



ІНФОРМАТИКА

8

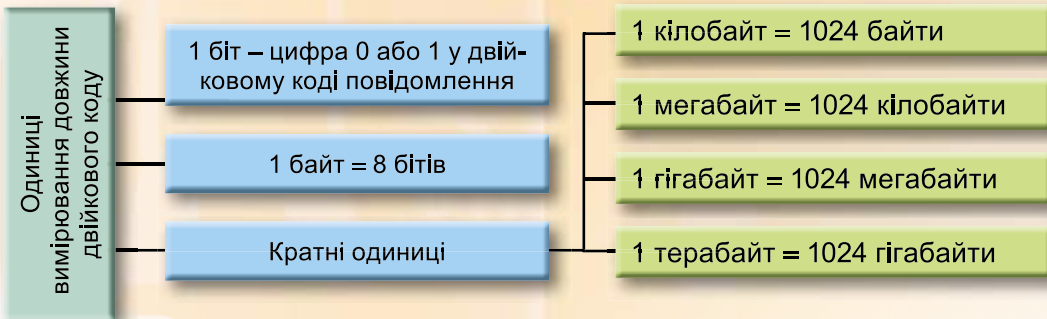
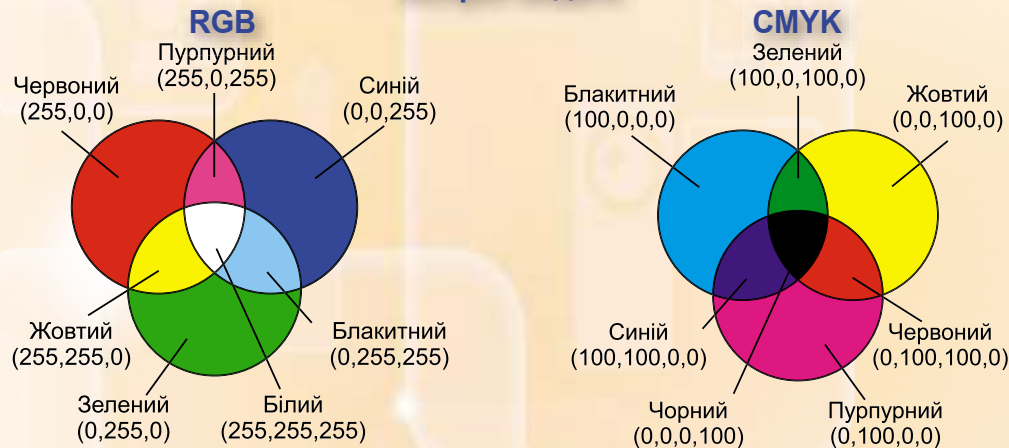


Найважливіше в розділі 1

Таблиці кодів символів

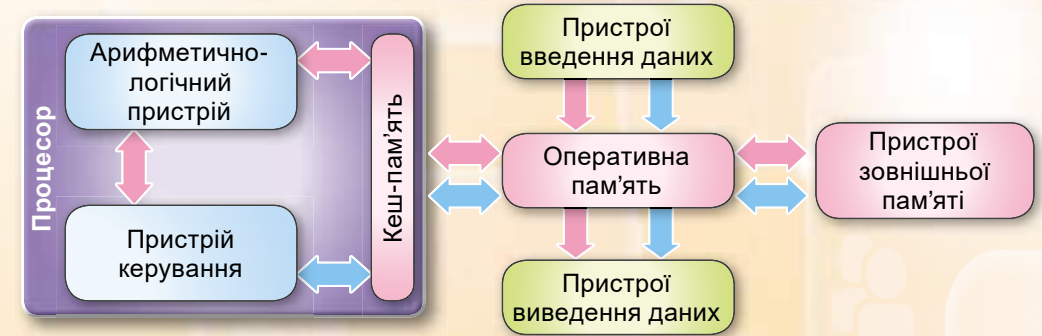
ASCII	KOI8-U, Windows-1251	Юнікод
<ul style="list-style-type: none"> 128 кодів символів коди літер латиниці, цифр, розділових знаків, спеціальних символів 	<ul style="list-style-type: none"> 256 кодів символів усі коди символів таблиці ASCII коди літер кирилиці 	<ul style="list-style-type: none"> 1 114 112 кодів символів усі коди символів таблиці ASCII коди літер усіх світових алфавітів

Колірні моделі



Найважливіше в розділі 2

Модель реалізації інформаційних процесів з використанням основних пристроїв комп'ютерів

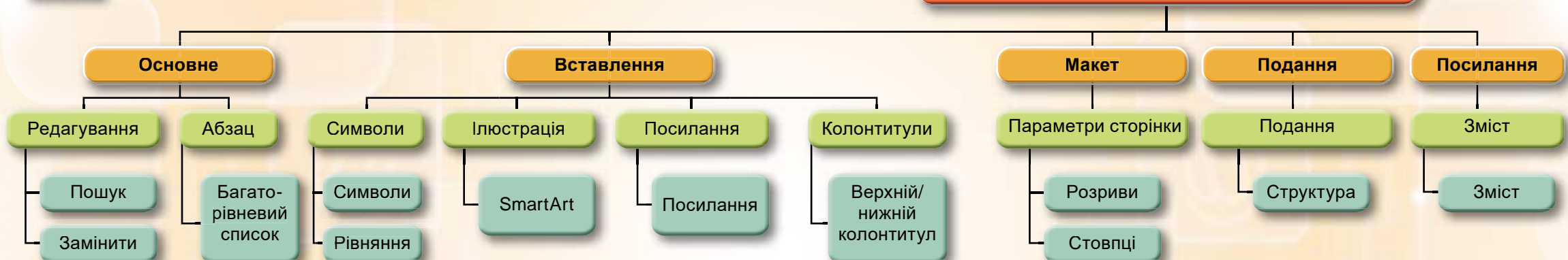


Етапи розвитку засобів реалізації інформаційних процесів

<p>Ручні засоби</p> <ul style="list-style-type: none"> Пальці рук Абак Рахівниця 	<p>Механічні засоби</p> <ul style="list-style-type: none"> Паскаліна Арифмометр Машина Беббіджа 	<p>Електричні засоби</p> <ul style="list-style-type: none"> Табулятор Електричний калькулятор 	<p>Електронні засоби</p> <ul style="list-style-type: none"> ABC ENIAC Персональний комп'ютер
--	---	--	--

Найважливіше в розділі 3

Основні операції в текстовому процесорі

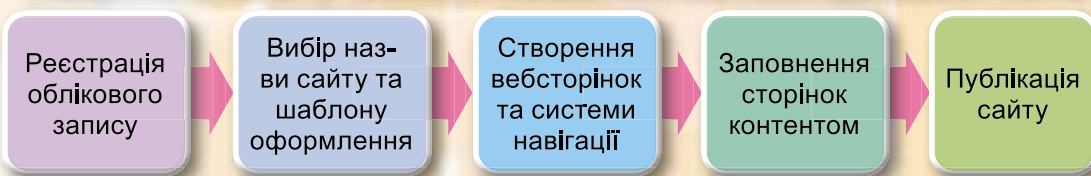


Найважливіше в розділі 4

Теги HTML

Тег	Призначення	Приклад
h1, ..., h6	Створення заголовка	<h1> Мова HTML</h1>
p	Створення абзацу	<p>Теги мови HTML – це команди, що визначають розмітку вебсторінки</p>
br	Розрив рядка	
a	Уставлення гіперпосилання	До другої сторінки
img	Уставлення зображення з файлу	
hr	Уставлення горизонтальної лінії	<hr>

Створення вебсайту засобами онлайн-системи конструювання сайтів

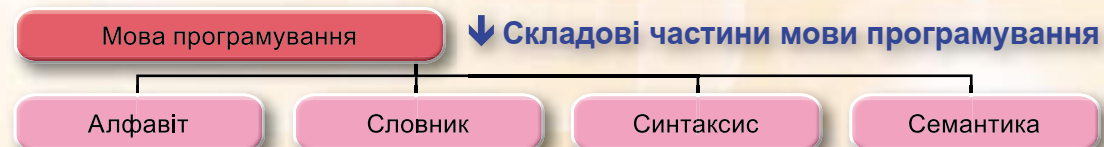


Найважливіше в розділі 5

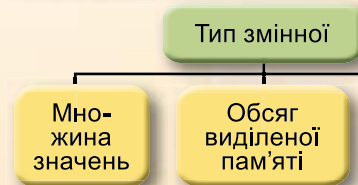
Програми



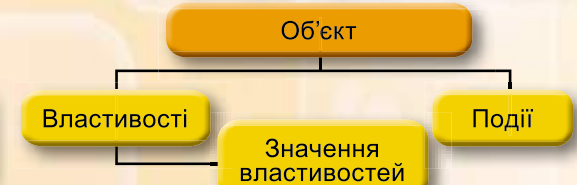
Найважливіше в розділі 6



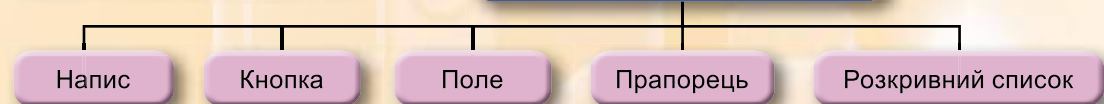
Тип змінної ↓



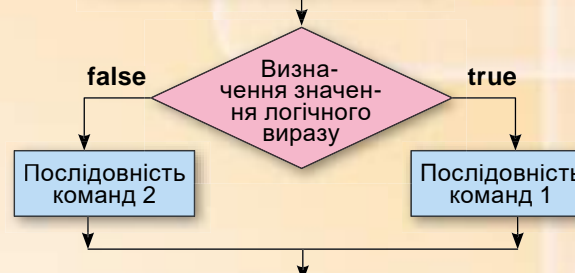
↓ Характеристики об'єкта



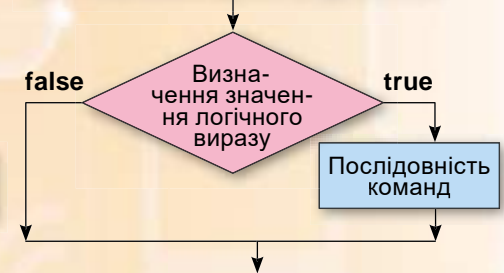
Компоненти на вікні ↓



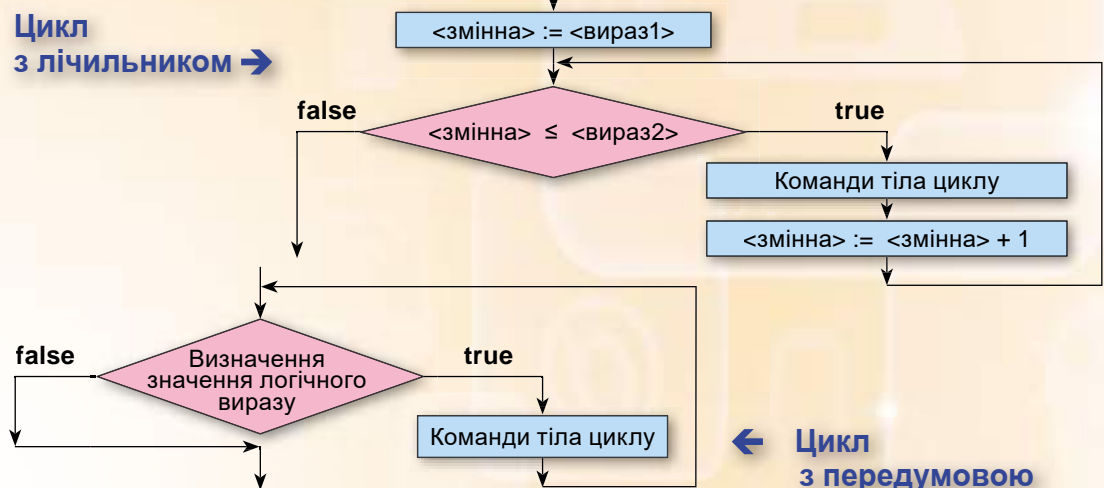
Повне розгалуження ↓



↓ Неповне розгалуження



Цикл з лічильником →



← Цикл з передумовою

ІНФОРМАТИКА

Підручник для 8 класу
закладів загальної середньої освіти



*Рекомендовано
Міністерством освіти і науки України*



Київ
«Гене́за»
2021

УДК 004(075.3)

I-74

Авторський колектив:

Йосиф Ривкінд, Тетяна Лисенко, Людмила Чернікова, Віктор Шакотько

*Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 22.02.2021 № 243)*

Видано за рахунок державних коштів. Продаж заборонено

Інформатика : підруч. для 8-го кл. закл. заг. се-
I-74 ред. освіти / Йосиф Ривкінд [та ін.]. — Київ : Генеза,
2021. — 256 с. : іл.

ISBN 978-966-11-1179-9.

УДК 004(075.3)

Навчальне видання

**РИВКІНД Йосиф Якович, ЛИСЕНКО Тетяна Іванівна,
ЧЕРНІКОВА Людмила Антонівна, ШАКОТЬКО Віктор Васильович**

ІНФОРМАТИКА

**Підручник для 8 класу
закладів загальної середньої освіти**

Рекомендовано Міністерством освіти і науки України

Видано за рахунок державних коштів. Продаж заборонено

Підручник відповідає Державним санітарним нормам і правилам
«Гігієнічні вимоги до друкованої продукції для дітей»

У підручнику використано ілюстративний матеріал з відкритих джерел інтернету, зокрема сайтів *vecteezy.com*, *depositphotos.com*. Усі матеріали в підручнику використано з навчальною метою відповідно до законодавства України про авторське право і суміжні права.

Відповідальна за випуск *Наталія Заблоцька*. Редактор
Наталія Дашко. Обкладинка *Олександра Павленка*. Макет, художнє
оформлення, комп'ютерна обробка ілюстрацій *Василя Марущинця*.
Технічний редактор *Цезарина Федосіхіна*. Комп'ютерна верстка
Юрія Лебедева. Коректори *Лариса Леуська, Любов Федоренко*

Формат 70×100/16. Ум. друк. арк. 20,8. Обл.-вид. арк. 19,56.

Тираж 248 696 пр. Вид. № 2186. Зам. № 21-04-2202

Видавництво «Генеза», вул. Тимошенка, 2-л, м. Київ, 04212.

Свідоцтво суб'єкта видавничої справи серія ДК № 5088 від 27.04.2016.

Віддруковано у ТОВ «ПЕТ», вул. Максиміліанівська, 17, м. Харків, 61024.

Свідоцтво суб'єкта видавничої справи серія ДК № 6847 від 19.07.2019.

© Ривкінд Й. Я., Лисенко Т. І.,
Чернікова Л. А., Шакотько В. В., 2021

© Видавництво «Генеза»,
оригінал-макет, 2021


ISBN 978-966-11-1179-9




ШАНОВНІ ВОСЬМИКЛАСНИЦІ ТА ВОСЬМИКЛАСНИКИ!


Ви продовжуєте вивчати цікавий і важливий предмет – **Інформатику**.

У попередніх класах ви вже навчилися виконувати операції з об'єктами операційної системи комп'ютера, створювати інформаційні моделі, опрацьовувати графічні, текстові, числові дані та мультимедійні презентації, листуватися, шукати відомості та колективно опрацьовувати документи в Інтернеті, складати алгоритми та програми для різних виконавців.



На уроках інформатики у 8-му класі ви розглянете кодування текстових і графічних даних, апаратне забезпечення комп'ютера та його призначення, продовжите опрацьовувати текстові та мультимедійні дані, створювати та публікувати в Інтернеті власні сайти. Ви навчитеся створювати відеофільми, опрацьовувати аудіодані, створювати проєкти в середовищі програмування. І в цьому вам допоможе цей підручник.


Вміст підручника поділено на розділи. Кожний розділ складається з пунктів, які, у свою чергу, поділяються на підпункти. На початку кожного пункту наведено запитання для повторення вивченого. Відповіді на них полегшать розуміння та засвоєння нового матеріалу. Ці запитання позначено .

Уважно читайте матеріал, викладений у підручнику. Найсуттєвіше зібрано в кінці кожного пункту в окремій рубриці **«Найважливіше в цьому пункті»** . Для відпрацювання практичних навичок у рубриці **«Працюємо з комп'ютером»**  наведено тренувальні вправи з детальними алгоритмами виконання завдань. Цю рубрику ви зможете знайти за посиланням <https://cutt.ly/8hedSjp> або за QR-кодом .






Наприкінці кожного пункту розміщено запитання для самоконтролю **«Дайте відповіді на запитання»** . Рекомендуємо вам після вивчення навчального матеріалу пункту дати відповіді на них. Біля кожного запитання стоїть позначка, яка означає, що правильна відповідь на це запитання відповідає:

- – початковому або середньому рівню навчальних досягнень;
- – достатньому рівню навчальних досягнень;
- * – високому рівню навчальних досягнень.

Так само позначено й рівні практичних завдань у рубриці **«Виконайте завдання»** , яку наведено в кінці кожного пункту. Завдання, які авторський колектив рекомендує для роботи вдома, позначено .



Завдання з позначкою  передбачають, що над їх виконанням доцільно попрацювати в парах або невеликих групах.

Крім основного матеріалу, пункти підручника містять рубрики:

-  «Для тих, хто хоче знати більше»;
-  «Чи знаєте ви, що...»;
-  «Цікаві факти з історії»;
-  «Для тих, хто працює з Linux»;
-  «Для тих, хто працює з Відеоредактором»;
- «Тлумачний словник термінів і понять» (на зеленому тлі).

У 8-му класі ви продовжите вивчення теми «Опрацювання текстових даних» на прикладі роботи з текстовим процесором **Microsoft Office 365**, опис цього самого матеріалу для **Microsoft Word 2010** розміщено в електронному додатку до підручника.

У розділі «Алгоритми та програми» в окремих підпунктах розглянуто два середовища створення проєктів:

-  «Для тих, хто вивчає Pascal»;
-  «Для тих, хто вивчає Python».

У кінці підручника розміщено **Словничок**, що містить означення основних понять.

Файли-заготовки для виконання завдань і практичних робіт, а також електронний додаток до підручника з інтерактивними вправами можна знайти в Інтернеті за адресою <http://sites.google.com/pu.org.ua/allinf> або QR-кодом.



***Бажаємо вам успіхів у вивченні найцікавішої
та найсучаснішої науки – ІНФОРМАТИКИ!***

Авторський колектив



Розділ 1. Кодування даних

У цьому розділі ви дізнаєтеся про:

- опрацювання даних як інформаційний процес
- кодування та декодування повідомлень
- кодування текстових і графічних даних
- таблиці кодів символів
- колірні моделі
- двійкове кодування
- одиниці вимірювання довжини двійкового коду

1.1. Опрацювання даних як інформаційний процес. Кодування та декодування повідомлень



1. Які інформаційні процеси ви знаєте? У чому сутність процесу опрацювання повідомлень?
2. Які існують способи подання повідомлень?
3. Що таке дані? Наведіть приклади даних.

Опрацювання даних як інформаційний процес

У сучасному суспільстві багато професій пов'язано з опрацюванням даних, поданих словами, числами, графічними зображеннями тощо. Бухгалтери здійснюють нарахування заробітної платні, спираючись на дані про обсяги виконаних робіт. Лікарі ставлять діагноз пацієнтам на основі результатів різних аналізів. Синоптики складають прогноз погоди, аналізуючи дані метеостанцій про рух повітряних мас. Астрономи розраховують траєкторії руху космічних об'єктів за результатами спостережень за їхнім положенням. Художник створює ілюстрації до книжки, ознайомившись з її змістом (мал. 1.1). Виконують опрацювання даних й учні/учениці, коли розв'язують задачі, пишуть твори, будують графіки функцій тощо.



Мал. 1.1. Опрацювання даних представниками різних професій

Як ви вже знаєте, **опрацювання даних** – це процес отримання нових даних з наявних.

Так, якщо розв'язують математичну задачу, то з наявних (початкових, вхідних) даних умови задачі отримують нові (вихідні) дані – розв'язок задачі. Для визначення переможців легкоатлетичного забігу дані опрацьовують, розміщуючи прізвища учасників/учасниць забігу в списку відповідно до часу подолання ними дистанції від найменшого до найбільшого значення часу. А під час побудови діаграми в табличному процесорі з наявних числових даних утворюються нові дані, подані графічним способом.

Опрацьовувати дані можна з використанням різноманітних пристроїв, у тому числі з використанням комп'ютера. Для цього потрібно подати дані у вигляді, придатному для опрацювання пристроями комп'ютера, та визначити правила, за якими повинно відбуватися опрацювання.

Кодування повідомлень

З метою зберігання, передавання, опрацювання, захисту повідомлень нерідко змінюють спосіб їхнього подання. Наведемо кілька прикладів:

- Усні повідомлення записують на папері, замінюючи звуки людської мови літерами алфавіту. При цьому виконується збереження повідомлень.
- Під час розмови по телефону звуки перетворюються на електричні сигнали. Це робить можливим передавання повідомлень на великі відстані.
- Розв'язуючи задачу на уроці математики, слова, що позначають числа, записують цифрами, а математичні операції – спеціальними знаками. Це спрощує опрацювання числових даних.
- Для захисту повідомлень від сторонніх осіб здійснюють шифрування, замінюючи літери в тексті повідомлення іншими літерами, числами або умовними позначеннями.

Код (лат. *codex* – книжка) – система умовних сигналів для передавання, опрацювання та зберігання повідомлень.

Процес заміни однієї послідовності сигналів, якою подано повідомлення, іншою послідовністю сигналів називають **кодуванням повідомлення**.

Кодування є одним із прикладів опрацювання даних. При цьому змінюється

спосіб подання, а зміст повідомлення не змінюється.

Кодування широко використовується в сучасному житті для подання повідомлень у вигляді, придатному для опрацювання не лише людиною, а й технічними пристроями. Так, відомості про товари кодують з використанням штрихкодів (мал. 1.2). Цифрами на штрихкоді позначено відомості про товар, країну та підприємство, що є виробником цього товару. Цей числовий код призначено для опрацювання людиною та використовується для впорядкованого зберігання даних про товари. А товщина ліній штрихкоду та відстань між ними кодують ті самі відомості. Цей графічний код сприймається спеціальними сканерами та опрацьовується з використанням комп'ютера.

Для кодування повідомлень визначають не лише набір сигналів, які будуть використані для заміни сигналів заданого повідомлення, а й правила, за якими здійснюється ця заміна.

Наприклад, з історії добре відомий шифр, яким користувався для секретного листування зі своїми

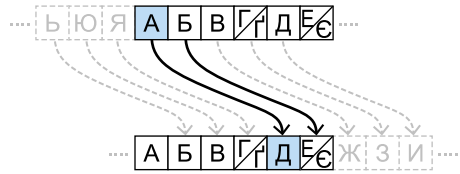


Мал. 1.2. Штрихкод товару



генералами римський імператор Гай Юлій Цезар (100 р. до н. е. – 44 р. до н. е.). Правила кодування полягають у тому, що кожна літера в тексті повідомлення замінюється іншою, що міститься в алфавіті на відстані кількох позицій від заданої літери (мал. 1.3).

Так, закодувавши повідомлення «*привіт*» з використанням шифру Цезаря зі зсувом на 4 літери праворуч, отримуємо повідомлення «*уфкелц*».



Мал. 1.3. Правила заміни символів за шифром Цезаря зі зсувом на 4 літери праворуч

Декодування повідомлень

Знаючи правило, за яким було закодовано повідомлення, можна його розкодувати. Процес отримання початкового повідомлення із закодованого називають **декодуванням повідомлення**.

Декодування повідомлень відбувається, коли ми читаємо вголос надрукований текст, виконуємо музичний твір по нотах, розшифровуємо повідомлення, визначаємо за штрихкодом країну, у якій виготовлено товар, тощо.

Так, для декодування повідомлення, закодованого шифром Цезаря зі зсувом на 4 літери праворуч, потрібно кожен літеру закодованого повідомлення замінити іншою, яку розміщено в алфавіті на 4 позиції ліворуч від заданої. Таким чином, декодувавши повідомлення «*зтдфкн зисб*», отримуємо «*добрий день*».

Кодування та декодування повідомлень є прикладами процесів опрацювання даних.



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/dhdeH0F> або QR-кодом.



Найважливіше в цьому пункті

Опрацювання даних – це процес отримання нових даних з наявних шляхом виконання дій за деяким алгоритмом.

Кодування повідомлення – це процес заміни однієї послідовності сигналів, якою подано повідомлення, іншою послідовністю.

Кодування повідомлень виконується з метою їх зберігання, передавання, подальшого опрацювання та захисту. При цьому змінюється спосіб подання, але зміст повідомлення не змінюється.

Для кодування повідомлень визначають набір сигналів, які будуть використані для заміни сигналів заданого повідомлення, та правила, за якими здійснюється ця заміна.

Декодування повідомлення – це процес отримання початкового повідомлення із закодованого.



Дайте відповіді на запитання

- 1°. У чому полягає процес опрацювання даних?
- 2°. Які нові дані можуть бути отримані в результаті опрацювання даних про:
 - а) кількість і ціну проданих у магазині товарів;
 - б) зріст учнів та учениць класу;
 - в) відстань між двома населеними пунктами та швидкість руху поїзда, що курсує між ними?

- 3°. У чому полягає процес кодування повідомлень?
- 4°. З якою метою здійснюють кодування повідомлень?
- 5*. З якою метою звуки музики кодують нотами; повідомлення про особливості руху по дорозі кодують дорожніми знаками?
- 6*. Повідомлення про що можна закодувати з використанням кольорів; графічних позначень?
- 7°. Що потрібно попередньо визначити для кодування повідомлення?
- 8*. За якими правилами у грі «Морський бій» кодують положення кораблів? З якою метою?
- 9°. У чому полягає процес декодування повідомлень?



Виконайте завдання

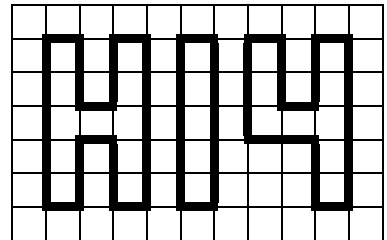
- 1°. Наведіть приклади кодування повідомлень, які використовуються в музиці, фізиці, хімії, географії та ін.
- 2°. Закодуйте з використанням шифру Цезаря зі зсувом на 4 літери праворуч повідомлення:
а) *інформатика*; б) *моя школа*; в) *кодування повідомлень*.



- 3°. Повідомлення закодовано шифром Цезаря зі зсувом на 3 літери праворуч. Декодуйте повідомлення:
а) *дзузфзра*; б) *егурґ тсесжґ*;
в) *рдґвоарїм укн усїтьсьґдфв*.

- 4°. Зображення, створене олівцем на аркуші паперу в клітинку, закодовано за таким правилом: стрілка вказує напрямок, у якому переміщується олівець по лінії сітки, а число вказує кількість клітинок для переміщення. Відтворіть закодоване зображення:
а) $\uparrow 5 \rightarrow 3 \downarrow 5 \leftarrow 1 \uparrow 4 \leftarrow 1 \downarrow 4 \leftarrow 1$;
б) $\leftarrow 3 \downarrow 5 \rightarrow 3 \uparrow 1 \leftarrow 2 \uparrow 1 \rightarrow 2 \uparrow 1 \leftarrow 2 \uparrow 1 \rightarrow 2 \uparrow 1$.

- 5°. Закодуйте наведене зображення літер (мал. 1.4) за таким правилом: стрілка вказує напрямок, у якому переміщується олівець по лінії сітки, а число вказує кількість клітинок для переміщення.
- 6*. Доповніть правила кодування, наведені в попередньому завданні, так, щоб можна було позначати переміщення олівця, при якому відбувається та при якому не відбувається малювання.
- 7°. Один з найдавніших шифрів розробив Полібій (III ст. до н. е.) – грецький історик, полководець, державний діяч. Цей шрифт отримав назву «квадрат Полібія». Кожна літера алфавіту (або пара літер, що відповідають близьким за вимовою звукам) міститься в таблиці. Під час кодування повідомлення кожна літера замінюється парою цифр – номером стовпця та номером рядка таблиці, на перетині яких вона розміщена. Для кодування повідомлень українською мовою можна використати таблицю із 6 стовпців і 5 рядків (мал. 1.5).



Мал. 1.4. До завдання 5

	1	2	3	4	5	6
1	А	Б	В	Г/Ґ	Д	Е
2	Є	Ж	З	И/Й	І/Ї	К
3	Л	М	Н	О	П	Р
4	С	Т	У	Ф	Х	Ц
5	Ч	Ш	Щ	Ь	Ю	Я

Мал. 1.5. Квадрат Полібія



Декодуйте з використанням наведеної таблиці повідомлення, закодоване за наведеним правилом:

- а) 53 43 13 52 21 52 42;
- б) 62 43 51 34 31 11 33 33 65.

8*. Закодуйте повідомлення, використавши правило кодування, наведене в попередньому завданні:

- а) *інформатика*;
- б) *повідомлення*.

9*. Запропонуйте власні правила кодування символів тексту та закодуйте з їх використанням повідомлення:

- а) *сьогодні гарний день*;
- б) *мені подобається футбол*.



10*. Знайдіть відомості та підготуйте повідомлення про способи кодування:

- а) азбука Морзе;
- б) семафорна азбука;
- в) шрифт Брайля;
- г) мова квітів «Селам».

1.2. Кодування текстових і графічних даних



1. Що таке *кодування повідомлень*? З якою метою здійснюють кодування повідомлень?
2. Наведіть приклади кодування текстових повідомлень.
3. Які види графіки ви знаєте? Охарактеризуйте їх.

Кодування текстових даних

Усі дані, які обробляються технічними пристроями, повинні бути закодовані певним чином. Для опрацювання текстових повідомлень з використанням комп'ютера символи повідомлення кодують числами.

Для кодування тексту використовують **таблиці кодів символів**, у яких кожному символу, що може бути використаний у текстовому повідомленні, поставлено у відповідність деяке число. У 1963 році у США було розроблено таблицю таких кодів символів, призначену для передавання повідомлень телетайпом. Пізніше вона стала стандартом для використання в комп'ютерній техніці й отримала назву **таблиця кодів символів ASCII** (англ. *American Standard Code for Information Interchange* – американський стандартний код для обміну інформацією).

У таблиці **ASCII** літерам англійського алфавіту, цифрам, розділовим знакам, символам редагування та форматування тексту поставлено у відповідність числа від 0 до 127 (табл. 1.1, с. 10).

Коди символів з таблиці **ASCII** можна знайти в додатку 1 за адресою <https://cutt.ly/NkBXn8o> або за адресою uk.wikipedia.org/wiki/ASCII.

Таблиця кодів символів **ASCII** містить коди літер лише англійського алфавіту. Для кодування літер інших алфавітів було розроблено інші таблиці кодів символів. Наприклад, таблиці **KOI8-U** (**KOI** – код обміну інформацією) та **Windows-1251** містять без змін усі коди таблиці **ASCII**, а також – коди літер кирилиці. Кодами цих літер є натуральні числа в діапазоні від 128 до 255.

Телетайп (лат. *tele* – віддалений, англ. *type* – друкування) – електромеханічна друкуюча машина, яку використовують для передавання текстових повідомлень дротами.

