьома точками і кнопка зі стрілкою вниз. Після клацання кнопки з трьома точками відкриється діалогове вікно для вибору відповідних значень. Наприклад, після активації властивості **Color** відкриється вікно **Колір**. Активація кнопки трикутника приведе до розкриття списку можливих значень цієї властивості.

У верхній частині вікна інспектора об'єктів міститься список імен усіх компонентів, розташованих у даний момент на формі. Після клацання мишею імені компонента у цьому списку виконується активація відповідного компонента на формі. Налаштування вікна інспектора об'єктів здійснюється за допомогою його контекстного меню.

Вікно редактора тексту (див. [рис. 2](#)) призначено для введення і редагування програмного коду. Фактично у редакторі тексту міститься структура (шаблон) майбутньої програми, що полегшує роботу програміста.



Ключові слова виділяються напівжирним шрифтом, знаки пунктуації — червоним кольором, коментарі — синім, а помилки — коричневим.

Вікно форми, або конструктор форми, призначено для створення інтерфейсу майбутньої програми, яку розробляє користувач. На початку роботи воно містить лише заголовок і кнопки управління вікном форми (див. [рис. 1](#)). Потім користувач наповнює її необхідними компонентами.

За замовчуванням на передньому плані міститься вікно редактора тексту, яке закриває форму. Активувати вікно можна за допомогою клавіші F12 або команди **Вигляд → Перемкнути форму/модуль**.

Для створення інтерфейсу на формі розміщують необхідні компоненти і встановлюють їхні властивості. Після розміщення на формі будь-якого компонента у вікні редактора тексту автоматично генерується відповідний програмний код. Користувач повинен доповнити його необхідними командами для реалізації відповідного алгоритму. Усі файли, з яких складається програмний код, називають **проект**.

Програми у середовищі Lazarus розробляють у три етапи ([рис. 3](#)).

Для створення інтерфейсу майбутньої програми у середовищі Lazarus використовують форму, а для введення і редагування програмного коду — вікно редактора. Після розміщення компонента на форму автоматично змінюється програмний код у вікні редактора тексту. Найчастіше використовують компоненти групи Standard, у якій містяться стандартні елементи інтерфейсу. Такими компонентами, зокрема, є: TButton (командна кнопка), TLabel (поле для розміщення однорядкових написів), TMemo (багаторядковий текстовий редактор) та ін.



Форма та інші об'єкти реагують на події. Перелік подій міститься в інспекторі об'єктів на вкладці **Події**. Найчастіше застосовується подія **OnClick**. Програміст розробляє код, який відповідно реагує на ту чи іншу подію. Такий код називають обробником події.

Програма у середовищі Lazarus є **проектом**. Проект об'єднує декілька файлів, із яких створюється єдиний файл, що виконується. У результаті компіляції з файлів проекту створюється єдиний файл, що виконується, з розширенням exe. Ім'я файла збігається з іменем проекту. Користувач розробляє програмний модуль, усі інші додаються до проекту автоматично. Заголовок модуля починається ключовим словом unit, за яким слідує ім'я модуля і крапка з комою. Розділ опису починається ключовим словом interface. Тут описуються компоненти програмного коду: типи, класи, процедури і функції. Розділ implementation (див. [рис. 1](#)) містить програмний код опрацювання даних, який розробляє користувач.



«Гарний токарь працює у декілька разів краще, ніж середній, а гарний програміст коштує у 10000 разів більше, ніж звичайний».

Білл Гейтс

Із детальним описом методики створення проекту в середовищі візуального програмування Lazarus ви можете ознайомитися на сайті interactive.ranok.com.ua



Запитання для перевірки знань

- 1 Які вікна має середовище Lazarus?
- 2 За допомогою яких команд виконується компіляція і запуск програми?
- 3 Для чого призначено вікно редактора тексту?
- 4 Для чого призначена панель інструментів головного вікна?
- 5 Як встановлюються події для об'єктів, розташованих на формі?
- 6 Для чого призначено вікно інспектора об'єктів?



Завдання для самостійного виконання

- 1 Запустіть середовище Lazarus. Відкрийте вікно форми. Змініть розміри і місце розташування вікна форми.
- 2 Розмістіть на формі компоненти: TLabel, TEdit, TMemo, TListBox, TButton. Для кожного з них установіть необхідні властивості за власним бажанням.
- 3 На формі з іменем **Моя робота** розмістіть два компоненти TLabel, компонент TEdit і компонент TButton. Установіть раціональні, на ваш погляд, їхні розміри. Самостійно визначте і встановіть для них необхідні властивості.
- 4 На формі з іменем **Спільна робота** розмістіть компоненти: TPopupMenu, TCheckBox, TMemo. Експериментальним шляхом спробуйте визначити їх функціональне призначення.

9.2. Поняття типів користувача і масиву



Приклад 1.

```
type autumn=(oct, nov, dec);
var a_1: autumn;
```

У цьому прикладі змінна `a_1` може набути будь-якого з трьох значень: `oct`, `nov`, `dec`.

```
Приклад 2. type letter='a'..'n';
var x: letter;
```

Змінній `x` можна присвоювати символи від 'a' до 'n'.

Приклад 3. Прикладом одно-вимірного масиву рядкового типу є назви пристроїв комп'ютера:

Номер елемента масиву	Значення елементів масиву
1	монітор
2	миша
3	принтер
4	клавіатура
5	сканер

Приклад 4.

```
var mas1: array [1..7] of integer; —
змінна mas1 оголошена одно-
вимірним масивом цілих чи-
сел, у якому 7 елементів типу
integer.
```

Пригадайте, які типи даних ви вивчали у 8 класі. Чому, на вашу думку, не можна в мовах програмування обмежитися лише простими типами даних?

У 8 класі ви вивчали базові типи даних середовища Lazarus. Нагадаємо, що базовими є прості (скалярні) типи. Існують п'ять стандартних простих типів: *цілі*, *дійсні*, *символьні*, *бульовські* й тип *дата-час*. Простим типом є також перелічуваний тип, але він оголошується користувачем (приклади 1, 2). Складені типи будуються з простих за певними правилами. Серед складених (структурованих) найчастіше застосовуються масиви, формування яких засновано на використанні інших типів.

Незважаючи на те що середовище Lazarus має потужну структуру стандартних (вбудованих у мову) типів даних, воно надає також можливість створення нових типів, які називають *користувацькими*. Структура їх оголошення така:

```
type <новий тип даних>=<визначення типу>;
var <список змінних: новий тип даних>;
```



Масив — це структурований тип даних, елементи якого мають один тип, наприклад `integer`, `char` та `in`.

Структура масиву може бути *одновимірною (лінійною)*, *двовимірною (табличною)* та *багатовимірною*. Загальну структуру одновимірного масиву можна позначити так:

$$x[1], x[2], x[3], \dots, x[n-1], x[n]$$

Місце елемента у масиві, тобто його порядковий номер, називають **індексом**. Індекс записують у квадратних дужках, наприклад запис `x[i]` означає *i*-й елемент масиву. Так, числа 105, 11, 173, 35, 40 можна розглядати одновимірним масивом цілих чисел, у якому п'ять елементів. У прикладі 3 видно, що першим елементом масиву є монітор, другим — миша.

Перш ніж опрацювати масив, його потрібно оголосити. Оголошення можна зробити такими способами.

- У розділі змінних. Структура оголошення одновимірного масиву в цьому розділі така:

```
var <ім'я змінної>: array [n1..n2] of <тип елементів масиву>;
```

Тут `array` — ключове слово, яке вказує, що змінна є масивом; у квадратних дужках визначається діапазон індексів: `n1` — індекс першого елемента масиву; `n2` — індекс останнього елемента масиву. Після слова `of` вказується тип елементів масиву (приклад 4).

Масив також може бути оголошений за такою структурою:

```
var <ім'я змінної>: array [n1..n2] of <тип елементів>=
(значення елементів);
```

Наприклад:

```
var mas3: array [1..4] of string=('байт', 'миша', 'файл', 'змінна');
```

- У розділі **типів**. Структура оголошення масиву така:
`type <ім'я типу>=array [n1..n2] of <тип елементів>;`
`var <ім'я змінної>: <ім'я типу>;`

Із наведеної структури видно, що після оголошення типу оголошується ім'я масиву.

Наприклад, `type mas4=array [0..5] of real; var s: mas4;`

- У полі **const**. Оголошення масиву має таку структуру:
`const <ім'я змінної>=array [n1..n2]`
`of <тип елементів>=(значення елементів);`

Наприклад, `const mas5=array [1..4] of integer=(33, 54, 2, 32);`

Індексами елементів масивів можуть бути дані будь-якого типу, в тому числі вирази, але найчастіше ними є цілі числа. Якщо індексом є змінна, то її необхідно оголосити константою.

Наприклад, `const n=7; var a: array [1..n] of integer;`

Будь-якому елементу масиву можна присвоїти певне значення за допомогою оператора присвоювання. Наприклад, `mas[4]:=5;` — четвертому елементу одновимірного масиву `mas` присвоєно значення 5.

Для одночасного опрацювання декількох типів даних у багатьох мовах програмування, в тому числі у мові Free Pascal, застосовується спеціальний тип, який називають записом. Формат опису змінної типу **запис** такий:

```
var <ім'я запису>: record
    ім'я1: тип1;
    ім'я2: тип2;
    ...
    ім'яN: типN;
end;
```

Наприклад,

```
var reds: record
    name: string [15]; //прізвище
    uspitch: real; //середній бал
end;
```

? Запитання для перевірки знань

- 1 Наведіть визначення масиву.
- 2 Які існують структури масивів?
- 3 Назвіть способи оголошення масивів.
- 4 Як здійснюється звернення до елементів масиву?
- 5 Яким може бути тип індексу елемента масиву?
- 6 Наведіть загальну структуру одновимірного масиву.
- 7 Як оголошується одновимірний масив у розділі оголошення змінних?
- 8 За якою структурою оголошуються масиви в розділі типів?
- 9 Наведіть приклад оголошення одновимірного масиву в розділі **const**.

💻 Завдання для самостійного виконання

- 1 Оголосіть у розділі змінних масив рядкового типу з елементами: файл, біт, папка, дані.
- 2 Оголосіть у розділі типів масив чисел: 2, 6, 2.1, 40, 7.
- 3 Оголосіть у полі **const** масив: процесор, пам'ять, плата, символ.
- 4 Визначте значення елементів `a[1]` і `a[3]` в результаті оголошення масиву:
 `var z: array [0..3] of string=('21', 'стіл', 'лист', '3.5');`
- 5 Визначте значення елемента `a[2]` в масиві:
 `const a: array [-1..3] of real=(105, 77, 2.4, 60, 107);`
- 6 Знайдіть помилки в оголошенні масиву:
 `var l_a: array [-3,2] of char;`
- 7 Знайдіть помилки в оголошенні масиву:
 `var mas: array [-1..2] of real=('байт', 'біт', 'миша');`

9.3. Введення і виведення значень елементів масиву



Наведіть означення масиву. Які існують способи оголошення масиву? Чи можна під час оголошення масиву присвоїти елементам масиву необхідні значення? Які, на ваш погляд, компоненти можна використовувати для введення даних у масиві?



Програміст пише у середньому 10–12 рядків програмного коду, які потрапляють у кінцевий продукт.

Елементи масиву можуть набувати значень різними способами: присвоюватися під час оголошення масиву, вводитися з клавіатури, присвоюватися під час виконання програми. Способи і методи введення і виведення елементів масиву для консольних і візуальних додатків відрізняються.

У візуальних додатках середовища Lazarus для формування масивів, уведення значень його елементів із клавіатури, а також виведення елементів найчастіше застосовуються багаторядкові текстові поля TMemo і TListBox, а також функції InputBox() і ShowMessage(). Можуть застосовуватися також і однорядкові текстові поля TEdit і TLabel. Розглянемо створення, введення і виведення елементів одновимірних масивів на конкретних прикладах.

- Використання компонента TMemo для введення і компонента TListBox для виведення значень елементів одновимірного масиву.

Приклад 1. Розмістимо на формі компоненти: TMemo, TListBox і TEdit. Надамо відповідним властивостям цих компонентів значень згідно з рис. 1. Уведемо з клавіатури 10 назв найбільших річок України: Дніпро, Десна, Дністер, Сіверський Донець, Псел, Західний Буг, Горинь, Прут, Прип'ять, Сейм. Для цього активуємо об'єкт Memo1, в інспекторі об'єктів виділимо властивість Lines і клацнемо в цьому рядку кнопку з трьома крапками. Відкриється вікно редактора рядків, у яке введемо назви десяти перелічених річок. Після натискання кнопки ОК відкриється вікно форми (рис. 1).

Встановимо також значення ssVertical (вертикальна смуга прокручування) для властивості

```
procedure TForm1.Button1Click(Sender: TObject);
var i:integer; //змінна цілого типу
    mas:array [1..10] of string; //масив рядкового типу
begin
    for i:=1 to 10 do //цикл уведення елементів масиву
        mas [i]:=Memo1.Lines [i-1]; //введення елементів масиву
    for i:=1 to 10 do //цикл виведення елементів масиву
        ListBox1.Items [i-1]:=mas [i]; //виведення елементів масиву
end;
end.
```

Рис. 2. Код для введення і виведення масиву з використанням компонентів TMemo і TListBox

ScrollBars об'єкта Memo1. Активуємо об'єкт Edit1, виберемо для нього подію OnClick і введемо код, зображений на рис. 2.

Після успішної компіляції програми виконуємо її. Результат виконання зображено на рис. 3. За допомогою смуг прокручування можна переглянути всі назви річок.

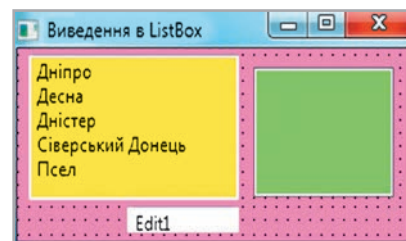


Рис. 1. Назви річок України у полі об'єкта Memo1

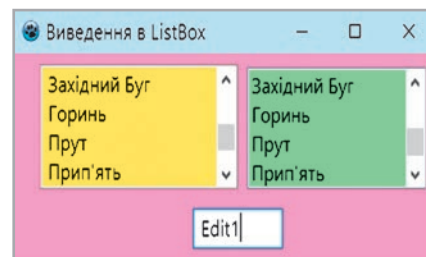


Рис. 3. Уведені та виведені назви річок України

- Формування масиву випадковими числами і виведення їх у текстові поля TListBox і TMemo.



Приклад 2. Генеруються 12 випадкових чисел з діапазону від 0 до 6. Із цих чисел формується масив, значення елементів якого виводяться в поле об'єкта ListBox1, а в поле об'єкта Memo1 виводяться всі елементи цього масиву, крім елементів, значення яких дорівнюють 6. Код, що реалізує це завдання, зображено на рис. 4. На формі розміщено об'єкти Memo1, ListBox1, Label1, Label2 і Button1.

Нагадаємо, щоб при кожному запуску програми генерувалися різні випадкові числа,

```
procedure TForm1.Button1Click(Sender: TObject);
var i:integer; //змінна цілого типу
    mas:array [1..12] of integer; //масив цілого типу
begin
    Randomize;
    for i:=1 to 12 do //цикл для створення масиву
    begin //початок операторних дужок
        mas[i]:=Random(7); //присвоєння значення елементу масиву
        ListBox1.Items.Add(IntToStr(mas[i])); //виведення елемента масиву
    end; //кінець операторних дужок
    for i:=1 to 12 do //цикл для виведення масиву
    if mas[i]<>6 then //чи дорівнює елемент масиву 6?
        Memo1.Lines.Add(IntToStr(mas[i])); //виведення елементів масиву
    end;
end.
```

Рис. 4. Код створення і виведення масиву випадкових чисел

на початку програми встановлено оператор Randomize. Без цього оператора кожного разу будуть генеруватися однакові числа.

На рис. 5 зображено один із можливих результатів виконання програми.

Зверніть увагу на те, що в поле об'єкта Memo1 виведені всі елементи масиву, крім елементів зі значенням 6.

Властивостям компонентів надано значень, які забезпечують їх зовнішній вигляд, зображений на рис. 5.

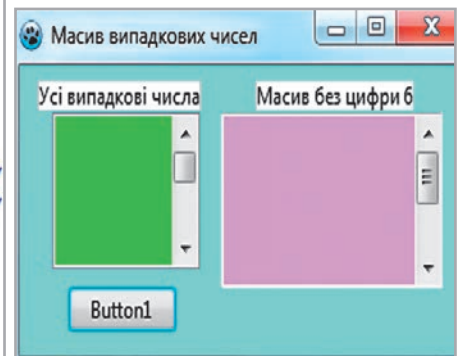


Рис. 5. Результати виведення масиву випадкових чисел

- Формування масиву на основі обчислених значень і виведення їх у декілька колонок текстового поля TListBox.



Приклад 3. Створимо масив, значеннями елементів якого є висота вільного падіння тіла після кожної із дванадцяти секунд (висоту обчислюють за формулою $h = \frac{gt^2}{2}$). Для того щоб значення елементів масиву виводилися колонками, властивості Columns об'єкта

```
procedure TForm1.Button1Click(Sender: TObject);
var t:integer; //змінна цілого типу
    mas:array [1..12] of real; //масив дійсних чисел
begin
    for t:=1 to 12 do //цикл для створення масиву
    mas[t]:=9.8*t*t/2; //створення масиву
    for t:=1 to 12 do //цикл для виведення масиву
    ListBox1.Items.Add(FloatToStr(mas[t])); //виведення масиву
    end;
end.
```

Рис. 6. Код створення і виведення масиву значень висоти вільного падіння тіла

ListBox1 надамо значення 2. Код реалізації цього завдання наведено на рис. 6.

На формі розміщено об'єкти ListBox1, Label1, Button1. Їхнім властивостям надано значень, щоб зовнішній вигляд компонента був орієнтовно таким, як зображено на рис. 7.

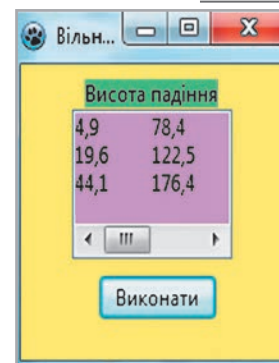


Рис. 7. У полі об'єкта ListBox1 виведено значення висоти вільного падіння тіла

- Формування масиву в процесі його оголошення і виведення в текстове поле TMemo.

Приклад 4. Варіант коду формування масиву цілих чисел у процесі його оголошення і виведення в поле об'єкта Memo1 подано на рис. 8.

```
procedure TForm1.Button1Click(Sender: TObject);
var i: integer;
    mas: array [1..12] of integer=(44, 32,
    3, 87, 55, 7, 21, 10, 90,75, 66, 53);
begin
    for i:=1 to 12 do
        Memo1.Lines.Add (IntToStr (mas[i]));
    end;
end.
```

Рис. 8. Код формування масиву в процесі його оголошення і виведення в поле об'єкта Memo1

Після успішної компіляції програми і її виконання отримаємо результат, зображений на рис. 9.

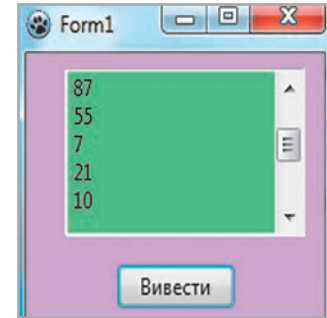


Рис. 9. Масив цілих чисел, виведений у поле об'єкта Memo1

- Уведення елементів масиву за допомогою функції InputBox() і виведення за допомогою функції ShowMessage().

Приклад 5. На рис. 10 зображено код із використанням функції InputBox() для циклічного введення елементів масиву рядкового типу і функції ShowMessage() для виведення елементів цього масиву.

У процесі введення даних із клавіатури зручно застосовувати функцію InputBox(), яка має таку структуру: InputBox (заголовок, підказка, рядок за замовчуванням), де: заголовок — це заголовок вікна (1) (рис. 11), яке висвітлюється на формі під час виконання функції; підказка — підказка для користувача (2) (рис. 11); рядок за замовчуванням — рядок для введення значення елемента масиву (3) (рис. 11). Якщо клацнути кнопку ОК, то елементу буде присвоєно введене значення (на рис. 12 це слово

алгоритм). Якщо натиснути кнопку Cancel, то функція повертає значення рядка за замовчуванням.

Під час виконання функції InputBox() виводиться вікно (див. рис. 11), програма призупиняє виконання, у рядок слід ввести значення певного елемента масиву, клацнути кнопку ОК і ввести значення наступного елемента. Після завершення введення всіх елементів виконання програми продовжується.

Функція ShowMessage() виводить на форму вікно із заданим текстом.

У вікно, зображене на рис. 12, виведено перший елемент масиву. Після кожного клацання кнопки ОК виводиться черговий елемент до завершення виведення всіх елементів.

```
procedure TForm1.Button1Click(Sender: TObject);
var i: integer; //змінна цілого типу
    mas: array [1..7] of string; //масив рядкового типу
begin
    for i:=1 to 7 do //цикл для введення масиву
        mas [i]:=InputBox ('Уведення', //уведення масиву
        'Увести елемент N'+IntToStr(i), ' ');
    for i:=1 to 7 do //цикл для виведення масиву
        ShowMessage ('Елемент N'+IntToStr(i) //виведення масиву
        +'-'+mas[i]);
    end;
end.
```

Рис. 10. Код введення й виведення масиву за допомогою функцій InputBox() і ShowMessage() відповідно

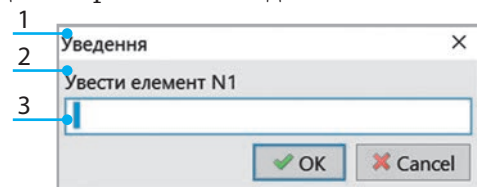


Рис. 11. Вікно функції InputBox() для введення значень елементів масиву

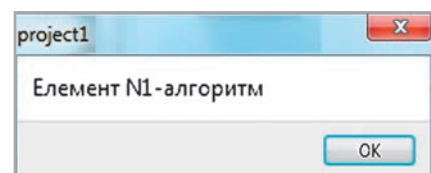


Рис. 12. Вікно функції ShowMessage() для виведення значень елементів масиву

- Уведення й виведення елементів масиву в консольних програмах.

Уведення значень елементів масиву в консольних програмах здійснюється за допомогою клавіатури. Для цього використовують оператори `readln` або `read` і оператори циклу. Спочатку вводиться значення першого елемента, потім — другого, третього і т. д. Виведення значень елементів масиву виконується за допомогою операторів `write` і `writeln`.



Приклад 6. На рис. 13 наведено код, за допомогою якого елементи одновимірного масиву (дійсні числа) вводяться за допомогою клавіатури, а потім виводяться на екран. Масив складається із 6 чисел.

Якщо з клавіатури будуть уведені числа: 12, 13.2, 3, 40.5, 5, 6.7, то на екран буде виведено результат виконання програми (рис. 14).

```
var
  i:integer;           //змінна цілого типу
  mas:array [1..6] of real; //масив дійсного типу
begin
  for i:=1 to 6 do    //цикл введення масиву
  begin              //початок операторних дужок
    writeln ('увести a[' , i, ']='); //повідомлення про введення
    readln (mas [i]) //введення елементів масиву
  end;               //кінець операторних дужок
  for i:=1 to 6 do    //цикл виведення масиву
    writeln ('mas[' , i, ']=', mas[i]); //виведення елементів масиву
    readln           //призупинення виконання програми
end.
```

```
mas[1]=12
mas[2]=13.2
mas[3]=3
mas[4]=40.5
mas[5]=5
mas[6]=6.7
```

Рис. 13. Код введення одновимірного масиву за допомогою клавіатури

Рис. 14. Результат виконання програми

? Запитання для перевірки знань

- 1 Які компоненти застосовують для введення і виведення масивів?
- 2 За допомогою яких операторів вводять елементи масиву в консольних додатках?
- 3 За допомогою яких операторів виводять елементи масиву в консольних додатках?
- 4 Які функції застосовують для введення і виведення даних у масив?
- 5 Поясніть використання функції `ShowMessage()`.
- 6 Поясніть порядок використання функції `InputDialog()`.
- 7 Як можуть формуватися масиви?
- 8 Поясніть методику використання компонента `TMemo` для введення даних.
- 9 Поясніть методику використання компонента `TListBox` для виведення даних.

📁 Завдання для самостійного виконання

- 1 Розробіть консольну програму для введення і виведення масиву, що містить слова: змінна, масив, дані, параметри, рекурсія.
- 2 Розробіть консольну програму формування масиву 10 цілих випадкових чисел з діапазону від 0 до 100.
- 3 Розробіть код для введення і виведення за допомогою компонента `TMemo` масиву, що містить 5 дійсних чисел.
- 4 Розробіть код із використанням компонента `TListBox` для формування і виведення масиву 10 випадкових чисел з діапазону від 0 до 20.
- 5 Розробіть код із використанням функцій `InputDialog()`, `ShowMessage()` і компонента `TListBox` для введення і виведення масиву цілих чисел.

9.4. Класичні алгоритми опрацювання числових одновимірних масивів

Класичними є алгоритми знаходження загальної суми значень елементів числового масиву, суми та кількості значень його елементів, що задовольняють певні умови. Для масивів із будь-якими типами даних класичними є також алгоритми пошуку елементів, що задовольняють задані умови, та алгоритми впорядкування елементів масивів.

9.4.1. Знаходження суми і кількості заданих елементів масиву



Назвіть компоненти і функції, які можна використовувати для введення даних у масив і виведення даних із нього. Яким із них ви надаєте перевагу і чому? Які, на ваш погляд, операції можна виконувати над значеннями елементів масиву?

Приклад 1. Нехай задано масив із чотирьох чисел: 2, 6, 24, 7. Тобто $n=4$, $a[1]=2$, $a[2]=6$, $a[3]=24$, $a[4]=7$. Сума обчислюється таким чином:

```
s=0;
i=1: s=s+a[1]=0+2=2,
i=2: s=s+a[2]=2+6=8,
i=3: s=s+a[3]=8+24=32,
i=4: s=s+a[4]=32+7=39.
```



Числові масиви — це масиви, значеннями елементів яких є цілі або дійсні числа.

• Знаходження загальної суми значень елементів масиву.

Нехай дано одновимірний масив цілих або дійсних чисел $a[1]$, $a[2]$, ..., $a[n]$. Обчислити суму значень елементів цього масиву можна методом послідовного накопичення. Сутність цього методу полягає в тому, що до початку додавання початкове значення суми вважається таким, що дорівнює нулю, тобто $s=0$. Потім до цього значення додається значення першого елемента масиву: $s=s+a[1]$. До отриманої суми додається значення другого елемента масиву: $s=s+a[2]$ і так далі до останнього елемента (приклад 1).

Розглянемо алгоритм знаходження суми значень елементів масиву, що задовольняють певну умову.

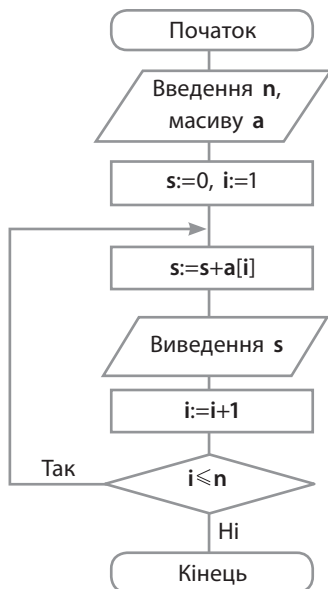


Рис. 1. Блок-схема алгоритму обчислення суми елементів одновимірному масиву з виведенням поточних значень суми

Крок 1	Увести масив a	Крок 5	$i:=i+1$
Крок 2	$s:=0$	Крок 6	Якщо $i \leq n$, то крок 4, інакше — крок 7
Крок 3	$i:=1$	Крок 7	Кінець
Крок 4	$s:=s+a[i]$		

Блок-схему алгоритму обчислення подано на рис. 1.

Приклад 2. Варіант коду обчислення суми значень елементів масиву цілих чисел

```
procedure TForm1.Edit1Click(Sender: TObject);
const n=6;           //константа типу змінної
var i, s:integer;    //змінні цілого типу
//оголошення і присвоєння значень елементам масиву
mas:array [1..n] of integer=(3,5,7,10,12,14);
begin
s:=0;               //початкове значення суми
Edit1.Text:='Сума='; //виведення 'Сума=' в об'єкт Edit1
for i:=1 to n do    //цикл обчислення суми
s:=s+mas[i];       //обчислення суми
Edit1.Text:=Edit1.Text+IntToStr(s); //виведення результату
end;
end.
```

Рис. 2. Код обчислення суми значень елементів масиву

із використанням компонента TEdit зображено на рис. 2. Для об'єкта Edit1 обрано подію OnClick. Після успішної компіляції і виконання програми отримаємо результат, зображений на рис. 3.

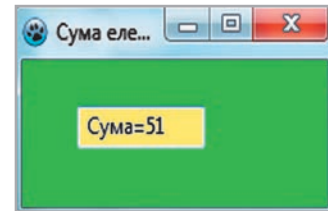


Рис. 3. Результат обчислення суми елементів масиву

- Знаходження кількості елементів, що дорівнюють значенню заданого.

Нехай у масиві $a[1], a[2], a[3], \dots, a[n]$ потрібно визначити кількість елементів, значення яких дорівнюють c . Сутність алгоритму полягає у послідовному порівнянні всіх елементів масиву, починаючи з першого елемента, із заданим значенням. Якщо значення елемента масиву дорівнює заданому, то показник кількості збільшується на одиницю.

Алгоритм знаходження кількості заданих елементів у масиві може бути таким:

Крок 1	Увести значення c і масив a	Крок 5	$m:=m+1$
Крок 2	$m:=0$	Крок 6	$i:=i+1$
Крок 3	$i:=1$	Крок 7	Якщо $i \leq n$, то крок 4, інакше — крок 8
Крок 4	Якщо $a[i]=c$, то крок 5, інакше — крок 6	Крок 8	Кінець

Як бачимо, змінній m , у якій підраховується кількість елементів, спочатку присвоюється значення нуль. Потім порівнюється значення елемента $a[1]$ зі значенням c . Якщо їхні значення збігаються, то значення змінної m збільшується на одиницю, інакше її значення не змінюється. На наступному кроці порівнюється значення елемента $a[2]$ зі значенням c і виконуються дії, аналогічні тим, що виконувалися на попередньому кроці. Потім порівнюється значення елемента $a[3]$ і так далі до $a[n]$. Блок-схему алгоритму подано на рис. 4.

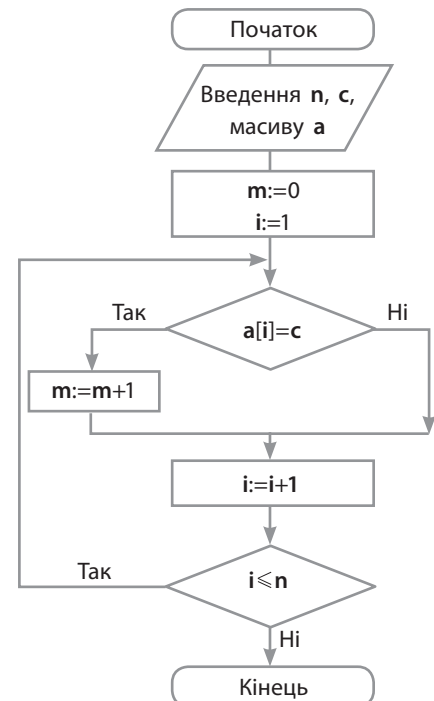


Рис. 4. Блок-схема алгоритму знаходження кількості елементів, що дорівнюють значенню заданого

Приклад 3. На рис. 5 зображено код, за допомогою якого створюється масив із 40 цілих випадкових чисел з діапазону від 0 до 6. Значення кожного елемента масиву виводиться в поле об'єкта Memo1 й одразу порівнюється з цифрою 4. Якщо значення елемента дорівнює 4, то значення змінної m збільшується на одиницю. Після виконання

40 циклів значення змінної m виводиться у поле об'єкта Label1. Для цього об'єкта обрано подію OnClick.

На рис. 6 зображено можливий результат виконання програми. У полі об'єкта Memo1 є повзунок, за допомогою якого можна переглянути, у якому порядку генерувалися всі випадкові числа.

```

procedure TForm1.Label1Click(Sender: TObject);
var i, m: integer; //змінні цілого типу
    mas: array [1..40] of integer; //масив цілого типу
begin
    m:=0; //початкова кількість чисел
    //при кожному виконанні програми будуть різні випадкові числа
    Randomize;
    for i:=1 to 40 do //цикл
    begin //відкриття операторних дужок
        mas[i]:=Random (7); //формування масиву
        Memo1.Lines.Add(IntToStr(mas[i])); //виведення масиву
        if mas[i]=4 then m:=m+1; //чи дорівнює елемент числу 4?
    end; //закриття операторних дужок
    Label1.Caption:='Цифр, що дорівнюють 4='+IntToStr(m); //виведення результату
end;
end.