Групи кодів символів таблиці ASCII

Діапазон кодів	Група символів	Приклад коду	Відповідний символ
Від 0 до 31, 127	Спеціальні символи	10	Символ, що відповідає перенесенню курсора на новий рядок
		13	Символ, що відповідає поверненню курсора на початок рядка
		27	Символ, що відповідає натисканню клавіші Esc
Від 32 до 64, від 91 до 96, від 123 до 126	Розділові знаки та цифри	32	Пропуск (пробіл)
		48	Цифра 0
		123	{
Від 65 до 90	Великі літери англійського алфавіту	65	A
		66	B
		90	Z
Від 97 до 122	Малі літери англійського алфавіту	97	a
		98	b
		122	z

Таблиця кодів символів **Windows-1251** є стандартом для кодування літер кирилиці в операційній системі **Windows**. У ній, наприклад, літера «а» українського алфавіту має код 224, літера «і» – код 179, літера «ґ» – код 180 та ін. Коди символи з таблиці **Windows-1251** можна знайти в додатку 2 за адресою <https://cutt.ly/NkBXn8o> або за адресою uk.wikipedia.org/wiki/Windows-1251.

Кирилиця – алфавіт для групи слов'янських мов, до яких належать українська, російська, білоруська, болгарська та інші мови. Англійська мова та деякі інші європейські мови мають алфавіт, який називають **латиницею**.

Цілих чисел від 0 до 255 вистачає, щоб закодувати символи двох алфавітів – латиниці та кирилиці – та деякі інші символи. Але для кодування символів інших алфавітів (грецького та арабського алфавітів, ієрогліфів тощо) потрібно значно більше значень кодів. Для них розроблено таблицю кодів символів **Юнікод** (англ. **Universal Code, Unicode** – універсальний код).

Таблиця **Юнікод** складається із 17 наборів по 65 536 значень кодів у кожному та дає можливість закодувати 1 114 112 різних символів, тобто майже всі символи писемності всіх світових мов. Як і в інших таблицях кодів, у **Юнікоді** перші 128 кодів відповідають таблиці **ASCII**. Окремий розділ у таблиці **Юнікод** містить коди літер кирилиці. Наприклад, літера «а» українського алфавіту має код 1072, літера «і» – код 1110, літера «ґ» – код 1169 та ін.

Порівняння структури різних таблиць кодів символів наведено в таблиці 1.2.



Порівняння структури таблиць кодів символів

Таблиця кодів символів	Кількість кодів символів	Символи з кодами від 0 до 127	Символи з кодами від 128 до 255	Символи з кодами, більшими за 255
ASCII	128	Літери англійського алфавіту, цифри, розділові знаки, спеціальні символи	Немає	Немає
KOI8-U	256	Як у таблиці ASCII	Символи кирилиці та деякі інші символи	Немає
Windows-1251	256	Як у таблиці ASCII	Символи кирилиці та деякі інші символи	Немає
Юнікод	1 114 112	Як у таблиці ASCII	Символи алфавітів різних мов світу та деякі інші символи	



Для тих, хто хоче знати більше

У табличному процесорі Excel є функції, призначені для визначення коду заданого символу, а також за заданим кодом – відповідного символу в кодових таблицях **Windows-1251** та **Юнікод**:

- **CODE** – визначення коду заданого символу в кодовій таблиці **Windows-1251**;
- **UNICODE** – визначення коду заданого символу в кодовій таблиці **Юнікод**;
- **CHAR** – визначення символу за його числовим кодом у таблиці **Windows-1251**;
- **UNICHAR** – визначення символу за його числовим кодом у таблиці **Юнікод**.



Для тих, хто хоче знати більше

У таблиці **Юнікод** містяться коди не лише літер і цифр, а й символів, які позначають торгові марки, грошові одиниці, символи транскрипцій, ідеограми тощо. Наприклад, кодом символу української грошової одиниці гривні ₴ є число 8372, кодом

ідеограми чоловік ♂ є число 4036002432, а ідеограми жінка ♀ – число 4036002433 тощо (коди цих символів не є їхніми порядковими номерами).

Для вставлення символів у текстовий документ можна застосовувати їхні коди з таблиці **Юнікод**. Для цього потрібно натиснути та утримувати клавішу **Alt** і набрати код символу на додатковій цифровій клавіатурі. Так, можуть бути вставлені, наприклад, символи *сніговик* ❄ – код 9731, *серце у вигляді квітки* 💖 – код 10087, *нота* 🎵 – код 9834, *шахова фігура чорний кінь* ♞ – код 9822 та ін. Знайти символ і відповідний йому код можна на сайті **Unicode® Character Table** (англ. *Unicode Character Table* – таблиця символів Юнікоду) (unicode-table.com). Але якщо вибраний вами символ відсутній серед шрифтів на вашому комп'ютері, то він не буде відображатися в документі.

Ідеограма (давньогрец. *idéa* – ідея, *úramma* – писемний знак, літера) – писемний знак, що передає, на відміну від літери, не звук певної мови, а деяке ціле слово, поняття, ідею.

Кодування графічних даних

У комп'ютерній графіці важливим є кодування кольорів.

Існують різні правила кодування кольору – **колірні моделі**.

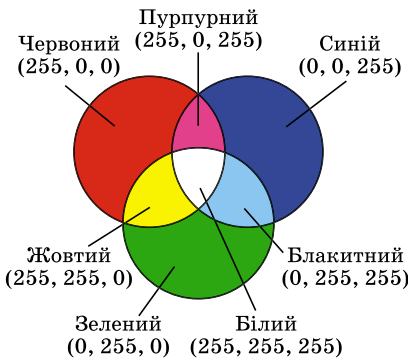
Колірна модель – це спосіб кодування різних кольорів спектра у вигляді впорядкованого набору числових значень певних **базових компонентів**.

Спектр (лат. *spectrum* – привид) – багатоколірна смуга, яка утворюється під час проходження білого світла через призму або якесь інше середовище, що заломлює світло.

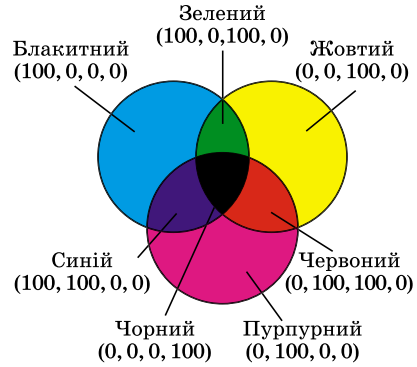
Найчастіше в комп'ютерній графіці використовують колірну модель **RGB**, якщо зображення буде відтворюватися на екрані, та модель **CMYK** для друку зображення.

У колірній моделі **RGB** (мал. 1.6) базовими компонентами є три кольори спектра – **червоний** (англ. *Red*), **зелений** (англ. *Green*) і **синій** (англ. *Blue*). Значення інтенсивності кожного компонента задається цілим числом від 0 до 255. Використовуючи модель **RGB**, можна закодувати $256^3 \approx 16,7$ млн кольорів.

У моделі **CMYK** використовують чотири базові компоненти: **блакитний** (англ. *Cyan*), **пурпурний** (англ. *Magenta*), **жовтий** (англ. *Yellow*), **чорний** (англ. *black*) кольори (мал. 1.7). Частка кожного з базових компонентів задається у відсотках (цілим числом від 0 до 100).



Мал. 1.6. Колірна модель RGB



Мал. 1.7. Колірна модель CMYK

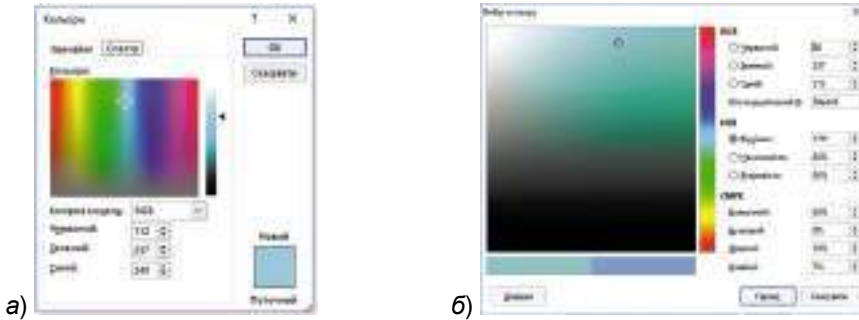
У таблиці 1.3 наведено приклади кодування кольорів.

Таблиця 1.3

Приклади кодування кольорів у колірних моделях RGB та CMYK

Колір	Модель RGB			Модель CMYK			
	R	G	B	C	M	Y	K
Білий	255	255	255	0	0	0	0
Сірий	122	122	122	0	0	0	50
Чорний	0	0	0	0	0	0	100
Смарагдовий	80	200	120	60	0	40	22
Рожевий	255	128	192	0	25	20	0

Ще одна популярна колірна модель **HSB** має три базові компоненти: **відтінок** (англ. *Hue*), **насиченість** (англ. *Saturation*) та **яскравість** (англ. *Brightness*).



Мал. 1.8. Визначення коду кольору під час роботи в програмах **Word** і **Excel** (а) та в програмі **LibreOffice Draw** (б)

Побачити код вибраного вами кольору можна під час роботи в різних програмних засобах. Наприклад, у програмах **Word** і **Excel** під час вибору кольору для заливки об'єктів подається код кольору в моделі **RGB** або **HSB** (мал. 1.8, а), а в програмі **LibreOffice Draw** – у моделях **RGB**, **HSB** та **CMYK** (мал. 1.8, б).

Кодування графічних даних виконується по-різному, залежно від виду графіки.

Як ви вже знаєте, растрове зображення складається з окремих пікселів. Для кодування растрового зображення визначається кількість пікселів у ньому й колір кожного пікселя кодують набором чисел.

Векторне зображення складається з окремих геометричних фігур (графічних примітивів): кіл, відрізків, кривих, прямокутників тощо. Кожен із графічних примітивів має: координати в площині зображення; форму, яку можна описати математичними формулами; параметри контуру та заливки, які кодуються числами. Кодування векторного зображення полягає в описі значень властивостей графічних примітивів, з яких складається зображення.

Наприклад, для побудови круга достатньо закодувати місце розміщення центра, радіус круга, товщину, стиль і колір лінії кола, колір заливки.



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/phdeK0J> або QR-кодом.



Найважливіше в цьому пункті

Для опрацювання текстових повідомлень з використанням комп'ютера символи повідомлення кодують числами. Для цього можуть використовуватися різні таблиці кодів символів, наприклад, **ASCII**, **KOI8-U**, **Windows-1251**, **Юнікод**.

Таблиця кодів символів **ASCII** містить коди 128 символів, **KOI8-U**, **Windows-1251** – 256 символів, **Юнікод** – 1 114 112 символів. У всіх таблицях кодів символів незмінними залишаються перші 128 кодів, що відповідають таблиці **ASCII**.

Колірна модель – це спосіб кодування різних кольорів спектра у вигляді впорядкованого набору числових значень певних **базових компонентів**.

Найчастіше використовують колірну модель **RGB**, якщо зображення буде відтворюватися на екрані, та модель **CMYK** для друку зображення.

Для кодування растрового зображення визначається кількість пікселів у ньому й колір кожного пікселя кодують набором чисел.

Кодування векторного зображення полягає в описі значень властивостей графічних примітивів, з яких складається зображення.



Дайте відповіді на запитання

- 1^о. Як кодують символи текстових повідомлень під час опрацювання їх з використанням комп'ютера?
- 2^о. Які таблиці кодів символів ви знаєте?
- 3^о. Які символи можуть бути закодовані з використанням таблиць кодів символів **ASCII**, **KOI-8U**, **Windows-1251**, **Юнікод**?
- 4*. Що спільного та чим відрізняються таблиці кодів символів **ASCII**, **KOI-8U**, **Windows-1251**, **Юнікод**?
- 5^о. Що таке колірна модель?
- 6^о. Які колірні моделі ви знаєте? Охарактеризуйте їх.
- 7^о. У чому полягає кодування растрового графічного зображення?
- 8^о. У чому полягає кодування векторного графічного зображення? Наведіть приклади.



Виконайте завдання



- 1^о. Повідомлення закодовано з використанням таблиці кодів символів **ASCII**. Декодуйте його (скористайтеся додатком 1, адреса <https://cutt.ly/NkBXn8o>):
а) 83 117 110; б) 104 111 109 101.
- 2^о. Закодуйте повідомлення кодами з таблиці кодів символів **Windows-1251** (скористайтеся додатком 2, адреса <https://cutt.ly/NkBXn8o>):
а) *Інформатика*; б) ваше прізвище та ім'я.
- 3*. Окремі слова в повідомленні закодовано з використанням таблиці кодів символів **Юнікод**. Знайдіть символи за їхніми кодами в таблиці та декодуйте повідомлення. Запишіть отриманий текст. Для пошуку символів можна використовувати сайт **Unicode® Character Table** (unicode-table.com):
Сьогодні зранку 9925. Синоптики обіцяють 9928. Збираючись у 127979, не забудьте взяти із собою 9730.



- 4*. Визначте, які символи будуть усталені до текстового документа шляхом уведення їхнього коду за натиснутої клавіші **Alt** для таких кодів: 1234, 3754, 8986, 9708, 9415, 10001, 10239.

- 5^о. Відкрийте файл **Розділ 1\Пункт 1.2\завдання 1.2.5.xlsx**. У клітинках електронної таблиці записано коди кольорів колірної моделі **RGB**. Зафарбуйте клітинки в кольори відповідно до наведених кодів. Значення кольорів установіть на вкладці **Спектр** у вікні вибору кольору заливки. Збережіть електронну таблицю у вашій папці у файлі з тим самим іменем.



- 6*. Створіть на аркуші **Аркуш 1** електронної таблиці зображення, розфарбувавши клітинки в різні кольори. Закодуйте зображення, розмістивши у відповідних клітинках на **Аркуші 2** коди кольорів зафарбованих клітинок отриманого зображення згідно з колірною моделлю **RGB**. Збережіть електронну таблицю у вашій папці у файлі з іменем **завдання 1.2.6.xlsx**.

1.3. Двійкове кодування



1. У чому полягають процеси кодування та декодування повідомлень?
2. Які таблиці кодів символів використовують для кодування текстових повідомлень?
3. Що означають префікси *кіло*, *мега*, *гіга*?



Двійкове кодування. Біт і байт

Як ви вже знаєте, для кодування даних використовують різні набори сигналів. Для кодування звуків мови використовують літери алфавіту. В українському алфавіті 33 літери, в англійському – 26. Для кодування числових значень використовуються 10 цифр. Для кодування команд, що регулюють порядок руху на пішохідних переходах, використовують 3 кольори світлофора.

Найменшу кількість сигналів для кодування повідомлень має азбука Морзе – лише 2 сигнали: *крапка* та *тире*.

З 1844 року азбуку, яку запропонував американський художник Семюел Морзе (1791–1872) і яку назвали на його честь, використовували для передавання повідомлень телеграфом (мал. 1.9).



Мал. 1.9. Семюел Морзе та телеграф

В азбуці Морзе літери текстових повідомлень кодуються послідовностями крапок і тире. Приклади кодування літер українського алфавіту наведено в таблиці 1.4.

Таблиця 1.4

Приклади кодування літер в азбуці Морзе

Літера	Значення коду в азбуці Морзе
а	· –
о	– – –
п	· – – ·
р	· – ·
т	–

Наприклад, слово *парта* кодується з використанням азбуки Морзе послідовністю сигналів «· – – · · – · – – · – –».

Кодування повідомлень з використанням сигналів лише двох видів називають **двійковим кодуванням**. Повідомлення, отримане в результаті двійкового кодування повідомлення, називають **двійковим кодом** повідомлення.

Двійкове кодування використовується і в сучасних комп'ютерах. У них усі повідомлення кодуються послідовностями сигналів двох видів. Кожний сигнал одного виду умовно позначається цифрою **0**, а другого виду – **1**.

Цифра 0 або 1 у двійковому коді повідомлення називається **бітом** (англ. *binary digit* – двійкова цифра).

З таблиці 1.4 видно, що в азбуці Морзе для кодування літери «т» використовується 1 сигнал – тире, для кодування літери «а» – 2, а для літери «п» – 4 сигнали. Під час передавання повідомлень телеграфом одну літеру від іншої відокремлюють проміжками в подачі сигналів. Для опрацювання даних з використанням комп'ютера це не зручно. Простіше відокремити одну літеру від іншої, коли на їх кодування буде використано однакову кількість сигналів.

Якщо для кодування повідомлення використовується один двійковий сигнал (біт) 0 або 1, то можна закодувати, наприклад:

- висновок про правильність твердження: *хибне* – 0 або *істинне* – 1;
 - стать людини: *жіноча* – 0 або *чоловіча* – 1;
 - стан вимикача: *вимкнено* – 0 або *ввімкнено* – 1
- тощо.

З двох бітів можна скласти 4 (2^2) різних кодів (00, 01, 10 і 11). Ними можна закодувати, наприклад, чотири сторони горизонту: 00 – північ, 01 – схід, 10 – південь, 11 – захід.

З трьох бітів можна скласти вже 8 (2^3) різних кодів (000, 001, 010, 011, 100, 101, 110, 111). Ними можна закодувати, наприклад, номери рядків або стовпців шахівниці.

Із чотирьох бітів можна скласти $2^4 = 16$ кодів, з п'яти – $2^5 = 32$ коди і т. д.

З восьми бітів можна скласти $2^8 = 256$ кодів, і цієї кількості кодів достатньо, щоб закодувати всі літери англійського та українського (або якогось іншого) алфавітів, арабські цифри, розділові знаки, знаки арифметичних дій, а також деякі інші символи. Саме така кількість кодів міститься, наприклад, у таблиці кодів символів **Windows-1251**.

Послідовність із восьми бітів називається **байтом** і позначається **Б**:

$$1 \text{ Б} = 8 \text{ бітів.}$$

Довжина двійкового коду повідомлень

Якщо символ повідомлення кодується послідовністю з 8 бітів, то довжина двійкового коду цього символу дорівнює 8 бітів або 1 Б.

Довжина двійкового коду повідомлення – це кількість бітів або байтів у двійковому коді цього повідомлення.

Наприклад, повідомлення *Інформатика – цікавий предмет!* містить 30 символів (включаючи символи пропусків, тире, знак оклику). Якщо кожний символ кодувати двійковим кодом завдовжки 1 Б, то довжина двійкового коду такого повідомлення дорівнюватиме 30 Б.

Як ви вже знаєте, у таблиці кодів символів **Windows-1251** кожний символ має код – ціле число від 0 до 255. У свою чергу, кожне із цих чисел може бути закодовано одним байтом. Приклади такого кодування наведено в таблиці 1.5.

Таблиця 1.5

Приклади двійкового кодування

Символ	Число	Відповідний байт
!	33	00100001
@	64	01000000
W	103	01100111
ю	254	11111110
я	255	11111111

Таким чином, кожен символ у таблиці кодів символів **Windows-1251** має двійковий код завдовжки 1 Б.

Щоб виконати двійкове кодування текстового повідомлення, можна кожний його символ замінити двійковим кодом, що відповідає числу з таблиці **Windows-1251**.



Саме так кодуються текстові документи, створені в текстовому редакторі **Блокнот**, при їх збереженні на носії даних, якщо під час збереження використовувати кодування **ANSI** (англ. **American National Standards Institute** – Американський національний інститут стандартів). Довжина двійкового коду повідомлення, що міститься в такому документі, буде дорівнювати в байтах кількості символів у текстовому документі. Треба врахувати, що кожне натискання клавіші **Enter** кодується двома спеціальними символами – символом переходу на новий рядок і символом повернення на початок рядка.

Під час кодування текстів з використанням кодової таблиці **Юнікод** довжина двійкового коду різних символів є різною – 1 Б, 2 Б, 4 Б і більше.

Графічні, звукові та відеоповідомлення для їх опрацювання комп'ютером також кодуються двійковими кодами.

Довжину двійкового коду повідомлень також називають **обсягом даних**.

Кратні одиниці вимірювання довжини двійкового коду

Для позначення довжини двійкового коду повідомлень використовують кратні одиниці вимірювання, які утворюються з використанням префіксів *кіло*, *мега*, *гіга*, *тера* та ін. Перелік цих префіксів для позначення кратних одиниць вимірювання визначено в Міжнародній системі одиниць (СІ). Там також наведено відповідні множники. Але історично склалося так, що ці префікси (*кіло*, *мега*, *гіга*, *тера*) в інформатиці трактуються інакше, ніж, наприклад, у фізиці:

$$1 \text{ кБ (кілобайт)} = 2^{10} \text{ Б} = 1024 \text{ Б};$$

$$1 \text{ МБ (мегабайт)} = 2^{10} \text{ кБ} = 2^{20} \text{ Б} = 1\,048\,576 \text{ Б};$$

$$1 \text{ ГБ (гігабайт)} = 2^{10} \text{ МБ} = 2^{20} \text{ кБ} = 2^{30} \text{ Б};$$

$$1 \text{ ТБ (терабайт)} = 2^{10} \text{ ГБ} = 2^{20} \text{ МБ} = 2^{30} \text{ кБ} = 2^{40} \text{ Б}.$$

Так сталося тому, що в комп'ютері використовуються двійкові коди, для опрацювання яких зручно оперувати степенями числа 2, а не степенями числа 10. І оскільки $2^{10} = 1024$, що приблизно дорівнює 1000, то саме $2^{10} \text{ Б} = 1024 \text{ Б}$ і стали називати **кілобайт**. Аналогічно 2^{10} кілобайтів стали називати **мегабайт** і т. д.

Одиниці вимірювання довжини двійкового коду використовують для оцінки розмірів файлів, ємності носіїв даних, об'єму даних, що передаються мережами, та швидкості їх передавання тощо.



Для тих, хто хоче знати більше

У 1999 році Міжнародна електротехнічна комісія ввела новий стандарт іменування двійкових префіксів. За цим стандартом 1 кБ вважається рівним 1000 Б, а величина 1024 Б отримала нову назву – 1 **кібібайт** (табл. 1.6).

Таблиця 1.6

Префікси та їх значення, затверджені Міжнародною електротехнічною комісією

Кількість байтів					
Префікси СІ			Бінарні префікси		
Назва	Скорочення	Значення	Назва	Скорочення	Значення
кілобайт	кБ	10^3	кібібайт	КіБ	2^{10}
мегабайт	МБ	10^6	мебібайт	МіБ	2^{20}
гігабайт	ГБ	10^9	гібібайт	ГіБ	2^{30}
терабайт	ТБ	10^{12}	тебібайт	ТіБ	2^{40}

Перехід на традиційні значення префіксів *кіло*, *мега*, *гіга*, *тера* та інші під час вимірювання довжини двійкового коду відбувається досить повільно. У цьому підручнику, як це склалося історично, будемо вважати, що 1 кілобайт = 1024 байти.

Перехід від одних одиниць вимірювання довжини двійкового коду до інших

Довжину двійкового коду повідомлень можна вказувати в бітах, байтах і в кратних їм одиницях. Розглянемо кілька задач на перехід під час визначення довжини двійкового коду повідомлення від одних одиниць вимірювання до інших.

Задача 1. Довжина двійкового коду повідомлення складає 4,5 МБ. Виразити це значення в байтах. Скільки символів у кодуванні **Windows-1251** буде закодовано?

Розв'язання. Для переходу від мегабайтів до байтів спочатку виразимо довжину двійкового коду цього повідомлення в кілобайтах. Ураховуючи, що 1 МБ = 1024 кБ, маємо $4,5 \text{ МБ} \cdot 1024 = 4608 \text{ кБ}$.

Переведемо кілобайти в байти, ураховуючи, що 1 кБ = 1024 Б:
 $4608 \text{ кБ} \cdot 1024 = 4\,718\,592 \text{ Б}$.

Відповідь: 4,5 МБ = 4 718 592 Б. Оскільки в таблиці **Windows-1251** один символ кодується одним байтом, то закодовано буде 4 718 592 символи.

Задача 2. Двійковий код повідомлення складається з 4 194 304 000 бітів. Виразити довжину двійкового коду цього повідомлення цілим числом у найбільших можливих одиницях.

Розв'язання. Переведемо біти в байти, ураховуючи, що 1 Б = 8 бітів:
 $4\,194\,304\,000 \text{ бітів} : 8 = 524\,288\,000 \text{ Б}$.

Отримане число більше за множник 1024, який використовується для переходу від байтів до кілобайтів. Тому переведемо байти в кілобайти:

$524\,288\,000 \text{ Б} : 1024 = 512\,000 \text{ кБ}$.

Аналогічно міркуючи, переведемо кілобайти в мегабайти:
 $512\,000 \text{ кБ} : 1024 = 500 \text{ МБ}$.

Отримане значення менше ніж 1024, тому перейти до більшої одиниці зі збереженням цілого результату неможливо.

Відповідь: 4 194 304 000 бітів = 500 МБ.

Задача 3. Обчислити наближено довжину двійкового коду тексту підручника з інформатики (не враховуючи форматування та малюнки).

Розв'язання. Вважатимемо, що текст закодовано з використанням таблиці кодів символів **Windows-1251**, тобто довжина двійкового коду кожного символа складає 1 Б.

У кожному рядку тексту міститься приблизно 60 символів, тобто довжина двійкового коду рядка складає приблизно 60 Б.

На кожній сторінці розміщено в середньому 50 рядків. Отже, довжина двійкового коду сторінки складає приблизно $60 \cdot 50 = 3000$ (Б).

У підручнику приблизно 250 сторінок, тобто довжина двійкового коду підручника становить $3000 \cdot 250 = 750\,000$ (Б).

Перейдемо до більшої одиниці вимірювання довжини двійкового коду:
 $750\,000 \text{ Б} : 1024 \approx 732,4 \text{ кБ}$.

Відповідь: довжина двійкового коду тексту підручника з інформатики становить приблизно 732,4 кБ.



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/whdeZob> або QR-кодом.



Найважливіше в цьому пункті

Кодування повідомлень з використанням сигналів лише двох видів називається **двійковим кодуванням**. Повідомлення, отримане в результаті двійкового кодування початкового повідомлення, називають **двійковим кодом** повідомлення.

Цифра 0 або 1 у двійковому коді повідомлення називається **бітом**.

Послідовність із восьми бітів називається **байтом**:

$$1 \text{ Б (байт)} = 8 \text{ бітів.}$$

Довжина двійкового коду повідомлення – це кількість байтів у двійковому коді цього повідомлення.

Кожному символу в таблиці кодів символів **Windows-1251** відповідає двійковий код завдовжки 1 Б.

$$1 \text{ кБ (кілобайт)} = 2^{10} \text{ Б} = 1024 \text{ Б}$$

$$1 \text{ МБ (мегабайт)} = 2^{10} \text{ кБ} = 2^{20} \text{ Б} = 1\,048\,576 \text{ Б}$$

$$1 \text{ ГБ (гігабайт)} = 2^{10} \text{ МБ} = 2^{20} \text{ кБ} = 2^{30} \text{ Б}$$

$$1 \text{ ТБ (терабайт)} = 2^{10} \text{ ГБ} = 2^{20} \text{ МБ} = 2^{30} \text{ кБ} = 2^{40} \text{ Б}$$



Дайте відповіді на запитання

- 1°. У чому полягає двійкове кодування повідомлень?
- 2°. Що таке 1 біт?
- 3°. Чому дорівнює 1 байт?
- 4°. Що таке довжина двійкового коду повідомлення?
- 5°. Якою є довжина двійкового коду кожного символу в таблиці кодів символів **Windows-1251**?
- 6°. Чому дорівнюватиме довжина двійкового коду повідомлення, створеного в текстовому редакторі **Блокнот** і збереженого в кодуванні **ANSI**? У кодуванні **UTF-8**?
- 7*. На скільки буде відрізняться довжина двійкового коду повідомлення, у якому зазначено ваше ім'я, якщо його створити в текстовому редакторі **Блокнот** і зберегти в кодуванні **ANSI** у двох варіантах: записаним в один рядок і записаним у стовпчик по одній літері в рядку?
- 8°. Чому дорівнює: 1 кілобайт; 1 мегабайт; 1 гігабайт; 1 терабайт?
- 9°. Яку математичну операцію потрібно виконати для переведення під час вимірювання довжини двійкового коду повідомлень:
 - а) з бітів у байти;
 - б) з байтів у кілобайти;
 - в) з мегабайтів у кілобайти?



Виконайте завдання

- 1°. Наведіть приклади, де використовується двійкове кодування повідомлень.
- 2°. Закодуйте з використанням азбуки Морзе слова *порт*, *рота*, *торт*. Порівняйте кількість сигналів у двійковому коді цих слів.

3°. Визначте довжину двійкового коду повідомлення за умови, що повідомлення закодовано з використанням таблиці кодів символів **Windows-1251**:

а) *Таблиця кодів символів Windows-1251.*

б) *В алфавіті азбуки Морзе два символи – крапка та тире.*



4°. Порівняйте довжину двійкового коду текстового повідомлення *Людина без друзів – що дерево без коріння*, створеного в текстовому редакторі **Блокнот**, якщо зберегти його з використанням різних кодових таблиць.

5°. Заповніть пропуски:

а) 5 Б = _____ бітів;

в) 10 кБ = _____ бітів;

б) 4096 Б = _____ кБ;

г) 3 МБ = _____ Б.



6°. Заповніть пропуски:

а) 2 Б = _____ бітів;

в) 2 кБ = _____ бітів;

б) 8192 Б = _____ кБ;

г) 100 МБ = _____ кБ.

7°. В одному рядку деякого тексту розміщується в середньому 60 символів, а на одній сторінці – 40 рядків. Скільки закодованих сторінок цього тексту можна взяти, щоб довжина двійкового коду не перевищила 8 МБ, якщо для кодування використати таблицю **Windows-1251**?

8°. Швидкість передавання даних локальною мережею складає 100 мегабіт за секунду. Скільки часу знадобиться для передавання цієї мережею відеофайлу, якщо довжина його двійкового коду складає 4 ГБ?

ПРАКТИЧНА РОБОТА № 1

«Розв'язування задач на визначення довжини двійкового коду повідомлень»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Обчисліть довжину двійкового коду деякого тексту за умови, що символи тексту закодовано з використанням таблиці кодів символів **Windows-1251**.

У тексті 32 сторінки, кожна сторінка тексту містить 48 рядків, у кожному рядку в середньому 56 символів. Виразіть довжину двійкового коду цього тексту цілим числом у найбільших можливих одиницях.

2. Визначте, використовуючи текстовий редактор **Блокнот**, довжину двійкового коду повідомлення *Практична робота № 1* за умови, що повідомлення закодовано з використанням таблиці кодів символів:

а) **Windows-1251** (кодування **ANSI**);

б) **Юнікод** (кодування **UTF-8**).

Збережіть текстові документи з повідомленням у файлах з іменами **практична 1 Windows.txt** і **практична 1 Юнікод.txt**.

3. Заповніть у табличному процесорі таблицю:

<i>Бітів</i>	<i>Байтів</i>	<i>Кілобайтів</i>	<i>Мегабайтів</i>
614 400			
	524 288		
		256	
			4

Збережіть результат у файлі з іменем **практична 1.xlsx**.

Розділ 2.

Апаратне забезпечення комп'ютера

У цьому розділі ви дізнаєтеся про:

- персональний комп'ютер
- пристрої комп'ютера, їх призначення та основні характеристики:
 - процесор
 - пристрої пам'яті
 - пристрої введення та виведення даних
 - мультимедійні пристрої
- історію обчислювальних і комп'ютерних пристроїв
- види сучасних комп'ютерів та їх застосування

2.1. Персональний комп'ютер, його основні складові



1. Які пристрої входять до складу комп'ютера? Для чого вони призначені?
2. Який пристрій комп'ютера виконує опрацювання даних? Де зазвичай він розміщується?
3. Назвіть інформаційні процеси. Які пристрої комп'ютера забезпечують збереження даних?

Комп'ютер та інформаційні процеси

Раніше на уроках з інформатики ви вже ознайомилися зі складовими персонального комп'ютера та основними інформаційними процесами. Розглянемо детальніше схему реалізації інформаційних процесів у комп'ютері з використанням його пристроїв (мал. 2.1). Дані потрапляють до внутрішніх запам'ятовуючих пристроїв (**внутрішньої пам'яті**) комп'ютера через пристрої введення даних (клавіатура, планшет, мікрофон, миша, сканер тощо) або із зовнішніх запам'ятовуючих пристроїв (**зовнішньої пам'яті**) – пристроїв для роботи з жорсткими магнітними та оптичними дисками, флешнакопичувачів, флешкарт тощо.