```

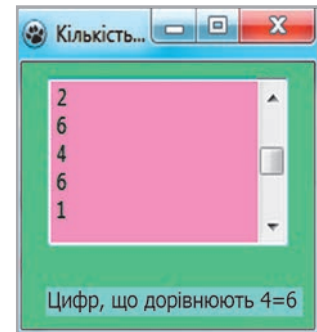


Рис. 5. Код обчислення кількості цифр 4, що містяться у масиві

Рис. 6. Результат обчислення кількості цифр у масиві

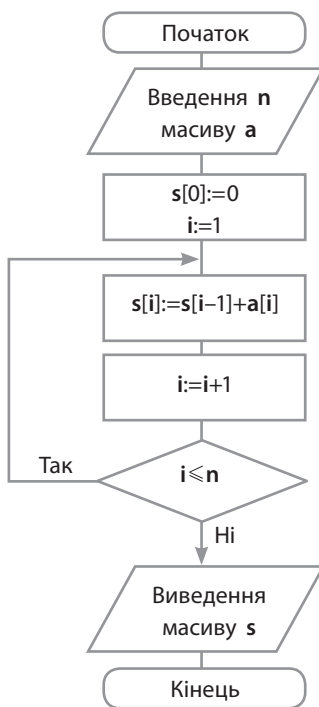


Рис. 7. Блок-схема алгоритму створення нового масиву



Запитання для перевірки знань

- 1 Назвіть класичні алгоритми опрацювання одновимірних масивів.
- 2 Поясніть сутність алгоритму обчислення суми значень елементів одновимірного масиву.
- 3 Накресліть блок-схему алгоритму обчислення кількості елементів масиву, значення яких дорівнюють заданому.
- 4 Накресліть блок-схему алгоритму обчислення в одновимірному масиві кількості елементів, значення яких дорівнюють заданому.



Завдання для самостійного виконання

- 1 Дано масив чисел: 71, 2, 7, 12, 4, 5, 17, 10. Розробіть програму обчислення суми чисел масиву, більших за 8.
- 2 Дано масив чисел: 66, 3, 12, 7, 9, 22, 44, 15. Розробіть програму обчислення кількості чисел, менших від 22.
- 3 На рис. 7 наведено блок-схему алгоритму створення нового масиву s із елементів заданого масиву a . Розробіть програму реалізації цього алгоритму.

9.4.2. Пошук даних у масиві

У повсякденному житті часто доводиться здійснювати пошук необхідного об'єкта серед багатьох із них. Чи можна такі процеси подати у вигляді загального алгоритму?

У масивах часто доводиться здійснювати пошук елемента, значення якого збігається із заданим. Існують різні методи пошуку потрібних елементів у масиві, всі вони базуються на **переборі елементів масиву**. Перебір може бути *повний* (прямий), за яким перевіряються всі елементи масиву, і *неповний* (скорочений). Нижче описано сутність двох найпростіших методів пошуку даних у масиві: лінійний і двійковий.

- **Лінійний пошук даних.**

 Лінійний пошук базується на прямому переборі елементів масиву.

Нехай дано масив $a[1], a[2], \dots, a[n]$ і значення c (ключ). Потрібно визначити, чи є у цьому масиві елемент, значення якого збігається зі значенням c .

Сутність лінійного пошуку така. Спочатку c порівнюється з $a[1]$. Якщо вони збігаються, робиться висновок, що елемент знайдено на першій позиції масиву, і на цьому пошук завершується. Інакше c порівнюється з $a[2]$ і робиться аналогічний висновок, потім — з $a[3]$ і так далі до $a[n]$.

Блок-схему алгоритму подано на рис. 1.

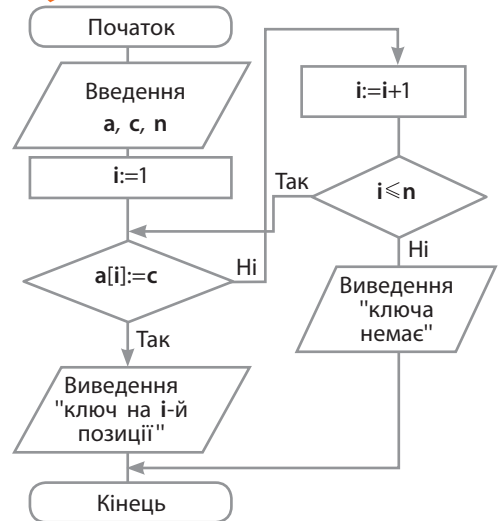


Рис. 1. Блок-схема алгоритму лінійного пошуку в одновимірному масиві, де i — лічильник кількості переглянутих елементів; n — загальна кількість елементів у масиві

Приклад 1. Визначити, чи є у масиві з 10 випадкових цілих чисел з діапазону від 0 до 6 число 5. Тобто достатньо знайти номер першої позиції, на якій розміщено такий елемент. Програмний код подано на рис. 2. У програмі використано об'єкти: `ListBox1` — для виведення елементів масиву, `Label1` — для виведення

повідомлення про наявність ключа, `Label2` — для виведення повідомлення про номер позиції, на якій розташовано ключ. Якщо ключа немає, виводиться нульова позиція. Для об'єкта `Label1` обрано подію `OnClick`.

На рис. 3 зображено один із результатів виконання програми.

```

procedure TForm1.Label1Click(Sender: TObject);
const n=10; //константа типу змінної
var mas: array [1..n] of integer; //масив цілого типу
    p:string; i, c, k: integer; //змінні різних типів
begin
  Randomize; //забезпечує генерування різних випадкових чисел
  c:=5; //значення ключа
  p:='Ключа немає'; //ознака відсутності ключа
  for i:=1 to n do //цикл створення масиву
  begin //початок операторних дужок
    mas[i]:=Random (7); //формування масиву
    ListBox1.Items.Add(IntToStr (mas[i])) //виведення масиву
  end; //кінець операторних дужок
  for i:=1 to n do //цикл аналізу масиву
  if mas[i]=c then begin //чи дорівнює елемент значенню c?
    k:=i; //запам'ятовується номер позиції
    p:='Ключ є'; //ознака - ключ знайдено
    Break end; //переривання циклу
  Label1.Caption:=p; //повідомлення про наявність ключа
  Label2.Caption:='Ключ на позиції-'+IntToStr(k); //виведення номера позиції
end; end.

```

Рис. 2. Код реалізації лінійного пошуку заданого значення у масиві

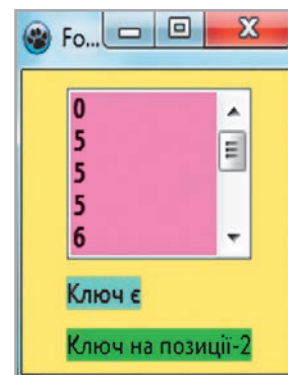


Рис. 3. Можливий результат пошуку елемента в масиві

Розглянутий метод лінійного пошуку є найпростішим. Але його програмна реалізація потребує більшого часу, оскільки в ньому не враховуються закономірності розташування елементів масиву. Якщо значення елементів масиву є впорядкованими за певною ознакою, пошук можна організувати раціональніше — методом двійкового пошуку даних.

- **Двійковий пошук даних.**



Двійковий пошук можна застосовувати лише до впорядкованих масивів.

Приклад 2. Нехай дано масив цілих чисел: 2, 5, 8, 9, 13, 15, 20, 21 і ключове значення $c=15$. У масиві міститься 8 елементів. Номер середнього елемента дорівнює 4, тому що $\lfloor(1+8)/2\rfloor=4$. Оскільки $a[4]<15$, далі пошук будемо проводити в частині масиву, що складається з елементів: 13, 15, 20, 21. У цьому масиві номером середнього елемента є число $\lfloor(5+8)/2\rfloor=6$. Значення шостого елемента дорівнює 15 і збігається зі значенням c . На цьому виконання алгоритму завершується.

Нехай дано масив $a[1], a[2], \dots, a[n]$, елементи якого впорядковані за зростанням їхніх значень. Потрібно визначити, чи є в цьому масиві елемент, значення якого збігається зі значенням c .

Сутність цього методу така. На кожному кроці масив ділиться навпіл та визначає, в якій із частин може міститися шуканий елемент, інша частина не розглядається. У процесі пошуку значення лівої і правої меж масиву будуть постійно змінюватися.

Позначимо поточне значення лівої межі масиву змінною l , а значення правої межі — змінною p (початкові значення змінних: $l=1$ і $p=n$). Спочатку в цьому масиві обирають елемент, розташований у середині масиву, — $a[i]$. Значення індексу середнього елемента можна визначити за формулою $i=\lfloor(l+p)/2\rfloor$ (квадратними дужками позначена ціла частина числа). Значення середнього елемента порівнюють із ключовим значенням c . Якщо $c=a[i]$, елемент знайдено. Якщо $c < a[i]$, то далі для пошуку обирають частину масиву, розташовану ліворуч від $a[i]$, у протилежному випадку — частину масиву, розташовану праворуч від $a[i]$. Для обраної частини процес повторюють (приклад 2).

Блок-схему алгоритму подано на рис. 4.

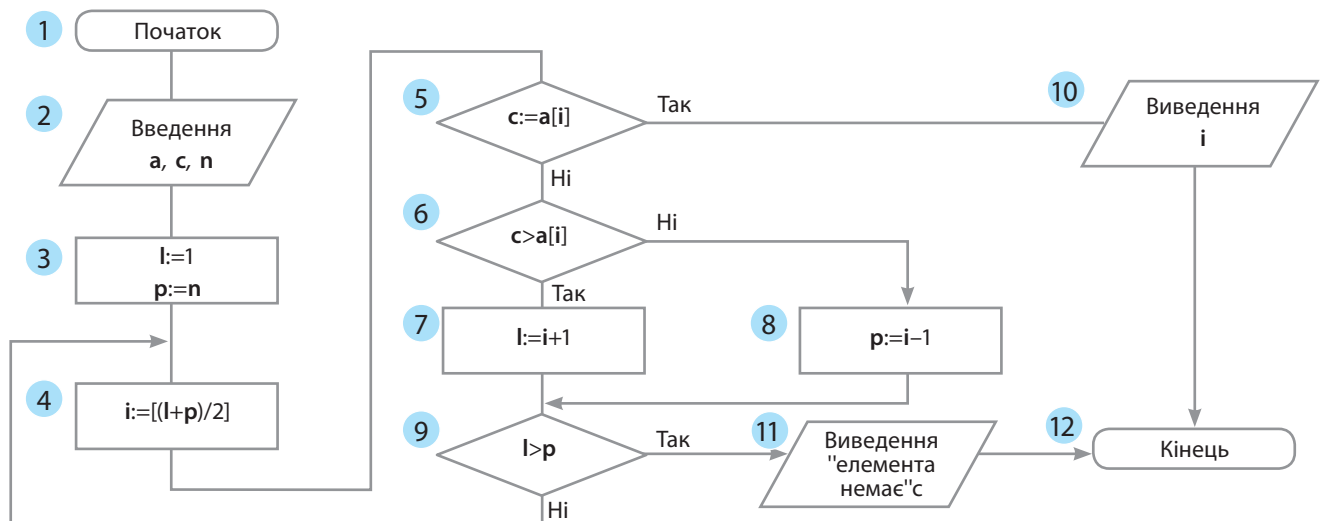


Рис. 4. Блок-схема алгоритму двійкового пошуку в масиві

Як бачимо, у блоці 3 змінній l надається початковий номер лівої межі масиву (він дорівнює 1), а змінній p — початковий номер правої межі (він дорівнює n). У блоці 4 обчислюється середній індекс розглядуваної ділянки масиву і його значення присвоюється змінній i .

У блоці 5 ключове значення s порівнюється зі значенням елемента, розташованого посередині поточної ділянки. Якщо ці значення збігаються, ключ знайдено, виводиться індекс i елемента масиву із ключовим значенням s (блок 10) і робота алгоритму завершується (блок 12). Але якщо згадані значення різні, керування передається блоку 6, де перевіряється, чи більше значення s за значення середнього елемента $a[i]$. Якщо $s > a[i]$, шуканий елемент розташований праворуч від $a[i]$ і значення правої межі не змінюється, а значенням лівої межі стає величина $i+1$ (блок 7). Якщо ж $s < a[i]$, то пошук ключового значення треба вести ліворуч від $a[i]$. У цьому випадку значення лівої межі не змінюється, а значенням правої межі стає величина $i-1$ (блок 8).

Алгоритм припиняє роботу у двох випадках: коли виконується умова $s = a[i]$ і коли значення лівої межі перевищує значення правої (блок 9). Настання другого випадку означає, що всі необхідні перевірки виконані, а ключового значення в заданому масиві не знайдено.



Метод двійкового пошуку значно швидший за метод лінійного пошуку. Проте, як уже згадувалося, двійковий пошук можна застосовувати лише до впорядкованих масивів.



Ада Августа Лавлейс розробила перші програми для аналітичної машини Беббіджа, заклавши тим самим теоретичні основи програмування. Саме вона вперше ввела поняття циклу операції. У наші дні Ада Лавлейс по праву називають першим програмістом у світі.



Приклад 3. На рис. 5 подано код програми, що реалізує алгоритм двійкового пошуку.

```

procedure TForm1.Button1Click(Sender: TObject);
const n=7; //константа типу змінної
var mas: array [1..n] of string= //масив рядкового типу
    ('алгоритм', 'байт', 'монітор',
    'ноутбук', 'принтер', 'сканер', 'файл');
i, l, p:integer; //змінні цілого типу
c:string; //змінна рядкового типу
vids:boolean=false; //змінна логічного типу
begin
    c:='ноутбук'; //термін, що відшукується
    l:=1; //початкова ліва межа
    p:=n; //початкова права межа
    while ((l<=p) and (not vids)) do //чи завершений пошук?
    begin //відкриваються операторні дужки
        i:=(l+p) div 2; //обчислення середнього індексу
        if mas[i]<c then l:=i+1 //пошук терміна i
        else if mas [i]>c then p:=p-1 //зміна меж масиву
        else vids:=true //ознака наявності терміна
        end; //закриваються операторні дужки
        if vids then Edit1.Text:=('термін на позиції-' + IntToStr(i))
        else Edit1.Text:=('термін відсутній'); //виведення результату
    end; end.

```

Рис. 5. Код реалізації двійкового пошуку даних у масиві

Результат виконання коду зображено на рис. 6.

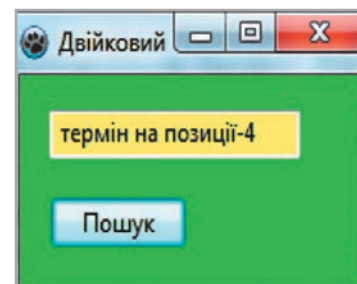


Рис. 6. Результат виконання коду двійкового пошуку даних

Приклад 5. Розглянемо покрокове виконання алгоритму пошуку мінімального елемента в масиві 37, 15, 41, 7, 21.

min:=37;

1-й цикл: 15<37? Так — min:=15

2-й цикл: 41<15? Ні

3-й цикл: 7<15? Так — min:=7

4-й цикл: 21<7? Ні.

Результат: min=7

• Пошук мінімального і максимального елементів.

Розглянемо сутність алгоритму пошуку мінімального елемента в масиві (пошук максимального елемента принципово не відрізняється від пошуку мінімального). Спочатку мінімальним вважається елемент, розташований на першій позиції. Його значення порівнюється зі значенням другого елемента. Якщо значення другого елемента менше від першого, то далі меншим вважається другий елемент. Потім значення меншого елемента порівнюється зі значенням третього елемента і так далі до останнього елемента. У результаті буде знайдено найменший елемент. Отже, алгоритм пошуку мінімального (максимального) елемента є циклічним алгоритмом (прикладі 5, 6).

Приклад 6. На рис. 7 наведено код пошуку мінімального і максимального елементів у одновимірному масиві.

```
procedure TForm1.Button1Click(Sender: TObject);
const n=12; //іменована константа
mas: array [1..n] of integer= //масив цілого типу
(41,20,8,60,11,31,5,17,50,16,33,15);
var i, min, max: integer; //змінні цілого типу
begin
max:=mas[1]; //початкове значення змінної max
min:=mas[1]; //початкове значення змінної min
for i:=2 to n do //цикл аналізу масиву
begin //відкриваються операторні дужки
if mas[i]>max then max:=mas[i]; //чи більше mas[i] максимального?
if mas[i]<min then min:=mas[i]; //чи менше mas[i] мінімального?
end; //закриваються операторні дужки
Edit1.Text:='Максимальним є число '+IntToStr(max); //виведення max
Edit2.Text:='Мінімальним є число '+IntToStr(min); //виведення min
end;
end.
```

Рис. 7. Код пошуку мінімального і максимального елементів масиву

Результат виконання коду зображено на рис. 8.

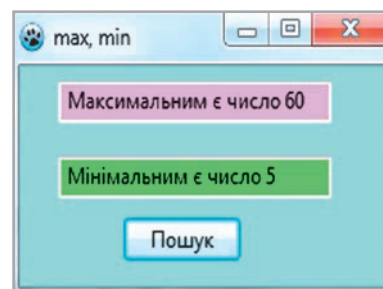


Рис. 8. Результат пошуку мінімального і максимального чисел

? Запитання для перевірки знань

- 1 Які існують найпростіші методи пошуку даних у масивах?
- 2 У чому полягає перевага двійкового пошуку над лінійним?
- 3 Поясніть сутність двійкового методу пошуку даних у масиві.
- 4 Накресліть блок-схеми: алгоритму лінійного пошуку даних; двійкового пошуку даних.

Завдання для самостійного виконання

- 1 Опишіть динаміку процесу пошуку числа 17 у масиві: 53, 21, 7, 40, 17, 4, 35.
- 2 У довільному порядку вводяться назви 5 обласних центрів України. Розробіть алгоритм і програму визначення: чи є в цьому переліку місто Суми.
- 3 Опишіть динаміку процесу пошуку мінімального і максимального чисел у масиві: 77, 65, 7, 21, 13, 40, 88, 51, 57.
- 4 Генерується 10 цілих випадкових чисел у діапазоні від 0 до 15. Розробіть алгоритм і програму пошуку мінімального і максимального з цих чисел. Масив випадкових чисел виведіть у поле об'єкта **ListBox1**.
- 5 Розробіть програму визначення третьої за довжиною річки України.

9.5. Упорядкування елементів масиву

Поясніть сутність лінійного і двійкового алгоритмів пошуку даних у масиві. У чому їх переваги й недоліки? Сформулюйте сутність алгоритму пошуку максимального елемента. За якими правилами, на вашу думку, можна впорядкувати за зростанням 4 цілі числа?



Упорядкувати масив означає розмістити його елементи за зростанням або спадання їхніх значень. Існує багато методів упорядкування одновимірного масиву. Далі розглянемо лише найпростіші методи, які широко використовують для розв'язування різноманітних задач. Такими методами є метод сортування вибором, метод сортування обміном і метод вставки. На основі цих методів побудовані інші, більш складні, методи впорядкування масивів.

- Упорядкування масиву методом вибору.

Нехай дано масив $a[1], a[2], \dots, a[n]$, який необхідно впорядкувати за зростанням його елементів. Сутність методу вибору така. Відшуковують елемент із максимальним значенням і міняють його місцем з останнім елементом масиву. Після цього останній елемент із подальшого розгляду виключають, а для перших $(n-1)$ елементів процедуру повторюють. Тобто аналізується масив $a[1], a[2], \dots, a[n-1]$, у якому також відшуковують максимальний елемент. Цей елемент міняють місцями з елементом $a[n-1]$. Подібні дії виконують у масиві $a[1], a[2], \dots, a[n-2]$, потім — у масиві $a[1], a[2], \dots, a[n-3]$ тощо. Отже, на кожному циклі алгоритму значення правої межі масиву зменшується на одиницю. Цей індекс будемо зберігати у змінній p . У табл. 1 наведено динаміку процесу впорядкування масиву 15, 12, 17, 8, 14, 6.

Загальне значення слова «сортування» — це розподіл елементів на групи за деякою ознакою, наприклад розподіл яблук по сортах за їх якістю, листів — за поштовими індексами, банок із фарбами — за кольором тощо. Проте у програмуванні під сортуванням розуміють упорядкування елементів (за деякою характеристикою), наприклад шиккування учнів за зростом на уроці фізкультури, розташування слів у словнику в алфавітному порядку, упорядкування точок площини за зростанням координати x тощо.

Таблиця 1. УПОРЯДКУВАННЯ МАСИВУ МЕТОДОМ ВИБОРУ

Цикли	Дії, що виконуються	Стан масиву					
1	Масив, у якому відшукується \max	15	12	17	8	14	6
	Масив після першої перестановки	15	13	6	8	14	17
2	Масив, у якому відшукується \max	15	13	6	8	14	
	Масив після другої перестановки	14	13	6	8	15	17
3	Масив, у якому відшукується \max	14	13	6	8		
	Масив після третьої перестановки	8	13	6	14	15	17
4	Масив, у якому відшукується \max	8	13	6			
	Масив після четвертої перестановки	8	6	13	14	15	17
5	Масив, у якому відшукується \max	8	6				
	Масив після п'ятої перестановки	6	8	13	14	15	17
	Упорядкований масив	6	8	13	14	15	17

Із таблиці видно, що перестановка елементів виконується після повного завершення поточного циклу перегляду масиву.

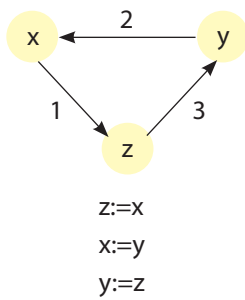


Рис. 1. Обмін значеннями між змінними x і y

Перестановка елементів — це обмін значеннями між двома змінними, між змінною з максимальним значенням і змінною, розташованою на останній справа позиції невпорядкованої частини масиву. Для виконання перестановки потрібна третя змінна, у якій тимчасово зберігатиметься значення однієї з двох змінних, значення яких необхідно поміняти місцями. Для перестановки в алгоритмі будемо використовувати змінну z (рис. 1). Перестановка значень елементів масиву деякою мірою нагадує процес переливання рідини з першої посудини в другу і з другої у першу. Для цього рідину переливають із першої посудини у третю, із другої у першу, а з третьої — у другу.

Блок-схему алгоритму подано на рис. 2.

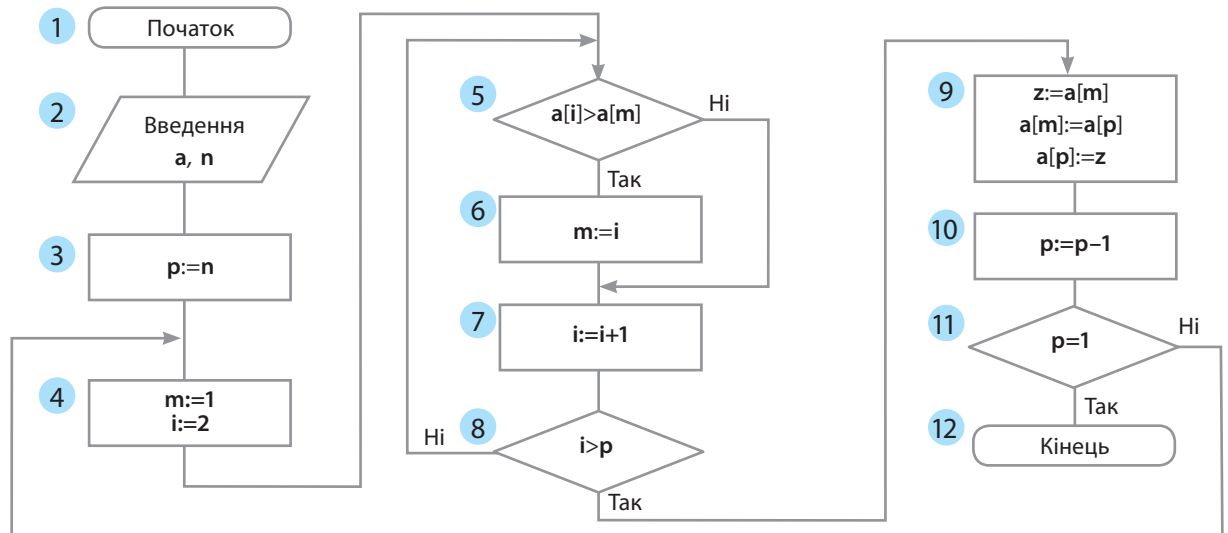


Рис. 2. Блок-схема алгоритму впорядкування масиву методом вибору

Початкове значення правої межі масиву (значення змінної p) дорівнює значенню змінної n . Змінна m містить індекс поточного елемента з максимальним значенням, а змінна i містить індекс елемента, що переглядається у даний момент. Оскільки перший елемент завжди спочатку вважається найбільшим, то у блоці 4 змінна m набуває значення 1, а змінна i — значення 2.

У блоці 5 перевіряється: чи більше значення поточного елемента $a[i]$ за значення поточного найбільшого елемента $a[m]$. Якщо $a[i]>a[m]$, то індекс поточного максимального елемента у блоці 6 набуває нового значення. Індекс поточного елемента масиву збільшується на одиницю у блоці 7. Якщо цей індекс менший або дорівнює правій межі (блок 8), то пошук максимального елемента триває далі. Якщо ж $i>p$, то це означає, що всю невпорядковану ділянку масиву переглянуто і в блоці 9 елемент із максимальним значенням міняють місцями з останнім. У блоці 10 значення правої межі ділянки, що переглядається, зменшується на одиницю. Ознакою завершення процесу виконання алгоритму є той факт, що індекс правої межі дорівнює одиниці (блок 11).

Метод сортування вибором є досить простим. Проте він найповільніший, оскільки в ньому не враховується те, що в заданому масиві деякі з елементів можуть бути вже впорядкованими. Навіть якщо масив буде повністю впорядкованим, кількість операцій в алгоритмі не зменшиться.

Приклад 1. Програмний код реалізації алгоритму впорядкування масиву методом вибору подано на рис. 3. У цьому коді масив

формується із 10 випадкових чисел у діапазоні від 0 до 30. На рис. 4 зображено один із можливих результатів виконання програмного коду.

```

procedure TForm1.Button1Click(Sender: TObject);
const n=10; //іменована константа
var mas: array [1..n] of integer; //масив цілого типу
p, i, m, z: integer; //змінні цілого типу
begin
  Randomize; //при кожному запуску випадкові числа різні
  for i:=1 to n do //цикл створення масиву
  begin //початок операторних дужок
    mas[i]:=Random (30); //формування масиву
    ListBox1.Items.Add(IntToStr(mas[i])) //виведення масиву
  end; //кінець операторних дужок
  p:=n; //початкове значення правої межі
  while p>1 do //цикл упорядкування масиву
  begin //початок операторних дужок
    m:=1; //початкове значення лівої межі
    for i:=2 to p do //цикл пошуку максимального елемента
    if mas[i]>mas[m] then m:=i; //який елемент більше?
    z:=mas[m]; mas[m]:=mas[p]; mas[p]:=z; //переміщення макс. елемента
    p:=p-1 end; //зменшення правої межі масиву
  for i:=1 to n do //цикл виведення масиву
  ListBox2.Items.Add(IntToStr(mas[i])); //виведення упорядкованого масиву
  end; end.

```

Рис. 3. Програмний код упорядкування масиву методом вибору

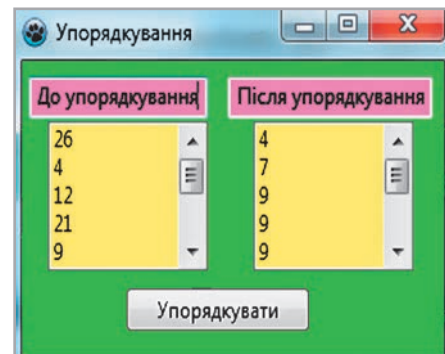


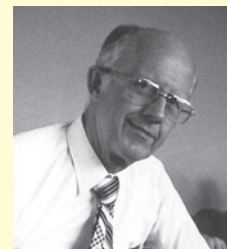
Рис. 4. Результат упорядкування масиву методом вибору

- **Упорядкування масиву методом обміну.**

Сутність упорядкування масиву за зростанням значень його елементів методом обміну така. Масив $a[1], a[2], \dots, a[n]$ переглядається зліва направо. Спочатку порівнюються елементи $a[1]$ і $a[2]$, потім — $a[2]$ і $a[3]$, $a[3]$ і $a[4]$, $a[4]$ і $a[5]$ і так далі до елементів $a[n-1]$ і $a[n]$. Кожного разу, коли попередній елемент більший за наступний, значення елементів міняються місцями. Зрозуміло, що після повного завершення першого перегляду всього масиву на останній позиції буде міститися елемент із максимальним значенням. Після цього елемент $a[n]$ з подальшого розгляду виключається, переглядається масив $a[1], a[2], \dots, a[n-1]$ і з його елементами виконуються аналогічні дії, в результаті чого на позицію $n-1$ буде переміщено другий за величиною елемент.

Процедура повторюється для масиву $a[1], a[2], \dots, a[n-2]$, потім — для масиву $a[1], a[2], \dots, a[n-3]$ і т. д. З описаного випливає, що сортування масиву завершується після повного впорядкування всіх елементів масиву. Ознакою того, що масив упорядкований, є те, що після завершення його перегляду жодної перестановки елементів не було.

Для свідомого розуміння сутності сортування масиву методом прямого вибору розглянемо динаміку впорядкування на конкретному прикладі.



У 1959 році Дональд Шелл запропонував удосконалення алгоритму прямих вставок. Його ідея — порівнювати елементи, розташовані на певній відстані один від одного, і покроково зменшувати цю відстань.



Приклад 2. Дано масив 41, 52, 58, 70, 66, який потрібно впорядкувати за зростанням значень елементів. У цьому масиві 3 перші елементи впорядковано. Для того щоб масив

був повністю впорядкований, потрібно поміняти місцями четвертий і п'ятий елементи. Для цього виконуються дії, наведені у табл. 2.

Таблиця 2. ПРОЦЕС УПОРЯДКУВАННЯ МАСИВУ

Зовнішній цикл	Внутрішній цикл	Елементи, що порівнюються	Значення елементів масиву					у (початкове значення true)
1	1	41 > 52?	41	52	58	70	66	true
	2	52 > 58?	41	52	58	70	66	true
	3	58 > 70?	41	52	58	70	66	true
	4	70 > 66?	41	52	58	70	66	false
			41	52	58	66	70	
Коментар: під час першого виконання зовнішнього циклу відбулась перестановка елементів масиву, після його завершення змінна у набула значення false . Тому починається другий зовнішній цикл, а змінна у набуває значення true								
2	1	41 > 52?	41	52	58	66	70	true
	2	52 > 58?	41	52	58	66	70	true
	3	58 > 66?	41	52	58	66	70	true
Упорядкований масив			41	52	58	66	70	
Коментар: після другого виконання зовнішнього циклу жодної перестановки не відбулося, тобто змінна у має значення true . Масив упорядкований, робота алгоритму завершена								

Метод сортування обміном у середньому діє швидше за метод сортування вибором, оскільки в ньому враховується можливість упорядкування масиву до того, як його буде переглянуто максимальну кількість разів.

Із табл. 2 видно, що після першого зовнішнього циклу масив є впорядкованим. Але змінна **у** має значення **false**, тому виконується другий зовнішній цикл. Після його завершення жодної перестановки елементів не було, тому процес упорядкування завершується. Звернемо особливу увагу на те, що на початку будь-якого зовнішнього циклу ця змінна набуває значення **true**. Після того як у внутрішньому циклі відбувається хоча б одна перестановка елементів, змінна **у** набуває значення **false** і до кінця цього циклу не змінюється.

На основі аналізу процесу впорядкування наведеного вище масиву розробимо *алгоритм упорядкування методом обміну*. В алгоритмі використано такі позначення: *p* — індекс правої межі поточної ділянки масиву; *y* — ознака наявності перестановки: на початку кожного зовнішнього циклу вона набуває значення **true**. Якщо після завершення зовнішнього циклу змінна **у** має значення **true**, це означає, що перестановок під час останнього перегляду не було, якщо **y=false**, то відбулася принаймні одна перестановка; *i* — індекс поточного елемента масиву; *z* — змінна, призначена для тимчасового зберігання значення елемента масиву під час перестановки елементів.

Крок 1	$p:=n-1$	початкове значення правої межі масиву	Крок 7	$i:=i+1$	підготовка до вибору чергового елемента масиву
Крок 2	$y:=true$	ознака того, що на початку зовнішнього циклу перестановок елементів не було	Крок 8	Якщо $i \leq p$, то крок 4, інакше крок 9	перевірка умови завершення внутрішнього циклу
Крок 3	$i:=1$	перший індекс масиву	Крок 9	$p:=p-1$	зменшення правої межі масиву
Крок 4	Якщо $mas[i]>mas[i+1]$, то крок 5, інакше — крок 7	порівняння сусідніх елементів	Крок 10	Якщо $y=false$, то крок 2, інакше — крок 11	перевірка умови зовнішнього циклу (чи впорядкований масив?)
Крок 5	$z:=mas[i];$ $mas[i]:=mas[i+1];$ $mas[i+1]:=z$	перестановка елементів	Крок 11	Кінець	
Крок 6	$y:=false$	ознака того, що відбулася перестановка елементів			



Приклад 3. Програму, що реалізує алгоритм сортування методом обміну, подано на рис. 5. У цьому коді масив формується з десяти

випадкових чисел з діапазону від 0 до 50. На рис. 6 зображено один із можливих результатів виконання програмного коду.

```

procedure TForm1.Button1Click(Sender: TObject);
const n=10; //іменована константа
var y: boolean; //змінна логічного типу
i, z, p: integer; //змінні цілого типу
mas: array [1..n] of integer; //масив цілого типу
begin
Randomize; //при кожному запуску випадкові числа різні
for i:=1 to n do //цикл формування масиву випадковими числами
begin //відкриваються операторні дужки
mas[i]:=Random(50); //створюється масив випадкових чисел
Memo1.Lines.Add(IntToStr(mas[i])) //виведення масиву
end; //закриваються операторні дужки
p:=n-1; //початкове значення правої межі масиву
repeat //зовнішній цикл упорядкування масиву
y:=true; //ознака: у масиві перестановок не було
for i:=1 to p do //початок внутрішнього циклу
begin //відкриваються операторні дужки
if mas[i]>mas[i+1] then //який елемент більше?
begin //відкриваються операторні дужки
z:=mas[i]; mas[i]:=mas[i+1]; mas[i+1]:=z; //переміщення елементів
y:=false //ознака: у внутрішньому циклі було переміщення
end; //закриваються операторні дужки
end; //кінець внутрішнього циклу
p:=p-1; //зменшення правої межі масиву
until y; //перевірка умови завершення зовнішнього циклу
for i:=1 to n do //цикл виведення упорядкованого масиву
Memo2.Lines.Add(IntToStr(mas[i])); //виведення елементів масиву
end; end.

```

Рис. 5. Програмний код упорядкування масиву методом обміну

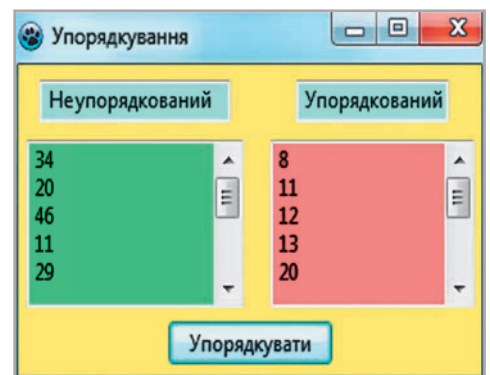


Рис. 6. Результат упорядкування масиву методом обміну

Приклад 4. Динаміку впорядкування масиву 33 14 70 17 наведено в таблиці:

Цикли	Стан масиву			
1	33	14	70	17
	14	33	70	17
2	14	33	70	17
	14	33	70	17
3	14	33	70	17
	14	17	33	70

• Упорядкування масиву методом вставки.

Метод сортування вставкою також є простим у реалізації. За швидкістю впорядкування він особливо ефективний для масивів невеликого розміру і для частково впорядкованих масивів.

Сутність цього методу така. Спочатку впорядковуються два перші елементи масиву. Потім на кожному наступному кроці береться наступний за порядком елемент і вставляється в уже впорядковану частину так, щоб ліворуч від нього всі елементи були не більшими, а сусідній правий — не меншим від нього. Якщо ця умова не виконується, береться наступний за порядком елемент і виконуються аналогічні дії (приклад 4).

Зазначимо, що в цьому параграфі розглянуто лише найпростіші методи впорядкування даних. Існують й інші, більш досконалі та складні методи, спрямовані насамперед на підвищення швидкості сортування.



Запитання для перевірки знань

- 1 Які переваги й недоліки мають методи впорядкування масиву вибором і обміном?
- 2 Як здійснюється перестановка двох елементів масиву?
- 3 Який вигляд матиме масив 21, 40, 5, 13, 18 після завершення першого зовнішнього циклу впорядкування масиву методом вибору?
- 4 Розробіть програму впорядкування назв районів свого міста (області) за алфавітом.
- 5 Поясніть сутність алгоритму впорядкування масиву методом вибору.
- 6 Поясніть сутність алгоритму впорядкування масиву методом обміну.
- 7 Поясніть сутність алгоритму впорядкування масиву методом вставки.



Завдання для самостійного виконання

- 1 Відомі прізвища і зріст 8 учнів вашого класу. Розробіть алгоритм і програму виведення на екран прізвищ учнів у порядку зменшення їх зросту. Використайте метод вибору.
- 2 Знайдіть в Інтернеті назви 7 найбільших за площею областей України. Розробіть алгоритм і програму впорядкування назв областей у порядку зменшення їх площ. Використайте метод вибору.
- 3 Опишіть у вигляді таблиці впорядкування масиву чисел 33, 60, 7, 12, 80, 41 у порядку їх зменшення. Використайте метод вибору; метод вставки.

Практична робота № 17

Тема. Розробка алгоритмів роботи з одновимірними масивами та їх реалізація у вигляді програм.

Завдання: розробити алгоритм і програму для реалізації одного із завдань, наведених у таблиці.

Обладнання: комп'ютер із встановленою програмою — середовищем програмування.

Таблиця. ЗАВДАННЯ ДЛЯ ВИКОНАННЯ ПРАКТИЧНОЇ РОБОТИ № 17

Варіант	Завдання	Початкові дані
1	N учнів класу виконують стрибки у довжину. Розробіть алгоритм і програму для визначення того, скільки учнів стрибнули далі за встановлений норматив c.	N=6 4.7; 3.8; 3.3; 3.6; 3.1; 3.9; c=3.5
2	Відомі результати змагань чемпіонату світу з жіночого біатлону. Розробіть алгоритм і програму визначення того, чи входить команда України у 6 кращих команд і, якщо є, яке місце вона посідає.	Норвегія, Німеччина, Україна, Росія, Франція, Швеція

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Виберіть один із варіантів завдань за вказівкою вчителя або самостійно. (У разі успішного виконання одного варіанта, виконайте інший.)
 2. Проаналізуйте умову задачі, побудуйте інформаційну модель.
 3. Розробіть алгоритм і реалізуйте його в середовищі програмування.
 4. Перевірте роботу програми із різними початковими даними.
 5. Виконайте програму із зазначеними в таблиці початковими даними. Проаналізуйте отриманий результат.
- **Зробіть висновок:** які компоненти доцільно використовувати для обчислення кількості заданих елементів у масиві.

Практична робота № 18

Тема. Класичні алгоритми для роботи з масивами та їх реалізація у вигляді програм.

Завдання: розробити алгоритм і програму для реалізації одного із завдань, наведених у таблиці.

Обладнання: комп'ютер із встановленою програмою — середовищем програмування.

Таблиця. ЗАВДАННЯ ДЛЯ ВИКОНАННЯ ПРАКТИЧНОЇ РОБОТИ № 18

Варіант	Завдання	Початкові дані
1	Дано масив з N цілих додатних чисел. Розробіть алгоритм і програму визначення, наскільки максимальне число більше за мінімальне.	N=7; 105, 34, 160, 31, 28, 60, 70
2	Знайдіть в Інтернеті прізвища космонавтів України. Розробіть алгоритм і програму впорядкування їхніх прізвищ в алфавітному порядку.	

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Виберіть один із варіантів завдань за вказівкою вчителя або самостійно. (У разі успішного виконання одного варіанта, виконайте інший.)
 2. Проаналізуйте умову задачі, побудуйте інформаційну модель.
 3. Розробіть алгоритм і реалізуйте його в середовищі програмування.
 4. Перевірте роботу програми із різними початковими даними, дібраними самостійно.
 5. Виконайте програму із зазначеними в таблиці початковими даними. Проаналізуйте отриманий результат.
- **Зробіть висновок:** які методи сортування і пошуку доцільно використовувати в цій роботі.

Розділ 10. СИМВОЛЬНІ ТА РЯДКОВІ ВЕЛИЧИНИ

10.1. Загальні відомості про символні та рядкові величини



Пригадайте, за допомогою яких кодів подаються символи в комп'ютері. Чи можуть символи бути значеннями елементів масиву? Що, на ваш погляд, називають рядком у мовах програмування?

Робота із символами кодової таблиці Unicode у середовищі Lazarus реалізована не повною мірою. Приміром, функція визначення кількості національних символів у рядку насправді визначає кількість байтів, які займає рядок. Тобто в процесі визначення кількості символів у рядку результат буде збільшено вдвічі. Тому далі в процесі опрацювання рядків і символів будемо використовувати переважно латинські літери.

Приклад 1. У результаті виконання фрагмента коду:

```
var predm: string;
begin
    predm:='Інфор'+'матика';
    ...
отримаємо слово: Інформатика.
```

Приклад 2. У результаті виконання фрагмента коду:

```
var a_1: string; b_1: char;
begin
    a_1:='процесор';
    b_1:=a_1[4];
    ...
змінна b_1 отримає символ ц.
```

Символ — це цифра, літера або інший знак клавіатури. Кожний символ має свій унікальний двійковий код. Нині для кодування символів найчастіше застосовують кодові таблиці ASCII і Unicode. Латинські літери розташовані у першій половині таблиці ASCII. Їхні коди збігаються з кодами цих літер у таблиці Unicode і є обов'язковими для всіх країн світу. Для кодування національних символів використовується друга половина таблиці ASCII (коди від 128 до 256), а також таблиця Unicode, у якій кожен символ кодується двома байтами.

Для роботи із символами у середовищі Lazarus використовується тип даних **char**. Символьні змінні оголошуються в розділі оголошення змінних, наприклад `var z, a_1: char`.



Рядок — це набір символів. Кількість символів у рядку називають його довжиною. Основним типом для роботи з рядками є тип даних **string**. Порядок оголошення рядкових змінних принципово не відрізняється від порядку оголошення інших типів даних, наприклад `var bas, b_1: string;`

Рядкові й символні константи беруться в одинарні лапки, наприклад 'а', 'Символ' '21'. Значення символним і рядковим змінним присвоюються за допомогою оператора присвоювання, наприклад `a_1:='к'; b_1:='Інформатика'`. Довжина символного рядка може бути обмежена. У цьому випадку під час оголошення змінної у квадратних дужках вказується максимальна кількість символів, які може містити ця змінна, наприклад `var str1: string [25];` (змінна `str1` може містити до 25 символів). Якщо довжина змінних типу `string` під час їх оголошення не вказана, максимальна кількість символів може сягати 255.



Однією з основних операцій над рядковими даними є їх об'єднання. Ця операція позначається символом «+» (приклад 1).

Рядок фактично є масивом символів, з якого можна отримати будь-який символ, вказавши його номер у квадратних дужках (приклад 2).

Приклад 3. Два учні грають у таку гру. Один учень каже: «Назви четверту літеру у назві найулюбленішої в усьому світі спортивної гри». Другий учень називає відповідну

літеру. У будь-якому випадку повідомляється про правильність відповіді, а також загадане слово — футбол. На рис. 1 наведено код моделювання цієї гри у консольному режимі.

```
const nazva='футбол'; //іменована константа
var x:char; //змінна символічного типу
y:string; //змінна рядкового типу
begin
writeln ('увести 4-ту літеру у назві спортивної гри');
readln (x); //уведення літери
//чи збігається введена літера з 4-ю літерою у слові
if x=nazva[4] then y:='правильно, це футбол'
else y:='неправильно, це футбол';
writeln (y); //виведення результату
readln //призупинення виконання програми
end.
```

Рис. 1. Програма моделювання гри учнів

Приклад 4. Розробити програму, за допомогою якої з літер слова facilities створюється слово file. Для цього необхідно зі слова facilities вибрати першу, четверту, п'яту і дев'яту літери

і виконати над ними операцію об'єднання. Код реалізації цього завдання наведено на рис. 2.

Результат виконання програмного коду подано на рис. 3.

```
procedure TForm1.Button1Click(Sender: TObject);
var a, b:string; //змінні рядкового типу
begin
a:='facilities'; //присвоювання значення змінній
Edit1.Text:='початкове слово '+a; //виведення початкового слова
b:=a[1]+a[4]+a[5]+a[9]; //створення нового слова
Edit2.Text:='отримане слово '+b //виведення нового слова
end;
end.
```

Рис. 2. Програмний код створення слова із літер іншого слова

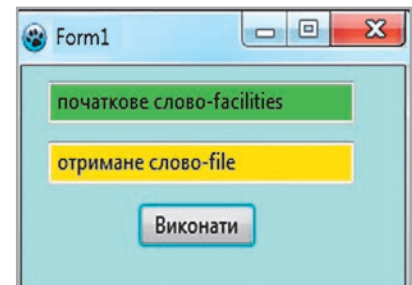


Рис. 3. Результат виконання коду створення нового слова

? Запитання для перевірки знань

- 1 Яку назву має тип даних для роботи із символами?
- 2 Яку назву має тип даних для роботи з рядками?
- 3 Наведіть приклад оголошення рядкових змінних.
- 4 Як присвоюються змінним символічні й рядкові константи?

📁 Завдання для самостійного виконання

- 1 Розробіть програму, за допомогою якої слова **Динамо**, **Шахтар** і **2:2** об'єднуються в одну конструкцію **Динамо — Шахтар 2:2**.
- 2 Розробіть програму для виведення на екран другої і восьмої літер у слові **calculate**.
- 3 Розробіть програму, за допомогою якої з літер слова **vertorization** створюється слово **var**.

10.2. Функції опрацювання рядків і символів



Поясніть правило запису рядкових і символних констант. Яку операцію над рядками ви вже знаєте? Які ще операції над рядками вам можуть знадобитися?

Середовище Lazarus має спеціальні функції для роботи з рядками і символами, основні з яких наведено у таблиці.

Таблиця. ОСНОВНІ ФУНКЦІЇ ОПРАЦЮВАННЯ РЯДКІВ І СИМВОЛІВ

Функція	Призначення	Тип аргументів	Тип результату
length (S)	Визначення довжини (кількості символів) рядка S	S — рядковий	Цілий
concat (S1, S2,...)	Об'єднання рядків S1, S2, ...	S1, S2, ... — рядковий	Рядковий
copy (S, n, m)	Копіювання m символів рядка S, починаючи з позиції n	S — рядковий, n — цілий, m — цілий	Рядковий
delete (S, n, m)	Вилучення m символів із рядка S, починаючи з позиції n	S — рядковий, n — цілий, m — цілий	Рядковий
insert (S1, S2, n)	Вставлення рядка S1 у рядок S2, починаючи з позиції n	S1 — рядковий, S2 — рядковий, n — цілий	Рядковий
pos (S1, S2)	Визначення номера позиції, з якої починається входження рядка S2 у рядок S1	S1 — рядковий, S2 — рядковий	Цілий
chr (x)	Повернення символу, код якого дорівнює x	x — цілий	Символьний
ord (c)	Повернення коду символу c	c — символний	Цілий

Рядкові змінні, константи і вирази можна порівнювати за допомогою операцій =, <, <=, >, >=, <>. У процесі порівняння рядків послідовно здійснюється порівнювання символів зліва направо. Рядки вважаються рівними, якщо вони мають однакову довжину й усі символи в них однакові. Меншим вважається той рядок, у якого менший код першого незбіжного символу. Наприклад, істинними є співвідношення: 'масив'<'маска', 'рядками'<'рядок'. Якщо один рядок повністю збігається з початком іншого, то більшим буде довший рядок. Наприклад, істинними є співвідношення: 'стовп'<'стовпець', 'багаторядковий'>'багато'.

Нижче розглянемо приклади використання функцій опрацювання даних рядкового і символного типів.



Нині більшість програм створюється командами. Але тільки один програміст відповідає за її архітектуру, інші — лише кодери.

- Функція **length (S)**.

Ця функція визначає довжину рядка S.

Приклад 1. Два слова вводять в поля об'єктів Edit1 і Edit2. Властивість Text об'єкта Edit1 набуває значення *amplification*, а властивість Text об'єкта Edit2 — значення *access*. Потрібно визначити, наскільки перше слово довше за друге. Результат вивести у поле об'єкта Edit3.

```
procedure TForm1.Button1Click(Sender: TObject);
var a1, a2:integer;           //змінні цілого типу
begin
  a1:=length (Edit1.Text);   //довжина рядка у полі Edit1
  a2:=length (Edit2.Text);   //довжина рядка у полі Edit2
  Edit3.Text:='Перше слово довше на '+ //уведення різниці
  IntToStr(a1-a2)+'символів '      //кількості символів
end;
```

Рис. 1. Програмний код визначення різниці довжин слів

Код реалізації цього завдання зображено на рис. 1.

Результат виконання програмного коду подано на рис. 2.

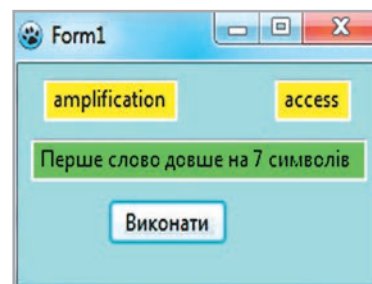


Рис. 2. Результат визначення різниці довжин слів

- Функція **concat (S1, S2, ...)**.

Ця функція об'єднує декілька рядків в один.

Приклад 2. Розробити код, за допомогою якого зі слів події, Код, OnCreate, опрацювання створюється речення: Код опрацювання події OnCreate.

Будемо вважати, що кожне слово окремо зберігається в полях об'єктів: слово події — у полі об'єкта Label1; слово Код — у полі

об'єкта Label2; слово OnCreate — у полі об'єкта Label3; слово опрацювання — у полі об'єкта Label4. Код створення речення подано на рис. 3.

Результат виконання коду зображено на рис. 4.

```
procedure TForm1.Button1Click(Sender: TObject);
var a1, a2, a3, a4:string;   //змінні рядкового типу
begin
  a1:=Label1.Caption+' ';   //слово поля об'єкта Label1
  a2:=Label2.Caption+' ';   //слово поля об'єкта Label2
  a3:=Label3.Caption;       //слово поля об'єкта Label3
  a4:=Label4.Caption+' ';   //слово поля об'єкта Label4
  Label5.Caption:=a2+a4+a1+a3; //виведення речення
end;
```

Рис. 3. Код створення речення з окремих слів

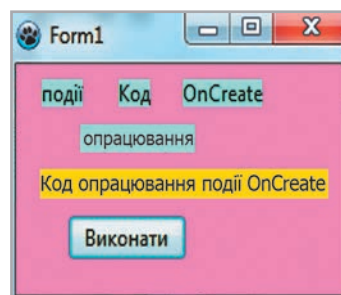


Рис. 4. Результат виконання коду створення речення

- **Функція copy (S, n, m).** Ця функція призначена для копіювання m символів із рядка S, починаючи із символу n.

Приклад 3. У поле об'єкта Label1 уводиться рядок. Потрібно скопіювати з нього m символів, починаючи із символу n, і присвоїти їх властивості Text об'єкта Edit1. На рис. 5 наведено програмний код реалізації цього завдання. До початку розробки коду властивості Caption

```
procedure TForm1.Button1Click(Sender: TObject);
var a1, a2: string; //змінні рядкового типу
begin
    //змінна a1 набуває значення властивості Caption
    //об'єкта Label1
    a1:=Label1.Caption;
    //змінній a2 присвоюється 6 символів змінної a1,
    //починаючи з 13-ої позиції
    a2:=copy (a1, 13, 6);
    Edit1.Text:=a2; //виведення результату
end;
end.
```

Рис. 5. Програмний код копіювання символів із рядка

об'єкта Label1 присвоюється речення 'application binary interface', у якому виділяються 6 символів, починаючи з позиції 13.

Результат виконання програмного коду зображено на рис. 6.

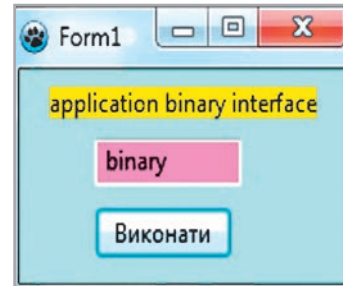


Рис. 6. Результат копіювання символів із рядка

- **Функція delete (S, n, m).** За допомогою цієї функції з рядка S вилучається m символів, починаючи з позиції n. Після вилучення результат залишається у рядку S.

Приклад 4. Вилучити слово aTop з оголошення type TAlign=(alNone, alTop, alBottom, alLeft). Вміст цього оголошення будемо зберігати у полі об'єкта Label1, а результат після

вилучення слова — у полі об'єкта Label2. Код реалізації цього завдання зображено на рис. 7.

На рис. 8 подано результат виконання цього програмного коду.

```
procedure TForm1.Button1Click(Sender: TObject);
var a1: string; //змінна рядкового типу
begin
    //змінна a1 набуває значення властивості Caption
    a1:=Label1.Caption;
    //з рядка a1 вилучається слово aTop
    Delete (a1, 24, 6);
    Label2.Caption:=a1; //виведення результату
end;
end.
```

Рис. 7. Програмний код вилучення слова з рядка

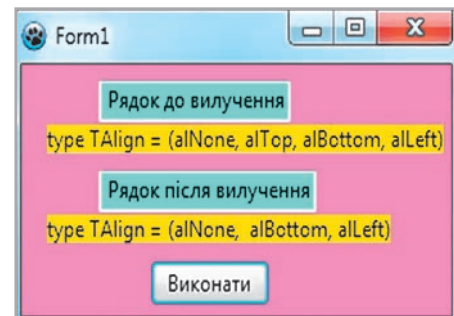


Рис. 8. Результат, отриманий після вилучення слова з рядка

- **Функція insert (S1, S2, n).** Ця функція забезпечує вставлення рядка S1 у рядок S2, починаючи з позиції n.

Приклад 5. У текст Actions speak louder вставити слова than words так, щоб було створено речення: Actions speak louder than words.

Для розв'язування завдання використаємо 6 компонентів TEdit і компонент TButton. До розробки програмного коду надамо властивості Text об'єкта Edit1 значення Текст до вставлення, властивості Text об'єкта Edit2 — значення Actions speak louder, властивості Text об'єкта

```
procedure TForm1.Button1Click(Sender: TObject);
var a1, a2:string; //змінні рядкового типу
begin
    //змінна a1 набуває значення властивості Text об'єкта Edit2
    a1:=Edit2.Text;
    //змінна a2 набуває значення властивості Text об'єкта Edit4
    a2:=Edit4.Text;
    insert (a2, a1, 15); //вставлення у рядок a1 рядка a2
    Edit6.Text:=a1; //виведення нового рядка
end;
end.
```

Рис. 9. Програмний код вставлення слів у речення

Edit3 — значення Текст, що вставляється, властивості Text об'єкта Edit4 — значення than words, властивості Text об'єкта Edit5 — значення Текст після вставлення. Програмний код реалізації сформульованого завдання подано на рис. 9.

Результат виконання програмного коду зображено на рис. 10.

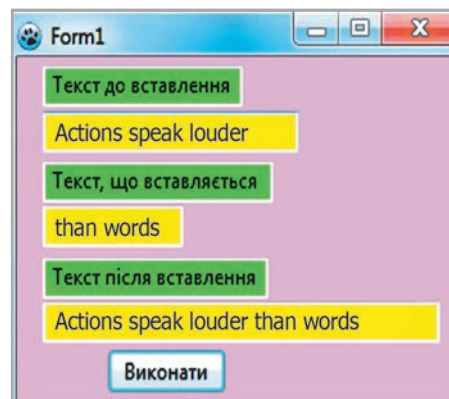


Рис. 10. Результат вставлення слів у речення

- **Функція pos (S1, S2).** Ця функція визначає номер позиції, починаючи з якої текст S2 входить до тексту S1.

Приклад 6. Розробити код, за допомогою якого у реченні The Dnipro's plunging shore відшукується номер позиції, починаючи з якої міститься слово plunging.

Для розробки програмного коду використаємо 5 компонентів TLabel і компонент TButton. На початку розробки програмного коду надамо властивості Caption об'єкта Label1 значення Текст, у якому відшукується інший текст,

```
procedure TForm1.Button1Click(Sender: TObject);
var a1, a2: string; n: integer; //оголошення змінних
begin
    //змінна a1 набуває значення властивості Caption об'єкта Label2
    a1:=Label2.Caption;
    //змінна a2 набуває значення властивості Caption об'єкта Label4
    a2:=Label4.Caption;
    n:=pos (a2, a1); //виведення номера позиції
    Label5.Caption:='Починається з позиції '
    +intToStr(n);
end;
end.
```

Рис. 11. Програмний код визначення позиції, з якої починається текст

і властивості Caption об'єкта Label3 — значення Текст, що відшукується. Властивості Caption об'єкта Label2 надамо значення The Dnipro's plunging shore, а властивості Caption об'єкта Label4 — значення plunging. Програмний код реалізації поставленого завдання зображено на рис. 11.

Результат виконання програмного коду подано на рис. 12.

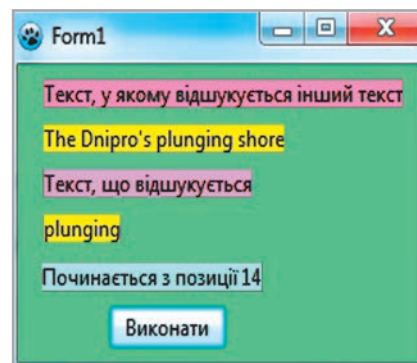


Рис. 12. Результат пошуку позиції, з якої починається заданий текст

- Функція **chr (x)**. Ця функція повертає символ, код якого дорівнює x.
- Функція **ord (c)**. Ця функція повертає код символу c.



Приклад 7. Розробити програмний код для визначення коду будь-якого латинського символу і визначення символу за відомим його кодом.

Розмістимо на формі 2 компоненти TEdit і компонент TButton. Властивості Text об'єкта Edit1 у програмі присвоїмо значення коду

```
procedure TForm1.Button1Click(Sender: TObject);
var x:char; //змінна символного типу
begin
  x:='a'; //змінній x присвоєно символ a
  //виведення коду символу
  Edit1.Text:='Цей символ має код '+IntToStr(ord(x));
  //виведення символу відповідного коду
  Edit2.Text:='Такий код має символ '+chr(97)
end;
end.
```

Рис. 13. Програмний код визначення символу й коду символу

англійського символу a. Властивості Text компонента Edit2 присвоїмо символ, код якого дорівнює 97. Для об'єкта Button1 встановимо подію OnClick. Програмний код реалізації цього завдання подано на рис. 13.

Результат виконання цього коду зображено на рис. 14.

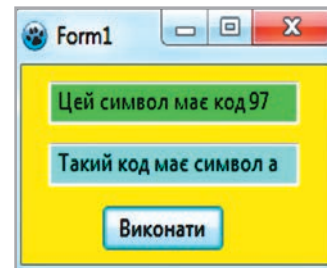


Рис. 14. Виведення коду символу й символу за його кодом

? Запитання для перевірки знань

- 1 Для чого призначена функція **chr (x)**?
- 2 Поясніть призначення аргументів функції **copy (S, n, m)**?
- 3 Яке призначення функції **concat (S1, S2,...)**?
- 4 Які дії виконує функція **insert (S1, S2, n)**?
- 5 Поясніть порядок виконання функції **delete (S, n, m)**.
- 6 Наведіть приклад використання функції **ord (c)**.
- 7 Наведіть приклад використання функції **copy (S, n, m)**.

💻 Завдання для самостійного виконання

- 1 Розробіть код обчислення кількості символів у рядку, не враховуючи символ пробілу. Перевірте роботу програми на прикладі рядка: **memory expansion card**.
- 2 Розробіть код, за допомогою якого після третього символу рядка вставляються два інші символи. Вставте, наприклад, символи **or** після третього символу рядка **infmation system**.
- 3 Розробіть алгоритм і код, за допомогою якого з рядка вилучаються всі слова, у яких є задана літера. Перевірте роботу програми на прикладі вилучення з рядка **multiple document interface** слів, у яких є літера **o**.
- 4 Розробіть код, за допомогою якого із окремо введених у довільному порядку слів створюється речення: **Алгоритм упорядкування рядків**.

10.3. Класичні алгоритми опрацювання рядків

Назвіть основні функції опрацювання рядків і символів. Як ви вважаєте, чи достатньо цих функцій для всебічного їх опрацювання? Якщо ні, то які ще операції необхідно виконувати над рядками? Наведіть конкретні приклади.



Над рядками можуть виконуватися різноманітні операції. Як правило, для цього застосовуються описані вище функції, а інколи й інші засоби.

До розповсюджених алгоритмів опрацювання символів можна віднести пошук заданого символу в рядку, пошук заданого підрядка, упорядкування елементів рядкової величини та ін. Слід також враховувати, що в реальних алгоритмах опрацювання рядків зазвичай використовується не одна окрема функція, а певна комбінація цих функцій.

- **Пошук заданого символу в рядку.**

Найпростіше здійснити пошук символу в рядку можна за допомогою функції `pos (S1, S2)`. Якщо після виконання цієї функції її значення дорівнює нулю, це означає, що символ у рядку відсутній. Приклад використання функції `pos (S1, S2)` для пошуку символу в рядку вже розглядався в попередньому параграфі. Можливий і такий варіант пошуку.

Рядок — це фактично масив символів. Кожний символ можна розглядати як елемент цього масиву. Це означає, що пошук заданого символу в рядку принципово не відрізняється, наприклад, від пошуку заданого числа в числовому масиві.

Нижче наведено в словесній формі *алгоритм пошуку символу в рядку* (де n — довжина рядка).



У світі існує близько 8500 мов програмування, але кількість мов, на яких пишуть програми професійні програмісти, не перевищує 10.

Крок 1	$i:=1$ (змінна i — лічильник символів у рядку)	Крок 5	Такий символ є, він розташований у позиції i
Крок 2	Якщо i -й символ дорівнює заданому символу, то крок 5, інакше — крок 3	Крок 6	Перейти до кроку 8
Крок 3	$i:=i+1$	Крок 7	Такого символу в рядку немає
Крок 4	Якщо $i \leq n$, то крок 2, інакше — крок 7	Крок 8	Кінець



Приклад 1. Прослідкуйте динаміку виконання алгоритму на прикладі пошуку в рядку East or West — home is best символу w. На рис. 1 зображено програмний код пошуку символу в рядку. У властивості Caption об'єкта Label1 зберігається текст: East or West — home is best. Властивість Caption об'єкта Label2 використовується

для виведення результату. Зверніть увагу на те, що пошук символу в тексті завершується одразу після знайдення потрібного символу (у даному випадку символу w) або після аналізу всіх символів тексту.

На рис. 2 подано результат пошуку символу w у наведеному вище рядку.

```

procedure TForm1.Button1Click(Sender: TObject);
var a1, a2:string; //змінні рядкового типу
    y:boolean; //змінна логічного типу
    i, n:integer; //змінна цілого типу
begin
    y:=false; //ознака відсутності символу в рядку
    a2:='Такого символу немає!'; //початковий результат пошуку
    a1:=Label1.Caption; //змінний a1 присвоюється значення рядка
    n:=length(a1); //визначення довжини рядка
    i:=0; //початковий номер символу
    repeat //початок циклу
    i:=i+1; //підготовка до вибору чергового символу
    if a1[i]='w' then //перевірка наявності символу в рядку
    begin //початок операторних дужок
    a2:='Такий символ є, він на позиції № ';
    y:=true; //ознака наявності символу в рядку
    end; //кінець операторних дужок
    until y or (i>n); //перевірка умови завершення циклу
    if y then Label2.Caption:=a2+IntToStr(i) //перевірка i
    else Label2.Caption:=a2 //виведення результату
end;
end.

```

Рис. 1. Програмний код пошуку заданого символу в рядку

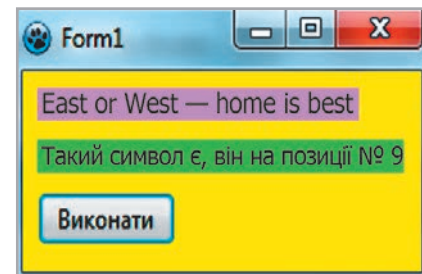


Рис. 2. Результат пошуку символу w в рядку

• Пошук заданого підрядка.

Пошук заданого підрядка в рядку можна виконати різними способами. Найпростіший із них — використання функції pos(S1, S2), яка фактично для цього і призначена. Результатом виконання цієї функції є номер позиції, починаючи з якої міститься підрядок у рядку. Якщо значення функції дорівнює нулю, то це свідчить про відсутність підрядка в рядку.

Розглянемо алгоритм пошуку заданого підрядка.

Крок 1	Увести a1, b1	Крок 4	Такий підрядок існує. Виведення n
Крок 2	n:=pos(b1, a1)	Крок 5	Такий підрядок відсутній.
Крок 3	Якщо n>0, то крок 4, інакше — крок 5	Крок 6	Кінець

Приклад 2. На рис. 3 зображено програмний код пошуку підрядка в рядку. На формі розташовано такі об'єкти: Edit1, Edit2, Label1, Label2, Label3 і Button1. До розробки програмного коду властивості Text об'єкта Edit1 надано значення Рядок, у якому відшукується підрядок, властивості Text об'єкта Edit2 — значення Підрядок, що відшукується, властивості Caption

об'єкта Label1 — значення Every country has its customs і властивості Caption об'єкта Label2 — значення country has. Нагадаємо, що у випадку відсутності підрядка в рядку функція pos набуває нульового значення.

Результат виконання програмного коду подано на рис. 4.

```

procedure TForm1.Button1Click(Sender: TObject);
var a1, b1, y:string; n:integer; //оголошення змінних
begin
  //змінна a1 набуває значення властивості Caption об'єкта Label1
  a1:=Label1.Caption;
  //змінна b1 набуває значення властивості Caption об'єкта Label2
  b1:=Label2.Caption;
  //визначення номера позиції входження підрядка в рядок
  n:=pos (b1, a1);
  if n>0 then //якщо підрядок відсутній у рядку, то n=0
    y:='Такий підрядок є, він починається з позиції №'
    +IntToStr(n)
  else y:='Такий підрядок відсутній';
  Label3.Caption:=y //виведення результату/
end;
end.