З внутрішньої пам'яті дані потрапляють до **процесора**, відбувається їхнє опрацювання. Результати опрацювання даних знову передаються до внутрішньої пам'яті, а звідти – до пристроїв виведення даних (монітор, принтер, планшет, звукові колонки тощо) або до пристроїв зовнішньої пам'яті.

Усі операції в комп'ютері, пов'язані з реалізацією інформаційних процесів, можливі тільки під керуванням відповідних програм.



Мал. 2.1. Схема реалізації інформаційних процесів у персональному комп'ютері

Процесор, його призначення

Ви вже знаєте, що опрацювання даних у комп'ютері виконується процесором (мал. 2.2). Він є пристроєм, який забезпечує виконання комп'ютерних програм. У процесорів, як у будь-якого виконавця алгоритмів, є свій набір команд.

Процесор комп'ютера є його основною складовою. Він містить **пристрій керування**, який забезпечує виконання команд комп'ютерної програми, та **арифметико-логічний пристрій**, який здійснює операції над даними. До складу сучасних процесорів входить ще й складова внутрішньої пам'яті – так звана **кеш-пам'ять**. Її призначення розглянемо дещо згодом.



Мал. 2.2. Процесор AMD Ryzen 7 з 8-ма ядрами

Властивостями процесора є: **тактова частота роботи процесора**, **кількість ядер**, **розрядність**, **обсяг кеш-пам'яті** тощо (табл. 2.1). Значення цих та інших властивостей визначають **швидкість опрацювання даних** процесором.

Таблиця 2.1

Приклади значень властивостей процесора для настільного й планшетного комп'ютерів

Властивість	Що характеризує	Значення властивостей процесора	
		Для настільного комп'ютера	Для планшетного комп'ютера
Модель	Ім'я, що надається фірмою-виробником	AMD Ryzen Threadripper 3990X	KIRIN970

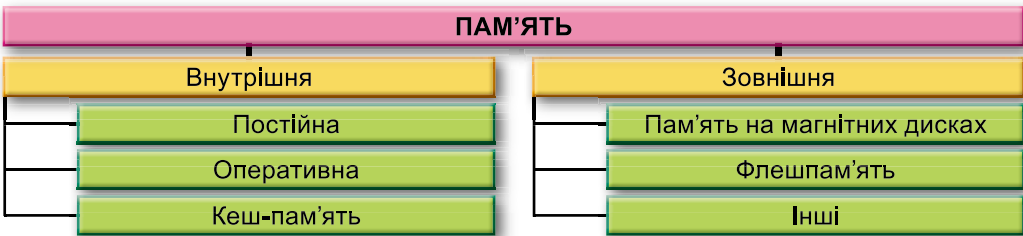


Продовження таблиці 2.1

Властивість	Що характеризує	Значення властивостей процесора	
		Для настільного комп'ютера	Для планшетного комп'ютера
Тактова частота	Частота керуючих сигналів, які узгоджують роботу пристроїв процесора	4,3 ГГц	2,36 ГГц
Кількість ядер	Кількість однакових за структурою процесорів, які об'єднано в один пристрій	64	8
Розрядність	Кількість двійкових розрядів, які може одночасно опрацювати процесор	64 біти	64 біти

Пам'ять комп'ютера. Внутрішня пам'ять

Пам'ять комп'ютера призначена для зберігання даних. Її поділяють на **внутрішню** та **зовнішню** (мал. 2.3). Основною властивістю всіх видів пам'яті є її ємність, яка вимірюється в байтах, кілобайтах, мегабайтах, гігабайтах, терабайтах тощо.



Мал. 2.3. Види пам'яті комп'ютера

Внутрішня пам'ять призначена для забезпечення роботи процесора. Її складові (оперативна пам'ять, постійна пам'ять, кеш-пам'ять), як і процесор, розміщуються на материнській платі комп'ютера (мал. 2.4).

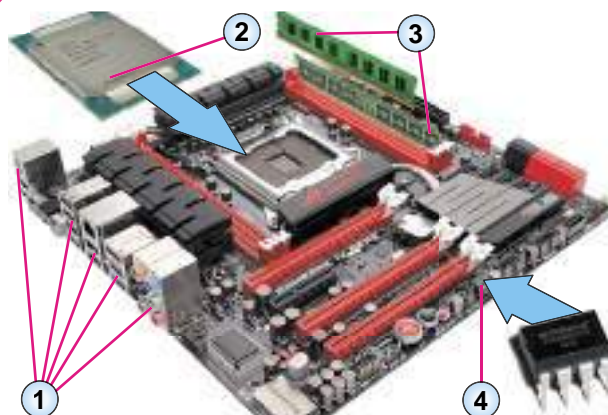
Оперативна пам'ять є основною в комп'ютері, у ній розміщуються програми й дані, які в подальшому опрацьовуються процесором. Щоб дані були опрацьовані процесором, вони попередньо повинні бути передані («завантажені») з пристроїв введення даних або запам'ятовуючих пристроїв до оперативної пам'яті.

Після вимкнення живлення комп'ютера всі дані з оперативної пам'яті зникають.

Оперативна пам'ять виготовляється у вигляді плати блоків мікросхем, яка встановлюється в спеціальні роз'єми (слоти) на материнській платі (мал. 2.4, 3). Зазвичай на материнській платі можна встановити кілька плат мікросхем оперативної пам'яті.

Материнська, або головна, плата (англ. *motherboard* або *mainboard*) – основна складова сучасних персональних комп'ютерів, яка забезпечує передавання даних між пристроями комп'ютера.

Слот (англ. *slot*) – щілина, паз, отвір.
Сокет (англ. *socket*) – заглиблення, гніздо, розетка.



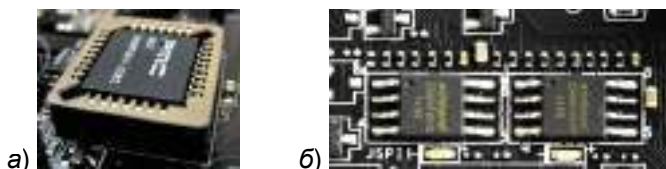
1. Роз'єми для підключення пристроїв введення та виведення даних
2. Процесор
3. Плати блоків мікросхем оперативної пам'яті
4. Мікросхема постійної пам'яті

Мал. 2.4. Материнська плата комп'ютера

Як уже зазначалося, до складу сучасних процесорів включають **кеш-пам'ять**. Вона призначена для прискорення обміну даними між оперативною пам'яттю та процесором. Розрізняють кеш-пам'ять першого, другого й третього рівнів. Ємність кеш-пам'яті першого рівня в більшості процесорів – 128 кБ. Ємності кеш-пам'яті другого й третього рівнів суттєво відрізняються в різних процесорах. На кінець 2020 року кеш-пам'ять другого рівня в процесорах для настільних персональних комп'ютерів становила від 1 до 32 МБ, а третього – від 2 до 256 МБ. Збільшення ємності кеш-пам'яті зазвичай збільшує швидкість процесора.

Програми та дані, які потрібні для виконання дій, пов'язаних з початком роботи комп'ютера, зберігаються в **постійній пам'яті**. Після ввімкнення комп'ютера дані з постійної пам'яті копіюються («завантажуються») в оперативну пам'ять і запускаються програми підготовки комп'ютера до роботи.

Постійна пам'ять виготовляється у вигляді однієї або двох мікросхем, що вставляються в спеціальні гнізда – сокети (мал. 2.5, а), або впаюються на материнській платі (мал. 2.5, б).



Мал. 2.5. Мікросхеми постійної пам'яті на материнській платі

Зовнішня пам'ять

Зовнішня пам'ять призначена для довготривалого зберігання значних масивів даних. Від внутрішньої пам'яті її відрізняє не тільки призначення, але й відмінності в значеннях окремих властивостей:

- збільшений обсяг даних (у персональних комп'ютерах – до 25 ТБ);
- збереження даних після вимкнення комп'ютера (енергонезалежність);
- можливість приєднання окремих пристроїв зовнішньої пам'яті без вимкнення комп'ютера тощо.

Основними носіями даних зовнішньої пам'яті є жорсткі магнітні диски та електронні схеми. Раніше в комп'ютерах як носії даних активно використовували гнучкі магнітні диски, оптичні диски та магнітні стрічки.



Носії даних розміщуються у відповідних пристроях. Жорсткі магнітні диски є носіями даних у пристроях і їх називають **накопичувачі на жорстких магнітних дисках** (НЖМД). Ці пристрої зовнішньої пам'яті можуть мати один або кілька металевих дисків, покритих шаром магнітної речовини (мал. 2.6). Запис даних виконується шляхом намагнічування ділянок поверхні диска з використанням електромагнітної головки, яка також і зчитує дані.



1. Поверхня одного з жорстких дисків
2. Електромагнітна головка

Мал. 2.6. Будова накопичувача на жорстких магнітних дисках

Властивості накопичувачів на жорстких магнітних дисках і приклади їх значень у сучасних пристроях подано в таблиці 2.2.

Таблиця 2.2

Значення властивостей сучасних НЖМД

Властивість	Значення властивості НЖМД для	
	настільних комп'ютерів	ноутбуків
Ємність	1–16 ТБ	0,5–5 ТБ
Діаметр жорстких дисків	3,5 дюйма	2,5 дюйма
Швидкість обертання жорстких дисків	5400–15 000 об/хв	5400–10 000 об/хв

Зовнішню пам'ять на електронних мікросхемах ще називають флешпам'яттю. Вона реалізується в пристроях пам'яті трьох основних типів:

- USB-флешнакопичувач («флешка»);
- флешкарта (карта пам'яті);
- твердотілий, або SSD-накопичувач.

USB-флешнакопичувачі використовують в основному для обміну даними між різними комп'ютерами. Для їх підключення до комп'ютера застосовують USB-роз'єми. Більшість USB-флешнакопичувачів мають ємність від 4 до 128 ГБ, хоча є пристрої і значно більшої ємності – до 2 ТБ.

Різноманітні **флешкарти** (мал. 2.7) здебільшого використовують як пристрої пам'яті для смартфонів, фото- та відеокамер тощо. Для їх підключення застосовують, залежно від типу флешкарти, відповідні слоти. Найпопулярнішими на сьогодні у зв'язку з їхнім широким використанням у смартфо-



Мал. 2.7. Флешкарти різних типів

нах є флешкарти типу **microSD**. Ємність флешкарт коливається в межах від 4 до 512 ГБ.

Твердотілі, або SSD-накопичувачі (мал. 2.8), мають таке саме призначення, як і накопичувачі на жорстких магнітних дисках. Здебільшого їх використовують як

SSD (англ. *Solid-State Drive*) – твердо-тілий (суцільний) пристрій.

основні пристрої зовнішньої пам'яті для мобільних комп'ютерів – ноутбуків, планшетних комп'ютерів. Порівняно з НЖМД SSD-накопичувачі мають більшу швид-

кість обміну даних, але й більшу вартість за однакової ємності. За максимальною ємністю SSD-накопичувачі також поки що поступаються накопичувачам на жорстких магнітних дисках – до 8 ТБ.



Мал. 2.8. SSD-накопичувач



Мал. 2.9. Підключення зовнішнього накопичувача

Для підвищення швидкодії комп'ютера за рахунок збільшення швидкості обміну даними значна частина ноутбуків оснащується двома пристроями зовнішньої пам'яті – більш «швидким» SSD-накопичувачем (для розміщення програм, які дуже часто використовуються комп'ютером, наприклад операційної системи) і значно «повільнішим» накопичувачем на жорстких магнітних дисках для забезпечення потрібної ємності зовнішньої пам'яті.

Для створення резервних копій даних, банку конфіденційних даних значного обсягу, переміщення великих обсягів даних без використання комп'ютерних мереж та інших цілей можуть використовуватися так звані *зовнішні накопичувачі* на жорстких магнітних дисках або SSD-накопичувачі. На відміну від *внутрішніх* пристроїв, які підключаються до материнської плати через відповідні роз'єми й розміщуються в системному блоці або корпусі мобільного комп'ютера, зовнішні пристрої підключаються до комп'ютера з використанням USB-роз'ємів (мал. 2.9).



Мал. 2.10. Завантаження IBM 350 Disk File в багажний відсік літака



Чи знаєте ви, що...

Один з перших накопичувачів на жорстких магнітних дисках розробила корпорація IBM, і в 1956 році його було представлено для продажу. Цей пристрій отримав назву **IBM 350 Disk File**, мав обсяг «цілих» 5 Мбайт і складався з 50 металевих дисків, вкритих шаром магнітної речовини. Діаметр дисків – 24 дюйми або 60,96 см, а маса всього пристрою перевищувала тону (мал. 2.10).



Чи знаєте ви, що...

Визначним ученим, який зробив суттєвий внесок у розробку технологій створення сучасних накопичувачів на магнітних дисках, є **Любомир Романків** (нар. 17.04.1931, м. Жовква, Львівська обл.) – американський учений українського походження, у минулому – провідний співробітник корпорації ІВМ. Він розробив технологію створення головок запису/зчитування даних з поверхні магнітних дисків, що значно зменшило витрати на виготовлення НЖМД. Ця технологія уможливила виготовлення пристроїв, що могли розміститися в корпусах персональних комп'ютерів. Л. Романків є співавтором понад 60 патентів.



Мал. 2.11. Любомир Романків



Для тих, хто хоче знати більше

Тривалий час як носії даних використовували оптичні диски, які відрізнялися ємністю та можливістю здійснювати запис і перезапис даних. Основні типи оптичних дисків, їх властивості та значення властивостей подано в таблиці 2.3.

Таблиця 2.3

Основні типи оптичних дисків

Позначення дисків, призначених для			Ємність дисків
тільки зчитування даних	зчитування та запису даних	зчитування, запису та перезапису даних	
CD-ROM	CD-R	CD-RW	640–800 МБ
DVD-ROM	DVD-R, DVD+R	DVD-RW, DVD+RW	9,4 ГБ
BD-ROM	BD-R	BD-RE	100 ГБ

У пристроях для роботи з **оптичними дисками** використовують промінь лазера для зчитування та запису даних.



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/5hdeX36> або QR-кодом.



Найважливіше в цьому пункті

Обов'язковими компонентами персонального комп'ютера є пристрої, які використовуються для **введення, виведення, опрацювання та зберігання** даних.

Процесор – це пристрій, який здійснює опрацювання даних. Він містить **пристрій керування**, який забезпечує виконання команд комп'ютерної програми, та **арифметично-логічний пристрій**, який здійснює операції над даними. Властивостями процесора є: **тактова частота роботи процесора, кількість ядер, розрядність, обсяг кеш-пам'яті** тощо (табл. 2.1). Значення цих та інших властивостей визначають **швидкість опрацювання даних процесором**.

Пам'ять комп'ютера призначена для зберігання даних, її поділяють на **внутрішню та зовнішню**.

Пристрої **внутрішньої пам'яті** (оперативна, постійна, кеш-пам'ять) розміщуються на материнській платі. Дані з оперативної пам'яті та кеш-пам'яті після вимкнення живлення зникають, а з постійної – ні.

Пристрої **зовнішньої пам'яті** (накопичувачі на жорстких магнітних дисках, пристрої, що використовують флешпам'ять, та ін.) призначені для довготривалого зберігання даних. Після вимкнення живлення дані з носіїв зовнішньої пам'яті не зникають.

Основною властивістю всіх видів пам'яті є її **ємність**, яка вимірюється в байтах, кілобайтах, мегабайтах, гігабайтах, терабайтах тощо.



Дайте відповіді на запитання

- 1°. Опишіть схему реалізації інформаційних процесів у комп'ютері за малюнком 2.1.
- 2°. Для чого призначено процесор? Назвіть пристрої, які входять до складу процесора.
- 3°. Який пристрій пам'яті включено до складу сучасних процесорів?
- 4°. Назвіть властивості процесора та наведіть приклади їхніх значень.
- 5°. Наведіть класифікацію пам'яті комп'ютера. Яка основна властивість пам'яті? Які одиниці вимірювання значень цієї властивості?
- 6°. Для чого призначена оперативна пам'ять? Де вона розміщується?
- 7°. Для чого призначена постійна пам'ять комп'ютера? У чому її відмінність від оперативної?
- 8°. Для чого призначена кеш-пам'ять?
- 9°. Назвіть пристрої зовнішньої пам'яті. Наведіть приклади значень властивостей кожного з них.
- 10*. Що спільного й у чому відмінності між пристроями зовнішньої пам'яті?



Виконайте завдання

- 1°. Використовуючи дані одного із сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад **hotline.ua**, запишіть значення властивостей трьох різних процесорів, які надійшли останнім часом у продаж у магазини України. Заповніть таблицю:





Властивість	Значення властивості процесора		
Модель			
Тактова частота			
Кількість ядер			
Розрядність			




- 2*. Дослідіть, використовуючи дані сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад **hotline.ua**, значення властивостей оперативної пам'яті з максимальною ємністю, що пропонуються для ноутбуків. Заповніть таблицю:

Властивість	Значення властивості		
Модель			
Виробник			

Властивість	Значення властивості		
Ємність			
Вартість			

- 3*.  Визначте значення властивостей процесора та пристроїв пам'яті вашого домашнього комп'ютера й заповніть таблиці результатів аналогічно до таблиць з рубрики **Працюємо з комп'ютером** до 2.1 (<https://cutt.ly/5hdeX36>).
- 4*.  За відомостями з Інтернету або каталогами комп'ютерних магазинів визначте значення властивостей процесорів, які доступні на ринку комп'ютерів вашого регіону, та заповніть таблицю, аналогічну до таблиці 2.1.
- 5*.  За відомостями з Інтернету або каталогами комп'ютерних магазинів порівняйте ємність кеш-пам'яті другого рівня процесорів з однаковою тактовою частотою і кількістю ядер різних виробників.
- 6*.  Дослідіть, чи можливо змінити дані в постійній пам'яті комп'ютера. Якщо так, то опишіть способи, як це зробити для домашнього комп'ютера.

2.2. Пристрої введення даних

- 1.  Які пристрої входять до складу комп'ютера? Як їх можна класифікувати?
- 2. Які пристрої введення даних ви знаєте? Для введення яких даних вони використовуються?
- 3. Які операції можна виконувати з використанням миші?

Види пристроїв введення даних

Як ви вже знаєте, до складу комп'ютерів входять пристрої введення та виведення даних.

Пристрої введення даних можна розподілити за типом даних, які вони опрацьовують. Так, можна виділити пристрої введення даних:

- текстових (**клавіатура**);
- графічних (**сканер, цифрова фотокамера, графічний планшет**);
- звукових (**мікрофон, цифровий диктофон**);
- відеоданих (**відеокамера, вебкамера, ТВ-тюнер**).

Ще однією групою пристроїв введення даних є пристрої для забезпечення керування роботою різноманітних програм. До них належать **миша, тачпад, мультимедійна (електронна) дошка, сенсорний екран, джойстик, геймпад** (англ. *gamepad* – ігровий майданчик) (мал. 2.12), **кейпад** (англ. *keypad* – майданчик для клавіш) (мал. 2.13), **руль, педалі, денспад** (англ. *dancepad* – майданчик для танців) (мал. 2.14) тощо.



Мал. 2.12. Геймпад



Мал. 2.13. Кейпад



Мал. 2.14. Денспад

У комп'ютерах, які використовують для навчальних і наукових цілей, можуть застосовуватися інші види пристроїв для введення даних. Наприклад, датчики для визначення значення температури, вологості повітря, наявності певних домішок у речовинах, швидкості руху різноманітних об'єктів тощо.

Розглянемо властивості (технічні характеристики), їх значення та класифікацію окремих пристроїв введення даних.

Клавіатура

Клавіатура призначена для введення символічних даних і команд. Залежно від значення певних властивостей клавіатури можна поділити на кілька груп:

- за призначенням: *стандартні* (для використання в стаціонарних комп'ютерах для введення символічних даних), *компактні* (для використання з мобільними пристроями малих розмірів), *мультимедійні* (для керування діями з мультимедійним контентом безпосередньо з клавіатури), *ігрові* (для керування діями в комп'ютерних іграх) тощо;
- за типом підключення до системного блока: *дротові* або *бездротові* (радіо, **Bluetooth**, **Wi-Fi** тощо).

Проекційна клавіатура (мал. 2.15) є ефективним пристроєм введення даних для мобільних пристроїв. Вона поєднує зручність введення даних з клавіатури стандартних розмірів і мобільність. Її зображення можна спроектувати на будь-яку плоску поверхню, а сам проектор має малі розміри. В іншому робота з такою клавіатурою нічим не відрізняється від роботи зі стандартною клавіатурою.



Мал. 2.15. Проекційна клавіатура



Цікаві факти з історії

У місті **Норвіч** (Велика Британія) у бруківці на вулиці Принцес можна побачити вирізьблену в камені клавіатуру (мал. 2.16). Місцеві жителі та гості міста ламають голову над тим, яким чином клавіатура з'явилась у бруківці, якій кілька сотень років. Чи це знак від позаземних цивілізацій, чи жарт студентів місцевого університету мистецтв?



Мал. 2.16. Клавіатура на вулиці Норвіча

Пристрої для забезпечення керування роботою програм

Маніпулятор **миша** використовують для забезпечення передавання команд від користувача комп'ютерним програмам.

Залежно від передбачених розробниками програм особливостей опрацювання натиснення тієї чи іншої кнопки миші, прокрутки її коліщатка може виконуватися певна команда меню, обиратися об'єкт, збільшуватись або зменшуватись масштаб відображення, відкриватися контекстне меню, вставлятися об'єкт тощо.

Маніпулятори миша поділяються залежно від значень таких властивостей:

- за призначенням: *для настільних ПК, для ноутбуків, ігрові* тощо;
- за типом підключення до системного блока: *дротові (USB, PS/2) або бездротові (радіо, Bluetooth, Wi-Fi)* тощо;
- за типом датчиків руху: *оптичні, лазерні, гіроскопічні* тощо;
- за кількістю кнопок: *2, 3–5, 6–9, 10 і більше* тощо;
- за типом корпусу: *симетрична стандартна, симетрична для шульги, ергономічна* тощо.



Мал. 2.17. Миша

На малюнку 2.17 зображено ігрову дротову лазерну ергономічну мишу, яка має 10 кнопок та одне коліщатко.

Тачпади (англ. *touch* – дотик, *pad* – площадка, майданчик, подушечка) є складовими практично всіх типів ноутбуків і нетбуків, проте можуть використовуватись і під час роботи з настільними ПК у вигляді додатково підключеного пристрою (мал. 2.18). Переміщення пальця по поверхні тачпада приводить до переміщення вказівника на екрані монітора. Одноразовий короткочасний дотик до поверхні тачпада аналогічний до одноразового клацання лівою кнопкою миші, а подвійний дотик – подвійного клацання лівою кнопкою миші. Ліва та права кнопки тачпада виконують ті самі функції, що й відповідні кнопки миші.



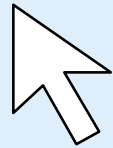
Мал. 2.18. Зовнішній тачпад



Для тих, хто хоче знати більше

Як змінити вигляд вказівника миші

Вказівник миші (тачпада) на екрані монітора комп'ютера за різних ситуацій має різне зображення. У більшості випадків він стандартно має вигляд стрілки (мал. 2.19). Однак, зовнішній вигляд вказівника можна змінити. Для цього слід виконати **Пуск** ⇒ **Настройки** ⇒ **Пристрої** ⇒ **Миша** ⇒ **Додаткові параметри миші**. І у вікні **Миша – властивості** відкрити вкладку **Вказівники** (мал. 2.20). У подальшому є кілька варіантів зміни зовнішнього вигляду вказівника:



Мал. 2.19. Вказівник

- змінити схему оформлення вказівників. Для цього слід:
 1. Відкрити список **Схема** (мал. 2.21).



Мал. 2.20. Вікно **Миша – властивості**



Мал. 2.21. Список **Схема**

2. Вибрати потрібну схему оформлення.
3. Вибрати кнопку **Застосувати**.
4. Закрити вікно;
- вибрати для кожного з режимів у списку **Настроювання** інше зображення. Для цього слід:
 1. Вибрати в списку назву режиму, зображення вказівника якого потрібно змінити, наприклад **Виділення тексту**.
 2. Вибрати кнопку **Огляд**.
 3. У вікні **Огляд** у списку папки **Cursor** вибрати потрібний файл із зображенням вказівника, наприклад **beam_im**.
 4. Ознайомитися зі збільшеним зображенням вибраного вказівника в полі **Перегляд**.
 5. Вибрати кнопку **Відкрити**.
 6. Повторити дії 1–5 для інших режимів вказівника.
 7. Вибрати кнопку **Застосувати**.
 8. Закрити вікно.

Перелік файлів із зображеннями вказівників, що під час інсталяції операційної системи розміщуються в папці **Cursor**, доволі обмежений. Однак користувач зможе знайти в Інтернеті великі колекції зображень, наприклад за ключовими словами «курсор миші». Колекції зображень вказівників – це набори файлів з розширенням імені **cur** чи **ani**. Ці файли бажано скопіювати в папку **Cursor** і для їх використання виконати наведену вище послідовність дій.

Сенсорні екрани використовують здебільшого в планшетних комп'ютерах, смартфонах, а також у різноманітних довідкових і презентаційних системах, платіжних терміналах. Уведення даних здійснюється дотиком пальця або стилуса до певних ділянок поверхні екрана. Екран «відчуває», у якому місці відбувся дотик, і передає відповідний сигнал комп'ютеру. Основні властивості – розмір екрана (довжина діагоналі може бути від 3 до 70 дюймів і більше) і принцип відстеження місця дотику.

Мультимедійні (електронні) дошки використовують здебільшого в закладах освіти, а також під час проведення різноманітних презентацій. Розрізняють дошки залежно від їх розмірів (довжина діагоналі 70–79, 80–89 чи понад 90 дюймів) і принципу визначення місця дотику на дошці.

Пристрої для введення графічних даних

Сканери (англ. *scanner* – той, що відстежує) використовуються для введення графічних даних у комп'ютер. На об'єкт, комп'ютерне зображення якого потрібно отримати, від спеціального пристрою направляється потік світла. Датчики світла аналізують інтенсивність і колір відбитого світла та перетворюють ці дані в електронну форму подання зображення.



Мал. 2.22.
Ручний сканер

Сканувати можна й текстовий документ, але в результаті буде отримано графічне зображення тексту. Це зображення можна перетворити на текст, використавши спеціальні програми – **системи оптичного розпізнавання тексту** (англ. **OCR – Optical Character Recognition** – оптичне розпізнавання символів).

Сканери поділяють на *ручні, настільні та проєкційні*.

Ручні сканери (мал. 2.22) мають малі розміри й зручні для введення невеликих за розмірами зображень. Під час сканування об'єкт залишається нерухомим, а переміщують сам сканер. Такі типи сканерів широко використовують у магазинах для сканування штрихкодів,

нанесених на товари, у поїздах для сканування QR-кодів на проїзних документах, у бібліотеках тощо.

Останнім часом набули розповсюдження ручні сканери з протяжним механізмом, що використовуються в комплекті з різними мобільними комп'ютерами для отримання якісних відсканованих зображень. Такі сканери мають невеликі розміри та живляться з використанням USB-підключення або від автономної батареї. Звичайно вони можуть бути підключені до мережі з використанням Wi-Fi-з'єднання. Один з таких сканерів подано на малюнку 2.23.

Серед **настільних сканерів** найпоширеніші **планшетні** (мал. 2.24, табл. 2.4). У них об'єкт, що сканується, нерухомо розміщується на склі. Сканування відбувається під час автоматичного переміщення механізму зчитування. Для сканування великої кількості сторінок із зображеннями чи текстами використовують сканери з автоматичною подачею сторінок і вбудованою пам'яттю для зберігання файлів відсканованих зображень.

Таблиця 2.4

Основні властивості планшетних сканерів

Властивість	Що характеризує	Одиниці вимірювання	Значення в сучасних сканерах, наприклад у моделі Epson V550 Photo
Формат	Розмір робочої області для сканування об'єктів	мм (формат)	216 мм × 297 мм (формат A4)
Роздільність	Здатність розпізнавати певну кількість точок на одному дюймі зображення	Точок на один дюйм – dpi (англ. <i>dot per inch</i> – точок на дюйм)	6400 × 9600
Кількість кольорів	Кількість кольорів, які може розпізнати сканер	Одиниці	$2,8 \cdot 10^{14}$ (48 бітів на піксель)

Крім планшетних, є настільні сканери, у яких об'єкт сканування протягується відносно нерухомого механізму зчитування (мал. 2.25).



Мал. 2.23. Ручний мобільний сканер



Мал. 2.24. Планшетний сканер



Мал. 2.25. Настільний сканер з протяжним механізмом



Мал. 2.26. Проекційний сканер

Проекційні сканери ще називають **безконтактними** або **документ-сканерами**. Їхньою особливістю є те, що зображення вони отримують, як у фотоапараті, відразу всієї сторінки. Об'єкт сканування розміщується під модулем сканування (мал. 2.26). Змінюючи відстань від об'єкта до модуля сканування, можна отримувати зображення об'єктів різних розмірів, а не тільки обмежуватися, наприклад, форматом A4. Значна частина таких сканерів має пристрій для підсвічування об'єкта сканування. Такі сканери активно використовують у бібліотеках для створення електронних копій книжок, журналів, текстових документів, фотографій, інших зображень.

Близький за принципом дії до сенсорного екрана **графічний планшет**. Він використовується для створення малюнків і введення інших графічних даних. Користувач створює малюнок, дотикаючись стилусом до поверхні графічного планшета. В окремих видах графічних планшетів робоча поверхня не відображає зображення, воно з'являється на екрані комп'ютера, до якого приєднано графічний планшет. В інших створене зображення відображається на робочій поверхні графічного планшета (мал. 2.27).



Мал. 2.27. Графічний планшет

Розрізняють графічні планшети за розміром робочої поверхні, роздільністю (від 2000 до понад 4000 пікселів на дюйм), а також кількістю варіантів реагування на силу натискання стилусом на робочу поверхню (від 512 до 2048).

Інші пристрої введення даних

Як уже було зазначено, для введення відео та звукових даних використовують **фото- і відеокамери, вебкамери, мікрофони** та інші мультимедійні пристрої.

Зазначимо, що основними властивостями фото- і відеокамер, вебкамер є їхня роздільність, яка визначає якість зображення. Для відеокамер якість зображення визначається стандартами **SD** (англ. *Standard Definition* – стандартна чіткість, 720 × 576 точок), **HD** (англ. *High Definition* – висока чіткість, 1280 × 720 точок), **Full HD** (англ. *Full High Definition* – повна висока чіткість, 1920 × 1080 точок), **Ultra HD** (англ. *Ultra High Definition* – надвисока чіткість, 3840 × 2160 точок).

Для **вебкамер** роздільність визначається такими самими, як і в інших цифрових камерах, значеннями кількості точок на дюйм, додатковою характеристикою є максимальна кількість кадрів, які може записувати камера за одну секунду. Зазвичай значення цієї властивості знаходиться в межах від 15 до 60 кадрів за секунду.

У **фотокамер** якість зображення залежить від оптичних властивостей об'єктива та кількості точок (пікселей) (від 10 до понад 45 мегапікселей), які може відтворити пристрій.

Мікрофон – пристрій, що здійснює перетворення звукових коливань в електричні. У комп'ютерах електричні коливання ще оцифровуються – перетворюються у форму, яка зручна для опрацювання іншими пристроями комп'ютера.