```

Рис. 3. Програмний код пошуку підрядка в рядку

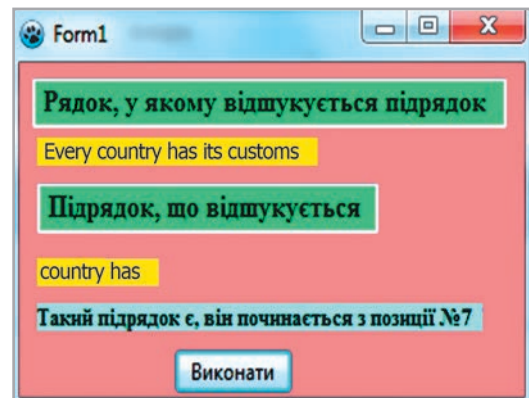


Рис. 4. Результат пошуку підрядка в рядку

- Упорядкування символів у рядку.

Раніше вже зазначалося, що в середовищі Lazarus існують деякі проблеми з використанням функцій опрацювання рядків і символів національних алфавітів. Тому будемо використовувати в основному латинські символи, для кодування яких застосовується перша половина кодової таблиці ASCII (символи з кодом від 0 до 127). Латинські символи (так само як і національні символи) кодуються в алфавітному порядку за зростанням (рис. 5). Це означає, що, наприклад, символ а має менший код, ніж символ b, а символ b — менший код, ніж символ c. Тому впорядкування символів рядка в алфавітному порядку не викликає труднощів, оскільки воно зводиться до простого порівняння їхніх кодів.

Оскільки рядок фактично є масивом символів, а кожний символ — двійковим числом, то алгоритм упорядкування символів у рядку принципово не відрізняється від розглянутих вище алгоритмів упорядкування чисел у числовому масиві.

65	A	97	a
66	B	98	b
67	C	99	c
68	D	100	d
69	E	101	e
70	F	102	f
71	G	103	g
72	H	104	h
73	I	105	i

Рис. 5. Фрагмент кодової таблиці ASCII



Приклад 3. На рис. 6 зображено код упорядкування символів у рядку методом вибору. На формі розташовано об'єкти: Edit1, Edit2, Label1, Label2, Button1. Властивості Text об'єкта Edit1 присвоєно значення Рядок до упорядкування символів, властивості Text об'єкта Edit2 — значення Рядок після упорядкування символів, властивості Caption об'єкта Label1 — значення copyleft media text. Виведення рядка

з упорядкованими символами здійснюється за допомогою властивості Caption об'єкта Label2.

Результат виконання програмного коду подано на рис. 7. Зверніть увагу на те, що у рядку впорядкованих символів відсутні пробіли. Насправді вони є і розташовані на початку цього рядка, тому що код пробілу має найменше значення серед усіх латинських символів.

```

procedure TForm1.Button1Click(Sender: TObject);
var st:string;           //змінна рядкового типу
    z:char;              //змінна символного типу
    m, n, i, p:integer;  //змінні цілого типу
begin
    //змінна st набуває значення властивості Caption об'єкта Label1
    st:=Label1.Caption;
    n:=length(st);      //визначення довжини рядка
    p:=n;                //початкове значення правої межі рядка
    while p>1 do        //зовнішній цикл
begin                    //відкриваються операторні дужки
    m:=1;                //початкове значення лівої межі рядка
    for i:=2 to p do    //внутрішній цикл
    if st[i]>st[m] then m:=i; //порівняння символів
    z:=st[m]; st[m]:=st[p]; st[p]:=z; //перестановка символів
    p:=p-1              //зменшення правої межі рядка
end;                    //закриваються операторні дужки
    Label2.Caption:=st  //виведення рядка
end;
end.

```

Рис. 6. Код упорядкування символів у рядку методом вибору

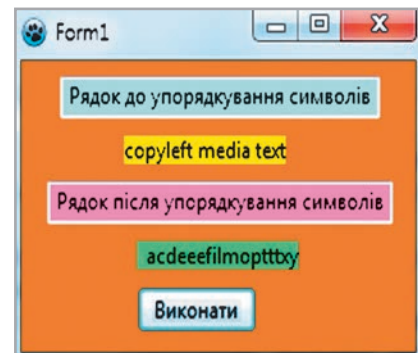


Рис. 7. Результат упорядкування символів рядка

? Запитання для перевірки знань

1. Визначте, яке значення набуває змінна **n** після виконання фрагмента коду:
`s1:='x_1 xor y1';`
`s2:='xor';`
`n:=pos(s1, s2);`
2. Сформулюйте алгоритм пошуку символу в рядку.
3. У чому полягає сутність алгоритму визначення кількості слів у рядку?
4. Сформулюйте сутність алгоритму виділення слів у рядку.

📁 Завдання для самостійного виконання

1. Розробіть алгоритм заміни у рядку одного символу іншим заданим символом. Наприклад, у рядку **game** замініть літеру **g** літерою **n**.
2. Задано послідовність латинських символів. Упорядкуйте їх в алфавітному порядку. Якщо в послідовності цих символів є символ **f**, після нього вставте два пробіли.
3. Рядок містить назви українських міст, записані англійськими літерами. Розробіть алгоритм і програмний код їх упорядкування за алфавітом. Перевірте роботу програми на прикладі назв міст: **Poltava, Sumy, Chernivtsi, Uzhhorod, Kyiv, Ternopil**.
4. Розробіть код упорядкування в алфавітному порядку назв п'яти найбільших країн світу за кількістю населення.

Практична робота № 19

Тема. Розробка алгоритмів з обробкою символьних та рядкових величин, їх реалізація у вигляді програм.

Завдання: розробити алгоритм і програму для реалізації одного із завдань варіантів 1–3.

Варіант 1. Чотири латинські літери розташовані в алфавітному порядку: **d, e, m, o**. Отримайте з них слово **modem**.

Варіант 2. Виправте помилки в рядку: **Current localtime inUkraine** так, щоб отримати рядок: **Current local time in Ukraine**.

Варіант 3. Вставте у рядок **A little is a dangerous thing** слово **knowledge** після слова **little**.

Обладнання: комп'ютер із встановленою програмою — середовищем програмування.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Виберіть один із варіантів завдань за вказівкою вчителя або самостійно. (У разі успішного виконання одного варіанта, виконайте інший на власний розсуд.)
 - 2 Проаналізуйте умову задачі.
 - 3 Розробіть алгоритм і реалізуйте його в середовищі програмування.
 - 4 Виконайте програму. Проаналізуйте отриманий результат.
- **Зробіть висновок:** які функції опрацювання рядків і символів доцільно застосовувати у цій роботі.

Практична робота № 20

Тема. Класичні алгоритми для роботи з рядками та їх реалізація у вигляді програм.

Завдання: розробити алгоритм і програму для реалізації одного із завдань варіантів 1–3.

Варіант 1. На скільки символів відрізняється найдовше слово від найкоротшого у заданому рядку, наприклад у такому: **We live in Ukraine?**

Варіант 2. Упорядкуйте слова рядка в алфавітному порядку. Проаналізуйте порядок виконання алгоритму на прикладі такого рядка: **resistor, library, raster, host, double**.

Варіант 3. Вилучте з рядка всі слова, які починаються з літери **b**, наприклад з рядка: **macro, bit, sampler, bar, save, basic**.

Обладнання: комп'ютер із встановленою програмою — середовищем програмування.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Виберіть один із варіантів завдань за вказівкою вчителя або самостійно. (У разі успішного виконання одного варіанта, виконайте інший на власний розсуд.)
 - 2 Проаналізуйте умову задачі.
 - 3 Розробіть алгоритм і реалізуйте його у середовищі програмування.
 - 4 Виконайте програму. Проаналізуйте отриманий результат.
- **Зробіть висновок:** які компоненти доцільно використовувати у цій роботі.

Розділ 11. ДВОВИМІРНІ МАСИВИ

11.1. Загальні відомості про двовимірні масиви



Наведіть означення одновимірного масиву. Як оголошуються одновимірні масиви? Які існують класичні алгоритми опрацювання значень елементів одновимірного масиву? Чим, на вашу думку, відрізняються двовимірні масиви від одновимірних?

Ви вже знаєте, що масив — це структурований тип даних, елементи якого мають один тип.



Двовимірний масив являє собою таблицю, що складається з фіксованої кількості рядків і стовпців.

Елемент двовимірного масиву має два індекси: номер рядка і номер стовпця (прикладі 1, 2).

Приклад 1. Двовимірний масив цілих чисел із трьома рядками і чотирма стовпцями може мати такий вигляд:

		Номери стовпців			
		1	2	3	4
Номери рядків	1	44	32	100	7
	2	55	41	72	53
	3	33	84	91	46

Приклад 2. Кількість пального, що доставляють з нафтобази на станцію заправки у робочі дні, можна подати у вигляді двовимірного масиву (таблиці):

		Дні тижня				
		понеділок	вівторок	середа	четвер	п'ятниця
Назви станцій	A3-20	6,4	7	5,5	6	6,2
	ЗЛ_2	10	5,5	6	6,4	7,4
	РК-5	7,8	4,8	5,7	6,6	7

У кожній клітинці вказана кількість пального (в тоннах), що постачається на станцію. Для опрацювання такого масиву потрібно дні тижня ототожнювати з номерами стовпців, а назви станцій — із номерами рядків.

Загальну структуру двовимірного масиву можна подати в такому вигляді:

$a[1,1], a[1,2], \dots, a[1,j], \dots, a[1,n]$

$a[2,1], a[2,2], \dots, a[2,j], \dots, a[2,n]$

...

$a[m,1], a[m,2], \dots, a[m,j], \dots, a[m,n]$

У квадратних дужках — індекси елемента масиву. Місце елемента у двовимірному масиві визначається двома індексами. Перший індекс — це номер рядка масиву, другий — номер стовпця.

Способи оголошення двовимірного масиву принципово відрізняються від оголошення одновимірного масиву. Різниця в тому, що в оголошенні двовимірного масиву через дві крапки вказується діапазон індексів рядків, а потім, відокремлюючи комою, — діапазон індексів стовпців (приклад 3).

Приклад 3.

```
var mas: array [1..4, 1..5] of char; —  
оголошено двовимірний масив символів mas, що має чотири рядки з номерами від 1 до 4 і п'ять стовпців із номерами від 1 до 5.
```

У розділі змінних структура оголошення двовимірного масиву така:

```
var <ім'я змінної>: array [діапазон рядків, діапазон стовпців] of <тип елементів>;
```

Оголосити двовимірний масив можна і так: оголосити новий тип даних, а потім описати змінну нового типу (приклад 4):

```
type <ім'я типу>=array [діапазон рядків, діапазон стовпців] of <тип елементів>;
```

```
var <ім'я змінної>: <ім'я типу>;
```

Для оголошення масиву можна використовувати константи, визначені у розділі оголошення констант. Наприклад:

```
const n=7; m=5;
```

```
var mas: array [1..n,1..m] of integer;
```

Для звернення до елемента двовимірного масиву слід вказати ім'я масиву і значення індексів у квадратних дужках. Наприклад, для звернення до елемента, розташованого у третьому рядку четвертого стовпця слід вказати: `mas[3,4]`.

Будь-якому символу можна присвоїти певне значення, наприклад: `mas[2,4]:=5` — елементу двовимірного масиву, розташованому у другому рядку четвертого стовпця, присвоєно символ 5.

Двовимірні масиви можна вводити як у консольному, так і у візуальному режимах. Як правило, для введення значень елементів двовимірного масиву використовують вкладені цикли. Внутрішній цикл використовується для введення значень елементів різних стовпців одного рядка, а зовнішній — для переходу до нового рядка. Для введення значень елементів масиву в консольному режимі використовується оператор `readln (read)`, а для виведення — оператор `writeln (write)`.

Розглянемо два приклади введення і виведення двовимірних масивів у консольних додатках.

Приклад 5. На рис. 1 зображено консольну програму формування двовимірного масиву цілими випадковими числами і виведення їх за допомогою оператора `write` по рядках.

```
const n=3; m=4;
var
  i,j:integer;           //змінні цілого типу
  mas: array [1..n,1..m] of integer; //двовимірний масив
begin
  Randomize;           //для генерування різних випадкових чисел
  for i:=1 to n do     //зовнішній цикл формування масиву
  for j:=1 to m do     //внутрішній цикл формування масиву
  mas [i,j]:=random (9); //формування масиву
  for i:=1 to n do     //зовнішній цикл виведення масиву
  begin
    for j:=1 to m do   //внутрішній цикл виведення масиву
      write (' mas[' ,i ,',' ,j ,']=', mas[i,j]); //виведення масиву
      writeln;         //призупинення виконання програми
    end;
  end;
end.
```

Рис. 1. Консольна програма формування елементів масиву випадковими числами

Після виконання програми на екран буде виведено імена елементів масиву та їхні значення (рис. 2).

```
mas[1,1]=6 mas[1,2]=0 mas[1,3]=2 mas[1,4]=4
mas[2,1]=5 mas[2,2]=4 mas[2,3]=8 mas[2,4]=3
mas[3,1]=6 mas[3,2]=1 mas[3,3]=2 mas[3,4]=5
```

Рис. 2. Результат виконання програми

Приклад 4.

```
type nov=array [1..8,1..10] of real;
var mas: nov; — оголошено числовий масив дійсного типу з іменем mas, у якому 8 рядків і 10 стовпців.
```



Брендан Айк, автор мови програмування JavaScript, що є стандартом для веб-програмування. Брав участь у створенні американської компанії Mozilla, яка розробляє браузер Firefox.

Приклад 6. На рис. 3 зображено консольну програму створення двовимірного масиву шляхом уведення значення кожного елемента з клавіатури і виведення цього масиву. Якщо

```
const n=3; m=4; //іменовані константи
var i, j: integer; //змінні цілого типу
    mas: array [1..n,1..m] of real; //двовимірний масив
begin
  for i:=1 to n do //зовнішній цикл уведення масиву
  for j:=1 to m do //внутрішній цикл уведення масиву
  begin //відкриваються операторні дужки
    write ('увести mas[' , i , ',' , j , '=' ); //повідомлення про уведення
    readln (mas[i,j]) //уведення чергового елемента масиву
  end; //закриваються операторні дужки
  for i:=1 to n do //зовнішній цикл виведення масиву
  for j:=1 to m do //внутрішній цикл виведення масиву
  writeln ('mas[' , i , ',' , j , '=' , mas[i, j]); //виведення масиву
  readln //призупинення виконання програми
end.
```

Рис. 3. Консольна програма введення масиву з клавіатури і його виведення

після запуску програми послідовно вводити з клавіатури числа від 1 до 12, то отримаємо результат, поданий на рис. 4.

```
mas[1,1]=1.000000000000000E+000
mas[1,2]=2.000000000000000T+000
mas[1,3]=3.000000000000000E+000
mas[1,4]=4.000000000000000E+000
mas[2,1]=5.000000000000000E+000
mas[2,2]=6.000000000000000E+000
mas[2,3]=7.000000000000000E+000
mas[2,4]=8.000000000000000E+000
mas[3,1]=9.000000000000000E+000
mas[3,2]=1.000000000000000E+001
mas[3,3]=1.100000000000000E+001
mas[3,4]=1.200000000000000E+001
```

Рис. 4. Результат виконання програми



Запитання для перевірки знань

- 1 Наведіть означення двовимірного масиву.
- 2 Наведіть загальну структуру двовимірного масиву.
- 3 Як здійснюється звернення до окремих елементів двовимірного масиву?
- 4 Як оголошується двовимірний масив у розділі змінних?
- 5 Як оголошується двовимірний масив у розділі типів?
- 6 Наведіть приклад оголошення двовимірного масиву з використанням іменованих констант.



Завдання для самостійного виконання

- 1 Визначте помилки в операторах:
 - a) `type mas=array [1..4,5] of real; var mas;`
 - b) `var a; array [3,4] of integer;`
- 2 Розробіть консольну програму введення і виведення двовимірного масиву розмірами 3×4 , елементами якого є назви геометричних фігур (трикутник, прямокутник та ін.).
- 3 Знайдіть в Інтернеті 8 найбільших озер України. Збережіть назви цих озер у двовимірному масиві, розмір якого визначте самостійно. Розробіть консольну програму реалізації цього завдання.

11.2. Введення і виведення значень елементів двовимірних масивів у візуальному режимі

Яку структуру мають двовимірні масиви і як вони оголошуються? Поясніть порядок звернення до окремих елементів двовимірного масиву. За допомогою яких операторів здійснюється введення і виведення значень елементів масиву в консольному режимі?



Середовище Lazarus має різноманітні компоненти для організації введення/виведення даних у візуальному режимі. Раніше вже були описані й наведені приклади використання компонентів TEdit, TLabel, TMemo, TListBox і функцій InputBox(), ShowMessage(). Ці засоби застосовувалися у процесі введення-виведення одновимірних масивів. Досить зручним для організації введення/виведення значень двовимірних масивів є компонент TStringGrid (таблиця рядків). Він визначений у модулі Grids, а його піктограма розташована на вкладці Additional палітри компонентів. Компонент переноситься на форму і стає об'єктом так само, як і інші компоненти. Основні властивості об'єкта наведено в табл. 1.

Таблиця 1. ОСНОВНІ ВЛАСТИВОСТІ ОБ'ЄКТА StringGrid

Назва властивості	Призначення властивості
Name	Визначає ім'я об'єкта
ColCount	Визначає кількість стовпців
RowCount	Визначає кількість рядків
Gells [col, row]	Вказує клітинку, розташовану на перехресті стовпця номер col і рядка номер row (стовпці нумеруються від 0 до ColCount - 1, а рядки — від 0 до RowCount - 1)
FixedCols	Визначає кількість лівих стовпців, які виділяються кольором і залишаються на місці у процесі прокручування горизонтальної смуги
FixedRows	Визначає кількість верхніх рядків, які виділяються іншим кольором і залишаються на місці у процесі прокручування вертикальної смуги
ScrollBars	Визначає наявність смуг прокручування
Options.goEditing	Значення властивості true дозволяє редагування масиву, а значення false забороняє його редагування
DefaultColWidth	Визначає ширину стовпців
DefaultRowHeight	Визначає висоту рядків
Height	Визначає висоту об'єкта
Width	Визначає ширину об'єкта
Font	Визначає шрифт, яким відображається вміст клітинок

Приклад 1. Розглянемо приклад уведення цифрових значень у клітинки об'єкта StringGrid1 за допомогою клавіатури, створення двовимірного масиву із цих елементів і виведення значень масиву у зворотному порядку в клітинки об'єкта StringGrid2.

Для цього розмістимо на формі два компоненти TStringGrid і компонент TButton. Надамо такі значення властивостям об'єктів StringGrid1 і StringGrid2: ColCount — 4; RowCount — 3; FixedCols — 0; FixedRows — 0; Options — goEditing — true.

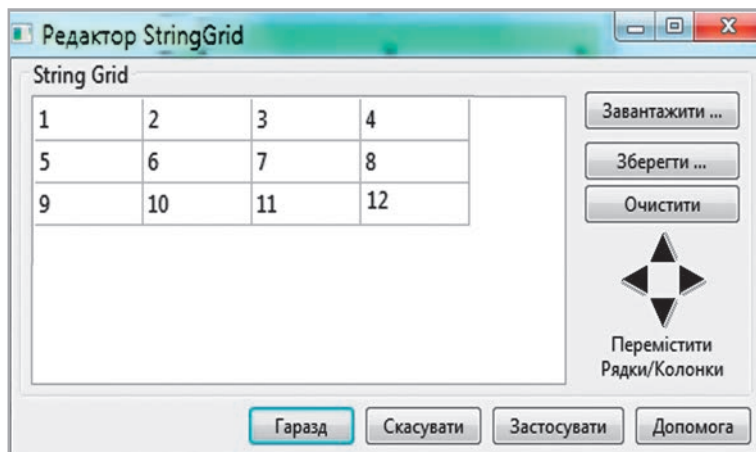


Рис. 1. Вікно редактора StringGrid

Значення інших властивостей залишимо за замовчуванням. Після цього двічі клацнемо поле об'єкта StringGrid1, розташованого на формі. Відкриється вікно Редактор StringGrid, зображене на рис. 1. У клітинки цього редактора введемо в 3 рядки і 4 стовпці цілочислові значення так, як показано на рис. 1.

Після завершення введення цифрових значень клацнемо кнопку Гаразд. Форма набуде вигляду, як на рис. 2.

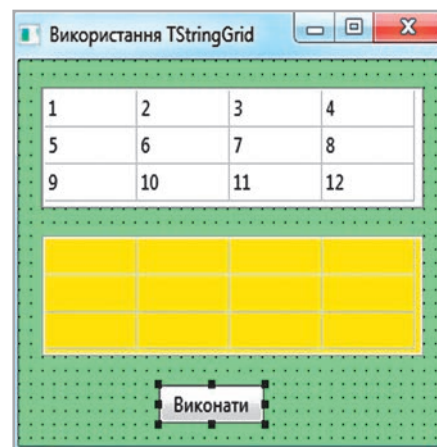


Рис. 2. Вміст об'єкта StringGrid1 після введення масиву

На рис. 3 зображено код, за допомогою якого послідовно зчитуються значення клітинок об'єкта StringGrid1, зі значень яких формується двовимірний масив mas розмірністю 3×4. Потім значення цього масиву виводяться у зворотному порядку в клітинки об'єкта StringGrid2.

```
procedure TForm1.Button1Click(Sender: TObject);
var i,j:integer; //змінні цілого типу
    mas: array [1..3,1..4] of integer; //двовимірний масив
begin
    for i:=1 to 3 do //зовнішній цикл формування масиву
        for j:=1 to 4 do //внутрішній цикл формування масиву
            //введення значень масиву із клітинок StringGrid1
            mas[i,j]:=StrToInt(StringGrid1.Cells[j-1,i-1]);
        for i:=1 to 3 do //зовнішній цикл виведення масиву
            for j:=1 to 4 do //внутрішній цикл виведення масиву
                //виведення елементів масиву в клітинки StringGrid2
                StringGrid2.Cells[j-1,i-1]:=IntToStr(mas[4-i,5-j]);
            end;
        end.
```

Рис. 3. Код введення і виведення масиву за допомогою об'єктів StringGrid1 і StringGrid2

Звернемо увагу на те, що у властивості Cells спочатку вказується номер стовпця, потім — номер рядка, а в масиві — навпаки.

Після успішної компіляції та виконання програми отримаємо результат, поданий на рис. 4.

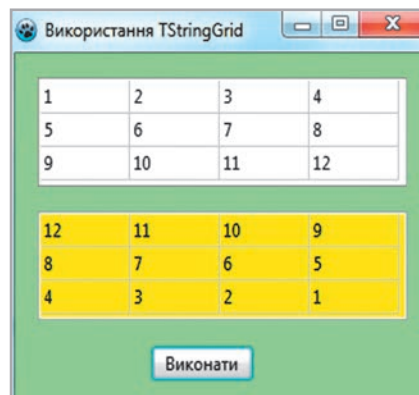


Рис. 4. Результати введення і виведення двовимірного масиву

Приклад 2. Створимо двовимірний масив розмірністю $n \times m$, елементами якого є цілі випадкові числа з діапазону від 0 до 20 і виведемо елементи цього масиву в клітинки об'єкта StringGrid1. Додамо на форму компонент TButton, для якого визначимо подію OnClick.

```
procedure TForm1.Button1Click(Sender: TObject);
const n=3; m=4; //іменовані константи
var i, j:integer; //змінні цілого типу
mas: array [1..n,1..m] of integer; //двовимірний масив
begin
  Randomize; //генерування різних випадкових чисел
  for i:=1 to n do //зовнішній цикл формування масиву
  for j:=1 to m do //внутрішній цикл формування масиву
  mas [i,j]:=Random (20); //формування масиву
  for i:=1 to n do //зовнішній цикл виведення масиву
  for j:=1 to m do //внутрішній цикл виведення масиву
  //виведення елементів масиву в клітинки StringGrid1
  StringGrid1.Cells[j-1,i-1]:=IntToStr(mas[i,j]);
end;
end.
```

Рис. 5. Код створення і виведення масиву в клітинки об'єкта StringGrid1

Значення основних властивостей об'єкта StringGrid1 обираємо такі самі, як у прикладі 1. Програму реалізації цього завдання наведено на рис. 5.

Результат виконання програми зображено на рис. 6.

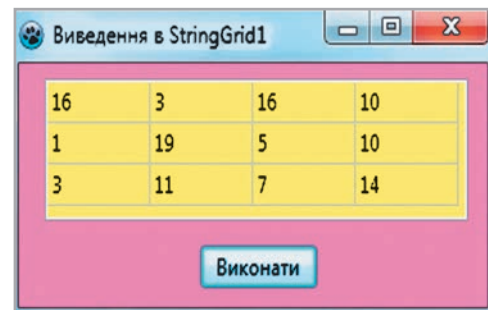


Рис. 6. Двовимірний масив випадкових чисел

Приклад 3. Перший член спадної геометричної прогресії дорівнює a , знаменник q . Обчислити 12 перших членів цієї прогресії і створити з їх значень масив розмірністю 3×4 . Елементи масиву розмістити в порядку зменшення їхніх значень. Вивести елементи масиву за допомогою компонента TStringGrid.

```
procedure TForm1.Button1Click(Sender: TObject);
var p, q: real; //змінні дійсного типу
i, j:integer; //змінні цілого типу
mas: array [1..3,1..4] of real; //масив чисел дійсного типу
begin
  p:=10; //значення першого члена прогресії
  q:=0.5; //значення знаменника прогресії
  for i:=1 to 3 do //зовнішній цикл створення масиву
  for j:=1 to 4 do //внутрішній цикл створення масиву
  begin
  mas[i,j]:=p; //елементу масива присвоюється значення
  p:=p*q; //обчислення члена прогресії
  end; //закриваються операторні дужки
  for i:=1 to 3 do //зовнішній цикл виведення масиву
  for j:=1 to 4 do //внутрішній цикл виведення масиву
  //виведення елементів масиву в клітинки StringGrid1
  StringGrid1.Cells[j-1,i-1]:=FloatToStr(mas[i,j]);
end;
end.
```

Рис. 7. Код створення масиву з членів спадної геометричної прогресії

Надамо основним властивостям об'єкта StringGrid1 такі значення, як і в попередніх прикладах. Програму реалізації завдання зображено на рис. 7.

Результат виконання програми наведено на рис. 8.

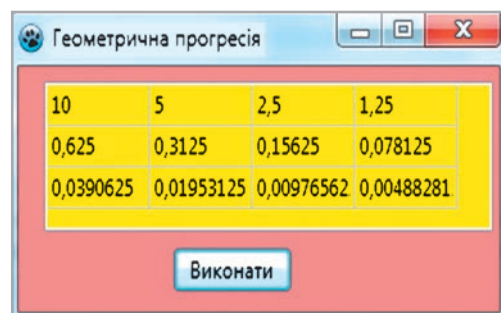


Рис. 8. Масив значень членів спадної геометричної прогресії

Якщо для динамічного масиву не вистачає пам'яті, виникає помилка **EOutOfMemory**. Якщо після використання динамічного масиву потреба в ньому відпадає, його можна знищити за допомогою процедури **Finalize** (ім'я масиву).

Lazarus забезпечує роботу не тільки зі статичними, але й із динамічними масивами. Для динамічних масивів значення їх розмірів під час оголошення масиву не задаються, вказується лише тип елементів масиву. Якщо розмір масиву невідомий, то не можна відвести для нього і пам'ять. Для відведення пам'яті й надання програмі можливості використовувати такі масиви виконується їх ініціалізація за допомогою функції **SetLength** (ім'я масиву, кількість елементів у масиві).

Інші способи введення значень елементів двовимірних масивів будуть розглянуті в наступних розділах.



Запитання для перевірки знань

- 1 Які компоненти забезпечують введення/виведення даних?
- 2 Який компонент зручний для введення/виведення двовимірних масивів?
- 3 Як нумеруються стовпці й рядки об'єкта **StringGrid**?
- 4 Яке призначення мають властивості **ColCount** і **RowCount**?
- 5 Назвіть основні властивості об'єкта **StringGrid**.
- 6 Для чого призначено вікно редактора об'єкта **StringGrid**?



Завдання для самостійного виконання

- 1 У школі 3 дев'яті класи. Від кожного класу в кросі на 1 км беруть участь по 4 учні. Використайте компонент **TStringGrid** для введення результатів забігу. Створіть відповідний двовимірний масив. Результати учнів кожного класу відобразіть за допомогою компонентів **TListBox**.
- 2 Чотири учасники гри, починаючи з першого, по черзі кидають 5 разів гральний кубик із цифрами від 1 до 6. Зареєструйте результати кидання кубика у масиві й виведіть результати за допомогою компонента **TStringGrid**.
- 3 Відомі результати зовнішнього незалежного оцінювання п'яти учнів з української мови, історії, математики і фізики. За допомогою функції **InputBox()** уведіть ці результати в масив і виведіть його значення за допомогою функції **ShowMessage()**.
- 4 У табл. 2 наведено 4 кращі команди чемпіонату України з футболу з 2011 до 2015 року. Використайте 2 компоненти **TStringGrid** для створення відповідного масиву та виведення його значень.

Таблиця 2

Рік	1-ше місце	2-ге місце	3-тє місце	4-те місце
2011–2012	Шахтар	Динамо	Металіст	Дніпро
2012–2013	Шахтар	Металіст	Динамо	Дніпро
2013–2014	Шахтар	Дніпро	Металіст	Динамо
2014–2015	Динамо	Шахтар	Дніпро	Зоря

11.3. Класичні алгоритми опрацювання значень елементів двовимірних масивів

Пригадайте, які компоненти вже використовувалися для введення і виведення значень елементів масиву. Які з них доцільно застосовувати для двовимірних масивів? Які алгоритми, на ваш погляд, можна назвати класичними для двовимірних масивів?

До класичних алгоритмів опрацювання двовимірних масивів можна віднести обчислення загальної суми значень елементів масиву, суми значень елементів головної і побічної діагоналей, суми значень елементів кожного рядка і кожного стовпця.

- Обчислення загальної суми значень елементів двовимірного масиву.

Сутність алгоритму обчислення загальної суми значень елементів двовимірного масиву розглянемо на прикладі такого масиву:

$$\begin{matrix} a[1,1], a[1,2], \dots, a[1,n] \\ a[2,1], a[2,2], \dots, a[2,n] \end{matrix}$$

$$\dots a[m,1], a[m,2], \dots, a[m,n]$$

Існують різні підходи до обчислення суми значень елементів масиву. Зокрема, можна знайти суму чисел у кожному рядку, а потім додати значення отриманих сум або обчислити суму чисел у кожному стовпці, а потім додати значення отриманих сум. Однак частіше застосовується метод «накопичення», сутність якого полягає в тому, що на початку значення суми дорівнює нулю. Потім до цього значення додається перший елемент першого рядка, до отриманої суми — другий елемент того самого рядка і так далі до останнього елемента цього рядка. До суми чисел першого рядка додається перший елемент другого рядка, потім другий елемент другого рядка і так далі до останнього елемента.

Приклад 1. Для двовимірного масиву:

$$\begin{matrix} a[1,1], a[1,2], a[1,3] \\ a[2,1], a[2,2], a[2,3] \end{matrix}$$

обчислення суми за допомогою цього методу буде здійснюватися так:

$$\begin{matrix} s[0]:=0; & s[3]:=s[2] + a[1,3]; & s[6]:=s[5] + a[2,3]. \\ s[1]:=s[0] + a[1,1]; & s[4]:=s[3] + a[2,1]; & \\ s[2]:=s[1] + a[1,2]; & s[5]:=s[4] + a[2,2]; & \end{matrix}$$

Приклад 2. Для двовимірного масиву цілих чисел:

$$\begin{matrix} 12, 21, 10 \\ 33, 22, 13 \end{matrix}$$

сума його елементів обчислюється у такому порядку:

$$\begin{matrix} s[0]:=0; & s[3]:=33+10=43; & s[6]:=98+13=111. \\ s[1]:=0+12=12; & s[4]:=43+33=76; & \\ s[2]:=12+21=33; & s[5]:=76+22=98; & \end{matrix}$$


Математик, професор Дортмундського коледжу (США) Джон (Янош) Кемені (на фото праворуч) разом із Томасом Курцем розробив мову програмування BASIC та мережеву систему користування декількома комп'ютерами одночасно (англ. *timesharing*). Є одним із піонерів викладання основ програмування.

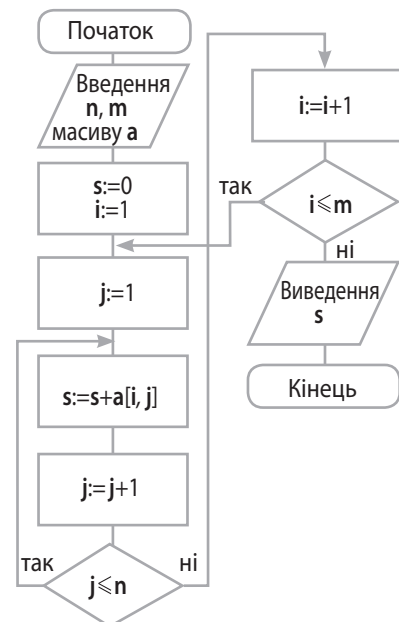


Рис. 1. Блок-схема алгоритму обчислення суми значень елементів двовимірного масиву

Блок-схему алгоритму зображено на [рис. 1](#).

У блок-схемі використано такі змінні: i , j — лічильники циклів (i — лічильник рядків, j — лічильник стовпців), m — кількість рядків у масиві, n — кількість стовпців у масиві, s — сума значень елементів масиву.