Мікрофони призначено для введення звукових даних до комп'ютера. Їх можна поділити на кілька груп залежно від:

- принципу дії (*динамічні, п'єзоелектричні, магнітоелектричні, електростатичні* тощо);
- системи під'єднання (*вмонтовані* (у корпуси ноутбуків, вебкамер тощо) і *відокремлені* (під'єднуються додатково до пристроїв комп'ютера, зазвичай з використанням USB-з'єднання), мал. 2.28);
- чутливості до гучності звуку тощо.



Мал. 2.28. Відокремлений мікрофон

Використання мікрофонів у комп'ютерній техніці набуло особливого розповсюдження з розвитком систем комп'ютерного зв'язку, використанням аудіо- та відеоконференцій, систем голосового введення текстів тощо.

Значення властивостей пристроїв введення даних постійно змінюються, вони вдосконалюються за дизайном і якістю здійснення операцій з введення даних, з'являються нові види пристроїв. Варто перед придбанням пристроїв ознайомитися з думками експертів і користувачів щодо їх використання. Такі матеріали регулярно розміщують в Інтернеті.



Найважливіше в цьому пункті

До пристроїв введення даних належать пристрої введення текстових (клавіатура), графічних (сканер, фотокамера, графічний планшет), звукових (мікрофон), відеоданих (відеокамера, вебкамера, ТВ-тюнер), а також пристрої для забезпечення керування об'єктами в різноманітних програмах – миша, тачпад, мультимедійна (електронна) дошка, сенсорний екран, джойстик, геймпад, кейпад, руль, педалі, танцювальний майданчик тощо.

Під час добору пристроїв варто зважати на їх призначення, значення основних властивостей і ціну.



Дайте відповіді на запитання

- 1^o. Які пристрої комп'ютера належать до пристроїв введення даних?
- 2^{*}. За значенням яких властивостей класифікують клавіатури? Наведіть приклади.
- 3^{*}. Які види маніпуляторів миша використовують у комп'ютерному класі вашої школи? Наведіть значення їх властивостей.
- 4^{*}. На значення яких властивостей слід звернути увагу, купуючи нову мишу для домашнього комп'ютера? Поясніть чому.
- 5^o. За допомогою яких пристроїв можна ввести графічне зображення в пам'ять комп'ютера?
- 6^{*}. Для чого використовують сканер? Які типи сканерів ви знаєте?
- 7^{*}. Які значення властивостей можуть мати сканери, що використовують у комп'ютерних класах школи? Який сканер ви б порекомендували для комп'ютерного класу?
- 8^{*}. Для чого призначено пристрої введення, які входять до складу мультимедійного обладнання комп'ютерного класу?
- 9^{*}. Чим, на вашу думку, відрізняється відеокамера від вебкамери? Що в них спільного?



Виконайте завдання

- 1^{*}. На основі відомостей, розміщених у підручнику, підготуйте повідомлення зі схемою класифікації пристроїв введення даних.
- 2^{*}. На основі відомостей, розміщених у підручнику, підготуйте слайд презентації зі схемою класифікації пристроїв введення даних. Збережіть презентацію у файлі з іменем **завдання 2.2.2** у вашій папці.
- 3^{*}. Використовуючи дані одного із сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад **hotline.ua**, визначте значення властивостей трьох різних маніпуляторів миша, що надійшли останнім часом у продаж у магазини України. Заповніть таблицю:



Властивість	Значення властивості миші		
Модель			
Тип підключення			
Тип датчиків руху			
Кількість кнопок			
Тип корпусу			
Призначення			



4*. Використовуючи дані сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад **hotline.ua**, визначте значення властивостей планшетних сканерів трьох різних виробників, що пропонуються магазинами України для використання вдома. Створіть у текстовому процесорі таблицю за зразком і заповніть її:

Властивість	Значення властивості сканера		
Модель, виробник			
Тип підключення			
Область сканування			
Роздільність			
Час попереднього сканування			



5*. За відомостями з Інтернету або каталогами комп'ютерних магазинів запропонуйте для шкільної бібліотеки дві моделі проєкційних сканерів для створення електронної бібліотеки. Свої пропозиції обґрунтуйте у відповідному текстовому файлі.



6*. Порадьте своєму товаришу (подрузі), якому (якій) подобається малювати, 2–3 моделі графічних планшетів. Пропозиції обґрунтуйте у відповідному текстовому файлі.



7*. Підготуйте повідомлення про використання мікрофонів різних типів для створення мультимедійних об'єктів з використанням комп'ютерів.

2.3. Пристрої виведення даних



1. Які пристрої входять до складу комп'ютера? Як їх можна класифікувати?
2. Які пристрої виведення даних входять до складу комп'ютера? Для виведення яких даних їх використовують?
3. Які пристрої є у вашому комп'ютерному класі? Для чого вони призначені?

Пристрої виведення даних, як і пристрої введення даних, можна поділити за типом даних, які вони опрацьовують. Так, можна виділити пристрої виведення даних:

- *текстових і графічних* (**монітор, принтер, плотер**);
- *звукових* (**навушники, звукові колонки**);
- *відеоданих* (**мультимедійні проєктори, екранні панелі**).

Певні типи пристроїв можуть виводити як графічні або текстові, так і відеодані (монітор, мультимедійні проєктори, екранні панелі тощо).

Монітор

Монітор є основним пристроєм для виведення даних у персональних комп'ютерах. Монітори поділяють за системою створення зображення на:

- **LCD-монітори** (англ. *Liquid Crystal Display* – рідкокристалічний дисплей), або **монітори на рідких кристалах**;
- **плазмові**;
- **OLED** (англ. *Organic Light Emitting Diode* – органічний світлодіод);
- монітори на **електронному чорнилі – e-ink** (англ. *Electronic ink* – електронне чорнило) тощо.



У більшості сучасних моніторів використовується рідкокристалічна технологія. **LCD-монітори** поділяють за:

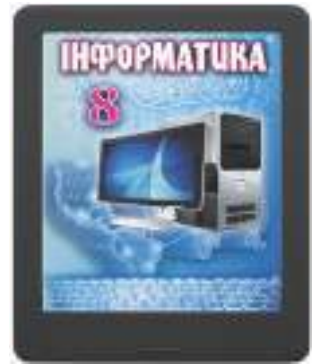
- принципом дії (*TN + film, PLS, різні версії IPS та VA*);
- довжиною діагоналі (від 19 до понад 37 дюймів);
- роздільністю (від 1280 × 1024 до 5120 × 2880 точок);
- часом відклику – реагування на команду зміни кольору пікселя екрана із чорного на білий (від 1 мс до 8 мс);
- співвідношенням довжин сторін екрана (16 : 9, 16 : 10, 21 : 9, 5 : 4).

Суттєвою характеристикою LCD-моніторів є тип джерела світла, яке використовується для підсвічування під час створення зображення на зовнішній поверхні екрана. У переважній більшості сучасних LCD-моніторів використовується **LED-технологія** (*Light Emitting Diode* – світлодіод) підсвічування екрана.

Плазмові монітори використовуються в основному для створення великих демонстраційних екранів, бо мають низьку роздільність, але високу яскравість зображення. Останнім часом ця технологія замінюється технологією на рідких кристалах.

Технологія **OLED** уже понад 15 років вважається перспективною для створення якісних моніторів завдяки високій контрастності та яскравості зображення, малий (до 1 см) товщині. Однак висока вартість цих моніторів, значне зниження яскравості через 3–5 років роботи суттєво знижують їх ринок збуту.

Висока контрастність чорно-білого зображення в моніторах на основі **електронного чорнила**, значні терміни роботи без додаткової підзарядки (кілька тижнів), відсутність мерехтіння екрана забезпечують комфортні умови для читання з екрана. Тому монітори, які створено за вказаною технологією, в основному використовують в електронних книжках (мал. 2.29) – планшетних комп'ютерах, що призначені для читання текстів. З 2016 року компанією **E Ink** розпочато випуск кольорових моніторів, однак їхньому широкому розповсюдженню у звичайних комп'ютерах заважає дуже низька швидкість оновлення екрана, мала кількість (4096) відтворюваних кольорів і потреба додаткового підсвічування.



Мал. 2.29. Електронна книжка на основі електронного чорнила

Відеоадаптер

Для опрацювання графічних даних, що виводяться на екран монітора, у комп'ютерах використовується спеціальний пристрій – **відеоадаптер**. Розрізняють *інтегровані відеоадаптери* та *у вигляді окремої плати*. Інтегрований відеоадаптер розміщується в мікросхемі процесора або на материнській платі. Його використовують у персональних комп'ютерах, які не призначені для якісного опрацювання великих обсягів графічних даних. Для цих цілей використовують відеоадаптери у вигляді окремої плати (мал. 2.30), що вставляється в один зі слотів материнської плати.



Мал. 2.30. Відеоадаптер

Інтегральний (лат. *integer* – цілий) – нерозривно зв'язаний, цілий.

Інтеграція (лат. *integratio* – поповнення) – об'єднання в ціле окремих частин.

Чіпсет (англ. *chip* – мікросхема, *set* – набір) – набір мікросхем, що забезпечує обмін даними між пристроями комп'ютера.

Такі відеоадаптери містять спеціальний графічний процесор і додаткову оперативну пам'ять – **графічну (відео-) пам'ять**. Їх розрізняють за типом процесора (наприклад, **AMD Radeon RX 5500 XT, NVIDIA GeForce RTX 2080**), обсягом пам'яті (від 2 до 24 ГБ і більше), системами під'єднання до материнської плати (**PCI Express 3.0, PCI Express 4.0**) та до монітора (**HDMI, D-Sub, DisplayPort, DVI-D**) тощо.

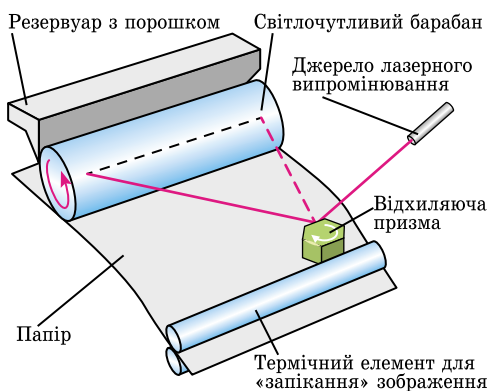
Пристрої для друкування

Для виведення текстових і графічних даних на папір, плівку або інші матеріали використовують **принтери та плотери**.

Основними властивостями принтерів є:

- розмір матеріалу, на який здійснюється друкування – **A4, A3**;
- кількість кольорів – **монохромні, багатокольорові**;
- роздільність – від **600 × 1200 до 5760 × 1440 точок на дюйм** і більше;
- швидкість друкування – від **1 до 80 сторінок за хвилину**;
- тип підключення до комп'ютера – **дротове або бездротове**.

Принтери поділяють за принципом дії на: **матричні, лазерні, струменеві, термосублімаційні, термічні, 3D-принтери** тощо. Найрозповсюдженішими є принтери, що використовують струменеву та лазерну технології, хоча останнім часом усе більшої популярності набувають термосублімаційні та 3D-принтери.



Мал. 2.31. Схема дії лазерного принтера

Лазерний принтер. Основним елементом лазерного принтера є циліндр, з нанесеним на нього світлочутливим матеріалом (мал. 2.31). Лазерна установка генерує тонкий світловий промінь, який потрапляє на барабан і змінює його електричний заряд у точці падіння. Далі на барабан наноситься тонер – спеціальна фарба в порошковому вигляді. Кількість порошку, що прилипає до поверхні барабана, залежить від величини заряду в певній точці. Під час обертання барабан притискається до аркуша паперу. Тонер прилипає до поверхні паперу й утворює на ньому потрібне зображення.

Для фіксації порошку на папері аркуш пропускають між роликками, які розігріті до температури близько 180 °С (так звана **печка** або **фюзер**). Тонер «запікається» на папері. Після цього йому ні волога, ні пряме сонячне проміння не завдають шкоди.

Для кольорового друкування використовують кольорові тонери.

Тонер (англ. *toner* – той, що надає відтінок) – порошок або крем, з використанням якого змінюють колір поверхні, на яку він наноситься.

Фюзер (англ. *fuse* – плавитися) – пристрій для плавлення, запікання.

Струменевий принтер. Більш точно зараз варто називати ці принтери **крапельними**. Технологія створення зображення в принтерах цього типу базується на нанесенні дуже малих крапель чорнил різного кольору на папір або інший матеріал. Залежно від способу генерації крапель розрізняють:

- **п'єзоелектричну технологію** – крапля видавлюється пластиною, що деформується під дією електричного струму (наприклад, принтери **Epson**);
- **бульбашкову, або термічну, технологію** – крапля видавлюється бульбашкою парів чорнила, що утворюються внаслідок швидкого нагрівання електро-термічного елемента (наприклад, принтери **Hewlett Packard, Canon**).



Суттєвим недоліком струменевих принтерів є вигорання чорнила під дією прямого сонячного проміння та нестійкість до вологи.

Сублімаційний друк може використовуватися для створення яскравих повнокольорових зображень на різних поверхнях – на папері, картоні, а головне – на тканині, склі, фаянсі тощо. Сублімація (лат. *sublimiter* – угору, прямо) – це фізичне явище переходу речовини з твердого стану в газоподібний, минаючи рідкий стан. Існує кілька варіантів сублімаційного друку:

Поліестер (грец. πολύ- – багато та нім. *Essig-Aether* – оцтовий ефір) – речовина, що використовується в тому числі для створення синтетичних тканин.

- **прямий «мокрый» друк** спеціальними (сублімаційними) чорнилами на поверхню, що містить поліестер, або поверхню, на яку попередньо нанесено шар речовини з поліестером;
- **прямий «сухий» друк** з використанням плівок різного кольору (мал. 2.32), з яких під час нагрівання тверда фарба випаровується та переноситься на поверхню, на якій утворюється малюнок;
- **непрямий друк**, на першому етапі якого відбувається друк з використанням струменевого принтера та сублімаційних чорнил на спеціальний сублімаційний папір. На другому етапі – створене на папері зображення прикладається до поверхні об'єкта, на якій потрібно створити зображення, і нагрівається до температури понад 100 °С. Фарба з паперу випаровується та утворює зображення на чашках, футболках, табличках тощо. У цьому випадку також використовуються тканини з поліестером або поверхні з попередньо нанесеним шаром речовини з поліестером.



Мал. 2.32. Сублімаційний принтер і набір кольорових плівок до нього



Мал. 2.33. Сублімаційний принтер

Прямий «мокрый» друк зазвичай застосовується для великих тиражів зображень на промислових поліграфічних підприємствах, прямий «сухий» друк – для друку фотографій високої якості (мал. 2.33), непрямий друк – для створення сувенірної продукції малими тиражами навіть у домашніх умовах.

Основна перевага сублімаційної технології друку – висока якість зображень і стійкість до вологи та світла.

Чи знаєте ви, що...

Одним з перших ідей виводити на тверду основу результати обчислень запропонував відомий англійський учений і винахідник Чарлз Беббідж. Під час проектування своєї першої обчислювальної машини Беббідж приєднав до неї друкуючий пристрій, який видавлював результат обчислень на мідній пластині. Таким чином отримане число могло довго зберігатися й унеможливило помилки, які виникали через неухважність тих, хто записував ці результати на папері.

Плотери (графобудівники) також можуть виводити на тверді носії текстові або графічні дані. Але в основному вони призначені для роботи з графікою. Використовують для друку креслень, ескізів, плакатів та інших зображень великих розмірів, зазвичай більших ніж 297 мм × 420 мм (формат А3).

Принципи, які лягли в основу конструкцій сучасних плотерів, мало чим відрізняються від принципів роботи принтерів. Найрозповсюдженішими є плотери, які використовують струменеву технологію.

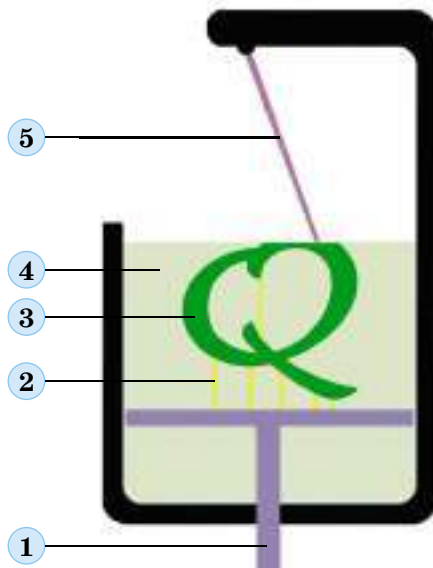
Тривимірні, або 3D-принтери, використовують для створення об'ємних об'єктів складної форми (мал. 2.34). Розрізняють тривимірні принтери за типом речовини, з якої «будується» об'єкт, і способом її отвердіння. Використовують кілька основних технологій:

- отвердіння спеціальної смоли під дією світла (лазерне випромінення або світлодіодне) – **SLA** (англ. *Stereolithography apparatus* – стереолітографічна апаратура), або стереолітографічна технологія (мал. 2.34);
- запікання (плавлення та отвердіння) спеціального порошку під дією променя лазера – **SLS**-технологія (англ. *Selective Laser Sintering* – вибіркоче лазерне спікання);
- отвердіння попередньо розплавлених пластичних мас – технологія **FDM** (англ. *Fused Deposition Modeling* – моделювання методом наплавлення).

FDM-технологія – найдешевша. Це зумовило її широке розповсюдження в принтерах, які використовують у домашніх умовах і в навчальних закладах. Однак **SLA**- та **SLS**-технології мають суттєву перевагу в точності виготовлення об'єктів і широко використовуються в промисловому виробництві.

Також розрізняють 3D-принтери за максимальними розмірами об'єктів (довжиною, шириною, висотою), які можна на них виготовити.

Порівняння характеристик трьох основних технологій за п'ятибальною шкалою подано в таблиці 2.5.



1. Стіл для побудови об'єкта
2. Додаткові елементи для частин об'єкта, що не мають опори
3. Об'єкт, що друкується (літера Q)
4. Речовина, що твердіє під дією випромінення
5. Промінь ультрафіолетового лазера

Мал. 2.34. Схема роботи SLA 3D-принтера



Порівняння характеристик технологій 3D-друку

	Назва технології		
	FDM	SLA	SLS
Роздільність	●●○○	●●●●●	●●●●○
Точність виготовлення	●●●●○	●●●●●	●●●●●
Шорсткуватість поверхні	●●○○○	●●●●●	●●●●○
Можливість створення складних виробів	●●●○○	●●●●○	●●●●●
Простота використання	●●●●●	●●●●●	●●●●○
Розміри об'єктів друку для настільних принтерів, до...	200 мм × 200 мм × 300 мм	300 мм × 335 мм × 200 мм	165 мм × 165 мм × 320 мм
Вимоги до приміщення	Кондиціонер або бажано спеціальна вентиляція	Достатньо офісного приміщення	Бажано розміщення в окремій майстерні

3D-принтери мають широке застосування в різних галузях людської діяльності (мал. 2.35–2.38). Особливо активно їх використовують у медицині.



Мал. 2.35. Створення зубного протеза в 3D-принтері



Мал. 2.36. Створення протеза грудної клітки в 3D-принтері



Мал. 2.37. Модель будинку, створена на 3D-принтері



Мал. 2.38. Створення моделі блока циліндрів автомобіля в 3D-принтері



Мал. 2.39. QR-коди посилань на відео про технології 3D-друку

Переглянути відео про технології 3D-друку можна за посиланнями: <https://cutt.ly/ag1ZJpO> чи <https://cutt.ly/Kg1ZZup> або використавши QR-коди (мал. 2.39).



Мал. 2.40. Створення моделі вежі з використанням 3D-ручки

3D-ручки – це нові пристрої виведення даних для «малювання» у просторі. Вони частіше використовують FDM-технологію 3D-принтерів і будують зображення шляхом видавлювання тонкого струменя (0,5–1,5 мм) розплавленої пластмаси, яка відразу твердіє, утворюючи об'ємний об'єкт (мал. 2.40). Є ручки, що використовують різнокольорові чорнила на основі смол, що твердіють під дією ультрафіолетового випромінювання. Такі ручки можуть тривалий час працювати без підключення до електричної мережі за рахунок акумуляторних батарей.

Переглянути відео про роботу з 3D-ручкою можна за посиланнями: <https://cutt.ly/Xg1ZCJ1>

чи <https://cutt.ly/6g1ZBJ2>, чи <https://cutt.ly/qg1Z1e0> або використавши QR-коди (мал. 2.41).



Мал. 2.41. QR-коди посилань на відео про роботу з 3D-ручками

Значення властивостей пристроїв постійно змінюються, вони вдосконалюються за дизайном і якістю здійснення операцій з виведення даних. Варто перед придбанням пристроїв ознайомитися з думками експертів і користувачів щодо їхнього використання. Такі матеріали регулярно розміщують в Інтернеті.

Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/AhdeMSZ> або QR-кодом.

Найважливіше в цьому пункті

До **пристроїв виведення даних** належать пристрої виведення текстових і графічних даних (**монітор, принтер, плотер**), звукових (**навушники, звукові колонки**), відеоданих (**мультимедійні проєктори, екранні панелі**) тощо.

Монітори поділяють за системою створення зображення на: **монітори на рідких кристалах**, або **LCD-монітори, плазмові, OLED**, на **електронному чорнилі – e-ink** тощо. У більшості сучасних моніторів використовується рідкокристалічна технологія.

Для опрацювання графічних даних, що виводяться на екран монітора, у комп'ютерах використовується спеціальний пристрій – **відеоадаптер**. Відеоадаптер містить спеціальний графічний процесор і додаткову оперативну пам'ять – **графічну (відео-) пам'ять**.

Для виведення даних на папір, плівку або інші матеріали використовують **принтери та плотери**.

Принтери поділяють за принципом дії на: **матричні, лазерні, струменеві, термосублімаційні, термічні, 3D-принтери** тощо. Найбільш розповсюдженими є принтери, що використовують струменеву та лазерну технології.

В останні роки активно розвиваються технології створення тривимірних об'єктів з використанням **3D-принтерів і 3D-ручок**.




Дайте відповіді на запитання


- 1^o. Які пристрої комп'ютера належать до пристроїв виведення даних?
- 2*. За значенням яких властивостей класифікують монітори? Наведіть приклади значень властивостей сучасних моніторів.
- 3*. Які види принтерів використовують у вашій школі? Як визначити значення їх властивостей?
- 4*. На значення яких властивостей слід звернути увагу, купуючи новий відео-адаптер для домашнього комп'ютера? Поясніть чому.
- 5*. Чим відрізняється принтер, який працює за лазерною технологією, від принтера на основі сублімаційної технології? Яка технологія дає змогу створити більш якісне кольорове зображення?
- 6^o. Які принтери використовують для створення сувенірної продукції малими тиражами?
- 7*. Для чого призначено пристрої виведення даних, що входять до складу мультимедійного обладнання комп'ютерного класу?
- 8*. Чим, на вашу думку, відрізняється 3D-принтер від 3D-ручки? Що в них спільного?
- 9*. Які значення властивостей повинен, на вашу думку, мати принтер для домашнього використання?



Виконайте завдання

- 1*. На основі відомостей, розміщених у підручнику, підготуйте схему класифікації пристроїв виведення даних.
- 2*. На основі відомостей, розміщених у підручнику, підготуйте слайд презентації зі схемою класифікації пристроїв виведення даних. Збережіть презентацію у файлі з іменем **завдання 2.3.2** у вашій папці.
-  3*. Використовуючи дані одного із сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад **hotline.ua**, визначте значення властивостей трьох різних моніторів, що надійшли останнім часом у продаж у магазини України. Заповніть таблицю:

Властивість	Значення властивості монітора		
Модель			
Виробник			
Довжина діагоналі, дюймів			
Тип матриці			
Максимальна роздільність, пікселів			
Час відклику, мс			

- 4*. На основі відомостей, розміщених у підручнику, підготуйте в одній з прикладних програм схему класифікації сучасних принтерів. Збережіть схему у файлі з іменем **завдання 2.3.4** у вашій папці.
-  5*. Дізнайтеся значення властивостей пристроїв виведення вашого домашнього комп'ютера та заповніть таблиці результатів аналогічно до таблиць рубрики **Працюємо з комп'ютером** до 2.3 (<https://cutt.ly/AhdeMSZ>).



6*. За відомостями з Інтернету або каталогами комп'ютерних магазинів визначте значення властивостей принтерів (струменевих і лазерних) трьох різних виробників, що пропонуються в Україні. Заповніть таблицю:

Властивість	Значення властивості принтера		
Модель			
Виробник			
Принцип дії			
Роздільність, пікселів			
Швидкість друку, сторінок за хвилину			



7*. Підготуйте презентацію про галузі використання сублімаційних принтерів.



8*. Підготуйте презентацію про галузі використання 3D-принтерів і 3D-ручок. Зверніть особливу увагу на промислове використання технологій 3D-друку.

2.4. Історія обчислювальних і комп'ютерних пристроїв



1. Які види комп'ютерів ви знаєте? Чим відрізняється їх використання?
2. Які ви знаєте приклади застосування комп'ютерів у різних галузях людської діяльності?
3. Назвіть прізвища українських учених, які зробили значний внесок у розвиток комп'ютерної техніки.

Етапи розвитку засобів реалізації інформаційних процесів

Розвиток людства безпосередньо пов'язаний з розвитком засобів передавання, опрацювання та зберігання повідомлень, у якому можна виділити кілька етапів (табл. 2.6).

Таблиця 2.6

Етапи розвитку засобів реалізації інформаційних процесів

Назва етапу	Період історії людства	Приклади носіїв даних, засобів передавання та опрацювання даних
Етап ручних засобів	Від стародавніх часів до середини XV ст.	<i>Носії даних</i> – глиняні дощечки, папіруси, береста, палиці із зарубками, картини тощо. <i>Засоби передавання</i> – сигнальні вогнища, барабанний бій, персональні посланці, голубина пошта, перші поштові служби для передавання державних документів. <i>Засоби опрацювання</i> – мозок і пальці людини, ручні обчислювальні прилади (абак, рахівниця, вузлики на мотузці тощо)
Етап механічних засобів	Від середини XV ст. до середини XIX ст.	<i>Носії даних</i> – книжки, газети, журнали, фотопластини. <i>Засоби передавання</i> – поштові служби. <i>Засоби опрацювання</i> – друкарські машинки, фотоапарати, арифмометри, машинки для обчислень, музичні машинки (типу шарманок), грамофони, пристрої для відтворення анімації

Назва етапу	Період історії людства	Приклади носіїв даних, засобів передавання та опрацювання даних
Етап електричних засобів	Від середини XIX ст. до 40-х років XX ст.	Носії даних – платівки, кіноплівки, магнітні плівки. Засоби передавання – телефон, телеграф, радіо. Засоби опрацювання – фонографи, електричні друкарські машинки, табулятори, електричні арифмометри, магнітофони, ротаринти
Етап електронних засобів	Від 40-х років XX ст. до наших днів	Носії даних – магнітні й оптичні диски, електронні мікросхеми. Засоби передавання – телебачення, комп'ютерні мережі, засоби мобільного зв'язку. Засоби опрацювання – електронні калькулятори, комп'ютер

Історія засобів опрацювання повідомлень

Розглянемо детальніше історію розвитку засобів опрацювання повідомлень, призначених для виконання обчислень. Першими засобами опрацювання числових повідомлень були пальці (мал. 2.42). Про це свідчать, наприклад, римські цифри (I, V, X). У стародавній слов'янській нумерації одиниці називалися «перстами», тобто пальцями, а назва цифри «5» походить від слова «п'ясть» (кисть руки).



Мал. 2.42. Рахунок за допомогою пальців руки

Значним кроком уперед у розвитку засобів обчислювання стало створення абака в V ст. до н. е. у Стародавній Греції. Ідею такого пристрою греки запозичили у стародавніх вавилонян. Більш пізній римський варіант абака зображено на малюнку 2.43. **Абак** – це дошка з паралельними лунками, у які вкладалися якісь предмети, найчастіше камінці. Від цього пішла назва як самого пристрою – **калькулятор** (лат. *calculus* – камінці), так і процесу **обчислення** (англ. *calculate* – обчислювати).



Мал. 2.43. Римський абак

Подібні пристрої в подальшому було створено в різних країнах: Китаї (**суаньпань**), Японії (**соробан**), Росії (рос. *счёты* – **рахівниця** (мал. 2.44)) – і використовувалися майже до кінця XX ст.



Мал. 2.44. Рахівниця

Разом з простими й доволі розповсюдженими пристроями типу абака ще в Давній Греції та Римі використовували різноманітні обчислювальні пристрої з використанням коліщат із зубцями. Про це свідчить знахідка із затонулого античного судна. Її назвали «Антикітерський механізм» – за назвою грецького острова,



Мал. 2.45. Антикітерський механізм: фрагмент знахідки та комп'ютерна модель

біля якого знайшли судно в 1902 р. Але тільки в другій половині ХХ ст. вчені змогли за уламками механізму відновити його будову й визначити призначення (мал. 2.45). Створений за 100–150 років до нашої ери пристрій використовувався як календар, а також для механічного визначення положення небесних тіл – Сонця, Місяця та відомих грекам планет – Меркурія, Венери, Марса, Юпітера, Сатурна.

На жаль, пристрої, подібні до антикітерського механізму, і наукові знання, потрібні для їх виготовлення, були на тривалий час утрачені. Активні роботи щодо вдосконалення обчислювальних пристроїв розпочалися тільки в ХVІІ ст. з розвитком промисловості, торгівлі, будівництва, військової справи.

У 1642 р. французький математик, фізик, інженер і філософ **Блез Паскаль** (1623–1662) (мал. 2.46) подав на розгляд Королівської ради свій механічний обчислювальний пристрій. У подальшому він створив кілька десятків таких пристроїв (мал. 2.47). Їх використовували для виконання додавання та віднімання чисел.

Згодом ідеї Паскаля були розвинуті та вдосконалені багатьма математиками та інженерами. Зокрема, німецький учений **Готфрід Лейбніц** (1646–1716) (мал. 2.48) створив перший у світі **арифмометр** – механічний обчислювальний пристрій, що виконував усі чотири арифметичні дії (мал. 2.49).



Мал. 2.46. Блез Паскаль



Мал. 2.47. Обчислювальний пристрій Паскаліна



Мал. 2.48. Готфрід Лейбніц



Мал. 2.49. Арифмометр Готфріда Лейбніца

Удосконалені варіанти арифмометрів різних конструкцій активно використовували впродовж кількох століть і були замінені електронними калькуляторами у другій половині ХХ ст.

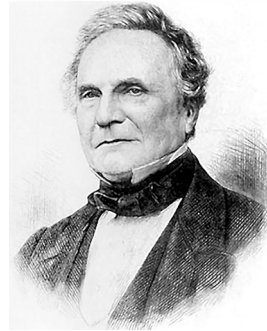
Одночасно з удосконаленням арифмометрів учені намагалися створити обчислювальні пристрої, які б могли автоматично здійснювати обчислення, виконуючи команди заздалегідь розробленої програми (мал. 2.50). Одним з перших ідей створення автоматизованої машини для обчислень висловив англійський математик і конструктор **Чарлз Беббідж** (1791–1871) (мал. 2.51). Він назвав її **аналітичною машиною**.

За проєктом Ч. Беббіджа аналітична машина мала такі складові частини:

1. «Склад» для зберігання чисел (у сучасній термінології – пам'ять).



Мал. 2.50. Машинка для обчислень І. Г. Мюллера



Мал. 2.51. Чарлз Беббідж

2. «Млин» для виконання арифметичних операцій над числами (у сучасній термінології – **арифметичний пристрій**).
3. Пристрій, який керує послідовністю виконання арифметичних операцій (у сучасній термінології – **пристрій керування**).
4. Пристрій для введення початкових (вхідних) даних.
5. Пристрій виведення результатів.