За допомогою внутрішнього циклу обчислюється сума чисел у рядку, а за допомогою зовнішнього циклу — перехід до чергового рядка. Використання вкладених циклів характерне для більшості програм опрацювання двовимірних масивів.

Приклад 3. Розмістимо на формі компонент TLabel і компонент TButton. Для об'єкта Button1 обираємо подію OnClick. Програму, що реалізує алгоритм обчислення загальної суми значень елементів масиву, зображено на рис. 2.

```

procedure TForm1.Button1Click(Sender: TObject);
const n=3; m=4;           //іменовані константи
var i, j:integer;        //змінні цілого типу
    s:real;              //змінна дійсного типу
    mas: array [1..n,1..m] of real; //двовимірний масив
begin
  for i:=1 to n do       //зовнішній цикл уведення масиву
  for j:=1 to m do       //внутрішній цикл уведення масиву
  //уведення значень із клавіатури і присвоювання їх елементам масиву
  mas [i,j]:=StrToFloat(InputBox ('Уведення',
  'Увести mas['+IntToStr(i)+'+', '+IntToStr(j)+'+', ' '));
  s:=0;                 //початкове значення суми
  for i:=1 to n do     //зовнішній цикл обчислення суми
  for j:=1 to m do     //внутрішній цикл обчислення суми
  s:=s+mas[i,j];       //обчислення суми
  Label1.Caption:='Сума='+FloatToStr(s); //виведення результату
end;
end.

```

Рис. 2. Програма обчислення суми значень елементів двовимірного масиву

Після успішної компіляції програми запустимо її на виконання й уведемо такі значення:

10,5	2	3	4
1	11	12,7	15
5	6	7	8

Результат виконання програми подано на рис. 3.

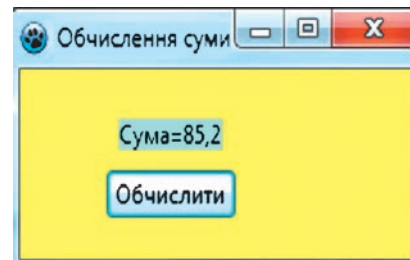


Рис. 3. Результат обчислення суми значень елементів двовимірного масиву

Приклад 4. Дано масив цілих чисел:

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Головна діагональ: числа 1, 6, 11, 16; побічна діагональ: числа 4, 7, 10, 13.

- Обчислення суми значень елементів масиву, розташованих на головній і побічній діагоналях.

Термін «головна діагональ» зазвичай використовується у масивах, у яких кількість рядків і стовпців однакова. **Головна діагональ** — це діагональ, що з'єднує лівий верхній елемент масиву з правим нижнім. Якщо, приміром, масив має розмірність 4×4 , то на головній діагоналі розташовані елементи $a[1,1]$, $a[2,2]$, $a[3,3]$ і $a[4,4]$ (приклад 4).

Побічна діагональ — це діагональ, що з'єднує правий верхній елемент масиву з лівим нижнім. На цій діагоналі для наведеного вище масиву розташовані елементи $a[1,4]$, $a[2,3]$, $a[3,2]$, $a[4,1]$.

Розглянемо алгоритм обчислення значень елементів, розташованих на головній діагоналі. Позначимо змінною s значення суми елементів, змінною n — кількість рядків і стовпців у масиві, змінною i — поточний номер рядка (стовпця), $mas[i, i]$ — значення елемента масиву на головній діагоналі. З урахуванням таких позначень алгоритм обчислення суми значень цих елементів, можна подати так:

Крок 1	Увести масив $mas[n,n]$	Крок 5	$i:=i+1$
Крок 2	$s:=0$	Крок 6	Якщо $i \leq n$, то крок 4, інакше — крок 7
Крок 3	$i:=1$	Крок 7	Виведення s
Крок 4	$s:=s+mas [i,i]$	Крок 8	Кінець



Приклад 5. Двовимірний масив розмірністю $n \times n$ набуває значень членів арифметичної прогресії, перший член якої позначимо змінною a , різницю прогресії — змінною d . На рис. 4 зображено програму створення двовимірного

масиву зі значень членів цієї прогресії й обчислення суми значень елементів головної діагоналі.

Результат виконання програми наведено на рис. 5.

```

procedure TForm1.Button1Click(Sender: TObject);
const n=4; //іменована константа
var i, j, a, d, s:integer; //змінні цілого типу
mas: array [1..n,1..n] of integer; //двовимірний масив
begin
  s:=0; //початкове значення суми
  a:=1; //значення першого члена прогресії
  d:=2; //різниця прогресії
  for i:= 1 to n do //зовнішній цикл створення масиву
  for j:=1 to n do //внутрішній цикл створення масиву
  begin //відкриваються операторні дужки
    mas [i,j]:=a; //елемент масиву набуває значення
    a:=a+d //обчислення поточного члена прогресії
  end; //закриваються операторні дужки
  for i:=1 to n do //цикл обчислення суми
  s:=s+mas[i,i]; //обчислення суми
  label1.Caption:='Сума= '+IntToStr(s); //виведення результату
end;
end.

```

Рис. 4. Програма обчислення значень елементів головної діагоналі масиву

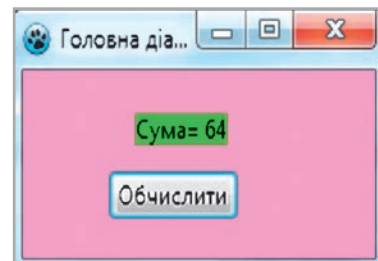


Рис. 5. Результат обчислення значень елементів головної діагоналі

Розглянемо алгоритм обчислення значень елементів побічної діагоналі масиву розмірністю $n \times n$. Використаємо ті самі позначення змінних, що й у попередньому алгоритмі.

Цей алгоритм можна подати так.

Крок 1	Увести масив <code>mas [n,n]</code>	Крок 6	<code>i:=i+1</code>
Крок 2	<code>s:=0</code>	Крок 7	<code>j:=j-1</code>
Крок 3	<code>i:=1</code>	Крок 8	Якщо <code>i<=n</code> , то крок 5, інакше — крок 9
Крок 4	<code>j:=n</code>	Крок 9	Виведення <code>s</code>
Крок 5	<code>s:=s+mas [i,j]</code>	Крок 10	Кінець

Приклад 6. Елементи масиву розмірністю 5×5 набувають значень випадкових цілих чисел. Для виведення цього масиву використаємо компонент `TStringGrid`. Для виведення суми значень побічної діагоналі — компонент `TLabel`. Для об'єкта `Button1` обрано подію `OnClick`. Для об'єкта `StringGrid1` встановлено такі

значення основних властивостей: `ColCount` — 5, `RowCount` — 5, `FixedCols` — 0, `FixedRows` — 0. Програму обчислення суми чисел побічної діагоналі зображено на рис. 6.

Результат виконання програми наведено на рис. 7.

```

procedure TForm1.Button1Click(Sender: TObject);
const n=5; //іменована константа
var i, j, s: integer; //змінні цілого типу
    mas: array [1..n,1..n] of integer; //двовимірний масив
begin
    Randomize; //для генерування різних випадкових чисел
    for i:=1 to n do //зовнішній цикл створення масиву
    for j:=1 to n do //внутрішній цикл створення масиву
    begin
        //елементу масива присвоюється ціле випадкове число
        mas [i, j]:=random (21);
        StringGrid1.Cells[j-1,i-1]:=IntToStr(mas[i,j]); //виведення масиву
    end; //закриваються операторні дужки
    s:=0; //початкове значення суми
    j:=n; //початковий номер стовпця
    for i:=1 to n do //цикл обчислення суми
    begin
        //відкриваються операторні дужки
        s:=s+mas [i, j]; //обчислення суми значень побічної діагоналі
        j:=j-1; //зменшення номера стовпця
    end; //закриваються операторні дужки
    Label1.Caption:='Сума= '+IntToStr(s); //виведення суми
end.

```

Рис. 6. Програма обчислення значень елементів побічної діагоналі

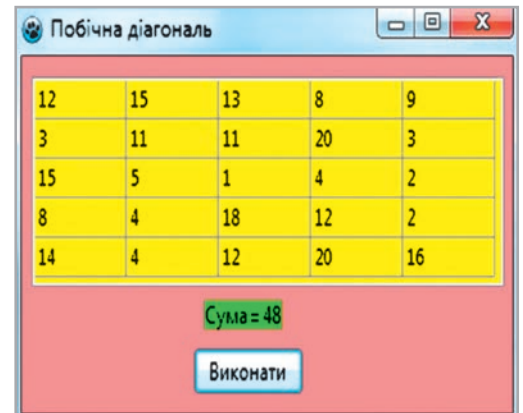


Рис. 7. Результат обчислення суми чисел побічної діагоналі

- Обчислення суми значень елементів кожного рядка масиву.

Алгоритм обчислення суми значень елементів в одному рядку двовимірного масиву принципово не відрізняється від алгоритму обчислення суми значень елементів одновимірного масиву. Тому задача обчислення суми значень елементів кожного рядка двовимірного масиву зводиться до обчислення суми значень елементів n одновимірних масивів. У процесі

розробки алгоритму використаємо такі змінні: n — кількість рядків, m — кількість стовпців, i — поточний номер рядка, j — поточний номер стовпця, s — сума значень елементів рядка. Цей алгоритм можна подати так.

Крок 1	Увести масив $mas[n,m]$	Крок 7	Якщо $j \leq m$, то крок 5, інакше — крок 8
Крок 2	$i:=1$	Крок 8	Виведення s
Крок 3	$s:=0$	Крок 9	$i:=i+1$
Крок 4	$j:=1$	Крок 10	Якщо $i \leq n$, то крок 3, інакше — крок 11
Крок 5	$s:=s+a[i,j]$	Крок 11	Кінець
Крок 6	$j:=j+1$		

Приклад 7. Програму реалізації алгоритму зображено на рис. 8. Для введення масиву використано функцію `InputBox()`, для виведення масиву — компонент `TStringGrid`, для виведення суми значень елементів рядків — компонент `TListBox`, для виконання програми — компонент `TButton` з подією `OnClick`. Для компонентів, розташованих на формі,

використано значення властивостей, аналогічних тим, які використовувалися у попередніх прикладах (звертаємо увагу на те, що для об'єкта `StringGrid1` значення властивостей `FixedCols` і `FixedRows` слід встановити такими, що дорівнюють нулю).

Результат виконання програми наведено на рис. 9.

```

procedure TForm1.Button1Click(Sender: TObject);
const n=3; m=4;           //іменовані константи
var i, j, s: integer;    //змінні цілого типу
    mas1:array[1..n,1..m] of integer; //двовимірний масив
begin
for i:=1 to n do        //зовнішній цикл уведення масиву
for j:=1 to m do        //внутрішній цикл уведення масиву
begin                  //відкриваються операторні дужки
//уведення значень елементів масиву
mas1[i,j]:=StrToInt(InputBox('Уведення',
'Увести mas1['+IntToStr(i)+'+', '+IntToStr(j)+'+', ' '));
//виведення масиву в клітинки TStringGrid1
StringGrid1.Cells[j-1, i-1]:=IntToStr(mas1[i,j]);
end;                  //закриваються операторні дужки
for i:=1 to n do        //зовнішній цикл обчислення суми рядків
begin                  //відкриваються операторні дужки
s:=0;                  //початкова сума рядка
for j:=1 to m do        //цикл обчислення суми поточного рядка
s:=s+mas1[i,j];        //обчислення суми значень рядка
ListBox1.Items.Add(IntToStr(s)); //виведення суми рядка
end;                  //закриваються операторні дужки
end;
end.

```

Рис. 8. Програма обчислення суми значень рядків двовимірного масиву

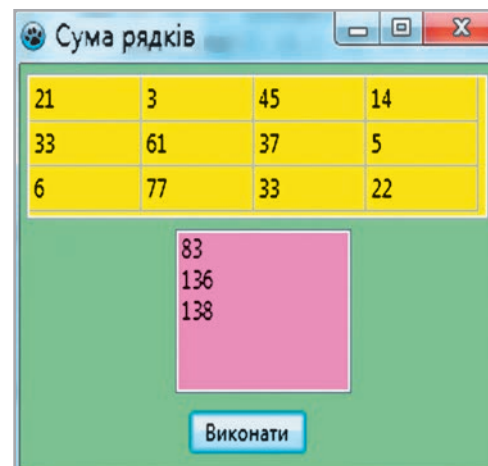


Рис. 9. Результат обчислення суми рядків двовимірного масиву

Гарний програміст у 10 разів продуктивніший, ніж середній, а відмінний — у 20–100 разів продуктивніший, ніж середній.

- Обчислення кількості заданих значень у масиві

Алгоритм обчислення кількості заданих значень у двовимірному масиві аналогічний алгоритму обчислення таких значень у одновимірному масиві. Різниця полягає лише в тому, що після завершення аналізу одного рядка необхідно здійснити перехід на наступний рядок. Розглянемо алгоритм обчислення кількості чисел, менших від значення c , у масиві $mas[n, m]$. Кількість таких чисел присвоюється змінній k .

Крок 1	Увести масив $mas[n,m]$, значення c	Крок 7	Якщо $j \leq m$, то крок 5, інакше — крок 8
Крок 2	$k:=0$	Крок 8	$i:=i+1$
Крок 3	$i:=1$	Крок 9	Якщо $i \leq n$, то крок 4, інакше — крок 10
Крок 4	$j:=1$	Крок 10	Виведення k
Крок 5	Якщо $mas[i,j] < c$, то $k:=k+1$	Крок 11	Кінець
Крок 6	$j:=j+1$		

Приклад 8. Програму реалізації алгоритму зображено на рис. 10. Масив формується випадковими числами з діапазону від 0 до 20. Значення елементів цього масиву виведено в клітинки об'єкта `StringGrid1`, а результат — в поле об'єкта `Label1`.

Результат виконання програми зображено на рис. 11. Зверніть увагу на те, що

у верхньому рядку і лівому стовпці записано номери рядків і номери стовпців. Це означає, що кількість стовпців і рядків у об'єкті `StringGrid1` на одиницю більша за кількість рядків і стовпців у масиві. Тому властивість `ColCount` повинна мати значення 5, властивість `RowCount` — значення 4, а властивості `FixedCols` і `FixedRows` — значення 1.

```

procedure TForm1.Button2Click(Sender: TObject);
const n=3; m=4; //іменовані константи
var i, j, k: integer; //змінні цілого типу
    mas: array [1..n,1..m] of integer; //двовимірний масив
begin
    Randomize; //генерування різних випадкових чисел
    for i:=1 to n do //зовнішній цикл створення масиву
    for j:=1 to m do //внутрішній цикл створення масиву
    begin
        //елементу масива присвоюється випадкове число
        mas[i,j]:=random(21);
        //значення елемента масиву виводиться в клітинки StringGrid1
        StringGrid1.Cells[j,i]:=IntToStr(mas[i,j]);
    end; //закриваються операторні дужки
    k:=0; //початкове значення кількості чисел
    for i:=1 to n do //зовнішній цикл обчислення кількості чисел
    for j:=1 to m do //внутрішній цикл обчислення кількості чисел
    if mas[i,j]<10 then k:=k+1; //перевірка й обчислення кількості чисел
    Label1.Caption:='Кількість чисел='+FloatToStr(k); //виведення результату
    end;
end.

```

Рис. 10. Програма обчислення кількості заданих значень у масиві

Рядки	1 стовпець	2 стовпець	3 стовпець	4 стовпець
1 рядок	12	2	16	19
2 рядок	19	17	7	3
3 рядок	4	12	2	9

Кількість чисел=6 Виконати

Рис. 11. Результат обчислення кількості заданих значень у масиві

- Пошук мінімального і максимального значень у двовимірному масиві

Сутність алгоритму пошуку максимального (мінімального) значення елемента двовимірного масиву полягає в тому, що спочатку максимальним (мінімальним) обирається значення елемента першого рядка першого стовпця. Це максимальне (мінімальне) значення послідовно порівнюється зі значеннями всіх елементів масиву. Кожного разу, коли поточне значення елемента більше за максимальне (менше від мінімального), поточне присвоюється максимальному (мінімальному). Процес порівняння елементів продовжується до останнього елемента масиву.

Розглянемо алгоритм пошуку максимального і мінімального значень елемента в масиві $mas[n, m]$. Значення максимального елемента зберігається у змінній max , а мінімального — у змінній min .



Поганий програміст не тільки не виконує свою роботу, але й створює проблеми, які доводиться вирішувати іншим.

Крок 1	Увести масив $mas[n,m]$	Крок 9	$min:=mas[i,j]$
Крок 2	$max:=mas[1,1]$	Крок 10	$j:=j+1$
Крок 3	$min:=mas[1,1]$	Крок 11	Якщо $j \leq m$, то крок 6, інакше — крок 12
Крок 4	$i:=1$	Крок 12	$i:=i+1$
Крок 5	$j:=1$	Крок 13	Якщо $i \leq n$, то крок 5, інакше — крок 14
Крок 6	Якщо $mas[i,j] > max$, то крок 7, інакше — крок 8	Крок 14	Виведення max і min
Крок 7	$max:=mas[i,j]$	Крок 15	Кінець
Крок 8	Якщо $mas[i,j] < min$, то крок 9, інакше — крок 10		



Приклад 9. На рис. 12 зображено програму реалізації алгоритму. У програмі за допомогою клавіатури у 3 рядки і 4 стовпці об'єкта `StringGrid1` вводяться цілі числа, з яких створюється масив $mas[3, 4]$. Значення

максимального і мінімального елементів виводиться за допомогою двох компонентів `TEdit`.

Результат виконання програми наведено на рис. 13.

```

procedure TForm1.Button1Click(Sender: TObject);
var i, j, max, min: integer; //змінні цілого типу
    mas: array [1..3,1..4] of integer; //двовимірний масив
begin
  for i:=1 to 3 do //зовнішній цикл створення масиву
    for j:=1 to 4 do //внутрішній цикл створення масиву
      //елементу масиву присвоюється значення клітинки StringGrid1
      mas[i,j]:=StrToInt(StringGrid1.Cells[j-1,i-1]);
      max:=mas[1,1]; //початкове значення змінної max
      min:=mas[1,1]; //початкове значення змінної min
      for i:=1 to 3 do //зовнішній цикл аналізу масиву
        for j:=1 to 4 do //внутрішній цикл аналізу масиву
          begin //відкриваються операторні дужки
            //чи більше значення поточного елемента за значення max
            if mas[i,j]>max then max:=mas[i,j];
            //чи менше значення поточного елемента від значення min
            if mas[i,j]<min then min:=mas[i,j];
          end; //закриваються операторні дужки
        //виведення максимального значення масиву
        Edit1.Text:='Максимальне значення='+IntToStr(max);
        //виведення мінімального значення масиву
        Edit2.Text:='Мінімальне значення='+IntToStr(min);
      end; end.

```

Рис. 12. Програма пошуку максимального і мінімального елементів

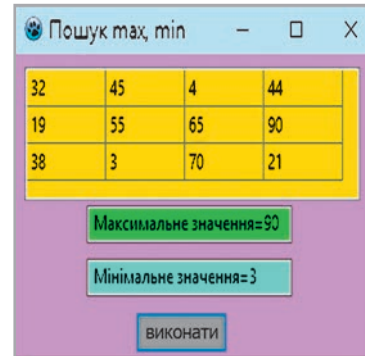


Рис. 13. Результат пошуку максимального і мінімального елементів



Запитання для перевірки знань

- 1 Що називають головною діагоналлю двовимірного масиву?
- 2 Накресліть блок-схему алгоритму обчислення загальної суми значень елементів двовимірного масиву.
- 3 Поясніть сутність алгоритму обчислення суми значень рядків двовимірного масиву.
- 4 Наведіть алгоритм обчислення суми значень елементів головної діагоналі двовимірного масиву.



Завдання для самостійного виконання

- 1 Із клавіатури за допомогою функції **InputBox()** у масив розмірністю $n \times m$ вводять додатні і від'ємні числа. Розробіть алгоритм і програму заміни додатних чисел одиницею, а від'ємних — нулем. Виведіть новий масив у поле об'єкта **StringGrid1**.
- 2 У таблиці наведено середні ціни реалізації продукції сільського господарства в окремих регіонах України у 2015 році (грн за тонну). Створіть масив із наведених даних. Розробіть алгоритм і програму обчислення середньої ціни продукції за цими регіонами.
- 3 Протягом п'яти днів від чотирьох господарств району на елеватор надходить зерно нового врожаю. Розробіть алгоритм і програму визначення, з якого господарства надійшло найбільше зернових.

Регіон	Зернові та зернобобові культури	Худоба та птиця (у живій масі)	Молоко та молочні продукти
Вінницька обл.	2781,7	25 725,5	4299,6
Житомирська обл.	2774,3	22 352,7	4368,2
Львівська обл.	2959,6	18 437,2	4330,4
Полтавська обл.	3368,7	22 788,3	4454,3
Чернігівська обл.	3046,1	21 877,5	4207,5

Практична робота № 21

Тема. Розробка алгоритмів з двовимірними масивами і їх реалізація у вигляді програм.

Завдання: розробити алгоритм і програму для реалізації завдання одного з наведених варіантів.

Обладнання: комп'ютер із встановленою програмою — середовищем програмування.

Варіант 1. Елементи двовимірного масиву розмірністю $n \times m$ набувають випадкових значень з діапазону від 0 до 20. Розробіть алгоритм і програму обчислення кількості чисел у масиві, менших від 15.

Завдання бажано виконувати у такій послідовності.

1. Розробити алгоритм реалізації завдання.

2. Розмістити на формі компоненти: TButton, TLabel і TStringGrid.
3. Установити необхідні властивості компонентів.
4. У програмі передбачити:
 - виведення значень елементів масиву в клітинки об'єкта StringGrid1;
 - для виведення результату використати об'єкт Label1.

Варіант 2. Відомі рейтинги восьми марок 3-річних автомобілів за два роки:

Рік	Lexus	Porsche	Buick	Toyota	Honda	Chevrolet
2015	1	6	2	3	5	10
2016	1	2	3	4	7	6

Із таблиці видно, що найвищий рейтинг як у 2015, так і у 2016 роках має марка автомобіля Lexus. Розробіть алгоритм і програму визначення середнього рейтингу кожної марки за два роки і розмістіть назви марок у порядку зменшення їх рейтингу.

Після розробки алгоритму бажано дотримуватися такого порядку виконання роботи.

1. Із клавіатури за допомогою функції InputBox() увести дані таблиці в клітинки об'єкта StringGrid1.

2. На основі даних, що містяться в об'єкті StringGrid1, створити двовимірний масив рейтингів.
3. Також на основі даних об'єкта StringGrid1 створити одновимірний масив марок автомобілів.
4. Опрацювати двовимірний масив, а саме: обчислити середнє значення рейтингів, із яких створити одновимірний масив за два роки.
5. Упорядкувати одновимірний масив рейтингів у порядку їх зростання, відповідно до яких розмістити у масиві марки автомобілів.
6. Вивести масив назв марок автомобілів, наприклад, у клітинки об'єкта ListBox1.

Хід роботи

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Виберіть один із варіантів завдань за вказівкою вчителя або самостійно. (У разі успішного виконання одного варіанта, виконайте інший).
2. Проаналізуйте умову задачі, побудуйте інформаційну модель.
3. Розробіть алгоритм і реалізуйте його у середовищі програмування за наведеним планом.
4. Виконайте програму. Проаналізуйте отриманий результат.
- **Зробіть висновок:** які компоненти можуть бути використані у цій роботі.

Практична робота № 22

Тема. Класичні алгоритми для роботи з двовимірними масивами, їх реалізація у вигляді програм.

Завдання: розробити алгоритм і програму для реалізації завдання одного з наведених варіантів.

Обладнання: комп'ютер із встановленою програмою — середовищем програмування.

Варіант 1. У шаховому турнірі брали участь 6 шахістів. Нижче у таблиці наведено результати турніру.

№ з/п	Прізвище	1	2	3	4	5	6
1	Таран	X	0	1	½	0	0
2	Колот	1	X	½	½	1	1
3	Пахов	0	½	X	0	½	0
4	Носов	½	½	1	X	1	½
5	Середа	1	0	½	0	X	0
6	Литвин	1	0	1	½	1	X

Розробити алгоритм і програму визначення прізвищ учасників, які виконали норматив першого розряду (для присвоювання першого розряду потрібно набрати не менше 50 % від максимальної можливої кількості очок).

Доцільно дотримуватися такого порядку виконання роботи.

1. Розробити алгоритм розв'язування.

Варіант 2. У таблиці наведено результати оцінювання учнів з окремих предметів.

№ з/п	Прізвище	Математика	Історія	Інформатика	Хімія
1	Раков	8	11	12	8
2	Юшко	10	11	10	9
3	Сахно	12	10	11	10
4	Темків	9	11	12	11
5	Смирнова	11	12	11	10

Визначте, з якого предмета учні мають найвищий середній бал успішності. Роботу рекомендується виконувати у такій послідовності.

1. Розробити алгоритм розв'язування.
2. Розмістити на формі компоненти: TLabel, TButton, TStringGrid.

Хід роботи

Під час роботи за комп'ютером дотримуйтесь правил безпеки.

1. Виберіть один із варіантів завдань за вказівкою вчителя або самостійно. (Можливе виконання завдання командою з двох-трьох учнів).
 2. Проаналізуйте умову задачі, побудуйте інформаційну модель.
 3. Розробіть алгоритм і реалізуйте його в середовищі програмування за наведеними вказівками.
 4. Виконайте програму. Проаналізуйте отриманий результат.
- **Зробіть висновок:** чи можна виконати цю роботу без операторів циклу.

Розділ 12. ДОПОМІЖНІ АЛГОРИТМИ

12.1. Загальні відомості про допоміжні алгоритми і підпрограми

Пригадайте, які функції, вбудовані у мову, ви вже використовували у програмах. Які переваги має використання таких функцій? Чи може, на вашу думку, користувач сам розробити необхідну функцію, якщо у самій мові вона відсутня?



! **Допоміжний алгоритм** — це заздалегідь розроблений алгоритм, до якого можна звертатися неодноразово з основного алгоритму в процесі розв'язування конкретної задачі.

Прикладом допоміжного алгоритму є алгоритм обчислення середньої заробітної плати працівників фірми, який використовується в алгоритмі оцінки ефективності її функціонування. Допоміжним може бути алгоритм пошуку мінімального числа в одновимірному масиві у процесі розробки алгоритму пошуку мінімального числа у двовимірному масиві.

✓ У мовах програмування допоміжні алгоритми реалізуються за допомогою підпрограм.

Із підпрограмами ви вже зустрічалися раніше у процесі розробки програм обчислення значень арифметичних та інших виразів, не акцентуючи увагу саме на цьому терміні. Наприклад, використовувалися функції обчислення кореня квадратного — $\text{sqrt}(x)$, абсолютного значення виразу $\text{abs}(x)$ та ін. Функції можуть використовуватися в одній програмі багаторазово у різних місцях цієї програми. Перелічені та інші функції і є підпрограмами. Але вони **вбудовані** в саму мову програмування і тому їх називають **стандартними підпрограмами**. Для їх використання достатньо вказати ім'я функції і значення аргументу.

Крім стандартних підпрограм сучасні мови програмування надають можливість програмісту самому **створювати підпрограми**. Головні переваги використання підпрограм наведено на [рис. 1](#).

Сутність використання підпрограм продемонстровано на [рис. 2](#).

Головні переваги використання підпрограм

- Суттєва економія трудовитрат програмістів за рахунок того, що підпрограма розробляється один раз, а використовується багаторазово у різних програмах
- Скорочення довжини програмного коду, а отже, зменшення обсягу пам'яті для її збереження за рахунок того, що в програмі не потрібно кожного разу наводити повний текст коду підпрограми (достатньо в основній програмі вказати її ім'я й аргументи)

Рис. 1. Переваги використання підпрограм

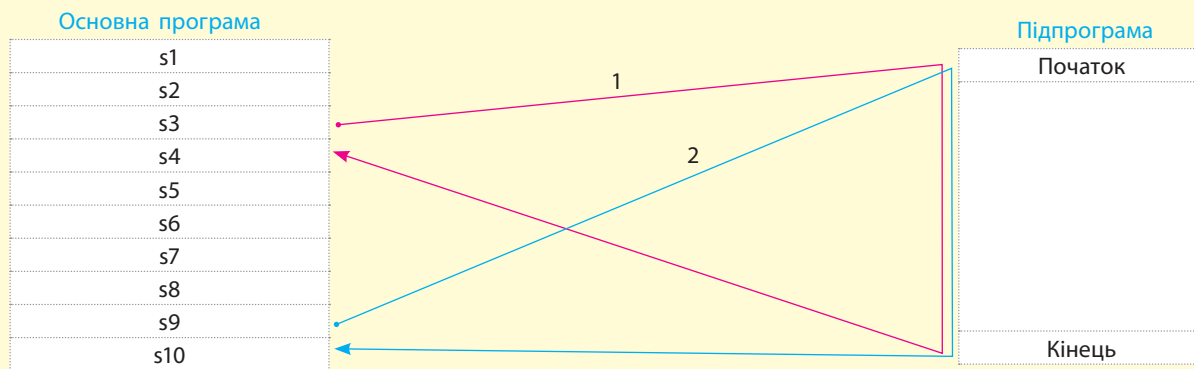


Рис. 2. Організація звернення до підпрограм

Підпрограми користувача обов'язково слід описати у **розділі опису**, дотримуючись встановленого синтаксису. Опис містить **заголовок підпрограми** і **тіло підпрограми**. У заголовку оголошується ім'я підпрограми й у круглих дужках параметри (якщо вони є).

Тіло підпрограми слідує безпосередньо за її заголовком. Під час звернення до підпрограм їм передаються конкретні дані для опрацювання.

Результат обчислення підпрограмою повертається в основну програму.

Із рис. 2 видно, що основна програма містить оператори s1–s10. Перший раз звернення до підпрограми здійснюється з оператора s2. Після завершення виконання підпрограми управління передається оператору основної програми, розташованому безпосередньо за оператором s2, тобто оператору s3. Другий раз звернення до підпрограми здійснюється з оператора s9, а повертається управління до оператора s10 основної програми. З основної програми можна неодноразово звертатися до багатьох підпрограм.



Отже, **підпрограма** — це логічно завершений фрагмент програми з наданим іменем, який має строго визначене призначення і до якого можна неодноразово звертатися з основної програми.

У кожній мові програмування існують свої правила розробки підпрограм користувача, звернення до них, отримання результатів, виходу з підпрограм і повернення в основну програму.

Одна підпрограма може містити опис інших підпрограм. Це означає, що з однієї підпрограми можна звертатися до декількох інших підпрограм.



Змінні, константи й типи даних, які використовуються у підпрограмі, можуть бути оголошені як в основній програмі, так і в підпрограмі. Якщо вони оголошені в основній програмі, то доступні у будь-якій підпрограмі і їх називають **глобальними**. Змінні, константи й типи, оголошені у підпрограмі, доступні лише цій підпрограмі і їх називають **локальними**.

Кожна змінна, константа й тип обов'язково повинні бути оголошені до моменту їх використання. Областю їх дії є та підпрограма, у якій вони оголошені. У програмі можуть використовуватися глобальні та локальні змінні з однаковими іменами. У цьому випадку вони сприймаються як різні змінні. Однак для запобігання плутанині краще цього уникати.



Катерина Логвинівна Ющенко написала перші програми для першої ЕОМ, створеної у НАН України під керівництвом С. О. Лебедева. За 40 років роботи в Інституті кібернетики ім. В. М. Глушкова НАН України нею створена широковідомо в Україні і за кордоном наукова школа теоретичного програмування.

Передавання аргументів формальним параметрам здійснюється у процесі виклику підпрограми. Наприклад, у підпрограмі обчислення квадратного кореня — $\text{sqrt}(x)$ змінна x — це формальний параметр. Після виклику цієї підпрограми змінна x набуде конкретного значення, наприклад 23, тобто воно є аргументом, з якого слід обчислити квадратний корінь.

Формальних параметрів у підпрограмі може бути декілька. Однак їхня кількість, тип і порядок розміщення повинні однозначно відповідати кількості, типам і порядку розміщення аргументів. Якщо ця вимога не виконується, транслятор повідомить про наявність синтаксичної помилки.

У середовищі Lazarus існують два різновиди підпрограм: **процедури** і **функції**. Принципова відмінність між ними полягає в тому, що результатом виконання підпрограми-функції завжди є один результат-значення, а результатом виконання підпрограми-процедури можуть бути декілька значень, одне значення або жодного. В основній програмі процедури і функції оголошуються між розділом оголошення змінних та операторною частиною.

Структура процедур і функцій аналогічна структурі програми. Як і основна програма, вони мають складатися із заголовка й операторів, розміщених між ключовими словами `begin` і `end`. Цю групу операторів називають **тілом процедури (функції)**. Процедура (функція) завершується словом `end`, після якого ставиться символ крапки з комою.

У підпрограмах розрізняють формальні та фактичні параметри. Змінні, вказані в заголовку підпрограми, називають **формальними параметрами**, або просто **параметрами**. Змінні, які вказані в операторі виклику підпрограми, називають **фактичними параметрами**, або **аргументами**.



Запитання для перевірки знань

- 1 Що називають допоміжним алгоритмом?
- 2 Які переваги дає використання підпрограм?
- 3 Які змінні називають глобальними й локальними?
- 4 Поясніть область дії локальних і глобальних змінних.
- 5 Поясніть сутність формальних і фактичних параметрів.
- 6 Які різновиди підпрограм існують у середовищі Lazarus?
- 7 Яка основна відмінність підпрограм-функцій від підпрограм-процедур?
- 8 Наведіть означення підпрограми.



Завдання для самостійного виконання

- 1 Основна програма займає 5 Кбайтів пам'яті, а підпрограма — 0,75 Кбайта. Із різних місць основної програми здійснюється звернення до підпрограми 6 разів. Визначте, на скільки орієнтовно більше пам'яті займала б ця програма за умови відсутності підпрограми.
- 2 Розроблено програму, яка займає 4 Кбайти пам'яті. Потім виникла необхідність у цій програмі тричі обчислити квадрат числа і двічі — модуль числа. Який орієнтовно обсяг пам'яті буде займати така програма, якщо кожного разу обчислювати ці значення безпосередньо у цій програмі?

12.2. Процедури



Що називають допоміжним алгоритмом і підпрограмою? Поясніть загальний принцип звернення до підпрограм. Які змінні називають глобальними й локальними? Поясніть сутність формальних і фактичних параметрів. Які різновиди підпрограм існують у середовищі Lazarus?

У структурі опису процедури обов'язковими є ключове слово **procedure**, ім'я процедури, ключові слова **begin**, **end** і тіло процедури. Інші складові використовуються за необхідності. Найчастіше процедура оголошується у розділі опису основної програми або іншої процедури.

Загальна структура опису процедури має такий вигляд:

```
procedure <ім'я процедури>[(список формальних параметрів)];
label <список міток>;
const <список констант>;
type <список типів>;
var <список змінних>;
begin
    //тіло процедури
end;
```

Список формальних параметрів — це перелік імен змінних і їхніх типів у заголовку процедури. Імена параметрів відокремлюються одне від одного комами, а перед іменем типу ставиться символ двокрапки. Параметри можуть мати будь-який тип, у тому числі й структурований. Якщо декілька параметрів мають однаковий тип, його можна вказати для всієї групи параметрів. У цьому випадку групи параметрів різних типів відділяються одна від одної символом крапки з комою.

Із синтаксису заголовка видно, що процедура може і не мати формальних параметрів. У такому разі в заголовку вказується лише її ім'я. Подібні процедури називають **процедурами без параметрів**. Процедури, які мають список формальних параметрів, називають **процедурами з параметрами**.

В оголошенні можуть використовуватися і var-параметри.

Службове слово var перед змінними у списку формальних параметрів свідчить про те, що ці змінні можуть набувати нові значення в процесі роботи підпрограми. Їх називають **параметрами-змінними**, а змінні без службового слова var — **параметрами-значеннями** (приклад 2).

Для звернення до процедури використовується оператор виклику такої структури:

```
ім'я процедури [(список фактичних параметрів)];
```

Фактичні параметри у списку фактичних параметрів відокремлюються один від одного комою. Якщо в описі процедури формальні параметри відсутні, то в разі її виклику фактичні параметри також повинні бути відсутні.

Нагадаємо, що між формальними і фактичними параметрами має бути повна відповідність. Це означає, що кількість формальних і фактичних параметрів, а також послідовність

Приклад 1.

Процедура без параметрів:
procedure minor;

Процедура з параметрами:
procedure nash_1 (a, b: real; i, j, k: integer);

Приклад 2. В оголошенні
procedure nash_2 (x,y: real; var z: real);
x, y — це параметри-значення,
z — параметр-змінна.

їх розміщення повинні бути однаковими, а тип кожного фактичного параметра має збігатися з типом відповідного формального параметра. Як фактичні параметри можуть бути використані константи або вирази.

У середовищі Lazarus допускається будь-який рівень вкладеності процедур і функцій. Так, процедура, оголошена в основній програмі, може містити оголошення внутрішніх процедур або функцій, які, у свою чергу, теж можуть містити оголошення підпрограм і т. д. Область дії імен, описаних у процедурі або функції, поширюється на всі вкладені процедури чи функції.



Приклад 2. Обчислити об'єми двох циліндрів за відомими радіусами (r) і висотами (h) за умови, що r , h — цілі числа. Об'єм циліндра збережемо у змінній v .

Алгоритм обчислення об'єма циліндрів з використанням допоміжного алгоритму може бути таким.

Основний алгоритм	
Крок 1	Увести змінні x_1 , x_2 , x_3 , x_4
Крок 2	$r:=x_1$
Крок 3	$h:=x_2$
Крок 4	Звернення до кроку 1 допоміжного алгоритму
Крок 5	Виведення v (об'єм першого циліндра)
Крок 6	$r:=x_3$
Крок 7	$h:=x_4$
Крок 8	Звернення до кроку 1 допоміжного алгоритму
Крок 9	Виведення v (об'єм другого циліндра)
Крок 10	Кінець

Допоміжний алгоритм	
Крок 1	$v:=3.14*r*r*h$
Крок 2	Повернення до основного алгоритму (до кроку, що слідує за кроком, з якого було звернення до допоміжного алгоритму)

Для реалізації алгоритму розмістимо на формі 4 компоненти TEdit, 2 компоненти TLabel і компонент TButton. Встановимо такі значення властивості Text: об'єкта Edit1 — 2, об'єкта Edit2 — 3, об'єкта Edit3 — 4, об'єкта Edit4 — 5. Об'єкти Label1 і Label2 використаємо для виведення результатів обчислення об'ємів циліндрів. Для об'єкта Button1 обираємо подію OnClick. Для обчислення об'єму циліндра

використаємо процедуру obem і глобальні змінні. Результат обчислення об'єму циліндра надається змінній v . Звернення до процедури здійснюється з основної програми двічі. Варіант програмного коду реалізації алгоритму зображено на рис. 1.

Результат виконання програми подано на рис. 2.

```

procedure TForm1.Button1Click(Sender: TObject);
var r,h:integer; //глобальні змінні цілого типу
v:real; //глобальна змінна дійсного типу
procedure obem; //процедура обем без параметрів
begin //початок тіла процедури обем
v:=3.14*r*r*h; //обчислення об'єму циліндра
end; //кінець процедури обем
begin //початок основної програми
r:=StrToInt(Edit1.Text); //значення радіуса першого циліндра
h:=StrToInt(Edit2.Text); //значення висоти першого циліндра
obem; //перше звернення до процедури обем
//виведення об'єму першого циліндра
Label1.Caption:='Першого циліндра='+FloatToStr(v);
r:=StrToInt(Edit3.Text); //значення радіуса другого циліндра
h:=StrToInt(Edit4.Text); //значення висоти другого циліндра
obem; //друге звернення до процедури обем
//виведення об'єму другого циліндра
Label2.Caption:='Другого циліндра='+FloatToStr(v);
end;
end.

```

Рис. 1. Програмний код з процедурою без параметрів

На рис. 3 зображено ще один варіант програмного коду реалізації того самого алгоритму. У цьому варіанті процедура обем міститься за межами обробника події об'єкта Button1.

```

var
Form1: TForm1;
r,h:integer; //глобальні змінні цілого типу
v:real; //глобальна змінна дійсного типу
implementation
{$R *.1fm}
{ TForm1 }
procedure obem; //процедура обем без параметрів
begin //початок тіла процедури обем
v:=3.14*r*r*h; //обчислення об'єму циліндра
end; //кінець процедури обем
procedure TForm1.Button1Click(Sender: TObject);
begin //початок основної програми
r:=StrToInt(Edit1.Text); //значення радіуса першого циліндра
h:=StrToInt(Edit2.Text); //значення висоти першого циліндра
obem; //перше звернення до процедури обем
//виведення об'єму першого циліндра
Label1.Caption:='Першого циліндра='+FloatToStr(v);
r:=StrToInt(Edit3.Text); //значення радіуса другого циліндра
h:=StrToInt(Edit4.Text); //значення висоти другого циліндра
obem; //друге звернення до процедури обем
//виведення об'єму другого циліндра
Label2.Caption:='Другого циліндра='+FloatToStr(v);
end; end.

```

Рис. 3. Програма з процедурою без параметрів за межами обробника подій

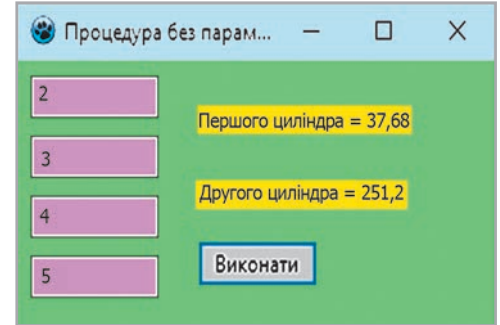


Рис. 2. Результати обчислення об'єму двох конусів

Разом із тим, вона розміщена вище за те місце програмного коду, де здійснено звернення до цієї процедури.

Процедури без параметрів мають недолік, який полягає в тому, що кожного разу перед зверненням до процедури необхідно виконати певну кількість операторів присвоювання. Ці оператори надають змінним, які використовуються у процедурі, певних значень. Для того щоб не виконувати згадані присвоювання, необхідно застосувати процедуру з параметрами. Структуру процедури з параметрами було наведено вище.

Як параметри-змінні у підпрограму слід передавати ті величини, що мають модифікуватися в операторах підпрограми, а як параметри-значення — ті величини, яким у підпрограмі не присвоюється жодних значень.



Лідерами в рейтингу мов програмування на кінець 2016 року за даними аналітичної компанії RedMonk є такі: JavaScript, Java, PHP, Python, C#, C++, Ruby.



Приклад 3. Розробити програму обчислення об'єму двох циліндрів за відомими висотою і радіусом основи з використанням процедури з параметрами. До процедури обчислення об'єму циліндра величину v (шуканий об'єм) передамо як параметр-змінну, а величини r і h (радіус і висоту циліндра) — як параметри-значення.

Варіант програми обчислення об'єму циліндра зображено на рис. 4.

Результат виконання програми подано на рис. 5.

На рис. 6 наведено програму, яка реалізує ту саму функцію, що й попередня програма, процедура з параметрами міститься за межами обробника подій.

```
var
  Form1: TForm1;
var res1, res2: real; //глобальні змінні дійсного типу
implementation
  {$R *.lfm}
  { TForm1 }
  procedure TForm1.Button1Click(Sender: TObject);
  procedure obem (r, h:integer; var res:real); //заголовок процедури обем
  begin
    //початок тіла процедури
    res := 3.14*r*r*h; //обчислення об'єму циліндра
  end;
  //кінець тіла процедури
  begin
    //початок основної програми
    //перше звернення до процедури обем
    obem(StrToInt(ListBox1.Items[0]), StrToInt(ListBox1.Items[1]), res1);
    //виведення об'єму першого циліндра
    Label1.Caption:='Першого циліндра =' +FloatToStr(res1);
    //друге звернення до процедури обем
    obem(StrToInt(ListBox1.Items[2]), StrToInt(ListBox1.Items[3]), res2);
    //виведення об'єму другого циліндра
    Label2.Caption:='Другого циліндра =' +FloatToStr(res2);
  end;
end.
```

Рис. 4. Програма з процедурою з параметрами усередині обробника подій

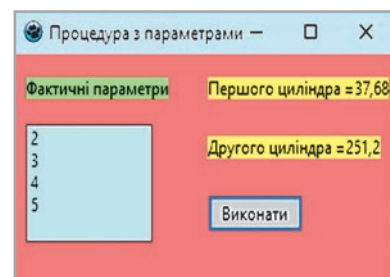


Рис. 5. Результат виконання програми з процедурою з параметрами



```

var
  Form1: TForm1;
var res1, res2:real; //глобальні змінні дійсного типу
implementation
procedure обем (r,h:integer; var res:real);//заголовок процедури обем
begin
  //початок тіла процедури
  res:=3.14*r*r*h; //обчислення об'єму циліндра
end; //кінець тіла процедури

{$R *.lfm}
{ TForm1 }
procedure TForm1.Button1Click(Sender: TObject);
begin
  //початок основної програми
  //перше звернення до процедури обем
  обем (StrToInt(ListBox1.Items[0]), StrToInt(ListBox1.Items[1]), res1);
  //виведення об'єму першого циліндра
  Label1.Caption:='Першого циліндра='+FloatToStr(res1);
  //друге звернення до процедури обем
  обем(StrToInt(ListBox1.Items[2]), StrToInt(ListBox1.Items[3]), res2);
  //виведення об'єму другого циліндра
  Label2.Caption:='Другого циліндра='+FloatToStr(res2);
end;
end.

```

Рис. 6. Програма з процедурою з параметрами за межами обробника подій



Запитання для перевірки знань

- 1 Яка інформація міститься в заголовку процедури?
- 2 Які різновиди формальних параметрів використовуються у процедурах?
- 3 Як здійснюється виклик процедур?
- 4 Які компоненти є обов'язковими в описі процедури?
- 5 Наведіть правила запису формальних параметрів процедури.
- 6 Наведіть загальну структуру опису процедури.
- 7 Яка різниця між параметрами-змінним і параметрами-значеннями?
- 8 Як пов'язані формальні і практичні параметри процедури?



Завдання для самостійного виконання

- 1 Розробіть алгоритм із використанням допоміжного алгоритму обчислення повної поверхні двох діжок із відомими радіусами і висотами. Реалізуйте алгоритм із використанням процедури без параметрів.
- 2 Складіть програму з використанням підпрограми, яка визначала б день тижня за його номером (1 — понеділок, 2 — вівторок і т. д.).
- 3 У банк поклали s грн під k % річних. Розробіть алгоритм і програму обчислення накопиченої суми через n років. У програмі використовуйте дві процедури без параметрів. За допомогою першої процедури вводяться вихідні дані, а за допомогою другої — обчислюється і виводиться накопичена сума.

12.3. Функції

Яку загальну структуру має процедура? Які складові цієї структури є обов'язковими? Поясніть сутність термінів «параметри-змінні» та «параметри-значення». Як здійснюється звернення до процедур?



Загальний опис функції складається із заголовка функції, розділу опису, тіла функції і має таку структуру:

```
function <ім'я функції>(<список формальних параметрів>):
<тип результату>;
label <список міток>;
const <список констант>;
type <список типів>;
var <список змінних>;
begin
    //тіло функції
end;
```

Як і в процедурі, обов'язковим тут є заголовок і тіло функції. Заголовок починається службовим словом **function** (приклад 1). Список формальних параметрів функції відповідає списку формальних параметрів процедури. Він містить імена формальних параметрів і їхні типи. Тип результату — це тип значення, що повертає функція. Повертатися можуть скалярні значення цілого, дійсного, логічного, символічного і посилального типів.

У тілі функції має міститися хоча б один оператор присвоєння, у лівій частині якого вказується ім'я даної функції. Так результат виконання функції повертається у програму, з якої цю функцію було викликано. Наприклад:

```
function func_1(a1, a2: integer): real;
begin
    func_1:=(2*a1-3*a2)/3;
end;
```

Розглянемо приклади використання функцій у програмах.



Приклад 2. Розробити алгоритм і програму обчислення значення виразу $(a-b)\sqrt{a^2+b^2}$ для двох пар значень змінних a і b з використанням підпрограми-функції.

Алгоритм розв'язування цієї задачі можна подати так.

У структурі опису функції обов'язковими є ключове слово **function**, ім'я функції, ключові слова **begin**, **end** і тіло функції. Інші складові використовуються за необхідності. Найчастіше функція оголошується у розділі опису основної програми або іншої підпрограми.



Приклад 1. Оголошення функцій:

- 1) **function** min (a,b: real; var c: real): real;
- 2) **function** ab (a: integer): char;
- 3) **function** con (var a,b: integer): integer;



Основний алгоритм		Допоміжний алгоритм	
Крок 1	Увести значення x_1 , y_1 , x_2 , y_2	Крок 1	$p := \sqrt{a^2 + b^2}$
Крок 2	$a := x_1$; $b := y_1$	Крок 2	$fun_1 := p * (a - b)$
Крок 3	Звернення до кроку 1 допоміжного алгоритму	Крок 3	Повернення до основного алгоритму (до кроку, що слідує безпосередньо за кроком виклику)
Крок 4	$y := fun_1$		
Крок 5	Виведення y	<p><i>Примітка:</i> Значення виразу $\sqrt{a^2 + b^2}$ збережено у локальній змінній p.</p>	
Крок 6	$a := x_2$; $b := y_2$		
Крок 7	Звернення до кроку 1 допоміжного алгоритму		
Крок 8	$z := fun_1$		
Крок 9	Виведення z		
Крок 10	Кінець		

Для реалізації алгоритму на формі розміщено об'єкти, наведені у таблиці.

Назва об'єкта	Призначення об'єкта	Назва об'єкта	Призначення об'єкта
Edit1	Висвітлення повідомлення Для першого звернення	Label6	Виведення результату z
Edit2	Висвітлення повідомлення Для другого звернення	Label7	Поява повідомлення $a =$ при першому зверненні
Label1	Введення значення x_1	Label8	Поява повідомлення $b =$ при першому зверненні
Label2	Введення значення y_1	Label9	Поява повідомлення $a =$ при другому зверненні
Label3	Введення значення x_2	Label10	Поява повідомлення $b =$ при другому зверненні
Label4	Введення значення y_2	Button1	Встановлення події OnClick
Label5	Виведення результату y	Label5	Виведення результату y

```

procedure TForm1.Button1Click(Sender: TObject);
var y, z:real; //глобальні змінні
function fun_1 (a, b:integer): real; //заголовок функції
var p:real; //локальна змінна
begin //початок тіла функції
    p:=sqrt(a*a+b*b); //обчислення кореня квадратного
    fun_1:=p*(a-b); //обчислення арифметичного виразу
end; //кінець тіла процедури
begin //початок основної програми
    //перше звернення до функції
    y:=fun_1(StrToInt(Label1.Caption), StrToInt(Label2.Caption));
    //виведення результату після першого звернення до функції
    Label5.Caption:='Результат 1-го звернення'+FloatToStr(y);
    //друге звернення до функції
    z:=fun_1(StrToInt(Label3.Caption), StrToInt(Label4.Caption));
    //виведення результату після другого звернення до функції
    Label6.Caption:='Результат 2-го звернення'+FloatToStr(z);
end;
end.

```

Рис. 1. Програмний код із функцією

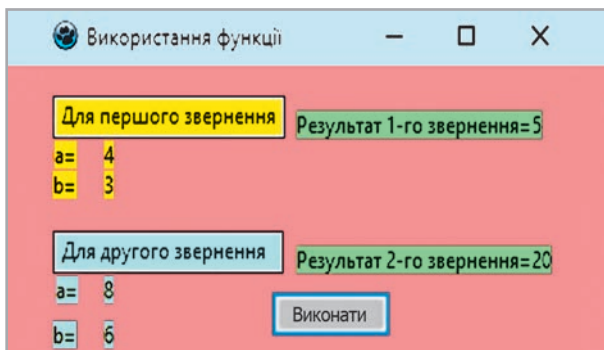


Рис. 2. Результати виконання програми з двома зверненнями до функції

У тілі функції може бути оператор, який присвоює значення, що повертається, стандартній змінній Result. Цю змінну оголошувати не потрібно, вона вже є в кожній функції. Змінній Result можна присвоювати значення вказаного у заголовку типу.

Приклад 3. Відомі значення довжини трьох лінійних відрізків: x , y і z . Потрібно розробити алгоритм і програму визначення: чи можуть дані відрізки бути сторонами трикутника. Якщо можуть, обчислити площу трикутника з цими сторонами.

Для розв'язання цієї задачі розробимо основний алгоритм і два допоміжні: у першому

допоміжному алгоритмі будемо визначати умову можливості існування трикутника, а у другому — обчислювати площу трикутника. Фактичні довжини відрізків позначимо змінними a , b , c , а формальні — відповідно змінними x , y і z .

Алгоритм розв'язування задачі можна подати так.



Основний алгоритм		Перший допоміжний алгоритм	
Крок 1	Увести значення a, b, c	Крок 1	Якщо $(x+y>z)$ and $(x+z>y)$ and $(y+z>x)$, то $umova:=true$, інакше $umova:=false$
Крок 2	$x:=a; y:=b; z:=c$	Крок 2	Перейти до кроку 4 основного алгоритму
Крок 3	Звернення до кроку 1 першого допоміжного алгоритму		
Крок 4	Якщо $umova=true$, то крок 1 другого допоміжного алгоритму, інакше — крок 5 основного алгоритму		
Крок 5	$plosha:='Не трикутник'$	Крок 1	$p:=(x+y+z)/2$
Крок 6	Виведення $plosha$	Крок 2	$plosha:=sqrt(p*(p-x)*(p-y)*(p-z))$
Крок 7	Кінець	Крок 3	Перейти до кроку 6 основного алгоритму

Для реалізації алгоритму розмістимо на формі компонент TEdit, 7 компонентів TLabel і компонент TButton. Об'єкт Edit1 призначений для виведення повідомлення Сторони трикутника, об'єкт Label1 — для виведення повідомлення $x=$, об'єкт Label2 — для виведення повідомлення $y=$, об'єкт Label3 — для виведення повідомлення $z=$, об'єкт Label4 — для введення

значення змінної a , об'єкт Label5 — для введення значення змінної b , об'єкт Label6 — для введення значення змінної c , об'єкт Label7 — для виведення результату, об'єкт Button1 — для надання події OnClick.

Програму реалізації алгоритму подано на рис. 3. Результат виконання програми зображено на рис. 4.

```

var
  Form1: TForm1;
  a,b,c: integer; s: real;           //глобальні змінні
implementation
function plosha (x,y,z:integer):real; //заголовок функції plosha
var p: real; //локальна змінна
begin //початок тіла функції plosha
  p:=(x+y+z)/2; //півпериметр трикутника
  plosha:=sqrt(p*(p-x)*(p-y)*(p-z)); //площа трикутника
end; //кінець тіла функції plosha
function umova (x,y,z:integer): boolean; //заголовок функції umova
begin //початок тіла функції umova
  if (x+y>z) and (x+z>y) and (y+z>x) then umova:=true else umova:=false;
end; //кінець тіла функції umova
{$R *.lfm}
{ TForm1 }
procedure TForm1.Button1Click(Sender: TObject);
begin //початок основної програми
  a:=StrToInt(Label4.Caption); //присвоєння значення змінній a
  b:=StrToInt(Label5.Caption); //присвоєння значення змінній b
  c:=StrToInt(Label6.Caption); //присвоєння значення змінній c
  if umova (a,b,c) then begin s:=plosha (a,b,c); //звернення до
    Label7.Caption:='Площа='+FloatToStr(s) end //функції i
  else Label7.Caption:=('Не трикутник'); //виведення результату
end; end.

```

Рис. 3. Програмний код із двома функціями



Рис. 4. Результат виконання програмного коду з двома функціями

Іноколи процедури або функції здійснюють взаємне звернення одна до одної. У такому випадку використовується **випереджальний опис** (приклад 4).



Приклад 4.

```
procedure an_1 (x: real); forward; //випереджальний опис
procedure an_2 (x: integer); //початок процедури an_2
begin
  an_1 (5); //виклик процедури an_1
end;
procedure an_1; //початок процедури an_1
begin
  an_2 (7); //виклик процедури an_2
end;
```

Сутність випереджального опису полягає в тому, що використувана підпрограма може бути описана тільки заголовком, одразу за яким записується стандартна директива **forward**. Опис тексту такої програми може бути у будь-якому місці розділу описів без повторення списку формальних параметрів.

? Запитання для перевірки знань

- 1 Для чого використовується змінна **Result**?
- 2 Яку структуру має заголовок функції?
- 3 Як повертається результат функції у програмі, що його викликала?
- 4 Які компоненти в описі функції є обов'язковими?
- 5 Наведіть загальну структуру опису функції.
- 6 Наведіть приклад опису функції.

📁 Завдання для самостійного виконання

- 1 Розробіть програму обчислення значення виразу $\frac{(a^2 + b^2)}{(c^2 + d^2) + 3c}$ з використанням функції обчислення значення виразу $x^2 + y^2$.
- 2 Дано дві групи цілих чисел: **a1, a2, a3** і **b1, b2, b3**. Розробіть алгоритм і програму визначення мінімального числа у кожній групі, використовуючи підпрограму визначення мінімального числа з двох.
- 3 Відомі довжини всіх сторін трьох трикутників. Розробіть алгоритм обчислення загальної площі трикутників із використанням допоміжного алгоритму. Розробіть програму реалізації алгоритму з використанням функції.

12.4. Використання масивів як формальних параметрів підпрограм

У чому полягає принципова різниця в описі процедури і функції? Поясніть, як результат виконання функції повертається в основну програму. Досі як формальні параметри підпрограм розглядалися змінні скалярних типів. Чи можуть, на вашу думку, ними бути інші типи даних?



У підпрограмах формальними параметрами можуть бути не тільки змінні, але й одновимірні масиви. Масиви можуть бути звичайними, але частіше використовуються так звані відкриті масиви.

Приклад 1.

Оголошення відкритих масивів:

- 1) mas_1: array of integer;
- 2) mas_2: array of char;



Профільний IT-ресурс DOU.UA провів у 2017 році опитування щодо популярності мов програмування, в якому взяли участь 8 тисяч спеціалістів, із яких 90 % проживає в Україні. Найпопулярнішими виявилися такі мови: Java, JavaScript, C#, PHP, Python, C++.

Функцію **high** можна використовувати як параметр циклу:
for i:=0 to high(mas_1) do



Відкритими називають масиви, які оголошуються у розділах **var** або **type** без діапазону індексів, вказується лише тип елементів.

Відкриті масиви оголошуються за такою структурою:
var <ім'я масиву>: array of <тип>; (приклад 1).

Оскільки після оголошення відкритого масиву його розмір невідомий, то для нього не можна виділити пам'ять. Тому й використовувати такий масив на цьому етапі не можна.

Розмір відкритих масивів можна задавати і змінювати під час виконання програми.

Виділення пам'яті масиву здійснюється за допомогою функції **SetLength**, яка має таку структуру:

SetLength (<ім'я відкритого масиву>, <розмір масиву>);

Звільнення пам'яті, виділеної для відкритого масиву, здійснюється за допомогою команди:

<ім'я відкритого масиву>:=nil;

Отже, відкритий масив є не статичним (для статичного масиву виділяється пам'ять на весь час виконання програми), а фактично динамічним. Для нього виділяється пам'ять після визначення розмірності масиву, а звільнити її можна в будь-який час після опрацювання елементів масиву.

Нумерація елементів відкритого масиву завжди починається з нуля. За допомогою функції **high** можна визначити індекс останнього елемента масиву (**high=Length-1**).

Перший індекс відкритого масиву дорівнює нулю, а останній можна отримати за допомогою стандартної функції:

high <ім'я відкритого масиву>;

Параметром функції **high** є ім'я відкритого масиву, а результатом — індексний номер останнього елемента масиву.

Наприклад, якщо виділено пам'ять для 100 елементів відкритого масиву, то функція **high** поверне число 99 (оскільки індексація починається з 0, і, відповідно, індекс останнього елемента масиву дорівнюватиме 99).

Відкриті масиви найчастіше використовуються як параметри підпрограм (процедур або функцій). Разом із тим їх можна використовувати й у звичайних програмах.

Приклад 2. Розробити програму обчислення суми значень елементів одновимірного відкритого масиву.

На формі розміщено такі об'єкти: **Label1**, властивість **Caption** якого має значення Довжина масиву; **Label2**, властивість **Caption** якого має значення Значення елементів масиву; **Label3** призначений для виведення результату;

ListBox1 — для виведення значень елементів створеного масиву; **Button1** — для встановлення події **OnClick**. Програму обчислення суми значень елементів масиву подано на рис. 1. Масив формується випадковими числами.

Результат виконання програми зображено на рис. 2.

```

procedure TForm1.Button1Click(Sender: TObject);
var mas: array of integer; //оголошення відкритого масиву
    i, n, s: integer;      //змінні цілого типу
begin
  Randomize;              //генерування різних випадкових чисел
  n:=StrToInt(Edit1.Text); //довжина масиву
  SetLength(mas, n);      //виділення пам'яті під масив
  for i:=0 to high(mas) do //циклічне створення масиву
  begin
    mas[i]:=Random(20);    //елементу масиву присвоюється значення
  ListBox1.Items.Add(IntToStr(mas[i])); //виведення масиву
  end;
  s:=0;                   //початкове значення суми
  for i:=0 to high(mas) do //циклічне обчислення суми
  s:=s+mas[i];             //обчислення суми значень елементів масиву
  mas:=NIL;               //звільнення пам'яті, виділеної під масив
  Label3.Caption:='Сума='+IntToStr(s); //виведення суми
  end;
end.

```

Рис. 1. Програма з відкритим масивом

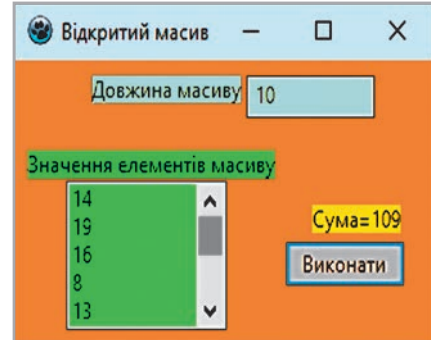


Рис. 2. Результат виконання програми з відкритим масивом

Як уже зазначалося, відкриті масиви зазвичай використовують для передавання у підпрограму масивів різних розмірів. Це дозволяє за допомогою тієї самої програми опрацювати масиви довільної довжини. Без використання відкритих масивів довелось би для кожного масиву іншої довжини розробляти власну програму.

Найчастіше у підпрограмах як формальні параметри використовують стандартні типи даних (integer, char та ін.). Але можна використовувати і нестандартні типи даних. У такому разі їх необхідно оголосити раніше оголошення підпрограм. Тип масиву оголошується за такою структурою:

```
type <тип масиву>=array [список індексів] of <тип елементів>;
```

Наприклад:

```
type skan = array [1..8] of byte;
```

Опис підпрограми, наприклад процедури, у якій як параметр використовується відкритий масив, має такий вигляд:

```
procedure <ім'я процедури> (<ім'я масиву>: <тип масиву>;
```

Наприклад:

```
procedure proc_1 (vas_3: skan);
```

Приклад 4. На рис. 3 наведено програму, в якій оголошено два масиви: mas1 і mas2. Формальними параметрами процедури proc є відкритий масив mas типу integer. Звернення до процедури з основної програми здійснюється двічі. Перший раз формальним параметрам процедури передаються значення масиву mas1 завдовжки 5, а у процесі другого звернення передаються значення масиву mas2 завдовжки 4. Під час першого звернення до процедури обчислюється кількість елементів масиву mas1, що дорівнює 4, а під час другого

звернення — кількість елементів масиву mas2, що також дорівнює 4. Отже, за допомогою однієї процедури опрацюються два різні за довжиною і змістом масиви, але їхні типи збігаються з типом відкритого масиву процедури. На формі розташовано об'єкти Label1 і Label2, за допомогою їхніх властивостей Caption здійснюється виведення результатів звернення до процедури. Для об'єкта Button1 обрано подію OnClick.

Результат виконання програми зображено на рис. 4.



Джеймс Гослінг, автор мови програмування Java, творець віконної системи NeWS, GoslingEmacs, один із розробників StarSeve. Наразі входить в команду відомого українського стартапа Jelastic як незалежний директор.


```

procedure TForm1.Button1Click(Sender: TObject);
var mas1:array [0..4] of integer=(4,4,3,4,5); //перший масив
    mas2:array [0..3] of integer=(5,4,4,3);    //другий масив
    m:integer;                                //глобальна змінна
//заголовок процедури, формальним параметром якої є масив
procedure proc (mas:array of integer);
var j:integer;                                //локальна змінна
begin
    //початок тіла процедури
    m:=0; //початкове значення кількості чисел, що відшукуються
    for j:=0 to high(mas) do //цикл аналізу масиву
        if mas[j]=4 then m:=m+1; //чи дорівнює значення елемента масиву 4
    end; //кінець тіла процедури
begin
    //початок тіла основної програми
    proc (mas1); //перше звернення до процедури
    Label1.Caption:='У першому масиві таких чисел='+IntToStr(m);
    proc (mas2); //друге звернення до процедури
    Label2.Caption:='У другому масиві таких чисел='+IntToStr(m);
end; //кінець тіла основної програми
end.

```

Рис. 3. Програма з процедурою, формальним параметром якої є масив

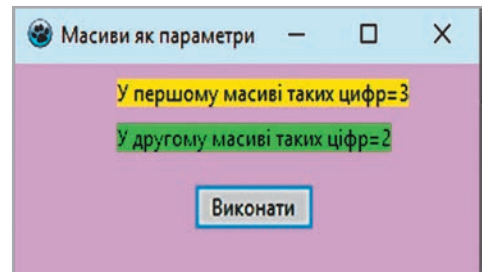


Рис. 4. Результат виконання процедури, формальним параметром якої є масив

Запитання для перевірки знань

- 1 Які масиви називають відкритими?
- 2 Для чого призначена функція `SetLength`?
- 3 Як можна звільнити пам'ять, виділену під відкритий масив?
- 4 За якою структурою оголошуються відкриті масиви?
- 5 Наведіть приклад оголошення відкритого масиву.
- 6 Як можна визначити максимальний індекс масиву?
- 7 Як викликається процедура, у якій параметрами є відкриті масиви?

Завдання для самостійного виконання

- 1 Розробіть програму для визначення різниці суми значень парних елементів одновимірного числового масиву і суми значень його непарних елементів. Використайте функцію, формальними параметрами якої є відкритий масив.
- 2 Розробіть програму пошуку мінімального числа у відкритому масиві. Кількість елементів масиву вводиться у процесі виконання програми. Масив формується випадковими цілими числами з діапазону від 0 до 75.
- 3 Розробіть програму з використанням процедури, формальним параметром якої є відкритий масив. За допомогою процедури обчислюється сума значень елементів трьох масивів різної розмірності. Елементи всіх трьох масивів набувають значень під час оголошень масивів.