Ада Лавлейс (1815–1852) (мал. 2.52), дочка відомого англійського поета Джорджа Байрона, працювала разом з Беббіджем над створенням проєкту аналітичної машини та його реалізацією. Вона вперше описала основні принципи розробки програм для обчислювальних машин. Ада Лавлейс стояла у витоків програмування і на її честь названо одну із сучасних мов програмування **Ada**.



Мал. 2.52. Ада Лавлейс

До кінця XIX ст. пристрої для обчислень були ручними або механічними. І тільки в кінці XIX ст. американський учений **Герман Голлеріт** (1860–1929) запропонував новий пристрій, робота якого базувалася на використанні електричного струму, – **табулятор** (мал. 2.53). Він був призначений для опрацювання даних перепису населення. Дані про конкретну людину не записувалися на аркуші паперу, а позначалися отворами в строго визначених місцях персональної карти – **перфокарти** (мал. 2.54).

Перфорація (лат. *perforate* – проривлювати) – пробивка отворів, сукупність отворів.



Мал. 2.53. Табулятор Германа Голлеріта

PERSONENSTATISTIK										HAUSHALTUNGSSTATISTIK										WOHNUNGSSTATISTIK									
Zählort		Geb. Jahr		Arbeitsort		Beruf		Eink.		Kind.		Verm.		Gew. Miet.		Dienst.		inf. Pers.		ART. F. D.		Wohn. räum.		Nutz.		Schlaf. räum.		Sonn. räum.	
O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.	O. A.	Gem.
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9

Мал. 2.54. Перфокарта, що використовувалася під час перепису населення в Німеччині в 1910 р.

Заснована Голлерітом у 1896 р. фірма з масового випуску табуляторів, зазнавши низку реорганізацій і зміну власників, із 14 лютого 1924 р. стала називатись **IBM Corporation**. Сьогодні – це один з основних виробників потужних комп'ютерів.

До середини ХХ ст. розроблялися різноманітні конструкції механічних й електромеханічних обчислювальних пристроїв. Вони давали змогу значно прискорити процес обчислення.



Мал. 2.55. Конрад Цузе

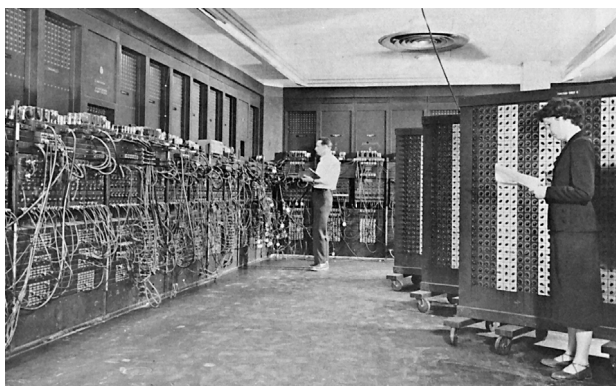
У кінці 30-х і в 40-х роках ХХ ст. вчені та конструктори різних країн створили якісно нові обчислювальні машини, у яких зазвичай використовувалися електричні пристрої. Так, у Німеччині в 1941 р. **Конрад Цузе** (1910–1995) створив першу обчислювальну машину на електромеханічних реле, яку назвав **Z3**. У ній використовувалося двійкове кодування. У 1950 р. він створив комп'ютер на електронних лампах – **Z4**.

У 1941 р. у США **Джон Вінсент Атанасов** (1903–1995) разом зі своїм асистентом **Кліфордом Беррі** (1918–1963) створили перший комп'ютер на електронних лампах з використанням двійкового кодування – **ABC** (англ. *Atanasoff Berry Computer*).

У 1944 р., також у США, **Говард Ейкен** (1900–1973) на замовлення IBM створив обчислювальну машину **Mark-1** на електромагнітних реле з автоматичним керуванням послідовністю операцій.

У 1943–1944 рр. в умовах цілковитої таємності вчені Великої Британії створили обчислювальну машину **Colossus**, призначену для дешифрування радіограм нацистської Німеччини. Від інших машин того часу її відрізняло те, що програма, за якою вона працювала, зберігалася в пам'яті самої машини, інші ж машини вводили по черзі команди із зовнішніх пристроїв. Розробку здійснила група англійських учених під керівництвом **М. Ньюмена** (1897–1984) та інженера **Т. Флауерса** (1905–1998), з урахуванням математичних підходів до дешифрування, які запропонував визначний англійський математик **Алан Тюрінг** (1912–1954).

У 1943–1946 рр. у США **Джон Моклі** (1907–1980) і **Преспер Еккерт** (1919–1995) створили **ENIAC** (англ. *Electronic Numerical Integrator and Calculator* – електронний цифровий інтегратор й обчислювач) – **електронну обчислювальну машину** (ЕОМ), що містила 18 000 електронних ламп, важила 30 тонн і виконувала п'ять тисяч операцій за секунду (мал. 2.56). Після завершення цього проєкту вони відразу почали роботу над новим комп'ютером на замовлення військового відомства США – **EDVAC** (англ. *Electronic Discrete Variable Automatic Computer* – елек-



Мал. 2.56. Обчислювальна машина ENIAC

тронний дискретно змінний автоматичний обчислювач). До розробки було залучено відомого американського математика **Джона фон Неймана** (1903–1957), який у статті «Попередня доповідь про машину EDVAC» сформулював основні принципи побудови універсальної обчислювальної машини. Незважаючи на те, що конструкцію комп'ютера розробили П. Еккерт і Д. Моклі, принципи його побудови стали відомі всьому світові як «принципи фон Неймана».

Широкого розповсюдження комп'ютери отримали після появи великої кількості порівняно дешевих обчислювальних машин – так званих *мініЕОМ*, які в подальшому стали називатися персональними комп'ютерами. Перші з них продавались у вигляді електронних конструкторів, з деталей яких користувач міг самостійно зібрати свій комп'ютер, використавши телевізор як монітор. Перші персональні комп'ютери розробили фірми:

- **MITS** (англ. *Micro Instrumentation and Telemetry Systems* – системи мікроінструментів і телеметрії) – комп'ютер **Altair 8800** (1975 р., мал. 2.57);
- **Apple Computer Company** – комп'ютер **Apple I** (1976 р., мал. 2.58);
- **IBM** – комп'ютер **IBM PC** (1981 р., мал. 2.59).



Мал. 2.57. Altair 8800



Мал. 2.58. Apple I



Мал. 2.59. IBM PC

Широке розповсюдження персональних комп'ютерів значно розширило коло задач, що розв'язуються з використанням ЕОМ.

Детальніше історію розвитку обчислювальної техніки можна розглянути за хронологічною таблицею (додаток 3, адреса <https://cutt.ly/NkBXn8o>).

Історія розвитку обчислювальної техніки в Україні

У 1951 р. в Києві під керівництвом **Сергія Олексійовича Лебедева** (1902–1974) було введено в дію універсальну ЕОМ з програмою, що зберігалася в пам'яті, – **МЭСМ** (рос. *Малая Электронная Счётная Машина* – Мала Електронна Обчислювальна Машина). Це була перша ЕОМ у Радянському Союзі. Вона мала такі характеристики:

- загальна кількість електронних ламп – близько 6000;
- двійкова система кодування даних;
- кількість розрядів – 16 для кодування числа й один для кодування знака;
- ємність запам'ятовуючого пристрою – 31 число та 63 команди;
- швидкодія – близько 3000 операцій за хвилину.

Перші програми для цієї ЕОМ складала **Катерина Логвинівна Ющенко** (1919–2001) (мал. 2.60).

У Києві розробку обчислювальних машин було продовжено в Інституті кібернетики, який очолив видатний учений **Віктор Михайлович Глушков** (1923–1982). Під його керівництвом:

- завершено розробку ЕОМ **Київ** (1959 р., безпосередні розробники: Б. В. Гнеденко, Л. М. Дашевський, К. Л. Ющенко);



Мал. 2.60.
К. Л. Ющенко

- створено серію універсальних ЕОМ **Дніпро** (1961 р., головний конструктор Б. М. Малиновський);
- створено ЕОМ для інженерних розрахунків **Промінь** (1963 р.);
- розроблено серію ЕОМ для інженерних розрахунків **МИР** (рос. *Машина для инженерных расчётов* – машина для інженерних розрахунків, 1960-ті роки);
- створено цілий ряд ЕОМ для військових цілей.

Піонер (англ. *pioneer* – перший поселенець, дослідник, першовідкривач) – людина, яка першою торує стежку в новій галузі діяльності.



Мал. 2.61. Лицьова та зворотна сторони медалі В. М. Глушкова «Комп'ютерний піонер»

в Радянському Союзі та Європі мікрокалькулятори **Електроніка** (мал. 2.62).

У 70–90-ті роки ХХ ст. Харківське НВО «Хартрон» і Київський радіозавод розробляли й виготовляли ЕОМ для ракетно-космічних комплексів.

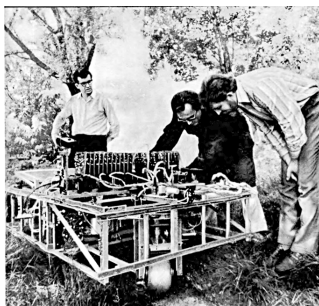
Розроблено й налагоджено (Київський науково-дослідний інститут радіоелектроніки та Київський завод «Буревісник», 70–80-ті роки ХХ ст.) серійний випуск ЕОМ «Карат» для керування кораблями й підводними човнами.

У 1975 р. в Інституті кібернетики АН УРСР під керівництвом М. М. Амосова спроектовано перший в СРСР автономний транспортний робот **ТАІР** (мал. 2.63), який міг рухатися в природному середовищі, оминаючи перепони.

Учені та інженери України протягом 60–80-років ХХ ст. розробили багато комп'ютерів для різноманітних ракетних комплексів. Так, для найбільшої у світі балістичної ракети **Р-36М2** («Сатана») виробництва Дніпропетровського НВО «Південний машинобудівний завод» спеціалістами Харківського НВО «Електроприлад» і ВО «Київський радіозавод» створено ЕОМ, яка забезпечувала керування її запуском і польотом (мал. 2.64).



Мал. 2.62. Мікрокалькулятор Електроніка



Мал. 2.63. Конструктори налагоджують робот ТАІР



Мал. 2.64. Пуск ракети Р-36М2



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/Uhde1FA> або QR-кодом.



Найважливіше в цьому пункті

Перші пристрої для проведення обчислень були створені людиною кілька тисячоліть тому. Вагомий внесок у розвиток обчислювальної техніки зробили Б. Паскаль, Г. Лейбніц, Ч. Беббідж, А. Лавлейс, Г. Голлеріт. Перші електронні обчислювальні машини були створені в 40–50-х роках ХХ ст.: К. Цузе (**Z4**), Д. Атанасовим і К. Беррі (**ABC**), Г. Ейкеном (**Mark-1**), групою англійських учених (**Colossus**), Д. Моклі та П. Еккертом (**ENIAC**), колективом радянських учених під керівництвом С. Лебедева (**МЭСМ**).

В Україні розроблено перші в Радянському Союзі комп'ютери. До розробки ЕОМ для різних галузей економіки та для керування військовою технікою були залучені десятки підприємств і організацій з різних куточків України. Провідною організацією став Інститут кібернетики Академії наук України під керівництвом В. Глушкова.



Дайте відповіді на запитання

- 1°. Які етапи розвитку інформаційних технологій ви знаєте?
- 2°. Як змінювалися засоби опрацювання повідомлень на різних етапах розвитку інформаційних технологій?
- 3°. У чому заслуга Б. Паскаля в розвитку пристроїв для проведення обчислень?
- 4°. Чим відрізнявся арифмометр Г. Лейбніца від пристрою Б. Паскаля?
- 5°. Назвіть складові універсальної машини для обчислень, яку запропонував Ч. Беббідж. Поясніть їх призначення.
- 6°. Чим характеризувався період створення перших електронних обчислювальних машин (40–50 рр. ХХ ст.)? Які вчені зробили внесок у розвиток електронних обчислювальних машин у цей період?
- 7°. Які перші персональні комп'ютери ви знаєте? Коли вони були створені?
- 8°. Яка роль українських учених у розвитку комп'ютерної техніки? Опишіть основні етапи розвитку комп'ютерної техніки в Україні.



Виконайте завдання

- 1°. Підготуйте хронологічну схему розвитку комп'ютерів, починаючи із середини ХХ ст. до кінця ХХ ст. Використайте для цього одну з прикладних програм, яку ви знаєте.
- 2°. Здійсніть в Інтернеті або друкованих ресурсах пошук відомостей про перші електронні обчислювальні машини та заповніть порівняльну таблицю:



Властивість	Значення властивості ЕОМ		
	Z4	Colossus	ABC
Розробники			
Кількість електронних ламп			
Використання двійкового кодування			
Зберігання програми в пам'яті			



3*. Порівняйте значення основних властивостей перших електронних обчислювальних машин: американської ENIAC та створеної в Україні МЭСМ. Результати порівняння запишіть у вигляді таблиці в одній з прикладних програм.



4*. Підготуйте повідомлення про розробку українськими вченими ЕОМ для керування атомними криголами.

5*. Підготуйте повідомлення про особливості програмування перших комп'ютерів, створених в Україні. Використайте в повідомленні спогади К. Л. Ющенко.

2.5. Види сучасних комп'ютерів та їх застосування



1. Які види комп'ютерів ви знаєте? Чим відрізняється їх використання?
2. Які вам відомі приклади застосування комп'ютерів у різних галузях людської діяльності?
3. Яких українських учених, які зробили вагомий внесок у розвиток комп'ютерної техніки, ви знаєте?
4. Що таке Інтернет речей, розумні пристрої?

Види сучасних комп'ютерів

Вам уже відомо, що найпоширенішими на сьогодні є **персональні комп'ютери** (ПК), серед яких розрізняють **стаціонарні** та **мобільні (портативні)**. Стаціонарні ПК, у свою чергу, поділяються за основним призначенням на **офісні, домашні, ігрові** тощо. А мобільні, залежно від особливостей конструкції, – на **ноутбуки, нетбуки, планшетні ПК, смартфони** тощо.

Для розв'язування задач, що потребують великих обсягів складних обчислень, наприклад для розрахунків прогнозу погоди на декілька днів чи тижнів, використовують потужні комп'ютери. Найбільші з них називають **суперкомп'ютерами**. Одночасно суперкомп'ютери можуть виконувати тисячі різноманітних задач і обслуговувати сотні тисяч користувачів по всьому світу.



Мал. 2.65. Комп'ютер Fugaku

На середину 2020 року найпотужнішим комп'ютером у світі був комп'ютер **Fugaku** (мал. 2.65), створений корпорацією Fujitsu для Центру обчислювальних наук RIKEN (Кобе, Японія) у цьому ж році. Центр досліджує і проводить високопродуктивні обчислення в моделюванні клімату, штучного інтелекту, у біології, у теорії катастроф, у фізиці елементарних частинок тощо.

У комп'ютері Fugaku використано процесори Fujitsu SoC A64FX з 48 ядрами в кожному, тактовою частотою 3,2 ГГц.

Загальна кількість процесорних ядер у всіх процесорах комп'ютера приблизно 7,3 млн. Цей комп'ютер може виконувати дії над даними зі швидкістю понад $4 \cdot 10^{17}$ операцій за секунду.

Як **сервери** використовують спеціальні комп'ютери. Такі комп'ютери потужніші та можуть зберігати більші обсяги даних, ніж персональні комп'ютери. Залежно від призначення та кількості користувачів, яким надають послуги такі сервери, вони можуть використовувати один спеціальний серверний процесор (наприклад,



сервер електронної пошти школи) або кілька сотень чи тисяч таких процесорів (наприклад, сервер банку або інтернет-провайдера).

Серед найпопулярніших персональних комп'ютерів варто виділити кишенькові комп'ютери, що поєднують функції мобільного телефону та комп'ютера, – **смартфони** (мал. 2.66). Кількість смартфонів у користуванні мешканців України останнім часом щорічно зростає майже на 20 %. Смартфон став універсальним помічником не тільки в забезпеченні спілкування, але й у використанні сервісів Інтернету. Цікавою тенденцією є доступ до інтернет-послуг з використанням смартфонів жителями малих сіл і міст України, де відсутній або дуже слабкий інтернет-зв'язок. За дослідженням оператора мобільного зв'язку **Київстар** у 2020 році жителі сіл у середньому за день використовували в 3–4 рази більше даних з Інтернету, ніж жителі великих міст. Наприклад, у селі Лошакова Гута (Чернігівська область) кожен власник смартфона за день використовував у середньому 41 380 МБ мобільного Інтернету, у той час як у м. Скадовську (Херсонська область) – 16 620 МБ, а у м. Дніпро – тільки 9360 МБ.



Мал. 2.66.
Смартфон

За даними сайту wearesocial.com на початку 2020 року понад 53 % користувачів використовують для підключення до Інтернету смартфони. Середньо статистичний користувач використовує Інтернет протягом майже 7 год на день або понад 40 % часу, коли ми не спимо.

Смартфони та Інтернет кардинально і швидко змінюють життя людей.

Основні властивості смартфонів і приблизні їх значення на середину 2020 року наведено в таблиці 2.7.

Таблиця 2.7

Значення основних властивостей смартфонів

<i>Властивість</i>	<i>Значення властивості</i>
Оперативна пам'ять, ГБ	4–12
Вбудована пам'ять, ГБ	64–512
Модель процесора	Exynos 9611; Kirin 810; Apple A13
Діагональ дисплея, дюймів	4,6–6,8
Основна камера, Мпксель	16–64
Стандарти зв'язку	4G; 5G
Операційна система	Android; iOS
Виробник	Samsung; Xiaomi; Apple; HUAWEI; OnePlus

Застосування комп'ютерів

У наш час комп'ютери застосовують у науці, промисловості, торгівлі, управлінні, банківській системі, освіті, медицині, транспорті, зв'язку, сільському господарстві, системі соціального забезпечення та інших галузях господарства, у побуті.

Одне з багатьох застосувань комп'ютерів у науковій сфері – це проведення так званих *комп'ютерних експериментів*. Адже проведення багатьох наукових досліджень пов'язано зі значними труднощами – матеріальними, технічними, енергетичними тощо. Особливості комп'ютерного моделювання ви вивчали в 7-му класі.

Важливу роль відіграє комп'ютер на виробництві. Моделювання і конструювання різноманітних виробів з використанням комп'ютера значно скорочує термін їх розробки, підвищує їх ефективність і якість, знижує вартість. Наприклад, якщо раніше, до застосування комп'ютера в проектуванні нової моделі автомобіля, від моменту виникнення ідеї до її виробництва проходило 5–6 років, то тепер цей час становить менше ніж один рік.

У сфері обслуговування комп'ютер використовується для зберігання та опрацювання різноманітних даних: текстів, таблиць, баз даних, малюнків і фотографій, мультимедійних даних. Важко сьогодні назвати таку установу, де б не використовували комп'ютери. Картотеки в бібліотеках і лікарнях, виконані на основі комп'ютерної бази даних, у багато разів надійніші й зручніші в роботі, ніж традиційні паперові. Бухгалтер теж сьогодні використовує комп'ютер і за кілька хвилин отримує результати, на обчислення яких раніше йому були потрібні години або дні. Фінансист, не виходячи зі свого кабінету, має змогу стежити за станом справ на біржі, простим натисненням на клавіші перевести на будь-який рахунок певну суму грошей.



Мал. 2.67. Держава в смартфоні

Усе більшу роль у комп'ютеризації всіх сфер життя людини відіграють смартфони. З їх використанням здійснюється більшість покупок в Інтернеті, керування «розумними» пристроями дома, обмін даними в мережах тощо.

Наприклад, в Україні з 2019 року реалізується програма «Держава в смартфоні» (мал. 2.67). У рамках цієї програми передбачено, що користувачі без відвідувань різноманітних офісів державних і фінансових установ зможуть:

- відкрити банківський рахунок;
- здійснити безготівкові платежі за різноманітні послуги державних і комунальних установ;
- надіслати звіти про свою комерційну діяльність в електронному вигляді, з використанням електронного цифрового підпису до державних органів;
- отримати гарантійні талони на різноманітну техніку та обладнання в електронному вигляді;
- отримати доступ до різноманітних реєстрів:
 - дозволів на міжнародні перевезення;
 - земельних ресурсів;
 - лісових ресурсів;
 - лікарських препаратів і термінів дії сертифікатів на них

тощо.

Для керування «розумним» будинком не потрібні додаткові комп'ютери, достатньо мати смартфон і засіб, що буде об'єднувати всі «розумні» пристрої в єдиний блок. Таким пристроєм є **хаб** (англ. *hub* – центр діяльності). На малюнку 2.68 подано один з варіантів підключення пристроїв «розумного» будинку. Використовуючи смартфон, господар будинку може:

- отримувати дані про характеристики безпеки всередині будинку – відсутність пожежі (сенсор диму, мал. 2.68, 7); прориву води (сенсор води, мал. 2.68, 6), витоку газу (сенсор газу, мал. 2.68, 4), проникнення сторонніх осіб у будинок (сенсор руху, мал. 2.68, 1) і вебкамера, мал. 2.68, 11) тощо;
- контролювати роботу побутових пристроїв – робота-пилососа (мал. 2.68, 13), очищувача повітря (мал. 2.68, 12), кондиціонера або іншого пристрою, використовуючи для цього бездротовий вимикач (мал. 2.68, 5);



- | | |
|--------------------------|------------------------------------|
| 1. Сенсор руху | 8. Хаб будинку |
| 2. Тепла підлога | 9. Бездротовий вимикач |
| 3. Сенсор температури | 10. «Розумні» лампочки |
| 4. Сенсор витoku газу | 11. Вебкамера |
| 5. Бездротова кнопка | 12. Очищувач повітря |
| 6. Сенсор наявності води | 13. Робот-пилосос |
| 7. Сенсор диму | 14. Смартфон з програмою керування |

Мал. 2.68. Схема керування пристроями «розумного» будинку

- керувати освітленням будинку, використовуючи для цього бездротовий вимикач освітлення (мал. 2.68, 9) та «розумні» лампи (мал. 2.68, 10).

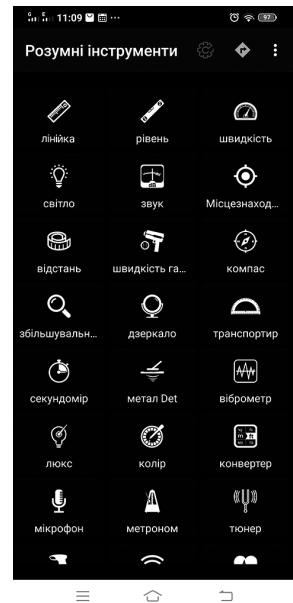
Усі ці пристрої вже сьогодні можна придбати і використовувати вдома.

Не виходячи з дому або ще по дорозі на роботу чи додому, можна здійснити замовлення їжі, товарів, квитків на транспортні засоби, бронювання готелей, туристичних поїздок, сплатити комунальні послуги, записатися на прийом до лікаря тощо.







Застосування смартфонів для навчання

Вам уже відомі можливості використання комп'ютерів для навчання, під час карантину ви активно використовували різноманітні освітні ресурси мережі Інтернет. Однак, використовуючи певні програми для смартфонів, можна навчатися в будь-якому місці та в будь-який час.

Зокрема, проводити різноманітні фізичні та математичні дослідження в себе вдома або в лісі чи в полі. Наприклад, одна з програм для операційної системи **Android** – **Розумні інструменти** (англ. *Smart Tools*) має такі інструменти для досліджень (мал. 2.69):



Мал. 2.69. Інтерфейс програми **Розумні інструменти**

- **лінійка**  – для вимірювання розмірів малих тіл з використанням екрана смартфона;
- **рівень**  – для встановлення точності розміщення горизонтальних і вертикальних площин;
- **відстань**  – для встановлення відстаней між об'єктами;
- **компас**  – для визначення сторін світу та орієнтації у незнайомій місцині;
- **люксметр**  – для вимірювання рівня освітленості;
- **металодетектор**  – для визначення наявності металів тощо.

У деяких версіях операційних систем «розумні» інструменти інстальовано за замовчуванням. Якщо це не зроблено, варто скористатися стандартним засобом отримання безкоштовних програм з використанням «магазину» програм.

Ще одним цікавим інструментом для навчання є програма, що поєднує певний об'єкт (маркер), зазвичай зображений на площині, з додатковими віртуальними елементами, об'ємним зображенням або тривимірною анімацією (мал. 2.70). Використовуючи відповідне програмне забезпечення для смартфонів (наприклад, JigSpace, Google Lens, Atom Visualizer, Civilisations AR, Futurio App, Skyscrapers AR), можна розглянути з різних боків історичну будівлю, скульптуру, переглянути, як працює певний механізм чи системи організму людини, переглянути експозицію відомого музею тощо. Головне, що для цього не потрібно нікуди їхати чи йти, а треба лише мати смартфон і відповідне програмне забезпечення. Можна й самому створювати матеріали з доповненою реальністю.



Мал. 2.70. Анімація мамонта в системі доповненої реальності

Основні напрями використання комп'ютерів

Основними напрямками використання комп'ютерної техніки є:

- виконання громіздких обчислень та обчислень з високою точністю;
- створення комп'ютерних моделей об'єктів і проведення комп'ютерних експериментів;
- забезпечення функціонування автоматизованих систем керування;
- забезпечення зберігання та опрацювання великих обсягів даних;
- забезпечення швидкого обміну даними;
- керування промисловою, побутовою та військовою технікою з використанням вбудованих комп'ютерів;



- підтримка вивчення навчальних предметів;
- організація дистанційного навчання учнів, студентів, фахівців, що особливо зручно для віддалених малих населених пунктів, для людей з обмеженими можливостями тощо.

Конфігурація комп'ютера під потребу

Залежно від того, для яких цілей буде використовуватися комп'ютер, добирають його складові (*конфігурацію*). При цьому особливу увагу звертають на значення таких властивостей:

- **потужність процесора**, що визначається значенням кількох властивостей – тактової частоти, кількістю ядер, обсягом кеш-пам'яті другого та третього рівнів. Що більші значення цих властивостей, то потужніше процесор;
- **обсяг оперативної пам'яті** – що більше значення, то продуктивність комп'ютера вище;
- **ємність накопичувача на жорстких магнітних дисках** – залежно від обсягів даних, з якими працює користувач, – для відео, аудіо і графічних даних потрібні НЖМД більшої ємності, ніж для роботи з текстовими даними;
- **наявність окремого відеоадаптера** – позитивно впливає на швидкість опрацювання відеоданих. А швидкість опрацювання даних відеоадаптером залежить від продуктивності процесора відеоадаптера та обсягу відео-пам'яті;
- **якість відображення даних монітором** – залежить від розмірів монітора, роздільності та швидкості відклику.

Звичайно, що можуть суттєво впливати на вибір комп'ютера і значення інших властивостей, таких як наявність диску SSD, значення властивостей материнської плати, тривалість роботи в автономному режимі (для мобільних пристроїв), наявність засобів підключення до різного типу комп'ютерних мереж тощо. На добір комп'ютерів буде суттєво впливати й їх вартість.

Зазвичай під час добору комп'ютерів, як і інших товарів, говорять про оптимальне співвідношення між ціною і функціональністю (якістю). Немає сенсу використовувати ігровий комп'ютер для офісної роботи – підготовки текстових документів, опрацювання ділової графіки, роботи з електронними таблицями. Ресурси такого потужного комп'ютера не будуть використані навіть на 10 %, а вартість може бути в 10 разів більшою. Домашні комп'ютери використовують, крім тих задач, що притаманні офісним, ще й для опрацювання мультимедійного контенту: відео та музика, фото, також вони активно використовуються для спілкування в Інтернеті, дистанційного навчання тощо. Тобто варто передбачити в складі цього типу комп'ютерів відповідні пристрої, з відповідними значеннями властивостей.

Значна частина інтернет-магазинів пропонує добирати комп'ютери, використовуючи так звані *розумні фільтри*. Наприклад, сайт **Hotline** (<https://hotline.ua/computer/nastolnye-kompyutery>) залежно від призначення пропонує такі групи комп'ютерів: *Компактний комп'ютер, Комп'ютер для роботи/навчання, Робоча станція, Базова оптимальна конфігурація, Універсальна оптимальна конфігурація, Базовий ігровий комп'ютер, Прогресивний ігровий комп'ютер*. Сайт **Розетка** (<http://rozetka.com.ua/computers-notebooks>) пропонує всього три групи комп'ютерів: *Початковий рівень, Для роботи та навчання, Ігрові комп'ютери (геймерські)*.

Вартість системних блоків стаціонарних комп'ютерів при цьому змінюється в доволі великому діапазоні – від 3,5 тис. грн до 400 тис. грн (за цінами 2020 року). Наведемо приклади значень основних властивостей настільних персональних комп'ютерів для запропонованої вище класифікації (табл. 2.8).

Приклади значень властивостей комп'ютерів різного призначення

Властивість	Значення властивості для комп'ютерів різного призначення		
	Офісний (для навчання та роботи)	Домашній	Ігровий
Тип комп'ютера	Стаціонарний	Стаціонарний	Стаціонарний
Модель процесора	AMD A6-9500	Intel Core i5-7400	Intel Core i9-9900K
Тактова частота процесора, ГГц	3,5	3	3,6
Обсяг оперативної пам'яті, ГБ	8	16	64
Ємність жорсткого диска, ГБ	1000	2000	4000
Тип відеоадаптера	Інтегрований AMD Radeon R5 Series	NVIDIA GeForce GTX 1050	NVIDIA GeForce RTX 2080 Ti
Обсяг відеопам'яті, ГБ	Додаткова відсутня	2	11
Тип звукового адаптера	Інтегрований	Інтегрований	Інтегрований
Монітор	LG 19M45A	ASUS VA24EHE	Samsung Curved C49HG90DMI
Діагональ монітора, дюймів	21,5	23,8	49
Роздільність	1920 x 1080	1920 x 1080	3840 x 1080
Час відклику, мс	5	5	1



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/Mhde2rC> або QR-кодом.



Найважливіше в цьому пункті

Серед комп'ютерів залежно від призначення та швидкості опрацювання виділяють **суперкомп'ютери**, **сервери** та **персональні комп'ютери**. Останні, у свою чергу, поділяються на **стаціонарні** та **мобільні**. Стаціонарні комп'ютери поділяють на **офісні**, **домашні**, **ігрові** тощо. А мобільні – на **ноутбуки**, **нетбуки**, **планшетні ПК**, **смартфони** тощо.

Усе більшу роль у комп'ютеризації всіх сфер життя людини відіграють смартфони. З їх використанням здійснюється більшість покупок в Інтернеті, керування «розумними» пристроями будинку, обмін даними в мережах тощо. Використовуючи певні програми для смартфонів, можна проводити різноманітні фізичні та математичні дослідження, отримати доступ до освітніх ресурсів в Інтернеті.

Залежно від того, для яких цілей буде використовуватися комп'ютер, добирають його складові (*конфігурацію*). При цьому особливу увагу звертають на значення таких властивостей: *потужність процесора*, *обсяг оперативної пам'яті*,



ємність накопичувача на жорстких магнітних дисках (для смартфонів – об'єм зовнішньої пам'яті), наявність окремого відеоадаптера (для смартфонів не враховується), якість відображення даних монітором.