12.5. Поняття рекурсії. Рекурсивні алгоритми

У всіх розглянутих раніше прикладах підпрограми викликалися з основної програми. А чи можуть, на ваш погляд, підпрограми викликати самі себе?



У програмуванні інколи використовують **рекурсивні підпрограми**. Виклик може виконуватися неодноразово. Рекурсивними підпрограмами можуть бути як процедури, так і функції.



Рекурсивною називають підпрограму, яка викликає сама себе.

Реалізація рекурсивних підпрограм заснована на структурі даних, що називається *стеком*. Важливо усвідомити, що в процесі кожного звернення до рекурсивної підпрограми для неї у стеці значень виділяється окремий фрагмент пам'яті. **Фрагментом** (або **блоком**) є певна кількість комірок пам'яті. У процесі першого звернення до підпрограми у вершину стека записується адреса команди основної програми, до якої потрібно повернутися після завершення всіх викликів рекурсивної підпрограми. У наступні блоки стека записуються значення фактичних параметрів процедури і дії, які потрібно виконати.

Зазначимо, що рекурсивні обчислення в багатьох випадках є неефективними і тому часто замінюються рекурентними обчисленнями. Метод рекурентного обчислення потребує меншого обсягу пам'яті й реалізується швидше за рекурсивний. Для свідомого розуміння такого висновку розглянемо детальніше сутність рекурентного і рекурсивного обчислень.

Нагадаємо, що рекурентне обчислення значення математичного виразу на кожному кроці здійснюється через його значення на попередньому кроці. Наприклад, *алгоритм обчислення значення виразу $y = x^n$* можна подати так.

Крок 1	Увести n, x	Крок 5	$i := i + 1$
Крок 2	$y := 1$	Крок 6	Якщо $i \leq n$, то крок 4, інакше — крок 7
Крок 3	$i := 1$	Крок 7	Виведення y
Крок 4	$y := y * x$	Крок 8	Кінець

Отже, після того як вираз $i \leq n$ із кроку 6 набуде значення `false`, виконання алгоритму завершується і змінна y буде містити результат обчислення. Для рекурсивного обчислення значення x^n (n — додатне ціле, x — може бути дійсне) потрібно обчислити значення x^{n-1} , оскільки $x^n = x^{n-1} \cdot x$. У свою чергу, для обчислення x^{n-1} потрібно обчислити x^{n-2} , оскільки



Макс Левчін (нар. 1975 р. в м. Києві) є автором багатьох стартапів, співзасновником та головним інженером PayPal, віце-президентом з розробки в компанії Google.

$x^{n-1} = x^{n-2} \cdot x$ і т. д. Процес обчислення слід припинити після обчислення x^0 , оскільки воно має значення 1. Але результат ще не отримано, отримано лише опорне значення функції. Для отримання кінцевого значення функції потрібно реалізувати зворотний процес обчислення, що міститься у стеку. Пояснимо сутність і необхідність цього процесу на прикладі обчислення значення виразу x^n . Рекурсивна функція його обчислення має такий зміст:

```
function st (x, n:integer): integer;
begin
  if n=0 then st:=1 else st:=st(x, n-1)*x;
end;
```

Ця рекурсивна підпрограма виконується у такій послідовності.

Крок 1	<p>Перший раз вона викликається з основної програми, наприклад, за допомогою команди <code>z=st(2, 3)</code>; (отже, <code>x=2</code> і <code>n=3</code>). У результаті цього виклику в стеку виділяється перший блок пам'яті (вершина стека) й управління передається умовному оператору.</p> <p>Нагадаємо, що у вершину стека записується адреса команди основної пам'яті, до якої слід повернутися після всіх звернень до процедури.</p> <p>Оскільки на цьому етапі вираз <code>n=0</code> набуває значення false, то виконується гілка else. Тут у виразі <code>st:=st(x, n-1)*x</code> робиться спроба обчислити значення <code>st(2,2)</code>, тобто в дійсності виконується звернення до самої функції <code>st</code> з новими значеннями параметрів. Тому обчислення переривається.</p>
Крок 2	<p>Викликається другий раз функція <code>st</code>, але вже з параметрами <code>st(2, 2)</code>. У стеку також виділяється наступний блок пам'яті, а в умовному операторі виконується гілка else і робиться спроба обчислити значення функції <code>st</code> з параметрами <code>st(2, 1)</code>.</p>
Крок 3	<p>Третій раз викликається функція <code>st</code> з параметрами <code>st(2, 1)</code>. У стеку також виділяється блок пам'яті і знову виконується гілка else з параметрами <code>st(2, 0)</code>.</p>
Крок 4	<p>Четвертий раз виклик виконується з параметрами <code>st(2, 0)</code>. У стеку виділяється четвертий блок пам'яті, але виконується гілка then, а функція <code>st</code> набуває значення 1. У результаті функція <code>st</code> рекурсивно більше не викликається і здійснюється повернення в основну програму, яка викликала цю підпрограму.</p>

У процесі повернення до основної програми виконуються такі дії. Зі стека вибирається опорне значення `st(2, 0)=1`, оскільки у стеку реалізується принцип «останній прийшов — перший вийшов». На наступному кроці обчислюється значення `st(2, 1)=1*2`, потім — `st(2, 2)=2*2`, нарешті — `st(2, 3)=2*4=8`. Це значення передається змінній `z` в основну програму.

Динаміку описаного процесу «заглиблення» і виходу зі стека для описаної функції наведено в таблиці.

Таблиця. ДИНАМІКА ВИКОНАННЯ РЕКУРСІЇ

Рівень рекурсії	Рекурсивне «заглиблення»	Рекурсивне повернення
0	Виклик $z:=st(2,3)$ ↓	→ Вихід $z:=8$
1	$st(2,3):=st(2,2)*2$ ↓	↑ $st(2,3):=2*4$ (8)
2	$st(2,2):=st(2,1)*2$ ↓	↑ $st(2,2):=2*2$ (4)
3	$st(2,1):=st(2,0)*2$ ↓	↑ $st(2,1):=1*2$ (2)
4	$st(2,0):=1$ →	

Класичним варіантом рекурсивної функції є обчислення факторіала заданого числа n , яке виконується за формулою $n!=n \cdot (n-1)!$, якщо $n > 0$ і $n!=1$, якщо $n=0$. Програму обчислення факторіала з використанням рекурсивної функції зображено на рис. 1.

Результат виконання програми наведено на рис. 2.

```
implementation
{$R *.lfm}
{ TForm1 }
function factorial (n:integer):integer; //заголовок функції
begin //початок тіла функції
  if n>1 then factorial:=n*factorial(n-1) //обчислення факторіала
  else factorial:=1 //заданого числа
end; //кінець тіла функції
procedure TForm1.Button1Click(Sender: TObject);
var m:integer; //змінна, від значення якої обчислюється факторіал
    fact :integer; //змінна для обчисленого значення факторіала
begin //початок тіла основної програми
  m:=StrToInt(Edit1.Text); //присвоєння значення змінній
  fact:=factorial(m); //звернення до функції factorial
  //виведення значення обчисленого факторіала
  Label2.Caption:='факторіал числа '+Edit1.Text+' =' +IntToStr(fact);
end;
end.
```

Рис. 1. Програма обчислення факторіала

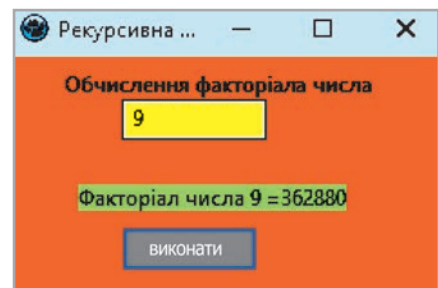


Рис. 2. Результат обчислення факторіала

Приклад. Дано два цілі додатні числа. Потрібно розробити програму для обчислення найбільшого спільного дільника з використанням рекурсивної функції.

Програму знаходження найбільшого спільного дільника двох цілих чисел без використання рекурсивних підпрограм ми вже розглядали раніше. Розв'язання цієї задачі

ґрунтується на використанні алгоритму Евкліда. Програму реалізації цього алгоритму з використанням рекурсивної функції наведено на рис. 3. Функція має ім'я *evk*, а числа, для яких обчислюється найбільший спільний дільник, позначені змінними n і m .

Результат виконання програми зображено на рис. 4.



```

var
Form1: TForm1;
implementation
{$R *.lfm}
{ TForm1 }
function evk(n, m:integer):integer; //заголовок функції evk
begin //початок тіла функції
if n=m then //перевірка умови завершення виклику функції
evk:=n //виклик функції
else if n>m then evk:=evk(n-m,m) //з різними значеннями
else evk:=evk(n,m-n); //параметрів
end; //кінець тіла функції
procedure TForm1.Button1Click(Sender: TObject);
var f, x,y:integer; //оголошення змінних
begin //початок тіла основної програми
x:=StrToInt(Edit1.Text); //значення змінної x
y:=StrToInt(Edit2.Text); //значення змінної y
f:=evk(x,y); //звернення до функції evk
Edit3.Text:=IntToStr(f); //виведення результату
end; //кінець тіла основної програми
end.

```

Рис. 3. Програмний код обчислення найбільшого спільного дільника

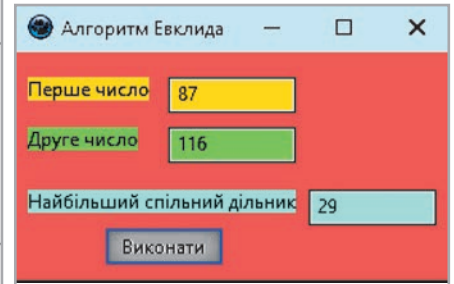


Рис. 4. Результат обчислення найбільшого спільного дільника

Отже, рекурсія є складним процесом, який вимагає додаткової пам'яті та часу її реалізації. Тому частіше використовується рекурентне обчислення. Однак без рекурсії інколи обійтися досить важко, оскільки без неї алгоритм має заплутану логіку.



Запитання для перевірки знань

- 1 Що називають рекурсією у програмуванні?
- 2 З якою метою у тілі рекурсії передбачається умовний оператор?
- 3 На якій структурі даних ґрунтується реалізація рекурсивних підпрограм?
- 4 Які недоліки має рекурсивне обчислення?
- 5 Поясніть сутність «заглиблення» й виходу зі стека у процесі реалізації рекурсії.



Завдання для самостійного виконання

- 1 Розробіть рекурсивну функцію для обчислення n перших членів заданої геометричної прогресії.
- 2 Дано десяткове натуральне число a . Розробіть рекурсивну функцію обчислення кількості цифр у цьому числі.
- 3 Розробіть програму з використанням рекурсивної функції для визначення мінімального елемента одновимірного числового масиву.
- 4 Розробіть програму пошуку мінімального елемента того самого масиву, який ви використовували для п. 3, без використання рекурсивної функції. Порівняйте розроблені програми і визначте, якій із них слід надати перевагу і чому.
- 5 Розробіть програму з використанням рекурсивної функції обчислення добутку значень елементів одновимірного числового масиву.

Практична робота № 23

Тема. Розробка алгоритмів з використанням допоміжних алгоритмів користувача та їх реалізація у вигляді програм.

Завдання: для кожного із завдань, наведених у таблиці, розробити алгоритм із використанням допоміжного алгоритму і програму реалізації алгоритму у візуальному режимі середовища Lazarus.

Варіант	Завдання	Початкові дані
1	Дано висоти і сторони основи трьох прямокутних паралелепіпедів. Обчисліть їх загальний об'єм.	$h_1=10, a_1=6, b_1=8$ $h_2=15, a_2=4, b_2=10$ $h_3= 8, a_3=7, b_3=12$
2	Дано гіпотенузи трьох прямокутних рівнобедрених трикутників. Обчисліть загальну площу цих трикутників.	$c_1=25, c_2=36, c_3=49$
3	Дано радіуси трьох кіл. Обчисліть їх загальну довжину.	$r_1=14, r_2=8, r_3=10$

Обладнання: комп'ютер із встановленою програмою — середовищем програмування.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

1. Виберіть один із варіантів завдань самостійно або за вказівкою вчителя. (У разі успішного виконання одного варіанта, виконайте інший.)
 2. Розробіть алгоритм розв'язування завдання.
 3. Розробіть програмний код реалізації алгоритму.
 4. Уведіть код програми в комп'ютер і виправте всі синтаксичні помилки.
 5. Виконайте програму для заданих початкових даних і доведіть, що програма функціонує правильно.
 6. Виконайте програму для інших початкових даних і переконайтеся, що отримано правильний результат.
- **Зробіть висновок:** чи існують такі значення початкових даних, для яких розроблена програма не працює.

Практична робота № 24

Тема. Розробка алгоритмів з використанням масивів як формальних параметрів і реалізація у вигляді програм.

Завдання: для кожного із завдань, наведених у таблиці, розробити алгоритм і програму його реалізації з використанням підпрограми, у якій формальними параметрами є масиви.

Варіант	Завдання	Початкові дані
1	Дано масив цілих чисел $a[1], a[2], \dots, a[n]$. Знайдіть у ньому кількість додатних і кількість від'ємних чисел.	43, -5, 0, 21, 23, -321, -66, 7, -59
2	На змаганнях у стрибках у воду з вишки спортсмена оцінюють 7 суддів. Максимальна і мінімальна оцінки не враховуються. Якщо максимальних або мінімальних оцінок декілька, не враховується по одній оцінці. Знайдіть середній бал виступу спортсмена.	9.5, 9.4, 9.5, 9.2, 9.5, 9.3, 9.2
3	Дано рядок символів. Знайдіть номери, на яких розташований символ s .	subscription

Обладнання: комп'ютер із встановленою програмою — середовищем програмування.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Виберіть один із варіантів завдань самостійно або за вказівкою вчителя. (У разі успішного виконання одного варіанта, виконайте інший.) Уважно проаналізуйте сутність завдання і тип даних.
 - 2 Розробіть у вигляді блок-схеми або у словесній формі алгоритм розв'язування завдання.
 - 3 Визначте необхідні компоненти та їхні властивості, розмістіть їх на формі й розробіть програмний код реалізації алгоритму.
 - 4 Уведіть код програми в комп'ютер. Спробуйте виконати компіляцію програми. Виправте всі синтаксичні помилки.
 - 5 Виконайте програму для заданих початкових даних і доведіть, що програма функціонує правильно.
 - 6 Виконайте програму для інших початкових даних і переконайтеся, що отримано правильний результат.
- **Зробіть висновок:** чим принципово відрізняється опрацювання двовимірних масивів від одновимірних.

Практична робота № 25

Тема. Розробка рекурсивних алгоритмів та їх реалізація у вигляді програм.

Завдання: для кожного із завдань, наведених у таблиці, розробити рекурсивний алгоритм і програму його реалізації.

Варіант	Завдання	Початкові дані
1	Дано рядок, який містить n символів. Розмістіть символи у зворотному порядку.	$n=7$ рядок: abcdefg
2	Дано n -розрядне ціле число. Розмістіть цифри числа у зворотному порядку.	$n=6$ 654321
3	У першому квадраті розташовано два прямокутники, сторони яких паралельні осям координат. Відомі координати вершин прямокутників. Визначте загальну площу обох прямокутників.	Вершини першого квадрата: $x_1=20, y_1=120, x_2=120, y_2=120,$ $x_3=120, y_3=40, x_4=20, y_4=40$ Вершини другого квадрата: $x_1=10, y_1=90, x_2=100, y_2=90,$ $x_3=100, y_3=10, x_4=10, y_4=10$

Обладнання: комп'ютер із встановленою програмою — середовищем програмування.

Хід роботи

Під час роботи з комп'ютером дотримуйтесь правил безпеки.

- 1 Виберіть один із варіантів завдань самостійно або за вказівкою вчителя. (У разі успішного виконання одного варіанта, виконайте інший.)
 - 2 Проаналізуйте умову завдання. Визначте загальний порядок розв'язання.
 - 3 Розробіть у словесній формі алгоритм розв'язування завдання.
 - 4 Розробіть програмний код реалізації алгоритму.
 - 5 Уведіть програму в комп'ютер. Виконайте компіляцію програми і виправте синтаксичні помилки.
 - 6 Виконайте програму для будь-яких початкових даних. Переконайтеся, що програма функціонує правильно.
 - 7 Виконайте програму для початкових даних, наведених у таблиці. Доведіть, що результат є правильним.
- **Зробіть висновок:** чи можна було виконати завдання за допомогою рекурентного обчислення.

Розділ 13. РОЗВ'ЯЗУВАННЯ КОМПЕТЕНТІСНИХ ЗАДАЧ

Які задачі цікавлять вас більше: ті, які поставлені у формалізованому вигляді, чи ті, які сформульовані у словесній формі, для яких спочатку потрібно розробити математичну модель? Наведіть приклади таких задач.



Задача № 1

У кожний період розвитку людства популярними були різні імена як серед жінок, так і серед чоловіків. Нині спостерігається тенденція до використання стародавніх імен, таких як Матвій, Кирило, Анастасія. Дослідіть, які імена найчастіше зустрічаються, наприклад, серед 11 учнів вашого класу. Розробіть необхідний алгоритм і програму визначення кількості різних імен.

Рекомендації. Існують різні алгоритми розв'язування цього завдання. Один із можливих алгоритмів ґрунтується на використанні допоміжного масиву, елементи якого під час уведення в основний масив імен учениць набувають значення **true**. Потім його відповідні елементи набувають значення **false**, якщо ім'я раніше вже зустрічалося в основному масиві або збігається з поточним іменем відповідного елемента основного масиву. Для розв'язання завдання можна також використати й різні засоби (компоненти). Доцільно дотримуватися такого порядку виконання завдання.

Крок 1

Розробити алгоритм.

Крок 2

Розмістити на формі необхідні компоненти, наприклад **TListBox**, **TEdit** і **TButton**.

Крок 3

Увести в об'єкт **ListBox1** імена учнів класу, наприклад у такому порядку: **Дмитро, Ольга, Олена, Дмитро, Аліса, Дмитро, Тетяна, Олена, Тетяна, Олег, Юрій** (рис. 1). Властивості **Text** об'єкта **Edit1** надати значення **Різних імен=**, а властивості **Caption** об'єкта **Button1** — значення **Виконати**. Установити подію **OnClick** для об'єкта **Button1**.

Крок 4

Розробити програмний код реалізації алгоритму. Результатом цього виконання повинні бути дані, наведені на рис. 2.

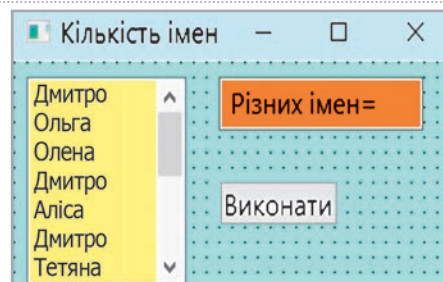


Рис. 1. Об'єкти **ListBox1**, **Edit1** і **Button1** на формі

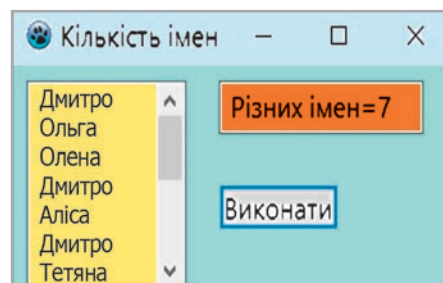


Рис. 2. Результат, який необхідно отримати

Крок 5

Порівняти розроблений вами програмний код із варіантом коду, зображеним на рис. 3, з метою виявлення переваг і недоліків кожного з них.

```

procedure TForm1.Button1Click(Sender: TObject);
var i, k, n : integer; //змінні цілого типу
    mas1: array [1..11] of string; //масив імен
    mas2: array [1..11] of boolean; //допоміжний масив
begin
    for i:=1 to 11 do //цикл створення масивів
    begin //відкриваються операторні дужки
        mas1[i] := ListBox1.Items[i-1]; //створення основного масиву
        mas2[i]:= true; //створення допоміжного масиву
    end; //закриваються операторні дужки
    k:=0; //початкова кількість імен
    for i:=1 to 11 do //цикл аналізу допоміжного масиву
    if mas2[i] then //аналіз елемента допоміжного масиву
    begin //відкриваються операторні дужки
        k:=k+1; //збільшення кількості імен
        for n:=i+1; to 11 do //цикл зміни допоміжного масиву
        //елемент допоміжного масиву набуває значення false, якщо
        //ім'я раніше вже зустрічалося
        mas2[n]:= (mas1[n]<>mas1[i]);
    end; //закриваються операторні дужки
    Edit1.Text:=Edit1.Text+IntToStr(k); //виведення кількості імен
end;
end.

```

Рис. 3. Варіант програмного коду реалізації завдання 1

Задача № 2

Дано двовимірний масив цілих чисел розмірністю $n \times m$. Розробіть алгоритм і програму визначення мінімального числа в кожному рядку і створення з них одновимірного масиву. Сформулюйте самостійно умову задачі, з якою ви зустрічалися в житті або можете зустрітися та яку можна формалізувати у сформульованій формі.

Задача № 3

Для підготовки до участі в чемпіонаті з легкої атлетики тренер поставив перед спортсменом завдання: перший тиждень пробігати дистанцію, довжина якої умовно дорівнює одиниці, а кожного наступного тижня збільшувати дистанцію на довжину, яка дорівнює початковій дистанції, поділеній на номер поточного тижня. Таким чином, математична модель бігу спортсмена така: $1, 1 + 1/2, 1 + 1/2 + 1/3, 1 + 1/2 + 1/3 + 1/4 + \dots$

Розробіть алгоритм і програму визначення номера тижня, коли дистанція буде більшою за 1,91.

Задача № 4

Розробіть програму для автоматизації пошуку ліків у базі аптеки. Одним із можливих варіантів може бути такий. Після запуску програми на формі повинні з'явитися компоненти, орієнтовний зміст яких зображено на рис. 4.

На формі розташовані такі об'єкти: ListBox1, Label1, Button1 і Button2. Для об'єкта Button1 встановлено значення Розпочати роботу властивості Caption. Усі інші об'єкти порожні. Після клацання кнопки Розпочати роботу відкривається вікно функції InputBox. Якщо ввести у це вікно, наприклад, латинську літеру b, то з масиву, в якому зберігаються всі назви ліків

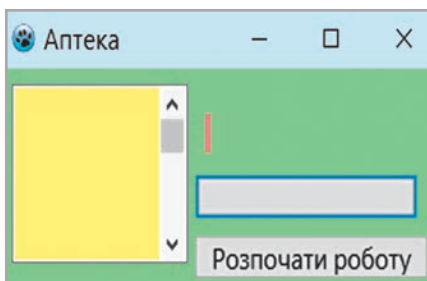


Рис. 4. Зміст форми після запуску програми

аптеки, вибираються всі ті ліки, назви яких починаються з цієї літери, і виводяться у поле об'єкта ListBox1 аналогічно тому, як зображено на рис. 5.

У полі об'єкта Button2 з'явилася повідомлення Завершити роботу, а поле об'єкта Button1 стало порожнім. Після клацання кнопки Завершити роботу повинно відкритися таке вікно, як зображено на рис. 4. Якщо в цьому вікні клацнути кнопку Розпочати роботу й увести літеру, на яку не починається жодна назва ліків у базі (масиві) аптеки, наприклад літеру w, то повинно відкритися вікно, орієнтовний зміст якого зображено на рис. 6.

На рис. 7 подано програму, що реалізує описаний сценарій.

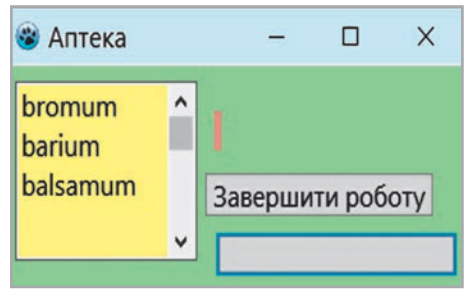


Рис. 5. Варіант змісту форми після введення букви

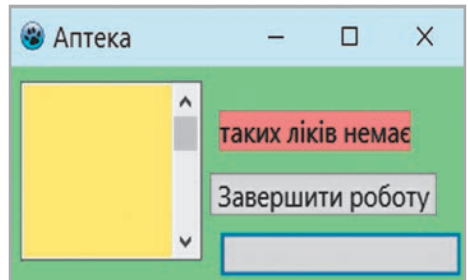


Рис. 6. Варіант змісту форми у випадку відсутності ліків

```

procedure TForm1.Button1Click(Sender: TObject); //перший обробник події
var mas: array [1..13] of string=('analginum', //формування
'codeinum', 'camphora', 'emplasfrum', 'magnesia', 'bromum', //масиву
'atropinum', 'barium', 'novocainum', 'mixtura', 'antiseptica', //назв
'balsamum', 'glycerinum'); //ліків
i, z:integer; a,b,x: string; //оголошення змінних
begin
z:=0; //ознака "ліки відсутні"
a:=InputBox ('Уведення', 'Увести першу літеру', ' ');
for i:=1 to 13 do //цикл аналізу масиву
begin //відкриваються операторні дужки
b:=mas[i]; //вибір чергової назви ліків
x:=copy (b, 1,1); //виділення першої літери у назві
if a=x then //чи збігається літера?
begin //відкриваються операторні дужки
z:=z+1; //збільшення кількості назв ліків
ListBox1.Items[z-1]:=b; //виведення назв ліків
end; end; //закриваються операторні дужки
if z=0 then Label1.Caption:='таких ліків немає';
Button1.Caption:=' '; //очищення Caption об'єкта Button1
Button2.Caption:='Завершити роботу';
end;

procedure TForm1.Button2Click(Sender: TObject); //другий обробник події
var j:integer; //оголошення змінної
begin
for j:=1 to 13 do //цикл очищення об'єкта ListBox1
ListBox1.Items[j-1]:=' '; //очищення об'єкта ListBox1
Button1.Caption:='Розпочати роботу';
Button2.Caption:=' '; //очищення Caption об'єкта Button2
Label1.Caption:=' '; //очищення Caption об'єкта Label1
end; end.
    
```

Рис. 7. Програмний код для провізора аптеки

Проаналізуйте програму, доведіть, що вона функціонує правильно для іншої бази ліків. Зверніть увагу на те, що в кодї програми використано 2 обробники подій. Перший обробник виконує функцію пошуку і виведення назв ліків, а другий виконує допоміжну функцію — встановлює нові значення й очищує поля компонентів.

Виберіть нові компоненти для реалізації завдання. Зробіть спробу розробити більш компактний програмний код і доведіть, що він функціонує правильно, або запропонуйте інший варіант підвищення продуктивності праці провізора аптеки.

Задача № 5

Із 1 до 10 березня учень спостерігав, на скільки збільшився світловий день. Розробіть алгоритм і програму визначення того, у які дні день збільшувався найбільше.

Розділ 14. СТВОРЕННЯ ТА РОЗРОБКА НАВЧАЛЬНИХ ПРОЕКТІВ

14.1. Структура проекту в середовищі Lazarus



Чи доводилося вам раніше брати участь у проектній діяльності з інших предметів? Яку функцію ви виконували в них?



Проект (від латин. *projectus* — «кинутий вперед») — це цілісна сукупність моделей та їхніх властивостей, описаних у формі, придатній для його реалізації. Процес розробки проекту називають проектуванням. **Проектування** — це процес визначення архітектури, компонентів, інтерфейсів та інших характеристик системи або її частини.

Етапи підготовки до проектів

- Визначення теми проекту і розбиття його на окремі незалежні, але взаємопов'язані частини (модулі)
- Формування груп для розробки проекту (з урахуванням інтересів учнів)
- Визначення основних етапів розробки кожного модуля проекту, строку їх виконання та порядку об'єднання окремих модулів у єдиний проект
- Формулювання технічних умов, яким повинен відповідати проект, переліку літератури, якою доцільно користуватися у процесі роботи над проектом, порядку його оформлення, тестування і захисту

Рис. 1. Загальний порядок підготовки до виконання навчальних проектів із програмування



Важливим принципом є комплексність, тобто врахування багатьох складових проекту: технологічної, санітарно-гігієнічної, кошторисної та інших. Зазвичай над проектом працює не один фахівець, а колектив працівників, тому кожний із них повинен дотримуватися єдиних вимог і норм проектування.

Які, на вашу думку, особливості має розробка проекту в середовищі Lazarus?



Правила і методика розробки проектів у кожній мові програмування має свої особливості.

Розглянемо деякі особливості створення навчальних проектів у середовищі візуального програмування Lazarus (рис. 2).

Проект у середовищі Lazarus — це програма, розроблена користувачем для розв'язування певної задачі. Проект об'єднує декілька файлів, із яких створюється єдиний файл, що виконується. Мінімальний склад файлів проекту такий:

1	Файл опису проекту (.lpr).
2	Файл проекту (.lpr). У ньому зберігаються всі основні параметри програми. Без цього файла неможливо виконати програму.
3	Модуль форми (.lfm). Містить опис усіх об'єктів, розташованих на формі. Файл містить також властивості тих об'єктів, які змінені програмістом.
4	Програмний модуль (.pas) — файл коду програми. Його можна поділити на окремі, менші за обсягом, модулі й кожний із них підключити до проекту.

Зазвичай проект містить також файл ресурсів (.lrs). У ньому можуть зберігатися дані різних типів.



У результаті компіляції з файлів проекту створюється **єдиний файл**, що виконується, з розширенням exe. Ім'я файла збігається з іменем проекту. Користувач розробляє програмний модуль, усі інші додаються до проекту автоматично.

Програмний модуль має таку загальну структуру:

```
unit <ім'я модуля>; //заголовок модуля
interface           //розділ опису (список модулів, типів,
                    //констант, змінних, процедур і функцій)
implementation     //розділ коду програми
end.                //кінець модуля
```

Розділ опису має таку загальну структуру:

```
interface
uses <список_модулів>;
type <список_типів>;
const <список_констант>;
var <список_змінних>;
procedure <ім'я процедури>;
...
function <ім'я_функції>;
```

Заголовок модуля починається ключовим словом **unit**, за яким слідує ім'я модуля і крапка з комою. Розділ опису починається ключовим словом **interface**. Тут описуються компоненти програмного коду: типи, класи, процедури і функції. Розділ **implementation** містить програмний код опрацювання даних, який розробляє користувач.

Особливості створення проектів у середовищі Lazarus

- Необхідність створення графічного інтерфейсу програми
- Розробка програмного коду мовою Free Pascal
- Компілювання, тестування, збереження і виконання проекту

Рис. 2. Особливості створення проектів у середовищі візуального програмування Lazarus

14.2. Варіанти тем навчальних проєктів у середовищі Lazarus



Пригадайте, з чого починається розробка програмного коду в середовищі Lazarus. Які компоненти ви використовували найчастіше? Як ви доводили правильність виконання програмного коду?

Сучасні програмні проєкти є досить складними. Успішна їх реалізація залежить від ретельного планування проєкту, розбиття його на окремі незалежні та взаємопов'язані модулі і визначення способів їх об'єднання у єдине ціле, методів тестування модулів і програми у цілому.

У цьому параграфі наведено три варіанти навчальних проєктів. Над кожним проєктом може працювати від 3 до 5 учнів. Учні призначають відповідальних за розробку окремих частин проєкту, його опис і інструкції для користувача. Звершується розробка проєкту його захистом.

Проєкт 1

Створіть для допомоги молодшим школярам каталог часто вживаних розширень файлів з описом.

У таблиці наведено деякі розширення, які автоматично присвоюються файлам програмами, що їх створюють, а також назви відповідних програм і назви типів файлів.

Розробіть проєкт, за допомогою якого дані, наведені у таблиці, виводяться у клітинки компонента `TStringGrid` в алфавітному порядку розширень. Виявіть переваги і недоліки розробленого програмного коду порівняно з наведеним.

Рекомендації. Бажано дотримуватися такого порядку дій.

Таблиця. РОЗШИРЕННЯ ФАЙЛІВ, ПРОГРАМ І ТИПІВ ФАЙЛІВ

Розширення	Програма	Тип файла
wav	MediaPlayer	Звуковий
txt	Блокнот	Неформатований текст
bmp	Paint	Растровий
doc	MS Word	Форматований текст
xls	MS Excel	Електронна таблиця

Крок 1	Розмістити на формі 2 компоненти <code>TStringGrid</code> і компонент <code>TButton</code> . В об'єкт <code>StringGrid1</code> увести дані, наведені у таблиці так, як зображено на рис. 1.	<table border="1"> <thead> <tr> <th>Розширення</th> <th>Програма</th> <th>Тип файла</th> </tr> </thead> <tbody> <tr> <td>wav</td> <td>Media player</td> <td>Звуковий</td> </tr> <tr> <td>txt</td> <td>Блокнот</td> <td>Неформ. текст</td> </tr> <tr> <td>bmp</td> <td>Paint</td> <td>Растровий</td> </tr> <tr> <td>doc</td> <td>MS Word</td> <td>Форм. текст</td> </tr> <tr> <td>xls</td> <td>MS Excel</td> <td>Електр. таблиця</td> </tr> </tbody> </table>	Розширення	Програма	Тип файла	wav	Media player	Звуковий	txt	Блокнот	Неформ. текст	bmp	Paint	Растровий	doc	MS Word	Форм. текст	xls	MS Excel	Електр. таблиця
Розширення	Програма	Тип файла																		
wav	Media player	Звуковий																		
txt	Блокнот	Неформ. текст																		
bmp	Paint	Растровий																		
doc	MS Word	Форм. текст																		
xls	MS Excel	Електр. таблиця																		
Крок 2	Розробити програмний код, за допомогою якого з даних, що містяться у клітинках об'єкта <code>StringGrid1</code> , створюється двовимірний масив з іменем <code>mas</code> , елементи якого мають тип <code>string</code> . Створений масив упорядкувати в алфавітному порядку розширення файлів.	<p>Рис. 1. Об'єкт <code>StringGrid1</code> з уведеними даними з Excel</p> <table border="1"> <thead> <tr> <th>Розширення</th> <th>Програма</th> <th>Тип файла</th> </tr> </thead> <tbody> <tr> <td>bmp</td> <td>Paint</td> <td>Растровий</td> </tr> <tr> <td>doc</td> <td>MS Word</td> <td>Форм. текст</td> </tr> <tr> <td>txt</td> <td>Блокнот</td> <td>Неформ. текст</td> </tr> <tr> <td>wav</td> <td>Media player</td> <td>Звуковий</td> </tr> <tr> <td>xls</td> <td>MS Excel</td> <td>Електр. таблиця</td> </tr> </tbody> </table>	Розширення	Програма	Тип файла	bmp	Paint	Растровий	doc	MS Word	Форм. текст	txt	Блокнот	Неформ. текст	wav	Media player	Звуковий	xls	MS Excel	Електр. таблиця
Розширення	Програма	Тип файла																		
bmp	Paint	Растровий																		
doc	MS Word	Форм. текст																		
txt	Блокнот	Неформ. текст																		
wav	Media player	Звуковий																		
xls	MS Excel	Електр. таблиця																		
Крок 3	Доповнити програмний код, розроблений на кроці 2, кодом виведення створеного масиву в клітинки об'єкта <code>StringGrid2</code> так, як зображено на рис. 2.	<p>Рис. 2. Об'єкт <code>StringGrid2</code> з елементами створеного масиву</p> <table border="1"> <thead> <tr> <th>Розширення</th> <th>Програма</th> <th>Тип файла</th> </tr> </thead> <tbody> <tr> <td>bmp</td> <td>Paint</td> <td>Растровий</td> </tr> <tr> <td>doc</td> <td>MS Word</td> <td>Форм. текст</td> </tr> <tr> <td>txt</td> <td>Блокнот</td> <td>Неформ. текст</td> </tr> <tr> <td>wav</td> <td>Media player</td> <td>Звуковий</td> </tr> <tr> <td>xls</td> <td>MS Excel</td> <td>Електр. таблиця</td> </tr> </tbody> </table>	Розширення	Програма	Тип файла	bmp	Paint	Растровий	doc	MS Word	Форм. текст	txt	Блокнот	Неформ. текст	wav	Media player	Звуковий	xls	MS Excel	Електр. таблиця
Розширення	Програма	Тип файла																		
bmp	Paint	Растровий																		
doc	MS Word	Форм. текст																		
txt	Блокнот	Неформ. текст																		
wav	Media player	Звуковий																		
xls	MS Excel	Електр. таблиця																		

Порівняти розроблений код із контрольним кодом, зображеним на рис. 3.
Виявити переваги і недоліки розробленої вами і наведеної програм.

Крок 4

```

procedure TForm1.Button1Click(Sender: TObject);
var i,j,m,p:integer; z:string; //оголошення змінних
    mas:array [1..5,1..3] of string; //оголошення двовимірного масиву
begin
    for i:=1 to 5 do //зовнішній цикл формування масиву
    for j:=1 to 3 do //внутрішній цикл формування масиву
    //введення масиву із клітинок StringGrid1
    mas [i,j]:=StringGrid1.Cells [j-1,i];
    p:=5; //початкове значення нижнього рядка
    while p>1 do //цикл упорядкування масиву
    begin //відкриваються операторні дужки
    m:=1; //початкове значення першого стовпця
    for i:=2 to p do //цикл пошуку максимального елемента
    //елемент якого рядка першого стовпця більше?
    if mas[i,1]>mas[m,1] then m:=i;
    z:=mas[m,1]; mas[m,1]:=mas[p,1]; mas[p,1]:=z; //переміщення
    z:=mas[m,2]; mas[m,2]:=mas[p,2]; mas[p,2]:=z; //усіх стовпців
    z:=mas[m,3]; mas[m,3]:=mas[p,3]; mas[p,3]:=z; //рядка
    p:=p-1; //зменшення номера рядка масиву
    end; //закриваються операторні дужки
    for i:=1 to 5 do //зовнішній цикл виведення масиву
    for j:=1 to 3 do //внутрішній цикл виведення масиву
    StringGrid2.Cells[j-1,i]:=mas[i,j]; //виведення елементів масиву
    end;
end.

```

Рис. 3. Програмний код реалізації завдання проекту 1

Проект 2

Щорічно в Україні для підготовки й формування команди з чотирьох учнів 9–11 класів для участі в міжнародній олімпіаді з інформатики проводяться збори, на які запрошуюються 8 учнів, нагороджених дипломами першого ступеня за результатами Всеукраїнської олімпіади поточного року. Усі учні після завершення зборів виконують контрольні завдання, які оцінюються максимальною сумою балів — 600. Учні, які набрали максимальну кількість балів, потрапляють до команди. Розробіть алгоритм і програму формування команди учнів.

Рекомендації. Бажано дотримуватися такого порядку дій.

Крок 1

Знайти в Інтернеті дані про учнів, нагороджених дипломами першого ступеня за результатами Всеукраїнської олімпіади з інформатики поточного року.

Крок 2

Розробити алгоритм реалізації завдання.

Крок 3

Розмістити на формі 2 компоненти **TStringGrid** і компонент **TButton**. Внести в об'єкт **StringGrid1** дані, знайдені в Інтернеті, про учнів і суму балів, набрану ними на олімпіаді. В об'єкт **StringGrid2** внести назви рядків і стовпців. Зразок оформлення об'єктів зображено на рис. 4.

nn	ПІБ	Школа	Клас	Бали
1	Зімін О.Л.	37, Київ	10	310
2	Петренко Н.К.	45, Харків	10	550
3	Петрік А.І.	90, Суми	9	470
4	Дрозд С.А.	5, Полтава	11	490
5	Сахно П.П.	12, Фастів	11	480
6	Каменев І.В.	30, Житомир	11	580
7	Зіненко В.І.	145, Київ	10	500
8	Рунов М.П.	11, Львів	11	470

nn	ПІБ	Школа	Клас	Бали
1				
2				
3				
4				

Рис. 4. Об'єкт **StringGrid1** з уведеними даними й об'єкт **StringGrid2**

Крок 4

Розробити і виконати програму реалізації алгоритму. Для правильно розробленої програми отримаємо результат, зображений на рис. 5.

nn	ПІБ	Школа	Клас	Бали
1	Каменев І.В.	30, Житомир	11	580
2	Петренко Н.К.	45, Харків	10	550
3	Зіненко В.І.	145, Київ	10	500
4	Дрозд С.А.	5, Полтава	11	490

Рис. 5. Результат виконання коду проекту 2

Крок 5

Проаналізувати розроблену програму і контрольну програму, зображену на рис. 6. Виявити переваги і недоліки розробленої вами і наведеної програм.

```

procedure TForm1.Button1Click(Sender: TObject);
var i, j, l, m: integer; z: string; //оголошення змінних
    mas: array[1..8, 1..4] of string; //оголошення масиву
begin
    for i:=1 to 8 do //зовнішній цикл створення масиву
        for j:=1 to 4 do //внутрішній цикл створення масиву
            mas[i, j]:=StringGrid1.Cells[j, i]; //введення елементів масиву
        l:=1; //початковий номер рядка
        while l<5 do //проаналізувати 4 рядки масиву
begin //відкриваються операторні дужки
    m:=l; //початковий номер макс. елемента
    for i:=l+1 to 8 do //цикл пошуку макс. елемента у 4 стовпці
        if mas[i, 4]>mas[m, 4] then m:=i; //який елемент більше?
    for i:=1 to 4 do //цикл переміщення і виведення рядків
begin //відкриваються операторні дужки
        //переміщення елементів чотирьох стовпців чергового рядка масиву
        z:=mas[m, i]; mas[m, i]:=mas[l, i]; mas[l, i]:=z;
        //виведення чотирьох стовпців чергового рядка масиву
        StringGrid2.Cells[i, l]:=mas[l, i];
        end; //закриваються операторні дужки
        l:=l+1; //збільшується значення номера рядка
    end; //закриваються операторні дужки
end;
end.

```

Рис. 6. Програмний код реалізації завдання проекту 2

Проект 3

У шкільній олімпіаді з інформатики бере участь різна кількість учнів 9–11 класів. Роботи учнів оцінюються загальною сумою від 0 до 50 балів. Розробіть алгоритм і програму визначення середнього балу результатів участі в олімпіаді учнів кожного класу.

Рекомендації. Найдоцільніше для реалізації цього завдання використати дві підпрограми. За допомогою однієї підпрограми можна створювати три масиви (за кількістю класів), у яких зберігаються отримані результати кожного учня класу, а за допомогою другої підпрограми — обчислювати середній бал учнів кожного класу. Для створення масивів необхідно використати підпрограму-процедуру, а для обчислення середнього балу — або функцію, або процедуру.

Оскільки від кожного класу в олімпіаді бере участь різна кількість учнів, то як

формальні параметри процедури краще використати відкритий масив.

Для розробки програми можна використовувати різні компоненти, наприклад TEdit, TLabel, TMemo, TListBox, TStringGrid та ін. Можна також використати функції InputBox(), ShowMessage() і компонент TButton. Для аналізу ефективності розробленого вами проекту на рис. 7 зображено програму одного з варіантів реалізації сформульованого завдання. У цьому варіанті використано дві процедури. Уведіть цю програму, переконайтеся, що вона функціонує правильно, і виявіть її недоліки.

```

procedure TForm1.Button1Click(Sender: TObject);
type mas=array of integer;           //тип масиву
var i,n,s:integer; sr:real; a:mas;    //оголошення змінних і масиву
procedure ved(var n:integer; var a:mas); //процедура уведення масиву
var j:integer;                       //локальна змінна
begin                                 //початок тіла процедури ved
  for j:=0 to (n-1) do                //цикл уведення масиву
    a[j]:=StrToInt(InputBox('Кількість ', 'балів учня ' //уведення
    +IntToStr(j+1), ' '));            //значень елементів масиву
  end;                                 //кінець процедури ved
procedure sum(n:integer; a:mas); //процедура обчисл. середн. значення
var j:integer;                       //локальна змінна
begin                                 //початок тіла процедури sum
  s:=0;                               //початкове значення суми елем. масиву
  for j:=0 to (n-1) do                //цикл обчислення суми елементів масиву
    s:=s+a[j];                       //обчислення суми елементів масиву
  sr:=s/n;                            //середнє значення суми елементів масиву
end;                                   //кінець процедури sum
begin                                 //початок тіла основної програми
  for i:=9 to 11 do                  //цикл опрацювання масиву
  begin                                //відкриваються операторні дужки
    n:=StrToInt(InputBox('Кількість учнів ', //кількість учнів класу,
    ' '+IntToStr(i), ' ')); //які брали участь в олімпіаді
    SetLength (a,n);                //виділення пам'яті масиву
    ved(n,a);                       //звернення до процедури ved
    sum (n,a);                       //звернення до процедури sum
    ShowMessage('Середній бал учнів класу ' //виведення
    +IntToStr(i)+' = '+FloatToStr(sr)); //середнього значення
    a:=NIL; end; end;                //звільнення пам'яті
end.

```

Рис. 7. Варіант програми реалізації завдання проекту 3

Після запуску програми на екрані з'явиться вікно, зразок якого наведено на рис. 8. У це вікно слід увести кількість учнів 9 класу, які брали участь в олімпіаді, і клацнути кнопку ОК.

Рис. 8. Вікно для введення кількості учнів у класі

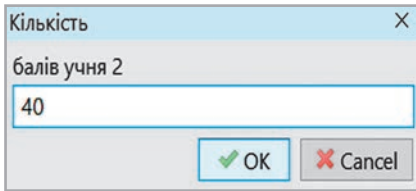


Рис. 9. Вікно для введення кількості балів, отриманих учнем

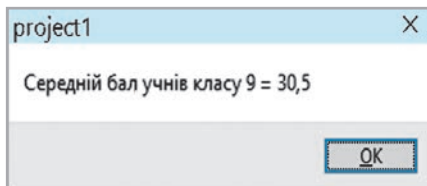


Рис. 10. Вікно із середнім балом учнів класу

Потім послідовно будуть з'являтися вікна, один зі зразків якого зображено на рис. 9, у які потрібно ввести кількості балів, отриманих кожним учнем цього класу.

Після завершення введення кількості балів, отриманих учнями 9 класу, на екран буде виведено середній бал учнів. Зразок такого вікна зображено на рис. 10.

В аналогічному порядку слід ввести дані про учнів 10 і 11 класів.

На рис. 11 подано програму, яка реалізує те саме завдання, але замість процедури, у якій обчислюється середній бал, використано функцію. Як впливає з аналізу цієї програми, вона суттєво відрізняється від першої програми, у якій як підпрограми використовувалися обидві процедури.

```

procedure TForm1.Button1Click(Sender: TObject);
type mas=array of integer;           //тип масиву
var i,n,s:integer; sr:real; a:mas;   //оголошення змінних і масиву
procedure ved(var n:integer; var a:mas); //процедура уведення масиву
var j:integer;                       //локальна змінна
begin                                 //початок тіла процедури ved
  for j:=0 to (n-1) do                //цикл уведення масиву
    a[j]:=StrToInt(TextBox('Кількість ', 'балів учня ' //уведення
    +IntToStr(j+1), ' '));            //значень елементів масиву
  end;                                 //кінець процедури ved
function sum(n:integer; a:mas):real; //функція обч. середн. значення
var j:integer;                       //локальна змінна
begin                                 //початок тіла функції sum
  s:=0;                               //початкове значення суми елем. масиву
  for j:=0 to (n-1) do                //цикл обчислення суми елементів масиву
    s:=s+a[j];                        //обчислення суми елементів масиву
  sum:=s/n;                           //середнє значення суми елементів масиву
end;                                   //кінець функції sum
begin                                 //початок тіла основної програми
  for i:=9 to 11 do                   //цикл опрацювання масиву
    begin                              //відкриваються операторні дужки
      n:=StrToInt(TextBox('Кількість учнів ', //кількість учнів класу,
      'класу '+IntToStr(i), ' ')); //які брали участь в олімпіаді
      SetLength(a,n);                 //виділення пам'яті масиву
      ved(n,a);                       //звернення до процедури ved
      sr:=sum(n,a);                   //звернення до функції sum
      ShowMessage('Середній бал учнів класу ' //виведення
      +IntToStr(i)+' = '+FloatToStr(sr)); //середнього значення
      a:=NIL; end; end;               //звільнення пам'яті
    end.

```

Рис. 11. Програмний код із процедурою і функцією

З урахуванням проведеного аналізу розробіть власну програму, у якій відсутні виявлені недоліки.

Комп'ютерний словник

База даних — сховище організованої сукупності даних різного типу, які відображають стан об'єктів певної предметної галузі та зв'язки між ними.

Вербальна модель — опис об'єкта, процесу, явища природною мовою.

Відкритий масив — масив, у розділі var або туре якого оголошується масив без діапазону індексів, вказується лише тип елементів.

Головна діагональ — діагональ масиву, що з'єднує лівий верхній елемент із правим нижнім.

Двійковий пошук — пошук даних у впорядкованому масиві.

Двовимірний масив — структура даних одного типу, що складається з фіксованої кількості рядків і стовпців.

Допоміжний алгоритм — алгоритм, який може неодноразово використовуватися в основному алгоритмі.

Експрес-сканування — швидке сканування, за допомогою якого сканується оперативна пам'ять і системні файли.

Електронний документообіг (обіг електронних документів) — сукупність процесів створення, опрацювання, відправлення, передавання, одержання, зберігання, використання та знищення електронних документів, які виконуються із застосуванням перевірки цілісності, а в разі необхідності — з підтвердженням факту одержання таких документів.

Запит — об'єкт бази даних, призначений для вибору з таблиці необхідних даних і подання їх користувачу в зручній формі.

Запит із параметрами — запит, у процесі виконання якого потрібно ввести деякі дані.

Запит на змінення — запит, за допомогою якого вносяться зміни в таблиці.

Звіт — об'єкт бази даних, призначений для виведення даних у зручному для користувача вигляді.

Інформатика — галузь людської діяльності, пов'язана з процесами опрацювання інформації за допомогою комп'ютера та середовища їх застосування.

Комп'ютерна безпека — стан захищеності систем опрацювання й зберігання даних, за якого забезпечується конфіденційність, доступність і цілісність даних

Інформаційна культура — елемент загальної культури людини, що є характеристикою особистості, яка проявляється в умінні цілеспрямовано працювати з інформацією, опануванні нових засобів інформаційно-комунікаційних технологій, дотриманні морально-етичних норм під час створення й використання інформаційних матеріалів.

Інформаційна модель — вербальний, графічний, математичний, табличний тощо опис об'єкта.

Інформаційна система — програмно-апаратний комплекс для зберігання даних, виконання над ними операцій, обумовлених специфікою предметної галузі, забезпечення зручного графічного інтерфейсу для доступу до необхідних даних та їх опрацювання.

Інформаційні процеси — процеси, пов'язані з пошуком, збиранням, збереженням, опрацюванням, передаванням, відображенням і захистом інформації.

Інформаційна технологія — сукупність методів і способів пошуку, збирання, накопичення, зберігання й опрацювання інформації на основі засобів обчислювальної і комунікаційної техніки.

Ключ таблиці — стовець (або кілька стовпців), який однозначно ідентифікує рядки.

Комп'ютерне моделювання — процес реалізації інформаційних моделей на комп'ютері.

Комп'ютерний вірус — шкідлива програма, що здатна до самокопіювання та вбудовування в код інших програм, системні ділянки пам'яті чи завантажувальні сектори, а також поширення своїх копій різними каналами зв'язку.

Лінійний пошук — пошук даних у масиві, заснований на прямому переборі елементів масиву.

Масив — структурований тип даних, елементи якого мають один тип.

Математична модель — система математичних рівнянь, формул, числових множин, які описують деякі властивості об'єкта, процесу або явища.

Метод вибору — метод упорядкування елементів масиву, заснований на пошуку максимального (мінімального) елемента й зміні місця його розташування.

Моделювання — процес створення моделі.

Модель — сукупність найважливіших властивостей об'єкта, процесу, явища, яка визначається метою її створення і може мати різні форми вираження.

Модель даних — спосіб об'єднання даних у структури.

Параметри підпрограми — дані, що передаються у підпрограму для опрацювання.

Підпрограма — логічно завершений фрагмент програми з присвоєним іменем, до якого можна звертатися неодноразово.

Побічна діагональ — діагональ масиву, що з'єднує правий верхній елемент із лівим нижнім.

Редактор StringGrid — вікно для введення елементів двовимірної масиву.

Рядок — набір символів певної довжини.

Символ — цифра, буква або інший знак клавіатури.

Синхронізація даних — процес перевірки та реалізації ідентичності вибраних даних на різних пристроях.

Стандартна підпрограма — підпрограма, вбудована у мову програмування.

Таблиця — об'єкт бази даних, у якому зберігаються дані різного типу.

Фактичні параметри — змінні, що вказані в операторі виклику підпрограми.

Фільтрування даних — вибір із таблиці записів, які містять задані значення в обраних полях.

Форма — об'єкт бази даних, за допомогою якого можна вводити й змінювати дані у таблицях.

Формальні параметри — змінні, вказані у заголовку підпрограми.

Шкідлива програма — спеціально розроблена програма для пошкодження та знищення файлів, даних, програмного забезпечення, викрадення особистих даних користувача.

avast! — антивірусна програма, що забезпечує високий ступінь захисту комп'ютера від вірусів та інших шкідливих програм.

chr (x) — функція повернення символу, код якого дорівнює x.

ColCount — кількість стовпців компонента TStringGrid.

concat (S1, S2) — функція об'єднання рядків S1 і S2.

copy (S, n, m) — функція копіювання m символів рядка S, починаючи з символу n.

delete (S, n, m) — функція вилучення з рядка S m символів, починаючи з позиції n.

InputBox() — функція, що застосовується для циклічного уведення елементів масиву.

insert (S1, S2, n) — функція вставлення в рядок S2 рядка S1, починаючи з позиції n.

length (S) — функція визначення довжини рядка.

ord (c) — функція повернення коду символу c.

pos (S1, S2) — функція визначення номера позиції, з якої починається входження рядка S2 у рядок S1.

RowCount — кількість рядків компонента TStringGrid.

RSS — множина XML-форматів, яку використовують для публікації та постачання інформації, що часто оновлюється.

ShowMessage() — функція, що застосовується для циклічного виведення елементів масиву.

TStringGrid — компонент, що використовується для введення/виведення двовимірних масивів.

Алфавітний покажчик

А

Авторське право 10

Б

База даних 37

Безпечний сайт 116

В

Вербальна модель 141

Видавнична система 23

Відкритий масив 214

Г

Геосервіс 102

Головна діагональ 192

Д

Двійковий пошук даних 162

Двовимірний масив 184

Домен 85

Доменне ім'я 95

Доменна система імен 97

Допоміжний алгоритм 201

Е

Електронний документ 12

Електронний документообіг 13

З

Закладка 128

Запит 56

— із параметрами 59

— на вибірку даних 56

— на змінення 62

— на обчислення 59

Захист даних 112

Звіт 70

І

Інтернет 90

Інтернет-провайдер 93

Інстант-месенджер 99

Інтернет-ресурс 107

Інформатика 4

Інформаційна безпека 110

Інформаційна культура 9

Інформаційна модель 141

Інформаційна система 8

Інформаційна технологія 5

Інформаційний процес 6

К

Канал зв'язку 81

Канал новин 130

Ключове поле 49

Компонент 149

— TListBox 154, 155

— TMemo 154, 155

— TStringGrid 187

Комп'ютерна мережа 76

— бездротова 82

— кабельна 81

— клієнт-серверна 77

— однорангова 77

Комп'ютерне моделювання 143

Комп'ютерний вірус 112

Л

Лінійний пошук даних 161

М

Макет 23

Масив 152

Маска підмережі 94

Математична модель 141
 Мережева спільнота 99
 Мережевий протокол 87
 Метод вибору 165
 Метод обміну 167
 Метод вставки 170
 Моделювання 140
 Модель 138, 140
 Модель даних 38

Н

Настільна видавнича система 23

П

Параметри підпрограми 203
 Право інтелектуальної власності 10
 Публікація 22
 Персональне навчальне середовище 120
 Підпрограма 202
 Побічна діагональ 192
 Програма розпізнавання тексту 17
 Програмне навчальне середовище 118
 Процедура 204

Р

Резервне копіювання даних 118
 Редактор StringGrid 187
 Рекурсивна підпрограма 217
 Робоча група 85
 Рядок 172

С

Сервіси Веб 2.0 100
 Сервіси Інтернету 99
 Синхронізація даних 132
 Стандартна підпрограма 201
 Сортування записів 54

Т

Таблиця 46
 Точка відновлення 118
 Топологія 78

Ф

Фактичні параметри 205
 Фільтрування записів 54
 Форма 64
 Формальні параметри 214
 Функція 203
 — chr (x) 174, 178
 — concat (S1, S2, ...) 174, 175
 — copy (S, n, m) 174, 176
 — delete (S, n, m) 174, 176
 — InputBox() 156
 — insert (S1, S2, n) 174, 177
 — length (S) 174, 175
 — ord (c) 174, 178
 — pos (S1, S2) 174, 177, 179, 180
 — ShowMessage() 156

Ш

Шаблон 25
 Шкідлива програма 112

І

IP-адреса 94

М

Mac-адреса 89

Р

RSS 130

U

URL-адреса 95

Зміст

Розділ 1. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ У СУСПІЛЬСТВІ

1.1. Інформатика та її вплив на розвиток суспільства	4
1.2. Інформаційні процеси та інформаційні системи	6
1.3. Поняття інформаційної культури й інтелектуальної власності	9
<i>Практична робота № 1</i>	11

Розділ 2. ЕЛЕКТРОННИЙ ДОКУМЕНТООБІГ

2.1. Поняття електронного документообігу	12
2.2. Апаратні засоби для забезпечення електронного документообігу	14
2.3. Програмне забезпечення електронного документообігу	17
<i>Практична робота № 2</i>	18
2.4. Програми автоматизованого перекладу текстів	19
<i>Практична робота № 3</i>	21

Розділ 3. КОМП'ЮТЕРНІ ПУБЛІКАЦІЇ

3.1. Поняття комп'ютерної публікації. Програмні засоби для створення публікацій	22
3.2. Види та структура публікацій. Основні складові публікацій	24
3.3. Створення комп'ютерної публікації	26
<i>Практична робота № 4</i>	29
3.4. Робота з текстовими об'єктами. Зв'язки між об'єктами публікації	30
3.5. Робота з графічними об'єктами	32
3.6. Етапи роботи з публікацією	34
<i>Практична робота № 5</i>	35
<i>Практична робота № 6</i>	35

Розділ 4. БАЗИ ДАНИХ

4.1. Поняття бази даних. Реляційна модель даних	37
4.2. Загальні відомості про систему Access 2016	42
4.3. Таблиці	45
4.4. Запити	56
4.5. Форми	64
4.6. Звіти	70
<i>Практична робота № 7</i>	73
<i>Практична робота № 8</i>	74
<i>Практична робота № 9</i>	74
<i>Практична робота № 10</i>	75

Розділ 5. МЕРЕЖЕВІ ТЕХНОЛОГІЇ

5.1. Класифікація комп'ютерних мереж	76
5.2. Апаратне і програмне забезпечення мереж	79
5.3. Канали зв'язку	81
5.4. Мережеві пристрої	83
5.5. Адресація в мережах	85
5.6. Мережеві протоколи	87
5.7. Призначення і структура Інтернету	90
5.8. Способи під'єднання до мережі. Функції провайдера	92

5.9.	Адресація в Інтернеті. Поняття IP-адреси, доменного імені та URL-адреси	94
5.10.	Адміністрування доменних імен	96
5.11.	Сучасні сервіси Інтернету	98
5.12.	Сервіси Веб 2.0	100
5.13.	Геосервіси	102
5.14.	Спеціальні інструменти пошуку	104
5.15.	Інтернет-ресурси для вивчення навчальних предметів	107
	<i>Практична робота № 11</i>	108
	<i>Практична робота № 12</i>	109

Розділ 6. ОСНОВИ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

6.1.	Класифікація загроз інформаційній безпеці. Морально-етичні і правові засоби захисту даних.	110
6.2.	Захист даних. Шкідливі програми, їх типи, принципи дії і боротьба з ними.	112
6.3.	Загрози при роботі в Інтернеті. Засоби браузера, призначені для гарантування безпеки	115
	<i>Практична робота № 13</i>	117
6.4.	Безпечне зберігання та видалення даних. Резервне копіювання та відновлення даних	118

Розділ 7. СТВОРЕННЯ ПЕРСОНАЛЬНОГО НАВЧАЛЬНОГО СЕРЕДОВИЩА

7.1.	Поняття персонального навчального середовища. Використання інтернет-середовищ для створення та публікації документів	120
7.2.	Організація та планування колективної діяльності. Використання офісних веб-програм для створення спільних документів	124
7.3.	Створення і використання спільних електронних закладок	128
7.4.	Канали новин	130
7.5.	Синхронізація даних	132
7.6.	Етапи створення веб-сторінок. Конструювання сайтів. Поняття мови розмітки гіпертексту	134
	<i>Практична робота № 14</i>	136
	<i>Практична робота № 15</i>	137

Розділ 8. МОДЕЛЮВАННЯ

8.1.	Поняття моделі та моделювання	138
8.2.	Класифікація моделей	141
8.3.	Комп'ютерне моделювання	143
8.4.	Використання програмних засобів для комп'ютерного моделювання	145
	<i>Практична робота № 16</i>	147

Розділ 9. ОДНОВИМІРНІ МАСИВИ

9.1.	Основи роботи у середовищі Lazarus (повторення)	148
9.2.	Поняття типів користувача і масиву	152
9.3.	Введення і виведення значень елементів масиву	154
9.4.	Класичні алгоритми опрацювання числових одновимірних масивів	158
9.5.	Упорядкування елементів масиву	165
	<i>Практична робота № 17</i>	170
	<i>Практична робота № 18</i>	171

Розділ 10. СИМВОЛЬНІ ТА РЯДКОВІ ВЕЛИЧИНИ

10.1. Загальні відомості про символічні та рядкові величини.....	172
10.2. Функції опрацювання рядків і символів	174
10.3. Класичні алгоритми опрацювання рядків	179
<i>Практична робота № 19</i>	183
<i>Практична робота № 20</i>	183

Розділ 11. ДВОВИМІРНІ МАСИВИ

11.1. Загальні відомості про двовимірні масиви	184
11.2. Введення і виведення значень елементів двовимірних масивів у візуальному режимі	187
11.3. Класичні алгоритми опрацювання значень елементів двовимірних масивів	191
<i>Практична робота № 21</i>	199
<i>Практична робота № 22</i>	200

Розділ 12. ДОПОМІЖНІ АЛГОРИТМИ

12.1. Загальні відомості про допоміжні алгоритми і підпрограми.....	201
12.2. Процедури	203
12.3. Функції	209
12.4. Використання масивів як формальних параметрів підпрограм	213
12.5. Поняття рекурсії. Рекурсивні алгоритми	217
<i>Практична робота № 23</i>	221
<i>Практична робота № 24</i>	221
<i>Практична робота № 25</i>	222

Розділ 13. РОЗВ'ЯЗУВАННЯ КОМПЕТЕНТІСНИХ ЗАДАЧ

<i>Задача № 1</i>	223
<i>Задача № 2</i>	224
<i>Задача № 3</i>	224
<i>Задача № 4</i>	224
<i>Задача № 5</i>	225

Розділ 14. СТВОРЕННЯ ТА РОЗРОБКА НАВЧАЛЬНИХ ПРОЕКТІВ

14.1. Структура проекту в середовищі Lazarus.....	226
14.2. Варіанти тем навчальних проектів у середовищі Lazarus	228

Відомості про користування підручником

№ з/п	Прізвище та ім'я учня / учениці	Навчальний рік	Стан підручника	
			на початку року	в кінці року
1				
2				
3				
4				
5				

Навчальне видання
РУДЕНКО Віктор Дмитрович
РЕЧИЧ Наталія Василівна
ПОТІЄНКО Валентина Олександрівна

«Інформатика для загальноосвітніх навчальних закладів з поглибленим вивченням інформатики»
підручник для 9 класу загальноосвітніх навчальних закладів

Рекомендовано Міністерством освіти і науки України

Видано за рахунок державних коштів. Продаж заборонено

Провідний редактор *І. Л. Морєва*. Редактори *Л. А. Каюда, Ю. М. Миронова*.
Художнє оформлення *В. І. Труфен*. Технічний редактор *А. В. Пліско*.
Комп'ютерна верстка *О. В. Сміян, С. В. Яшиш*. Коректор *Н. В. Красна*.

В оформленні підручника використані зображення,
розміщені в мережі Інтернет для вільного використання

Підписано до друку 20.07.2017. Формат 84×108/16.
Папір офсетний. Гарнітура Шкільна. Друк офсетний.
Ум. друк. арк. 25,20. Обл.-вид. арк. 32,76. Тираж 5366 прим. Заказ № 2912.

ТОВ Видавництво «Ранок»,
вул. Кібальчича, 27, к. 135, Харків, 61071.
Свідоцтво суб'єкта видавничої справи ДК № 5215 від 22.09.2016.
Адреса редакції: вул. Космічна, 21а, Харків, 61145.
E-mail: office@ranok.com.ua. Тел. (057) 701-11-22, тел./факс (057) 719-58-67.

Надруковано у друкарні ТОВ «Місар +»,
пров. Сімферопольський, 6, Харків, 61052.
Свідоцтво суб'єкта видавничої справи ДК № 3461 від 14.04.2009.
Тел. +38 (057) 764-83-04. E-mail: kt@mitsar.com.ua

9

Інформатика

ДЛЯ ЗАГАЛЬНООСВІТНІХ НАВЧАЛЬНИХ ЗАКЛАДІВ
З ПОГЛИБЛЕНИМ ВИВЧЕННЯМ ІНФОРМАТИКИ

Структура підручника

- Навчальний матеріал
 - основний матеріал
 - додатковий матеріал
 - приклади практичного застосування знань
 - вправи для самостійного виконання
 - запитання для перевірки знань
- Практичні роботи
- Комп'ютерний словник
- Алфавітний покажчик

Особливості підручника

- Дворівневість подання теоретичного матеріалу:
 - визначений програмою — в основному тексті,
 - додатковий — у спеціальних рубриках
- Описи практичних робіт
- Приклади практичного застосування набутих знань з ІКТ
- Система вправ для закріплення навичок роботи за комп'ютером

Інтернет-підтримка

- Онлайн-тестування за кожним розділом
- Додаткові матеріали до уроків

ВИДАВНИЦТВО
РАНОК



Інтернет-підтримка
interactive.ranok.com.ua