Дайте відповіді на запитання

- 1^о. Які комп'ютери називають персональними? Які види комп'ютерів входять до персональних?
- 2^о. Як поділяються персональні комп'ютери залежно від призначення? Яку ви знаєте класифікацію цих комп'ютерів за значенням інших властивостей?
- 3*. Чим відрізняються різні види мобільних (портативних) комп'ютерів?
- 4^о. Що таке суперкомп'ютер? Для яких цілей використовуються такі комп'ютери? Які значення властивостей цих комп'ютерів є основними?
- 5^о. Що таке смартфон? Чому він має таку назву і для яких цілей використовується?
- 6*. Чому все більше користувачів у всьому світі використовують для підключення до мережі Інтернет смартфони? Чи продовжиться, на вашу думку, ця тенденція в майбутньому?
- 7*. Чому існує значна відмінність у значеннях властивостей різних комп'ютерів? Наведіть приклади класифікації різних видів комп'ютерів.
- 8^о. Які ви знаєте приклади використання смартфонів у навчанні? Чи використовували ви смартфон із цією метою?
- 9^о. Які ви знаєте напрями використання комп'ютерів? Наведіть приклади.
- 10^о. Що впливає на вибір комп'ютера для певних сфер застосування?



Виконайте завдання

- 1^о. Створіть схему класифікації комп'ютерів, використавши будь-яку прикладну програму.
-  2^о. Доберіть комп'ютер для використання як домашній мультимедійний центр. Свій вибір обґрунтуйте.
- 3*. Підготуйте для учнів 7-го класу практичну роботу з фізики з використанням «розумних» інструментів смартфона, наприклад з теми «Вимірювання фізичних величин».
-  4*. Підготуйте повідомлення про можливості використання доповненої реальності на уроках одного з навчальних предметів 8-го класу.
-  5*. Створіть добірку програм доповненої реальності для смартфонів, які можна використати в навчальних цілях. Результати оформте у вигляді презентації або відеоподкасту.
-  6*. Знайдіть відомості про найпотужніші суперкомп'ютери на даний час. Створіть таблицю зі значеннями п'яти найбільш потужних суперкомп'ютерів. Опишіть, де використовується найпотужніший суперкомп'ютер.
- 7^о. Створіть у текстовому процесорі таблицю, аналогічну до таблиці 2.8. Використовуючи дані з інтернет-магазинів, заповніть таблицю актуальними значеннями властивостей персональних комп'ютерів різного призначення. Проаналізуйте, які значення властивостей суттєво змінилися. Чому це відбулося?

«Конфігурація комп'ютера під потребу»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

- Доберіть конфігурацію комп'ютерів для людей різних професій і різних сфер застосування:
 - письменник – працює у своєму кабінеті, використовує комп'ютер для створення своїх прозових творів, пошуку відомостей, для надсилання й отримання електронних листів тощо;
 - студент – використовує комп'ютер для підготовки навчальних завдань, а також для створення власної фонотеки, створення і редагування навчальних відеофільмів;
 - кореспондент інтернет-видання – використовує комп'ютер для написання і своєчасного надсилання в редакцію повідомлень з новинами під час від'їздів у різні частини світу.
- За результатами виконання завдання створіть у текстовому процесорі файл з таблицею значень властивостей комп'ютерів. Заповніть цю таблицю значеннями.

Властивість	Значення властивості для комп'ютерів		
	письменника	студента	кореспондента
Тип комп'ютера			
Модель процесора			
Тактова частота процесора, ГГц			
Обсяг оперативної пам'яті, ГБ			
Ємність жорсткого диска, ГБ			
Монітор			
Діагональ монітора, дюймів			
Роздільність			
Час відклику, мс			

- У цьому самому файлі обґрунтуйте ваш вибір конфігурації комп'ютерів.
- Файл з таблицею та обґрунтуванням збережіть у вашій папці з іменем **практична 2.docx** і надішліть на електронну пошту вчителя/вчительки.

Розділ 3.

Опрацювання текстових даних

У цьому розділі ви дізнаєтеся про:

- системи опрацювання текстів та їх можливості
- формати текстових документів
- вставлення в текстовий документ символів, які відсутні на клавіатурі
- пошук і замінювання фрагментів тексту
- роботу з кількома документами одночасно
- створення в текстовому документі багаторівневих списків, розділів, стовпців, колонтитулів
- стилі та їх використання
- перегляд структури документа та автоматичне створення його змісту
- створення гіпертекстових посилань у текстовому документі
- спільне використання документів
- створення, редагування та форматування спеціальних графічних об'єктів

3.1. Системи опрацювання текстів. Пошук і замінювання фрагментів тексту

1. Що таке *текстовий процесор*? Яке його основне призначення та можливості?
2. З яких об'єктів складається текстовий документ? Опишіть їх властивості.
3. Як у текстовому документі виділити фрагмент тексту?

Системи опрацювання текстів, їх призначення та основні можливості

Вивчаючи інформатику в 5-му класі, ви вже ознайомилися з роботою в текстовому процесорі **Microsoft Word 2010**, а саме: вводили текст, редагували і форматували його, вставляли до тексту графічні зображення та таблиці, друкували текстовий документ.

Крім текстового процесора **Word**, існує багато різноманітних програм, які призначені для створення та опрацювання текстів, їх називають **системами опрацювання текстів**. Усі вони належать до прикладного програмного забезпечення загального призначення. Вибір тієї чи іншої програми залежить від потреб користувача.

Використовуючи ці програми, можна створювати текстові документи, зберігати їх, швидко вносити зміни в раніше підготовлені документи. Можна переставляти фрагменти тексту, копіювати їх у різні місця документа, вставляти в текст частини

з інших документів, а за потреби – скасовувати виконані зміни. У більшості систем опрацювання текстів автоматизовано пошук орфографічних і синтаксичних помилок. Використання цих програм значно полегшує оформлення тексту, додавання в текст формул, малюнків, таблиць тощо. Нарешті, створивши електронний текстовий документ один раз, можна в будь-який момент отримати потрібну кількість його копій як електронних, так і паперових.

Більшість з них підтримує технологію **WYSIWYG** (англ. *What You See Is What You Get* – що бачиш, те й отримаєш), яка дає змогу відображати документ на екрані в такому самому вигляді, у якому він буде надрукований.

Використовують системи опрацювання текстів у всіх галузях людської діяльності для підготовки різноманітних документів, поліграфічних видань, буклетів, плакатів, макетів книжок тощо.

Основним об'єктом опрацювання в цих програмах є **текстовий документ**, у якому можуть міститися різні об'єкти: символи, слова, речення, абзаци, сторінки, малюнки, таблиці, діаграми та ін. Кожен з об'єктів текстового документа має свої властивості (табл. 3.1).

Таблиця 3.1

Об'єкти текстового документа



Об'єкт	Властивості об'єкта
Символ	Шрифт, розмір, колір, накреслення, видозміна, зсув, інтервал між символами та ін.
Абзац	Відступ від краю лівого і правого полів, відступ першого рядка, вирівнювання, міжрядковий інтервал, відстань між абзацами та ін.
Сторінка	Розмір аркуша сторінки, її орієнтація, розміри полів, нумерація, колонтитули та ін.
Малюнок	Тип, розмір, кольори, положення, обтікання текстом та ін.
Таблиця	Кількість рядків і стовпців, їх висота і ширина та ін.




Над кожним з об'єктів система опрацювання текстів може виконувати певний набір операцій з редагування об'єктів і змінення значень їх властивостей (форматування).

Найбільш поширеними форматами текстових файлів є формати **TXT**, **RTF**, **DOCX**, **PDF**, **ODT** тощо. Характеристику цих текстових форматів наведено в таблиці 3.2.

Таблиця 3.2

Характеристика найпоширеніших форматів текстових документів

Формат	Стандартний значок	Характеристика
TXT		У файлі зберігається тільки текст з розбиттям на абзаци, без форматування. Може опрацьовуватися різними текстовими редакторами, які працюють під різними операційними системами
DOCX		У файлі зберігаються текст, малюнки, вставлені об'єкти, значення їх властивостей. Забезпечує значне стиснення файлів порівняно з іншими форматами (до 70 %) та покращує відновлення файлів після пошкодження. Стандартний формат збереження текстових файлів у текстовому процесорі Word (починаючи з 2007 року)

Формат	Стандартний значок	Характеристика
RTF		У файлі зберігаються текст, малюнки, вставлені об'єкти, значення їх властивостей. Використовується різними системами опрацювання текстів для різних операційних систем
PDF		Для перегляду та редагування файлу використовується програма Adobe Reader або браузері. У файлі зберігаються форматування, усі вставлені об'єкти
HTML		Для збереження текстових документів у форматі вебсторінок. Зображення значка залежить від програми браузера, який встановлено основним в операційній системі для перегляду файлів HTML
ODT		Стандартний формат текстових файлів, створених текстовим процесором Open Writer

У 8-му класі продовжимо знайомство з можливостями таких програм на прикладі текстового процесора **Microsoft Word**. Матеріал, наведений у підручнику, описує роботу з версією **Office 365** і може застосовуватися для всіх версії, починаючи з версії **Word 2013** і вище. В електронному додатку до підручника розміщено аналогічні матеріали для **Microsoft Word 2010**.

Стандартним форматом файлів текстових документів, підготовлених з використанням текстового процесора **Word**, є формат **DOCX**, але програма підтримує й інші формати текстових документів. Щоб зберегти текстовий документ не в стандартному форматі, слід у вікні **Збереження документа** відкрити список **Тип файлу** і вибрати потрібний формат.

Вставка в текстовий документ спеціальних символів

Ви вже вмiєте вводити в документ текст і оформлювати його певними шрифтами. Але шрифт має більше символів, ніж вміщується на клавіатурі. Такі символи, які не вмістилися на клавіатурі, можна також вставити в документ.

У текстовому процесорі **Word** можна в документ вставляти літери інших алфавітів, математичні символи, умовні позначки та ін. Наприклад, © ≤ β Σ € ∞ ≈ ☺ 🎵 §. Це вам знадобиться під час підготовки матеріалів до уроків з математики, фізики, хімії та ін. Так само, як у **Word**, ці символи вставляються і в презентації, створені в **PowerPoint**.

Для вставлення в текст цих символів слід виконати такий алгоритм:

1. Установити курсор в потрібне місце тексту.
2. Вибрати на вкладці **Вставка** в групі **Символи** кнопку

Символ 

3. Вибрати потрібний символ у списку, що відкриється (мал. 3.1).



Мал. 3.1.
Список символів

Якщо потрібний символ відсутній у наведеному списку, то слід:

1. Вибрати команду **Інші символи**.
2. Переглянути перелік символів у діалоговому вікні **Символ** (мал. 3.2) на вкладках **Символи** та **Спеціальні символи**. За потреби можна в списку поля **Шрифт** або списку **Набір** вибрати інше значення, щоб вивести у вікні інший набір символів.
3. Вибрати потрібний символ.
4. Вибрати кнопку **Вставити**.




Мал. 3.2. Вікно **Символ**

Тим символам, які будуть траплятися в тексті часто, можна в цьому вікні **Символ** призначити певні сполучення клавіш (вибрати кнопку **Сполучення клавіш** і заповнити поля вікна **Настроювання клавіатури**). Після їх натискання потрібний символ буде вставлятися в поточне місце тексту.

У текстовому процесорі **Word** у документ можна вставляти і символи у вигляді малюнків. Наприклад,



Для цього слід вибрати в списку шрифтів графічні шрифти **MS Outlook**, **Webdings**, **Wingdings** та ін. Під час натискання на клавіші клавіатури в текст уводяться не літери, а маленькі піктограми.

Також до спеціальних символів належать і приховані символи, які ви вже використовували в 5-му класі. Нагадаємо, що до **прихованих** (інша назва **недрукованих**) належать символи, які вводяться користувачем з клавіатури під час набору тексту, але зазвичай вони не відображаються на екрані і не виводяться на папері при друкуванні. Якщо включити **режим відображення прихованих символів** вибором на вкладці стрічки **Основне** в групі **Абзац** кнопки **Відобразити всі знаки** , то у відповідних місцях тексту з'являться спеціальні позначення цих символів (табл. 3.3).

Редагування тексту та форматування всіх символів, які вставлені в текст будь-яким способом, здійснюються за правилами, які ви вже знаєте.

Таблиця 3.3

Приклади прихованих (недрукованих) символів

Символ	Клавіші введення	Позначення	Символ	Клавіші введення	Позначення
Пропуск	Пропуск	.	Нерозривний пропуск	Ctrl+Shift+пропуск	°
Кінець абзацу	Enter	¶	Нерозривний дефіс	Ctrl+Shift+дефіс	—
Табуляція	Tab	→	Розрив рядка	Shift+Enter	↵
М'який перенос	Ctrl+дефіс	¬	Розрив сторінки	Ctrl+EnterРозрив сторінки.....



Для тих, хто хоче знати більше

Деякі послідовності символів під час введення **Word** автоматично замінює на спеціальні символи. Приклади таких послідовностей наведено в таблиці 3.4.

Таблиця 3.4

Уведення спеціальних символів



Послідовність уведених символів	Спеціальний символ	Послідовність уведених символів	Спеціальний символ
Пропуск, дефіс, пропуск	– (тире)	Крапка, крапка, крапка	...
(c)	©	-->	→
(r)	®	<--	←
(tm)	™	<==	↩
:)	☺	<=>	↔
:	☹	==>	➔

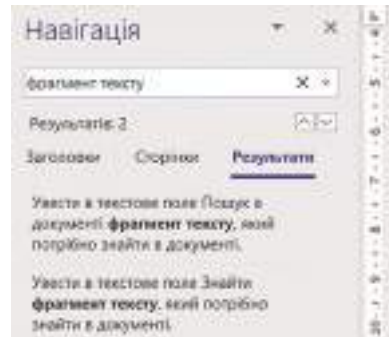
Режим автоматичної заміни вказаних послідовностей символів на спеціальні символи в програмі **Word** встановлено за замовчуванням. Користувач може за власним бажанням додати до списку автозамін свої елементи, вибравши кнопку **Автовиправлення** в діалоговому вікні **Символ**.


Автоматичний пошук і заміна фрагментів тексту

Текстовий процесор **Word** може виконувати автоматичний пошук у документі деякого фрагмента тексту (символа, слова, фрази). Наприклад, перевірити наявність у тексті конкретного слова або фрази.

Для цього слід:

1. Вибрати кнопку **Пошук**  Пошук  у групі **Редагування** вкладки **Основне**, яка відкриває панель **Навігація** у лівій частині вікна (мал. 3.3).
2. Увести в текстове поле **Пошук в документі** фрагмент тексту, який потрібно знайти в документі.

Мал. 3.3. Панель **Навігація** з результатами пошуку

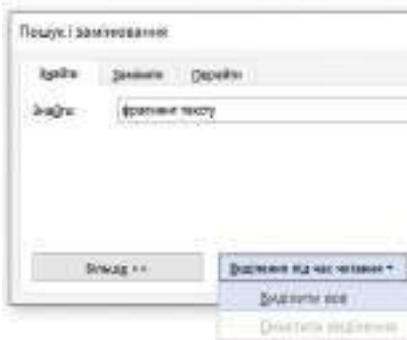
У результаті в панелі **Навігація** буде показана кількість входжень заданого фрагмента до тексту, виведені речення, у яких міститься шуканий фрагмент, і з'являться кнопки  для переміщення між знайденими фрагментами. У тексті всі знайдені фрагменти будуть підсвічені кольором.

У текстовому процесорі також можна виконати розширений пошук. Для цього слід:

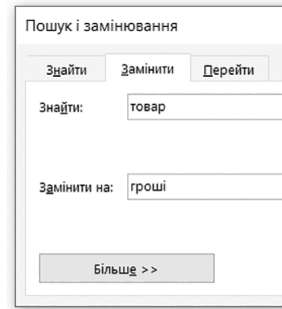
1. Вибрати в списку кнопки **Пошук**  у групі **Редагування**

вкладки **Основне** команду **Розширений пошук**, яка відкриває вікно **Пошук і замінювання** з поточною вкладкою **Знайти** (мал. 3.4).

2. Увести в текстове поле **Знайти** фрагмент тексту, який потрібно знайти в документі.
3. Виконати одну з дій залежно від потреби:
 - вибрати кнопку **Знайти далі**, щоб знайти по черзі кожне входження фрагмента в текст;
 - виконати послідовність дій **Виділення під час читання** \Rightarrow **Виділити все**, щоб одразу виділити в тексті всі входження вказаного фрагмента.



Мал. 3.4. Вікно вкладки **Знайти**




Мал. 3.5. Вікно вкладки **Замінити**

Крім того, вибравши кнопку **Більше**, можна вказати параметри пошуку: напрямки пошуку (вперед від положення курсора, назад від положення курсора, всюди по тексту), пошук слів з урахуванням регістру, слова повністю або частково тощо.

У текстовому процесорі **Word** можна здійснювати пошук не тільки тексту за зразком, а й різні спеціальні знаки (знак кінця абзацу, символ табуляції, знак кінця розділу, знак розриву сторінки тощо), а також текст заданого формату (шрифту, кольору, мови тощо). Для виконання такого пошуку в діалоговому вікні слід вибрати кнопку **Спеціальний** або **Формат** і вибрати потрібний об'єкт із запропонованого списку.

Зауважимо, під час уведення зразка слід уважно стежити за правильністю запису, оскільки пошук фрагмента здійснюється за абсолютним збігом усіх символів. І виконання операції пошуку не розповсюджується на фрагменти документа, які містяться у вбудованих об'єктах (малюнках, діаграмах тощо).

Якщо ж у документі потрібно замінити деякий фрагмент тексту на інший, то слід виконати такий алгоритм:

1. Вибрати в групі **Редагування** вкладки **Основне** кнопку **Замінити** .
2. Увести в поле **Знайти** (мал. 3.5) потрібний фрагмент тексту (наприклад, *товар*).
3. Увести в поле **Замінити на** текст для заміни цього фрагмента (наприклад, *гроші*).

Щоб заміна відбувалася автоматично в усьому документі, слід вибрати кнопку **Замінити все**, а для вибіркової заміни – вибрати кнопку **Знайти далі** і потім за потреби кнопку **Замінити**.


Така можливість текстового процесора корисна в багатьох випадках, наприклад: для автоматичної заміни в усьому документі одного прізвища чи назви закладу на інші; заміна кількох пропусків на один під час редагування текстів, збережених з Інтернету; об'єднання кількох абзаців в один під час заміни знака *кінця абзацу* на *пропуск*.

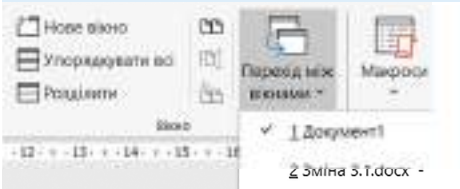


Для тих, хто хоче знати більше

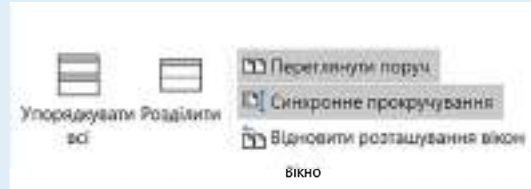
Особливості роботи з кількома вікнами документів

Для керування відкритими вікнами в програмі **Word** існують спеціальні засоби, які розташовані на вкладці **Подання** в групі **Вікно** (мал. 3.6). Так, переключення між вікнами можна здійснити, виконавши такий алгоритм:

1. Вибрати кнопку **Перехід між вікнами**  в групі **Вікно** на вкладці **Подання**.
2. У списку відкритих вікон вибрати потрібне (поточний документ позначається символом ✓).




Мал. 3.6. Згорнута група **Вікно**




Мал. 3.7. Команди групи **Вікно**

Для розділення вікна документа на дві частини слід виконати послідовність дій **Подання** ⇒ **Вікно** ⇒ **Розділити**. На екрані з'явиться смуга розділення вікна, яку слід перетягнути в потрібне місце. У кожній частині вікна – зверху та знизу від смуги розділення – є своя смуга прокручування, використовуючи яку можна відобразити на одному екрані дві різні частини документа.

Для повернення до роботи в одному вікні потрібно виконати команду **Зняти поділ** у групі **Вікно** вкладки **Вигляд** або двічі клацнути на смугі розділення вікна.

Коли робота відбувається з двома версіями одного і того самого документа, що містяться в різних файлах і відкриті в різних вікнах текстового процесора, зручно відобразити їх на екрані поруч. Для виконання цієї операції потрібно виконати **Подання** ⇒ **Вікно** ⇒ **Переглянути поруч**  і вибрати в списку відкритих документів потрібні. Після цього екран буде розділено вертикально навпіл і в кожній із цих частин буде відображено вікно одного з вказаних документів. Прокручування документів і змінення масштабу перегляду у двох вікнах буде відбуватися синхронно.

Щоб відмінити синхронне прокручування, слід вибрати відповідну кнопку в групі **Вікно** (мал. 3.7). Для скасування режиму відображення документів поруч слід повторно вибрати кнопку **Переглянути поруч**  у групі **Вікно**.



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/vhde9vY> або QR-кодом.

**Найважливіше в цьому пункті**

Існує багато різноманітних програм, які призначені для створення та опрацювання текстів, їх називають **системами опрацювання текстів**.

Використовуючи ці програми, можна створювати текстові документи, зберігати їх, редагувати та форматовувати їхні об'єкти, здійснювати автоматизовану перевірку правопису, друкувати текстові документи. Використання цих програм значно полегшує оформлення тексту, додавання в текст формул, малюнків, таблиць тощо.

Основним об'єктом опрацювання в цих програмах є **текстовий документ**, у якому можуть розміщуватися різні об'єкти: символи, слова, речення, абзаци, сторінки, малюнки, таблиці, діаграми та ін. Кожен з об'єктів текстового документа має свої властивості.

Стандартним форматом файлів текстових документів, підготовлених у текстовому процесорі **Word**, є формат **DOCX**. Крім того, **Word** забезпечує роботу з документами, збереженими в інших форматах, наприклад, **RTF, TXT, DOC, PDF, HTML, ODT** тощо.

У текстовому процесорі **Word** можна в документ вставляти спеціальні символи, які відсутні на клавіатурі, – літери інших алфавітів, математичні символи, умовні позначки та ін. Вибрати потрібні символи можна у вікні **Символ**, яке відкривається виконанням **Вставка** ⇒ **Символи** ⇒ **Символ** ⇒ **Інші символи**.

З використанням текстового процесора **Word** можна здійснювати автоматичний пошук і змінення в документі деякого фрагмента тексту (символа, слова, фрази). Для цього слід виконати **Основне** ⇒ **Редагування** ⇒ **Пошук/Замінити** і ввести потрібний текст у відповідні поля. Крім того, вибравши кнопку **Більше**, можна вказати додаткові параметри пошуку: напрямок пошуку, пошук слів з урахуванням регістру тощо.




Текстовий процесор **Word** надає користувачу можливість одночасно працювати з кількома відкритими документами. Це дає змогу користувачу переглядати документи, порівнювати їх зміст, уставляти фрагменти одного документа в інший, об'єднувати їх тощо. Для керування відкритими вікнами в програмі **Word** існують спеціальні засоби, які розміщені на вкладці **Подання** в групі **Вікно**.

**Дайте відповіді на запитання**

- 1°. Що називають системами опрацювання текстів? Які їх основні можливості?
- 2°. У яких форматах може бути збережено файл текстового документа? Який формат є стандартним? У чому особливості цих форматів?
- 3°. Які спеціальні символи можна вставити в текстовий документ? Як це зробити? Коли це може вам знадобитися?
- 4°. У яких випадках і для чого слід використовувати режим відображення прихованих символів? Як його увімкнути?
- 5°. Як здійснити пошук фрагмента тексту в документі? Наведіть приклади використання такої операції.
- 6°. Як здійснити автоматичну заміну в документі фрагмента тексту на інший? Наведіть приклади використання такої операції.
- 7°. Які є способи одночасного відображення вікон відкритих документів на екрані? Як їх установити? Коли зручно це використовувати?
- 8*. Обговоріть з батьками, як вони використовують у своїй діяльності системи опрацювання текстів. Наведіть ці приклади у спільному обговоренні з однолітками.
- 9*. Що, на вашу думку, стимулювало розвиток технологій опрацювання тексту і що ви розумієте під терміном «безпаперові технології»?



**Виконайте завдання**

- 1°. Електронний документ, створений у текстовому редакторі, містить 560 символів. Чому дорівнює довжина його двійкового коду за використання таблиці кодування **KOI-8**? Розрахунки запишіть у зошит.
- 2°. Знайдіть на жорсткому диску вашого комп'ютера текстовий файл формату **TXT** і відкрийте його в текстовому процесорі **Word**. Запишіть у зошит послідовність виконаних дій.
-  3°. Створіть новий текстовий документ і введіть наведений текст. Збережіть документ у вашій папці у файлі з іменем **завдання 3.1.3.docx**.

Сума трьох кутів трикутника $\alpha + \beta + \gamma = 180^\circ$. Площа круга дорівнює $\frac{1}{4}\pi d^2$.

Я народилася під знаком зодіаку Риб ♉. Мені ☺ слухати 🎵, читати 📖, працювати на 💻, вирощувати 🌱. Але я дуже ☹ літати ➦.

- 4°. Відкрийте текстовий файл (наприклад, **Розділ 3\Пункт 3.1\зразок.docx**). Знайдіть і виділіть у тексті всі літери **a**. Замініть їх на символ **@**. Збережіть документ у вашій папці у файлі з іменем **завдання 3.1.4.docx**.



- 5°. Відкрийте текстовий файл (наприклад, **Розділ 3\Пункт 3.1\зразок.docx**). Замініть усі слова *комп'ютер, комп'ютера, комп'ютеру, ...* на слово **ПК**. Збережіть документ у вашій папці у файлі з іменем **завдання 3.1.5.docx**. Поспілкуйтесь зі своїми однокласниками/однокласницями, який алгоритм заміни вони вибрали. Спробуйте інші способи виконання завдання, які використали ваші друзі/подруги. Оцініть, який з названих способів кращий.



- 6°. З'ясуйте, використовуючи **Довідку**, як знайти в тексті всі слова, що починаються з великої літери **M**. Виконайте це на практиці в текстовому файлі, який запропонує вчитель/вчителька (наприклад, **Розділ 3\Пункт 3.1\зразок.docx**). Збережіть документ у вашій папці у файлі з іменем **завдання 3.1.6**.



- 7°. Поясніть, як, використовуючи операцію автоматичної заміни, швидко об'єднати всі абзаци тексту в один. Виконайте це на практиці в текстовому файлі, який запропонує вчитель/вчителька (наприклад, **Розділ 3\Пункт 3.1\зразок.docx**). Збережіть документ у вашій папці у файлі з іменем **завдання 3.1.7.docx**.

- 8°. Відкрийте в текстовому процесорі **Word** три текстових файли (наприклад, з папки **Розділ 3\Пункт 3.1** файли **зразок.docx**, **зразок1.docx**, **зразок2.docx**). Опрацюйте переключення між вікнами, використовуючи кнопки цих вікон на **Панелі задач** і кнопкою **Перехід між вікнами**. Створіть новий текстовий документ і скопіюйте в нього заголовки трьох відкритих документів. Збережіть документ у вашій папці у файлі з іменем **завдання 3.1.8.docx**.



- 9°. Об'єднайтесь зі своїми однокласниками/однокласницями в групи по троє. Знайдіть інформацію про три інструменти, за допомогою яких люди записували тексти в різні часи. Підготуйте презентацію з п'яти слайдів для представлення вашої відповіді. Презентуйте свою групову роботу в класі.



3.2. Створення в текстовому документі багаторівневих списків, розділів, стовпців, колонтитулів



1. Які види списків можна створювати в текстовому документі? Як створити однорівневі списки?
2. Які режими відображення текстових документів ви знаєте? Яке їх призначення?
3. Яке призначення маркерів абзацу на горизонтальній лінійці?

Створення багаторівневих списків у текстовому документі

Ви вже знаєте, що одним з особливих видів форматування абзаців текстового документа є оформлення їх у вигляді **списків**, які використовують для подання переліків об'єктів, опису порядку дій тощо. У текстовому процесорі **Word** можна створювати списки трьох видів: *маркований, нумерований, багаторівневий*.


Марковані та нумеровані списки ви вже створювали в 5-му класі, тому більш детально ознайомимося зі створенням багаторівневих списків, приклади оформлення яких наведено на малюнку 3.8.

Пори року:	Пори року:	Пори року:
1. Зима	1. Зима	1. Зима
• Грудень	а) Грудень	1.1. Грудень
• Січень	б) Січень	1.2. Січень
• Лютий	в) Лютий	1.3. Лютий
2. Весна	2. Весна	2. Весна
• Березень	а) Березень	2.1. Березень
• Квітень	б) Квітень	2.2. Квітень
• Травень	в) Травень	2.3. Травень
3. Літо	3. Літо	3. Літо
4. Осінь	4. Осінь	4. Осінь

Мал. 3.8. Приклади оформлення багаторівневих списків



У багаторівневому списку окремі елементи можуть бути підпорядковані іншим елементам списку. Максимальна кількість вкладень елементів багаторівневого списку – 9 рівнів. І для кожного рівня списку можна встановити свій вид маркування чи нумерації.

Для створення багаторівневого списку слід:

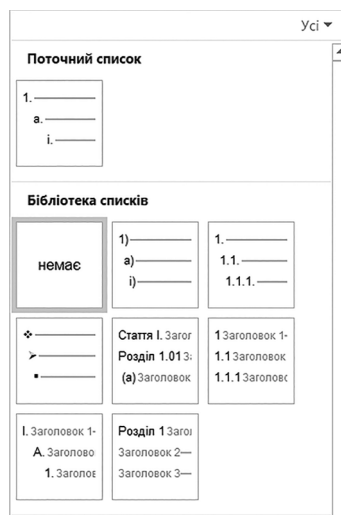
1. Установити курсор у тому місці тексту, де буде створюватися багаторівневий список.
2. Відкрити на **Стрічці** на вкладці **Основне** в групі **Абзац** список кнопки **Багаторівневий список**  (мал. 3.9).

3. Вибрати потрібний вигляд списку в розділі **Бібліотека списків**.

Після вибору виду списку в тексті з'явиться перший номер (маркер) нового списку.

4. Увести окремі елементи списку, змінюючи рівень вкладеності деяких елементів списку за допомогою кнопок **Зменшити відступ**  (перехід на рівень вище) або **Збільшити відступ**  (перехід на рівень нижче), які розташовані на **Стрічці** в групі **Абзац**. Це саме можна зробити, натискаючи клавіші **Tab** і **Shift+Tab**.

Змінити рівень вкладеності можна в будь-який момент – і під час створення, і в подальшому під час редагування списку. Редагування та форматування багаторівневих списків здійснюються аналогічно до однорівневих.



Мал. 3.9. Відкритий список кнопки **Багаторівневий список**


Створення розділів у текстовому документі

Зазвичай у текстовому документі всі сторінки документа відформатовано однаково (розмір аркуша, розміри полів, орієнтація сторінки тощо). Однак під час роботи з документами (зазвичай з багатосторінковими) може виникнути потреба, щоб деякі сторінки документа мали інші параметри форматування. Наприклад, у тексті з книжковою орієнтацією сторінок потрібно вставити таблицю чи малюнок на сторінці з альбомною орієнтацією або деякий фрагмент документа потрібно розмістити в стовпці. Таке форматування документа можливо зробити, розділивши документ на розділи.

Щоб поділити текстовий документ на розділи, у тому місці, де починатиметься інший розділ, потрібно вставити **розрив розділу**. Якщо розділів у документі повинно бути кілька, то для кожного з розділів вставляється свій розрив. За відсутності розривів розділів текстовий документ розглядається як один розділ.

Існує кілька видів розривів розділів:

- з наступної сторінки – розділ починається з початку наступної сторінки;
- неперервна – розділ починається в поточному місці документа;
- з парної сторінки – розділ починається з наступної парної сторінки;
- з непарної сторінки – розділ починається з наступної непарної сторінки.

Створення розділів бажано виконувати в режимі перегляду **Макет друку** і за включеного режиму відображення *прихованих* (недрукованих) символів. Потім потрібно виконати **Макет** ⇒ **Параметри сторінки** ⇒ **Розриви**  і вибрати в списку потрібний вид з переліку **Розриви розділів**. У місці розриву розділів з'являється прихований (недрукований) символ **Розрив розділу**, наприклад

————— Розрив розділу (з наступної сторінки) —————.

Для видалення розриву розділів потрібно видалити в документі прихований символ **розриву розділу**. При цьому текст над місцем видаленого розриву стає частиною наступного розділу і його форматування автоматично змінюється відповідним чином.

Створення стовпців у текстовому документі

Текстовий процесор **Word** надає можливість розмістити текст документа в кількох стовпцях. Як це, наприклад, роблять у газетах і журналах (мал. 3.10).

Віртуальна реальність — високорозвинута форма комп'ютерного моделювання, яка дозволяє користувачеві зануритись у штучний світ і безпосередньо діяти в ньому за допомогою спеціальних сенсорних пристроїв, які пов'язують його руки з аудіовізуальними ефектами. При цьому зорові, слухові, дотикові і моторні відчуття користувача замінюються їх імітацією, що її генерує комп'ютер. Характерними

ознаками віртуальної реальності є: моделювання в реальному масштабі часу; імітація оточення з високим ступенем реалізму; можливість діяти на оточення і мати при цьому зворотний зв'язок.

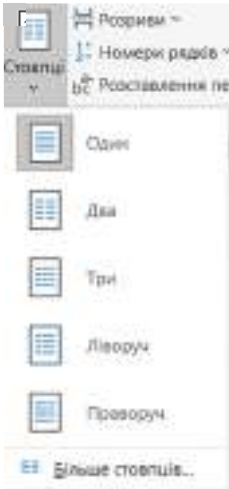
Штучний простір, створений комп'ютерами, який має всі ознаки реальності як такої, що піддається проникненню і трансформації ззовні. При цьому у віртуальній реальності можливі комунікації не лише з іншими людьми, але і з

віртуальними, штучними персонажами.

Термін «віртуальність» уперше з'явився в XVII сторіччі, в розробках класичної механіки, як позначення математичного експерименту, обмеженого об'єктивною реальністю, зокрема, накладеними зовнішніми обмеженнями і зв'язками. А власне формулювання «віртуальної реальності» трапляється вже в XIX сторіччі.

Уривок з Вікіпедії

Мал. 3.10. Приклад тексту, який розміщено в трьох стовпцях

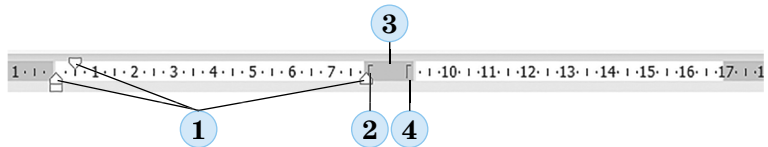


Мал. 3.11. Вибір кількості стовпців у списку кнопки **Стовпці**

Щоб деякий фрагмент тексту розмістити в кількох стовпцях, його потрібно виділити та виконати послідовність дій **Макет** ⇒ **Параметри сторінки** ⇒ **Стовпці**. Потім у списку слід вибрати необхідну кількість стовпців (мал. 3.11). Текст автоматично буде розміщено відповідно до вибраного способу.

Щоб скасувати розміщення фрагмента тексту в стовпцях, цей фрагмент потрібно виділити і встановити для нього режим відображення в один стовпець. При цьому в тексті перед першим рядком і після останнього рядка фрагмента тексту залишаються приховані символи **Розрив розділу**, які бажано видалити.

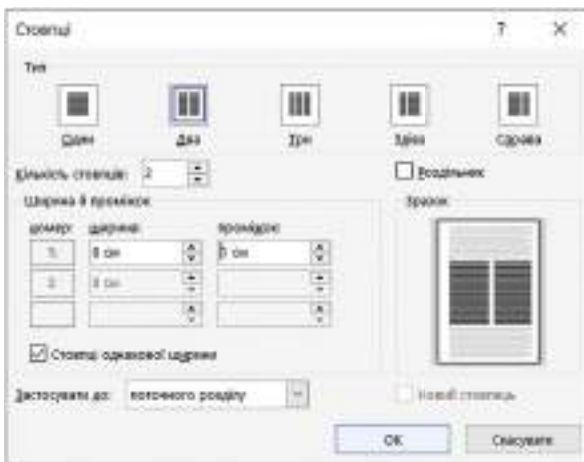
Коли курсор знаходиться в тексті, розміщеному в стовпцях, на горизонтальній лінійці, крім маркерів абзацу, з'являються *маркери меж стовпців* (мал. 3.12), перетягуючи які можна змінювати ширину стовпців і відстань між ними.



1. Маркери абзаців
2. Маркер правої межі першого стовпця
3. Проміжок між стовпцями
4. Маркер лівої межі другого стовпця

Мал. 3.12. Маркери меж колонок на горизонтальній лінійці

Форматування стовпців можна виконати й у діалоговому вікні **Стовпці** (мал. 3.13), яке відкривається **Макет** ⇒ **Параметри сторінки** ⇒ **Стовпці** ⇒ **Більше стовпців**. Тут можна встановити кількість стовпців, ширину кожного з них, відстань між окремими стовпцями, наявність розділювача (вертикальна лінія між стовпцями). Після встановлення значень властивостей слід вибрати кнопку **ОК**.



Мал. 3.13. Вікно **Стовпці**

Створення колонтитулів у текстовому документі


У багатосторінкових документах часто використовують **колонтитули** (франц. *colonne* – стовпець, колонка, лат. *titulus* – заголовок) – це службові повідомлення, які розміщують на полях сторінки і повторюють на інших сторінках документа.

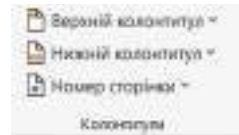
Розрізняють *верхній*, *нижній* і *бічні* колонтитули. Колонтитули можуть містити номери сторінок, назву документа або поточного розділу, прізвище автора, графічні зображення тощо.

Колонтитули першої сторінки, парних і непарних сторінок можуть розрізнятися. Також можуть бути різними колонтитули різних частин документа. Наприклад, у цьому підручнику різним розділам відповідають різні колонтитули.

На сторінки текстового документа у **Word** можна вставляти колонтитули, скориставшись готовою колекцією шаблонів колонтитулів або створити власні колонтитули, які можна зберегти в колекції. Відображаються колонтитули в документі тільки в режимі **Макет друку**. Опрацювання основного тексту документа під час роботи з колонтитулами неможливе.

Для створення на кожній сторінці документа колонтитула з готової колекції шаблонів слід:


1. Вибрати на **Стрічці** вкладку **Вставлення**.
2. Вибрати в групі **Колонтитули** одну з команд **Верхній колонтитул** або **Нижній колонтитул** (мал. 3.14).
3. Вибрати в запропонованому списку шаблонів колонтитулів потрібний варіант.
4. Увести потрібний текст у поля шаблону, використовуючи елементи керування тимчасової вкладки **Верхній і нижній колонтитул**, яка з'являється на **Стрічці** (мал. 3.15).
5. Закрити колонтитул, виконавши **Верхній і нижній колонтитул** ⇒ **Конструктор** ⇒ **Закрити** ⇒ **Закрити колонтитули** .




Мал. 3.14. Група Колонтитули



Мал. 3.15. Шаблон верхнього колонтитула і тимчасова вкладка Верхній і нижній колонтитул

Вміст колонтитула вводиться в поле, яке обмежене штриховою лінією. Використовуючи кнопки груп **Колонтитули** та **Вставлення**, у колонтитул можна вставити номер сторінки – кнопка **Номер сторінки** , поточну дату та час – кнопка **Дата та час**, рисунок – кнопка **Зображення** та ін. Розміщення вмісту колонтитула відносно верхнього чи нижнього краю сторінки регулюється відповідними лічильниками в групі **Розташування**.

Для переходу з поля верхнього колонтитула до поля нижнього колонтитула і назад використовують кнопку **Перейти до Верхнього (Нижнього) колонтитула**  в групі **Навігація**.

Створений колонтитул можна застосувати до всього документа, до парних чи непарних сторінок або до першої сторінки. Вибрати область застосування створеного колонтитула можна в групі **Параметри** вкладки **Верхній і нижній колонтитул**. Установлення значень деяких властивостей колонтитулів також можна здійснити і на вкладці **Макет** діалогового вікна **Параметри сторінки** вкладки **Макет**.

Якщо колонтитули для різних частин документа повинні бути різними, то перед їх створенням документ слід поділити на *розділи*.

Для змінення вставленого колонтитула слід виконати **Вставлення** ⇒ **Колонтитули** ⇒ **Верхній (Нижній) колонтитул** ⇒ **Змінити верхній (нижній) колонтитул** і виконати потрібне редагування. Для видалення колонтитула слід виконати **Вставлення** ⇒ **Колонтитули** ⇒ **Верхній колонтитул (Нижній колонтитул)** ⇒ **Видалити верхній (нижній) колонтитул**.



Для тих, хто хоче знати більше

Табуляція та її використання

Ви знаєте, що натискання на клавішу **Tab** переміщує курсор уздовж рядка на задану відстань. За замовчуванням ця відстань становить 1,25 см. Натисканню клавіші **Tab** відповідає прихований символ →.






Використовуючи клавішу **Tab**, можна розташувати об'єкти тексту в певних місцях рядка, установивши для цього на лінійці кілька позначок табуляції в потрібних місцях. Простір між об'єктами в рядку, який утворюється після натискання клавіші **Tab**, можна заповнити послідовністю деяких символів (крапками, тире, підкресленням тощо). Наприклад:

1) Зима	Весна	Літо	Осінь
Грудень	Березень	Червень	Вересень
Січень	Квітень	Липень	Жовтень
Лютий	Травень	Серпень	Листопад

2) Директор ЗАТ _____ І.В. Сидоренко

- 3) Розділ 1 1 с.
- Розділ 2 8 с.
- Розділ 3 15 с.

У текстовому процесорі **Word** можна використовувати кілька типів табуляції, які вибираються кнопкою **Тип табуляції**, що розміщена ліворуч від горизонтальної лінійки. Якщо послідовно вибирати цю кнопку, то тип табуляції буде змінюватися і можна встановити:

-  – вирівнювання зліва;
-  – вирівнювання по центру;
-  – вирівнювання справа;
-  – вирівнювання за розділювачем.
-  – табуляція з рискою;

Після цього на лінійці відмічають позицію табуляції, вибираючи потрібну позицію вказівником миші. У результаті таких дій на лінійці у вибраних місцях з'являються відповідні позначки (мал. 3.16).



Тип табуляції

Позначки табуляції

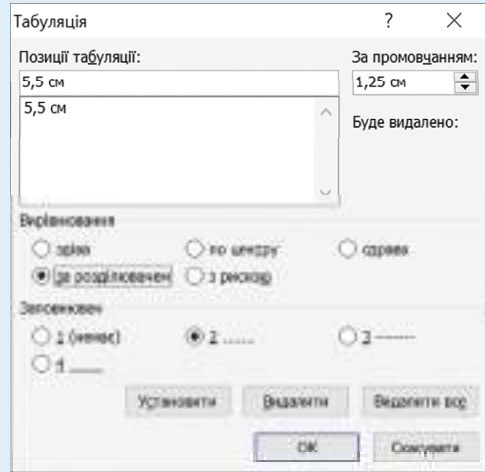
Мал. 3.16. Горизонтальна лінійка з позначками табуляції

Для зміни значень властивостей табуляції потрібно відкрити діалогове вікно **Табуляція** (мал. 3.17), двічі клацнувши будь-яку позицію табуляції на лінійці. У цьому вікні для кожної позиції табуляції можна встановити значення таких її властивостей:

- відстань від межі лівого поля (за замовчуванням вимірюється в сантиметрах);
- спосіб вирівнювання тексту відносно позиції табуляції;
- символ-заповнювач простору між об'єктами в рядку.

Установивши значення властивостей для нової позиції табуляції, слід вибрати кнопку **Установити** і нова позиція буде внесена до списку **Позиції табуляції**. Щоб змінити значення властивостей вже встановленої позиції табуляції, слід вибрати її в списку **Позиції табуляції** і встановити нові значення її властивостей.

Кнопка **Видалити** цього вікна видаляє зі списку вибрану позицію табуляції. Усі встановлені позиції табуляції можна видалити кнопкою **Видалити все**.



Мал. 3.17. Вікно встановлення позицій табуляції



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/uhde8j2> або QR-кодом.



Найважливіше в цьому пункті

Абзаци тексту можна оформити у вигляді списків, зокрема багаторівневих, які використовуються для наочного подання ієрархічного переліку об'єктів. Для цього потрібно виділити фрагмент тексту, виконати **Основне** ⇒ **Абзац** ⇒ **Багаторівневий список** і вибрати в **Бібліотеці списків** потрібний вид. Змінити рівень вкладеності елементів списку можна кнопками **Зменшити відступ** або **Збільшити відступ** , які розташовані на **Стрічці** в групі **Абзац**.

Щоб поділити текстовий документ на розділи в тому місці, де починатиметься інший розділ, потрібно вставити **Розрив розділу**. Створення розділів бажано виконувати в режимі перегляду документа **Макет друку** і при включеному режимі **відображення прихованих (недрукованих) символів**. Далі слід виконати **Макет** ⇒ **Параметри сторінки** ⇒ **Розриви** і вибрати в списку потрібний тип з переліку **Розриви розділів**. У місці розриву розділів з'являється прихований (недрукований) символ **Розрив розділу**. У документі, поділеному на розділи, можна виконати різне форматування окремих частин документа.

Щоб оформити фрагмент тексту в текстовому документі в кілька стовпців, слід виділити потрібний текст, виконати **Макет** ⇒ **Параметри сторінки** ⇒ **Стовпці** ⇒ **Більше стовпців**, установити у вікні **Стовпці** кількість колонок, ширину кожної з них, відстань між окремими колонками, наявність розділювача.

У багатосторінкових документах часто використовують **колонтитули** – це службові повідомлення, які розміщують на полях сторінки і повторюються на інших сторінках документа. У **Word** розрізняють верхній, нижній і бічні колонтитули. Вони можуть містити номери сторінок, назву документа або поточного розділу, прізвище автора, графічні зображення тощо. Колонтитули першої сторінки, парних і непарних сторінок можуть розрізнятися. Також можуть бути різними колонтитули різних розділів документа.

Відображаються колонтитули в документі тільки в режимі **Макет друку**. Опрацювання основного тексту документа під час роботи з колонтитулами неможливе. Для створення колонтитулів використовують елементи керування вкладки **Вставлення** групи **Колонтитули**.



Дайте відповіді на запитання

- 1°. Який список є багаторівневий? Як його створити?
- 2°. Як змінити рівень вкладеності елементів списку?
- 3°. Як розмістити текст у кілька стовпців? Значення яких властивостей колонок можна змінити?
- 4°. Що таке розділи? Для чого вони застосовуються?
- 5°. Як можна встановити ширину стовпців, відступи між ними, їх кількість?
- 6°. Що таке колонтитули? Які види колонтитулів можуть бути на сторінці?
- 7°. Яке призначення колонтитулів? Які об'єкти можна вставити в колонтитули?
- 8*. Як установити різні колонтитули для різних сторінок документа?
- 9°. Що таке табуляція? У яких випадках її використовують?
- 10°. Які, на вашу думку, об'єкти текстового документа, представлені в пункті 3.2, сприяють кращій візуалізації та структуруванню тексту? Поясніть свою думку.



Виконайте завдання

- 1°. Створіть новий документ за зразком. Установіть такі відступи елементів списку: відступ першого рівня – 2 см, відступ другого рівня – 3 см від лівого поля документа. Збережіть документ у вашій папці у файлі з іменем **завдання 3.2.1.docx**. Порівняйте свою роботу зі зразком і оцініть свій результат.
- 2*. Створіть новий документ і оформте у ньому один багаторівневий список, використавши наведений нижче текст. Оформте список на власний розсуд. Збережіть документ у вашій папці у файлі з іменем **завдання 3.2.2.docx**.

- | |
|---|
| <ol style="list-style-type: none"> 1. Сніданок <ol style="list-style-type: none"> 1.1. Чай/кава 1.2. Бутерброд з маслом 2. Обід <ol style="list-style-type: none"> 2.1. Борщ 2.2. Картопляники 2.3. Салат з капусти 2.4. Компот 3. Вечеря <ol style="list-style-type: none"> 3.1. Кефір 3.2. Булочка |
|---|

Кольори умовно поділяють на **теплі** (червоний, оранжевий, жовтий) і **холодні** (блакитний, синій, фіолетовий), а також на **легкі** та **важкі**.

Легкі кольори – прозорі, холодні, світлі, асоціюються з небом і простором.

Важкі кольори – темні, насичені, густі; до таких належать: чорний, темно-сірий, фіолетовий, темно-червоний.

- 3°. Відкрийте вказаний учителем/учителькою файл (наприклад, **Розділ 3\ Пункт 3.2\зразок 3.2.3.docx**). Кожен пункт текстового документа оформте як окремий розділ, який починається з нової сторінки. Установіть таке форматування для розділів документа:

- непарні розділи: орієнтація сторінки – книжкова, поля – по 2 см, нумерація сторінок – зверху по центру;
- парні розділи: орієнтація сторінки – альбомна, поля – по 1 см, нумерація сторінок – знизу зліва.

Збережіть документ у вашій папці у файлі з іменем **завдання 3.2.3.docx**. Яке із завдань було для вас складнішим за інші? Яким чином ви вирішили ці утруднення, щоб успішно виконати завдання?



4*. Відкрийте вказаний учителем/учителькою файл (наприклад, **Розділ 3\Пункт 3.2\зразок 3.2.4.docx**). У відкритому документі розмістіть текст другого абзацу у два стовпці однакової ширини, текст третього абзацу – у три стовпці різної ширини з розділювачем. Збережіть документ у вашій папці у файлі з іменем **завдання 3.2.4.docx**.

5*. Відкрийте вказаний учителем/учителькою файл (наприклад, **Розділ 3\Пункт 3.2\зразок 3.2.5.docx**). Створіть колонтитули: у верхньому запишіть поточну дату з вирівнюванням по центру, у нижньому – ваше прізвище з вирівнюванням по лівому краю. Збережіть документ у вашій папці у файлі з іменем **завдання 3.2.5.docx**.



6*. Відкрийте вказаний учителем/учителькою файл (наприклад, **Розділ 3\Пункт 3.2\зразок 3.2.6.docx**). Уставте в документ колонтитули: верхній на непарних сторінках – *назва текстового документа*, нижній на парних сторінках – *поточна дата і час*. Збережіть документ у вашій папці у файлі з іменем **завдання 3.2.6.docx**.

ПРАКТИЧНА РОБОТА № 3

«Створення текстового документа, що містить об'єкти різних типів»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте текстовий документ *Інструкція з безпеки життєдіяльності та санітарно-гігієнічних норм* з файлу **зразок пр3.docx**, який міститься в папці **Розділ 3\практична 3**.
2. Оформте текст у вигляді багаторівневого списку: текст синього кольору – I рівень, текст фіолетового кольору – II рівень. Установіть нумерацію для списку в такому вигляді $\overset{1.}{\text{_____}}$
 $\underset{1)}{\text{_____}}$.
3. Замініть у тексті всі англійські літери **V** українською літерою **В**.
4. Замініть у тексті всі коми на буквосполучення КОМ, а крапки – на КРПК.
5. Розмістіть текст абзацу зеленого кольору у два стовпці однакової ширини 7 см.
6. Створіть у документі верхній колонтитул з текстом заголовка документа і сьогоднішньою датою; нижній колонтитул з вашим прізвищем.
7. Збережіть документ у вашій папці у файлі з іменем **практична робота 3.docx**.

3.3. Використання стилів у текстових документах.

Структура текстового документа



1. Які властивості сторінок ви знаєте? Як установити їх значення?
2. Які є режими перегляду текстового документа у **Word**? Як їх установити?
3. Як структуровано текст вашого підручника з інформатики? Які об'єкти є на його сторінках?

Стилі в текстовому документі


Створивши текстовий документ, його потрібно оформити належним чином: установити потрібні значення властивостей для об'єктів різних типів – тексту, таблиць, малюнків тощо. Якщо документ великий і має велику кількість різноманітних об'єктів, то форматування кожного окремого об'єкта може зайняти багато часу. Тим більше, якщо часто трапляються об'єкти з однаковим форматуванням, – стежити за однаковим оформленням однотипних елементів документа досить складно.

Для пришвидшення оформлення документів та їх об'єктів зручно використовувати *стилі*. **Стиль** – це набір значень властивостей об'єктів певного типу, який має власне ім'я. Наприклад, *стиль абзаців* визначає значення властивостей вирівнювання тексту на сторінці, відступи від меж аркуша, міжрядковий інтервал тощо; *стиль таблиць* – колір і стиль ліній меж таблиці, колір заливки клітинок таблиці тощо.

Текстовий процесор **Word** за замовчуванням має власну бібліотеку професійно розроблених стилів, так звані *експрес-стилі*, у яких значення властивостей об'єктів підібрано гармонійно, з урахуванням основних вимог до дизайну. Такі стилі розроблено для об'єктів різних типів текстового документа – текстових об'єктів, таблиць, графічних об'єктів тощо (табл. 3.5).

Таблиця 3.5

Приклади експрес-стилів об'єктів різних типів у Word

Тип об'єкта	Ім'я стилю	Зразок оформлення об'єкта	Деякі значення властивостей
Текст	Заголовок 4	Для швидкого оформлення тексту дуже зручно використовувати стилі	Шрифт – <i>Cambria</i> Розмір – 12 Колір – <i>чорний</i> Накреслення – <i>напівжирний курсив</i> Міжрядковий інтервал – <i>1,5 рядка</i> Відступ першого рядка – <i>немає</i> Вирівнювання – <i>за шириною</i> Інтервал перед абзацом – <i>3 пт</i>
Таблиця	Таблиця – сітка 4 – акцент 6		Колір ліній – <i>зелений</i> Колір заливки – <i>перший рядок зелений</i> Стиль лінії – <i>суцільна</i> Ширина лінії – <i>0,5 пт</i>
Фігура	Помаранчевий, акцент 2		Колір заливки – <i>світло-коричневий</i> Прозорість – <i>0 %</i> Тип лінії – <i>суцільний</i> Ширина лінії – <i>1 пт</i> Колір лінії – <i>темно-коричневий</i> Тіньові ефекти – <i>знизу справа</i>

Використання стилів дає змогу однією дією одразу змінити значення кількох властивостей об'єкта. Наприклад, для того щоб відформатувати виділений фрагмент тексту зі значеннями властивостей, які має стиль **Заголовок 4** (табл. 3.5), потрібно виконати 8 окремих дій – вибрати відповідні елементи керування на вкладках або мініпанелі для встановлення значень кожної властивості, а під час застосування стилю **Заголовок 4** – тільки одну дію для вибору стилю з бібліотеки стилів.

Колекцію стилів користувач може доповнювати власними стилями, створюючи їх на основі вже існуючих або наново. Тоді при збереженні документа з ним автоматично зберігаються і застосовані стилі, тобто під час подальшого відкриття документа зовнішній вигляд документа не буде змінюватися.

Використання експрес-стилів

Ескізи експрес-стилів текстових об'єктів документа (символів, абзаців) відображено на вкладці **Основне** в групі **Стилі** (мал. 3.18).

Кожний ескіз із цього списку має оформлення, яке відповідає вибраному стилю, а поруч з іменем стилю може міститися значок, який показує тип об'єктів, до яких цей стиль застосовується: до абзаців – ¶, до символів – а, до символів і абзаців – ¶а.

Відповідні групи елементів керування для вибору експрес-стилів інших об'єктів (таблиць, рисунків, графічних зображень тощо) розміщено на тимчасових вкладках **Конструктор таблиць**, **Формат зображення**, **Формат фігури**, які призначені для опрацювання об'єктів певного типу (мал. 3.19).



Мал. 3.18. Ескізи експрес-стилів для текстових об'єктів



Мал. 3.19. Ескізи експрес-стилів для рисунків і таблиць

У вікні списку експрес-стилів завжди відображається кілька ескізів, які використовувалися останніми. Для перегляду інших ескізів потрібно використати кнопки чи смуги прокручування цього списку. Для відкриття всього списку потрібно вибрати кнопку **Додатково** .

Для застосування експрес-стилю потрібно виділити об'єкт і вибрати в списку потрібний ескіз. *Звертаємо вашу увагу*, що після наведення вказівника на ескіз застосовується попередній динамічний перегляд вибраного формату і можна переглянути вигляд об'єкта із застосуванням вибраного стилю.

Стилі можна перейменувати або видалити, скориставшись командами контекстного меню ескізу стилю.

Для відмови від застосування експрес-стилю та встановлення значень властивостей об'єкта за замовчуванням слід вибрати команду **Очистити форматування** у вікні стилів.

Теми документа та їх застосування

Стильове оформлення може бути застосовано не тільки до окремих об'єктів, а й до всього текстового документа як єдиного цілого.

Тема документа – це стиль оформлення текстового документа, який має ім'я і визначає для його об'єктів значення деяких властивостей (колір, розмір і шрифт символів, спосіб вирівнювання абзаців, міжрядковий інтервал, ефекти графічних зображень тощо).



Мал. 3.20. Бібліотека вбудованих тем документа

У **Word** є бібліотека вбудованих тем, список яких відкривається вибором на вкладці **Конструктор** у групі **Форматування документа** кнопки **Теми** (мал. 3.20).

Тему можна вибрати на початку створення документа або в процесі його підготовки. Під час перегляду теми відбувається динамічний перегляд вибраного стилю. Вибір теми зумовить зміну стилів оформлення таких об'єктів документа, як текст, таблиці, деякі графічні зображення тощо. Усі об'єкти будуть оформлені в єдиному професійно-розробленому стилі. *Слід пам'ятати*, що змінення теми документа впливає тільки на ті об'єкти, які перед цим було відформатовано з використанням стилів.

Правила стильового оформлення документів

Готуючи офіційні документи різних типів (накази, довідки, звіти, листи тощо), слід дотримуватись єдиних вимог і правил щодо їх оформлення, структури та змісту, які встановлюються нормативними актами – *державними стандартами*. Виконання цих стандартів забезпечує:

- оперативне та якісне складання і виконання документів;
- зручне сприймання документа людиною, яка буде його опрацьовувати;
- організацію швидкого пошуку документів;
- здійснення комп'ютерної обробки ділових паперів.

В Україні використовують кілька основних стандартів оформлення документів, які постійно поповнюються. Наприклад, ДСТУ 4163-2003 «Уніфікована система організаційно-розпорядчої документації. Вимоги до оформлювання документів» (<https://cutt.ly/shevOsc>). Відповідно до цього стандарту:

- службові документи оформляються на папері формату A4 (210 см × 297 см) і A5 (210 см × 148 см);
- окремі види документів, можуть оформлюватися на папері формату A3 (297 см × 420 см) і A6 (105 см × 148 см);
- мінімальні розміри полів становлять: ліве – 30 мм, праве – 10 мм, верхнє та нижнє – по 20 мм;
- шрифт основного тексту – *Times New Roman*, розмір – 12–14 пт, міжрядковий інтервал – 1–1,5 рядка, відступ першого рядка абзацу складає 1,25 см;
- зображення логотипа розміщується біля *лівого* поля на одному рівні з назвою організації.

Крім того, у стандарті визначено структуру та правила оформлення списків, таблиць, дат, нумерації та інших об'єктів.

Крім державних вимог до оформлення документів, існують відомчі стандарти – вимоги до оформлення наукових праць, тез доповідей на наукових конференціях, шкільних підручників тощо. Ці правила публікуються у відомчих актах і заздалегідь доводяться до відома авторів документів. Вони можуть дещо відрізнятись від загальноприйнятих і враховувати також поліграфічні особливості видання. Наприклад, правила оформлення науково-дослідницьких робіт МАН (<https://cutt.ly/FkvLwBG>):

- обсяг документа – до 30 друкованих сторінок, аркуші формату А4, формат файлу текстового редактора *Microsoft Word*, на одній сторінці – 40±2 рядки;
- шрифт – *Times New Roman*, 14 пт, *полуторний інтервал*, вирівнювання *по ширині*;
- абзацний відступ – *стандартний комп'ютерний*;
- поля: верхнє – 20 мм; ліве та нижнє – не менше ніж 20 мм, праве – 10 мм;
- заголовки відокремлюються від тексту зверху і знизу *подвійним* інтервалом; заголовки структурних частин друкуються великими літерами по центру рядка; заголовки підрозділів друкують малими (крім першої) літерами з абзацу. Крапку в кінці заголовка не ставлять;
- нумерація сторінок: арабськими цифрами у правому верхньому куті без знака №, першою сторінкою є титульна, яка не нумерується.

Окремі організації для внутрішніх документів можуть розробляти свій власний корпоративний або фірмовий стиль оформлення документів – відповідним чином підібрані кольорова гама та шрифти символів, визначені правила оформлення абзаців документа, вигляд і розміщення логотипа організації тощо. Відповідно до цих вимог розробляються фірмові бланки та шаблони документів. Наприклад, фірмовий стиль компанії представлено на малюнку 3.21.



Мал. 3.21. Приклад фірмового стилю компанії

Якщо документ готується для приватних цілей, також бажано, щоб він був оформлений в єдиному стилі. Для цього слід дотримуватися загальних правил стильового оформлення текстів:

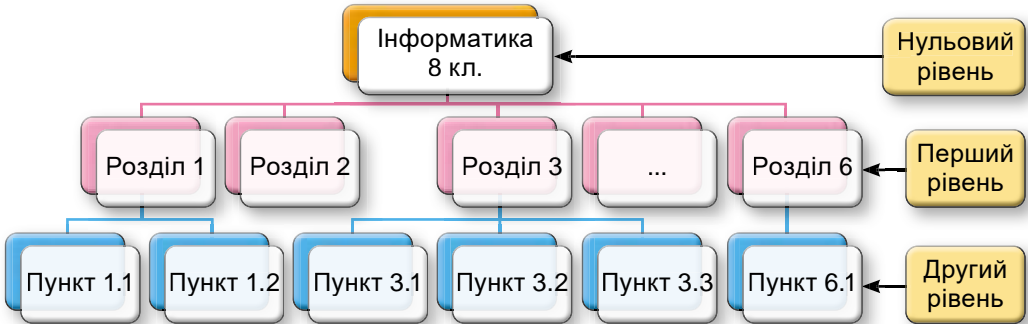
- основний текст документа бажано оформлювати в одному стилі, для виділення заголовків, окремих смислових фрагментів можна використовувати інший формат;
- кількість різних кольорів і шрифтів у документі не повинна перевищувати трьох;
- розмір символів і міжрядковий інтервал потрібно підбирати такими, щоб текст можна було легко прочитати, без напруження очей;
- кольорова гама повинна відповідати призначенню документа – вітальна листівка може бути оформлена яскравими, насиченими кольорами, а звичайний лист до друга – у більш спокійних тонах;
- однотипну структуровану інформацію бажано подавати в таблицях;
- графічні зображення в документі (малюнки, діаграми, схеми) повинні доповнювати зміст тексту, роз'яснювати його окремі моменти.

Структура документа

Багатосторінкові документи (реферати, курсові роботи, брошури, книжки тощо) для зручності ділять на структурні частини – розділи, параграфи, пункти тощо, створюючи таким чином структуру документа. Такі структури документів вам траплялися, наприклад, на уроках української літератури, коли складали план написання твору.

Структура документа – це ієрархічна схема розміщення складових частин документа.


Якщо як приклад розглянути структуру підручника (мал. 3.22), то на верхньому (нульовому) рівні знаходиться назва документа, на першому рівні – назви розділів, другий рівень складають назви пунктів, третій – назви підпунктів і т. д.



Мал. 3.22. Приклад структури документа

Створення структури документа та її використання

Створення структури текстового документа в текстовому процесорі **Word** відбувається із застосуванням стилів оформлення текстових об'єктів **Заголовок 1**, **Заголовок 2**, ..., **Заголовок 9**. Для цього потрібно назви всіх структурних частин документа відформатувати з використанням відповідного стилю. А саме, до назв структурних частин першого рівня (розділів) застосувати стиль **Заголовок 1**, назви частин другого рівня (пунктів) відформатувати із застосуванням стилю **Заголовок 2** і т. д.

Для перегляду структури документа використовують режим перегляду **Структура**, який устанавлюється вибором кнопки **Структура**  на вкладці **Подання** у групі **Подання**.

У цьому режимі зовнішній вигляд документа відрізняється від його вигляду в режимі **Макет друку**, у якому ви звикли працювати (мал. 3.23). У режимі

Програма ЮНЕСКО "Інформація для всіх"

1. Преамбула.




- Інформація і знання є величезним загальнолюдським завоюванням. Вони необхідні для просування освіти, науки, культури і комунікацій, для збагачення культурної різноманітності і відкритості управління. Мандат ЮНЕСКО "сприяти вільному руху цієї словом і справою" ясно указує на роль, яку організація покликана грати в тому, щоб зробити інформацію і знання доступною для всіх, з кінцевою метою подолання розриву між інформаційно-багатими й інформаційно-бідними.
- Для досягнення цієї мети, відповідно до свого конституційного мандата ЮНЕСКО повинна "підтримувати, зберігати, збільшувати, і поширювати знання", гарантуючи і забезпечуючи "збереження і захист" зафіксованих знань людства. Більш того, Організація повинна сприяти "співпраці між народами... у обмін публікаціями" і іншими інформаційними матеріалами і ініціювати "способи міжнародної співпраці, розраховані на те, щоб дати народам всіх країн доступ до опублікованих матеріалів, незалежно від місця і способу їх публікації".
- Міжнародна інтелектуальна співпраця є надзвичайно важливою в нашої світу, який все більш обумовлюється розвитком інформаційних і комунікаційних технологій (ІКТ) і збільшеною конвергенцією їх застосувань. Наслідки цього розвитку, особливо в освіті, науці, культурі і комунікаціях, самими різними способами, прямо або побічно, впливають на всі країни-члени ЮНЕСКО, групи населення і окремих людей.
- Технологічна конвергенція сама, що посилюється, по собі не сприяє ні глобальному обміну моральними, культурними і суспільними цінностями, ні розвитку ґрунтовного доступу до інформації для всіх.

Програма ЮНЕСКО "Інформація для всіх"

- 1. Преамбула.**
 - Інформація і знання є величезним ...
 - Для досягнення цієї мети, відповідно до свого ...
 - Міжнародна інтелектуальна співпраця є ...
 - Технологічна конвергенція сама, що посилюється, ...
 - Нові способи доступу, обробки і захисту інформації ...
 - У цьому контексті ЮНЕСКО з своїм мандатом - ...
- 2. Мандат.**
 - Програма "Інформація для всіх" повинна зіграти ...
 - Програма повинна сприяти зменшенню розриву між ...
 - В рамках програми повинні робитися прями ...
 - Пріоритетним напрямом Програми повинна бути ...
- 3. Цілі програми.**
 - У дусі Міжнародної Декларації Прав Людини, яка ...
 - Як одна з основних програм ЮНЕСКО програма ...
 - Зокрема програма "Інформація для всіх" націлена ...
 - Заохочення і розширення доступу до інформації за ...
 - Роз'язок міжнародної рефлексії і дискусій по ...
 - Сприяння тренінгу, безперервній освіті і навчання ...

Мал. 3.23. Вигляд тексту документа в режимі **Макет друку** та **Структура**


Структура текст структурних частин документа може відображатися в так званому *згорнутому вигляді* і на екран виводяться тільки заголовки структурних частин. Для кращої візуалізації структури рівні документа мають різний відступ від лівого поля. Біля кожного рівня документа є відповідна позначка:

-  – якщо в рівня є підрівні або внутрішній текст;
-  – якщо в рівня відсутній внутрішній текст;
-  – якщо цей текст має найнижчий рівень (так званий *основний текст*, до якого не застосовувалися стилі заголовків).

Включення цього режиму зумовлює появу на **Стрічці** додаткової вкладки **Структура** (мал. 3.24), елементи керування якої використовують для відображення тексту тільки потрібних рівнів документа, переміщення структурних частин у документі, змінення їх рівнів, видалення розділів тексту або додавання нових тощо.



Мал. 3.24. Вкладка **Структура**



Зручність перегляду документа в режимі **Структура** полягає в тому, що є можливість налаштувати відображення будь-якого рівня документа, приховуючи при цьому ті рівні, які займають нижчу ієрархію. Для цього слід на вкладці **Структура** у групі **Знаряддя структури** вибрати потрібний рівень зі списку **Показати рівень**. Якщо потрібно розгорнути зміст деякої окремої структурної частини документа, слід у тексті документа двічі клацнути на значку  біля відповідного заголовка. Повторення цих дій приховає внутрішній текст.








Якщо внутрішній текст має великий розмір шрифту чи є занадто довгий, що заважає роботі, можна відключити відображення форматування (зняти позначку прапорця **Показати форматування тексту** на вкладці **Структура**), а також відображення всіх рядків тексту, крім першого (установити позначку прапорця **Показати лише перший рядок**).

У режимі **Структура** зручно редагувати структуру документа, змінюючи рівень текстових фрагментів і послідовність їх розміщення, використовуючи елементи керування групи **Знаряддя структури** (табл. 3.6).




Таблиця 3.6

**Призначення елементів керування групи Знаряддя структури
вкладки Структура**

Елемент керування	Призначення
Підвищити до заголовка 1 	Для підвищення рівня ієрархії виділеного фрагмента тексту до найвищого. Автоматично до тексту буде застосовано стиль <i>Заголовок 1</i>
Знизити до звичайного тексту 	Для зниження рівня ієрархії виділеного фрагмента тексту до найнижчого. Автоматично до тексту буде застосовано стиль <i>Звичайний</i>

Елемент керування	Призначення
Підвищити рівень  Знизити рівень 	Для переміщення виділеного фрагмента тексту на рівень вище або нижче з відповідною зміною стилю
Вгору  Вниз 	Для змінення місця розташування виділеного фрагмента тексту без зміни рівня ієрархії
Розгорнути  Згорнути 	Для відображення або приховування внутрішнього тексту для виділеного (поточного) заголовка
	Для встановлення для поточного фрагмента тексту певного рівня, який обирається зі списку з відповідною зміною стилю

Слід пам'ятати, що виділення або переміщення заголовка супроводжується виділенням і переміщенням внутрішнього тексту.

Упорядкування тексту можна виконувати і перетягуванням структурних елементів документа за позначки  ,  та  . На екрані відображуються вертикальні чи горизонтальні лінії, які вказують на нове місце фрагмента тексту.

Структуру документа можна надрукувати. На папері вона виглядатиме так само, як і на екрані. Формати абзаців ігноруються, формати символів відтворюються повністю.



Працюємо з комп'ютером

Виконайте тренувальні завдання за посиланням <https://cutt.ly/jhde4U7> або QR-кодом.


Найважливіше в цьому пункті

Для прискорення оформлення документів та їх об'єктів зручно використовувати стилі. **Стиль** – це набір значень властивостей певного типу об'єктів, який має власне ім'я. Текстовий процесор **Word** має власну бібліотеку професійно розроблених стилів, так звані **експрес-стилі**, у яких значення властивостей об'єктів підбрано гармонійно, з урахуванням основних вимог до дизайну. Такі стилі розроблено для різних типів об'єктів текстового документа – тексту, таблиць, графічних об'єктів тощо. Кожен експрес-стиль у бібліотеці має ім'я. Ескізи експрес-стилів текстових об'єктів документа відображено на вкладці **Основне** в групі **Стилі**, для інших об'єктів (таблиць, графічних зображень) – у групі **Стилі** на вкладці **Конструктор таблиць, Формат зображення, Формат фігури**.

Тема документа – це стиль оформлення текстового документа, який має ім'я і визначає для його об'єктів значення деяких властивостей. У **Word** є бібліотека вбудованих тем, список яких відкривається вибором на вкладці **Конструктор** у групі **Форматування документа** кнопки **Теми**.

Готуючи текстові документи різних типів, слід дотримуватись єдиних вимог і правил щодо їх оформлення, структури та змісту, які встановлюються державними або відомчими стандартами, корпоративними вимогами, правилами ведення ділової документації.

Структура документа – це ієрархічна схема розміщення складових частин документа. Найчастіше структуру створюють для багатосторінкових документів, розділяючи їх на розділи, параграфи, пункти.

Створення структури документа відбувається із застосуванням стилів **Заголовок 1, Заголовок 2, ..., Заголовок 9** – назви всіх структурних частин документа слід відформатувати з використанням відповідного стилю. Для перегляду структури документа використовують режим перегляду **Структура**, який встановлюється вибором кнопки **Структура**  на вкладці **Подання** у групі **Подання**. Зручність перегляду документа в режимі **Структура** полягає в тому, що є можливість налаштувати відображення будь-якого рівня документа, приховуючи при цьому ті рівні, які займають нижчу ієрархію.

У режимі **Структура** зручно редагувати структуру документа, змінюючи рівень текстових фрагментів і послідовність їх розміщення, використовуючи елементи керування групи **Знаряддя структури**.



Дайте відповіді на запитання

- 1°. Що таке стилі? До яких об'єктів можна застосовувати стильове оформлення у **Word**?
- 2°. Які властивості об'єктів документа визначають стилі? Для чого їх використовують?
- 3°. Як застосувати стиль до різних об'єктів текстового документа?
- 4°. Які загальні правила стильового оформлення документів?
- 5°. Що таке теми документа? Які властивості документа вони характеризують?
- 6°. Для чого призначено режим відображення документа у вигляді **Структури**? Як його можна встановити?
- 7°. Що таке структура документа? Наведіть приклади структурованих документів.
- 8°. Яке призначення стилів заголовків? Скільки таких стилів є в текстовому процесорі **Word**?
- 9°. Які зміни можна внести в документ у режимі **Структура**?



Виконайте завдання

- 1°. Відкрийте вказаний учителем/учителькою файл (наприклад, **Розділ 3 Пункт 3.3.1 зразок 3.3.1.docx**). Відформатуйте об'єкти текстового документа з використанням таких стилів:
 - заголовки – стиль *Назва книги*;
 - текстові фрагменти – стиль *Строгий*;
 - таблиці – стиль *Таблиця-список 1 (світла), акцент 2*;
 - малюнок – стиль *Металевий овал*.



- 2°. Відкрийте вказаний учителем/учителькою файл (наприклад, **Розділ 3 Пункт 3.3.2 зразок 3.3.2.docx**), у якому до об'єктів 1 та 2 текстового документа було застосовано стильове оформлення. З'ясуйте значення властивостей об'єктів, які встановлюють використані стилі. Запишіть результати в зошит:

№ об'єкта	Тип об'єкта	Стиль об'єкта	Значення властивостей об'єкта
1			

3°. Відкрийте вказаний учителем/вчителькою файл (наприклад, **Розділ 3\Пункт 3.3\зразок 3.3.3.docx**). Визначте, яку тему оформлення документа застосовано. Змініть тему документа на *Галерея*, потім – на *Інтеграл*. Перегляньте оформлення об'єктів документа в обох випадках. З'ясуйте, які значення властивостей об'єктів документа змінюються під час змінення теми документа. Висновок запишіть у зошит. А яка тема документа вам більше до вподоби? Чому саме вона?



4*. Знайдіть в Інтернеті текст державного стандарту ДСТУ 4163-2003 та з'ясуйте правила оформлення в офіційних документах таблиць і рисунків. Запишіть їх у зошит. Розпитайте в близьких, які вимоги до оформлення документів існують у них на роботі.

5°. Відкрийте вказаний учителем/вчителькою файл (наприклад, **Розділ 3\Пункт 3.3\зразок 3.3.5.docx**). Відформатуйте його згідно з вимогами до ділової документації, які описано в параграфі. Збережіть документ у вашій папці у файлі з іменем **завдання 3.3.5.docx**.



6*. Об'єднайтеся у групи по троє. Розробіть корпоративний стиль оформлення документів для громадської організації «Світ чудес». Запишіть його основні вимоги в зошит. Відформатуйте вказаний учителем/вчителькою текстовий документ (наприклад, **Розділ 3\Пункт 3.3\зразок 3.3.6.docx**) із застосуванням розроблених вимог. Збережіть документ у вашій папці у файлі з іменем **завдання 3.3.6.docx**. Презентуйте свою роботу іншим. Яка робота однокласників вам сподобалася? Обґрунтуйте свій вибір.

7°. Відкрийте вказаний учителем/вчителькою документ (наприклад, **Розділ 3\Пункт 3.3\зразок 3.3.7.docx**). Відформатуйте документ, використовуючи такі стилі:

- заголовки 1 рівня – стиль *Заголовок 1*;
- заголовки 2 рівня – стиль *Заголовок 2*;
- заголовки 3 рівня – стиль *Заголовок 3*;
- інші фрагменти тексту – стиль *Звичайний*.

Перегляньте структуру відформатованого документа. Збережіть документ у вашій папці у файлі з іменем **завдання 3.3.7.docx**.

8*. Відкрийте вказаний учителем/вчителькою документ (наприклад, **Розділ 3\Пункт 3.3\зразок 3.3.8.docx**). Перегляньте його в режимі **Структура**. Упорядкуйте розділи документа за їхніми номерами. За потреби установіть необхідні рівні для фрагментів тексту. Збережіть документ у вашій папці у файлі з іменем **завдання 3.3.8.docx**.

3.4. Зміст документа. Гіперпосилання. Колективна робота над документом



1. Що таке *стили*? Для чого їх використовують?
2. Що таке *гіперпосилання*? Для чого і де вони використовуються? Як створити гіперпосилання в презентації?
3. Яким чином здійснюється спільна робота з документами в хмарному середовищі **Google**? Які права доступу можна надати користувачам?

Автоматизоване створення змісту документа

У попередніх пунктах ви ознайомилися з деякими можливостями опрацювання текстових документів – перегляд структури документа, створення розділів, вико-

ристання стильового оформлення та ін. Крім того, у текстовому процесорі **Word** існують й інші засоби для спрощення процесу створення та оформлення великих і складних документів, перегляду їх структурних частин.

Розглянемо, як створюється в текстовому документі його автоматизований зміст.

Зміст документа – це перелік назв структурних частин документа з відповідними номерами сторінок, де цей фрагмент тексту розпочинається (мал. 3.25).

Зміст


Програма ЮНЕСКО «Інформація для всіх»	1
1. Преамбула	1
2. Мандат	2
3. Цілі програми	2
4. Основні розділи програми	3
Розділ 1. Розвиток інформаційної політики на міжнародному, регіональному і національному рівнях	3
Розділ 2. Розвиток людських ресурсів, навиків і умінь у століття інформації	3
Розділ 3. Посилення ролі інституцій в забезпеченні доступу до інформації	3
Розділ 4. Розвиток інструментів, способів і систем інформаційного менеджменту	4
Розділ 5. Інформаційні технології для освіти, науки, культури і комунікацій	4
5. Принципи здійснення програми	4
1. Партнерство і співпраця	4
2. Оцінки	5
3. Принципи здійснення проекту	5


Мал. 3.25. Приклад змісту документа

Зазвичай зміст документа створюється на його початку або в кінці. Кожен елемент змісту є гіперпосиланням на відповідний заголовок у документі.

Якщо для кожної із структурних частин документа використано форматування із застосуванням стилів *заголовків*, то зміст документа можна створити автоматично.

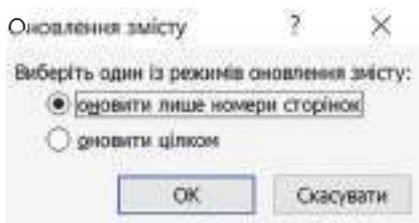
Для цього слід:

1. Установити курсор у тому місці документа, де потрібно розмістити зміст документа.
2. Виконати **Посилання** ⇒ **Зміст** ⇒ **Зміст** .
3. Вибрати зі списку вбудованих зразків потрібний варіант оформлення змісту (мал. 3.26).

Створений зміст можна використовувати для швидкого переміщення по документу, пошуку потрібних структурних частин – розділів, пунктів тощо. Для цього слід у змісті документа вибрати вказівником потрібний заголовок. *Зверніть увагу*, що завжди після наведення вказівника на гіперпосилання курсор набуває вигляду вказівного пальця , але в текстовому процесорі **Word**, на відміну від презентацій і вебсторінок, для переходу за гіперпосиланням потрібно втримувати натиснутою клавішу **Ctrl**.



Мал. 3.26. Список вбудованих зразків оформлення змісту



Мал. 3.27. Вікно **Оновлення змісту**

зміни вносилися в заголовки структурних елементів документа) або оновити лише номери сторінок (якщо змінювалося розташування структурних елементів за рахунок редагування чи форматування документа).

Щоб видалити зміст документа, його слід виділити і натиснути клавішу **Delete**.

Створення гіперпосилань у текстовому документі

Зміст документа – зручний об'єкт текстового документа, але він забезпечує перехід усередині документа тільки до його структурних елементів. Переходи до інших об'єктів у текстовому документі можна зробити з використанням **гіперпосилань**, з якими ви вже ознайомилися під час роботи з вебсторінками і створювали їх у презентаціях.

Гіперпосилання можуть бути створені для переходу в межах одного документа (на фрагмент тексту, зображення, таблицю, схему, бібліографічні джерела або інші об'єкти) – це так зване **внутрішнє** гіперпосилання. Також за допомогою гіперпосилання можна зробити перехід з текстового документа на зовнішні ресурси (вебсайти, матеріали на **Google Диску**, сторінку в соціальних мережах, папку в локальній мережі, на відправлення листа електронною поштою тощо), створивши **зовнішнє** гіперпосилання.

Наприклад, у текстовому документі зі звітом про виконання вами групового проекту з фізики можна не розміщувати в тексті всі ваші напрацювання, а вставити у звітний документ гіперпосилання на матеріали проекту, які розміщені на **Google Дисках** учасників/учасниць вашої групи або на інших ресурсах: на електронну таблицю з обчисленнями та діаграмами, на презентацію перебігу проекту, на фотоальбом тощо. Це зробить ваш звіт більш структурованим і компактним, і дає змогу зекономити час на його підготовку.

Гіперпосилання в тексті потрібно пов'язати з якимось об'єктом документа – словом, фрагментом тексту, малюнком тощо. Для переходу за гіперпосиланням потрібно буде вибрати цей об'єкт вказівником за натиснутої клавіші **Ctrl**. Текст гіперпосилання в текстовому документі, як і на вебсторінках, і в презентаціях, виділяється іншим кольором символів і підкреслюється.

Перед створенням внутрішніх гіперпосилань усі об'єкти документа, на які буде здійснюватися перехід, потрібно відформатувати з використанням стилів **Заголовок** різних рівнів або створити відповідні **закладки**.




Для тих, хто хоче знати більше

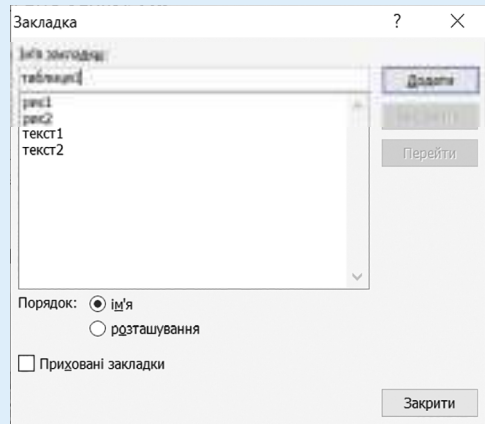
Закладка позначає місце в текстовому документі та має унікальне ім'я. Імена закладок мають починатися з літери і можуть містити літери, цифри та різні символи без пропусків. Закладку можна створити для будь-якого об'єкта текстового документа – фрагмента тексту, графічного

зображення, таблиці, змісту тощо. Використовують закладки для швидкого переходу до потрібних об'єктів документа та для створення гіперпосилань на ці об'єкти.

Для створення закладки слід:

1. Виділити об'єкт тестового документа, для якого створюється закладка.
2. Відкрити вкладку **Вставлення** та в групі **Посилання** вибрати кнопку **Закладка** .
3. Увести ім'я закладки у відповідному полі вікна **Закладка** (мал. 3.28).
4. Вибрати кнопку **Додати**.

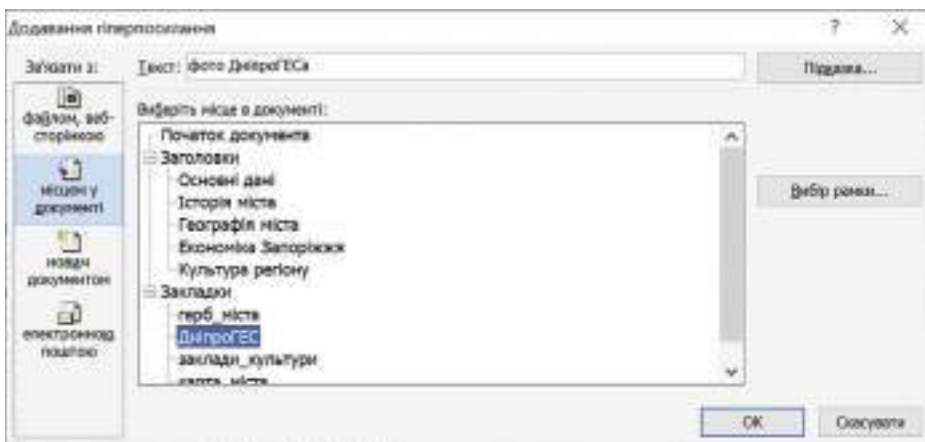
Щоб видалити закладку, слід відкрити вікно **Закладка**, вибрати у списку потрібну закладку і натиснути кнопку **Видалити**.



Мал. 3.28. Вікно **Закладка**

Щоб створити в текстовому документі внутрішнє гіперпосилання, слід:

1. Відформатувати об'єкт документа, на який буде здійснюватися перехід за гіперпосиланням, з використанням будь-якого стилю **Заголовок** або створіть для нього **закладку**.
2. Виділити в документі об'єкт, з яким буде пов'язано гіперпосилання (фрагмент тексту, малюнок тощо).
3. Виконати **Вставлення** ⇒ **Посилання** ⇒ **Посилання**.
4. Вибрати у вікні **Додавання гіперпосилання** в області **Зв'язати з:** кнопку **місцем у документі** (мал. 3.29).

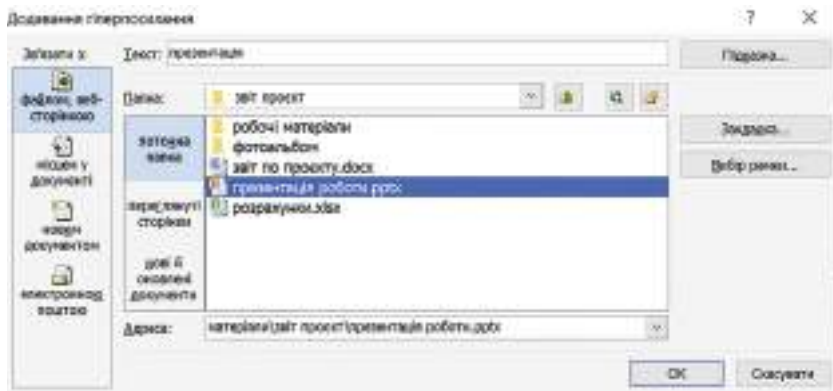


Мал. 3.29. Вікно **Додавання гіперпосилання** для вставлення внутрішнього гіперпосилання

5. Вибрати в області **Виберіть місце в документі** в списку **Заголовки** або **Закладки** об'єкт документа, на який має здійснитися перехід.
6. Вибрати кнопку **ОК**.

Для створення зовнішнього гіперпосилання на файл у локальній мережі чи на носії даних локального комп'ютера слід:

1. Виділити в документі об'єкт, з яким буде пов'язано гіперпосилання (фрагмент тексту, малюнок тощо).
2. Вибрати у вікні **Додавання гіперпосилання** в області **Зв'язати з:** кнопку **файлом, вебсторінкою**.
3. Відкрити в списку **Папка** потрібну папку та вибрати потрібний файл (мал. 3.30).
4. Вибрати кнопку **ОК**.



Мал. 3.30. Вікно **Додавання гіперпосилання** для вставлення гіперпосилання на зовнішній файл

Після вибору такого гіперпосилання буде відкрито вікно програми, яка призначена для роботи з файлами цього типу, і в ньому відкриється вибраний файл. Наприклад, якщо під час створення гіперпосилання було вибрано файл презентації, то після вибору гіперпосилання відкриється вікно програми **Microsoft PowerPoint** і в нього буде завантажено вибраний документ.

Щоб у текстовому документі вставити гіперпосилання на вебсторінку, слід:

1. Виділити в документі об'єкт, з яким буде пов'язано гіперпосилання (фрагмент тексту, малюнок тощо).
2. Вибрати у вікні **Додавання гіперпосилання** в області **Зв'язати з:** кнопку **файлом, вебсторінкою**.
3. Увести або скопіювати адресу потрібної вебсторінки в поле **Адреса** (мал. 3.31).
4. Вибрати кнопку **ОК**.



Мал. 3.31. Вікно **Додавання гіперпосилання** для вставлення гіперпосилання на вебсторінку

Після вибору в текстовому документі такого гіперпосилання буде відкрито вікно браузера, у якому відобразиться вебсторінка, адреса якої була введена в поле **Адреса**.

Уставлені гіперпосилання можна змінювати, копіювати або видаляти, використовуючи команди контекстного меню цього посилання (мал. 3.32).



Мал. 3.32. Команди контекстного меню гіперпосилання


Спільна робота над документами

Сучасні текстові процесори мають засоби для організації спільної роботи над документами кількома користувачами. Такого роду засоби потрібні для забезпечення можливості різним користувачам, учасникам групової роботи співпрацювати в підготовці різних документів, але в той самий час відстежувати зміни, що вносять у документ співавтори.

Ви вже знаєте можливості спільної роботи з документами з використанням хмарних сервісів **Google** в онлайн-режимі, де різні користувачі, маючи відповідний доступ, можуть колективно створювати документи, переглядати та редагувати їх, коментувати та пропонувати зміни, обговорювати в чаті спірні питання. При цьому співавтори можуть перебувати територіально в різних місцях і здійснювати цю роботу одночасно або в зручний для них час.

Такі самі засоби для організації спільної роботи з документами має й остання версія текстового процесора **Word**, який входить до пакета програм **Microsoft Office 365**, що є аналогом хмарних сервісів **Google**.

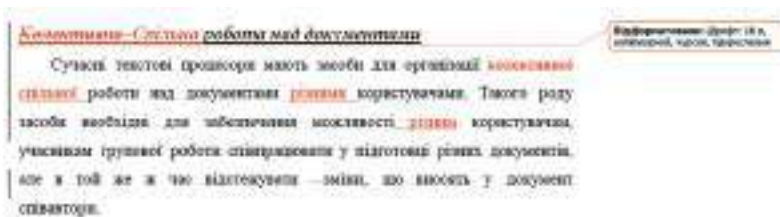
У текстовому процесорі **Microsoft Word**, який встановлено на локальному комп'ютері, також існують засоби для колективної роботи. Але здійснюється робота з документом не одночасно, а різні користувачі вносять свої правки в документ по черзі, виконуючи це офлайн на своєму локальному комп'ютері та надсилаючи змінені документи іншим співавторам для ознайомлення та подальшого опрацювання.

Таке опрацювання текстових документів у **Word** здійснюється в режимі **рецензування**. Щоб увімкнути цей режим, потрібно виконати послідовність дій **Рецензування** ⇒ **Відстеження** (мал. 3.33) і вибрати кнопку **Виправлення** .



Мал. 3.33. Групи команд вкладки **Рецензування**

Якщо після ввімкнення цього режиму здійснювати редагування та форматування документа, то всі внесені зміни будуть відображатися в документі. Дії з редагування документа відображаються в самому тексті документа, а на правому полі – виконані операції форматування.





Мал. 3.34. Вигляд фрагмента тексту в режимі рецензування

Наприклад, якщо в режимі рецензування видалити слово і замінити його на інший фрагмент, то воно з тексту не видаляється, а закреслюється, а той текст, що вставляється, виділяється іншим кольором (мал. 3.34).

Зміни, унесені в документ різними користувачами, будуть відображатися в документі різними кольорами. Під час збереження рецензованого документа всі зміни будуть збережені разом з ним.

Після відкриття такого документа автор документа (або інший користувач) побачить унесені зміни, може ознайомитися з ними і вирішити – прийняти чи відхилити ці правки. При відхиленні – унесені правки зникають, а при прийнятті – вносяться в документ остаточно.

Для опрацювання окремого виправлення потрібно виділити правку і вибрати на вкладці **Рецензування** в групі **Зміни** кнопку **Прийняти**  або кнопку **Відхилити** . Для переходу до попереднього або наступного виправлення слід вибрати відповідно кнопки **Назад** або **Далі**.

Якщо автор документа ознайомився з правками і згоден з ними, то він може вибрати в списку кнопки **Прийняти** команду **Прийняти всі зміни в документі** або в списку кнопки **Відхилити** команду **Відхилити всі зміни в документі**, якщо він не згоден з усіма виправленнями.

Коли всі виправлення будуть опрацьовані, на екрані з'являється повідомлення «Документ не містить приміток або виправлень». Після цього режим рецензування може бути вимкнено повторним вибором кнопки **Виправлення** і нові правки вже не будуть відстежуватись.



Для тих, хто хоче знати більше

У документі, який передбачається для рецензування, автор документа може обмежити права користувачів під час спільного опрацювання документа, а саме: заборонити форматування документа з використанням стилів, установити можливості тільки читання або надання відповідей на запитання форми в документі, визначити конкретних користувачів, які можуть редагувати документ, установити пароль на внесення змін. Усі ці налаштування можна вибрати на панелі **Обмеження редагування**, яка відкривається кнопкою **Обмежити редагування** в групі **Захист** вкладки **Рецензування**.



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/Uhde5qg> або QR-кодом.



Найважливіше в цьому пункті

Зміст документа – це перелік назв структурних частин документа з відповідними номерами сторінок, де цей фрагмент тексту розпочинається. Зазвичай зміст документа створюється на його початку або в кінці. Кожен елемент змісту є гіперпосиланням на відповідний заголовок у документі.

Якщо для кожної із структурних частин документа використано форматування із застосуванням стилів **заголовків**, то зміст документа можна створити автоматично, виконавши **Посилання** ⇒ **Зміст** ⇒ **Зміст**. Якщо під час роботи з документом його текст чи структура змінювалися, то зміст документа потрібно оновити, виконавши **Посилання** ⇒ **Зміст** ⇒ **Оновити таблицю**.

Переходи до будь-яких об'єктів у текстовому документі можна зробити з використанням **гіперпосилань**. Гіперпосилання можуть бути створені для переходу в межах одного документа (*внутрішнє* гіперпосилання) або на зовнішні ресурси (*зовнішнє* гіперпосилання). Гіперпосилання в тексті пов'язуються з якимось об'єктом документа – словом, фрагментом тексту, малюнком тощо. Текст гіперпосилання виділяється іншим кольором символів і підкреслюється.

Перед створенням внутрішніх гіперпосилань усі об'єкти документа, на які буде здійснюватися перехід, потрібно відформатувати з використанням стилів **Заголовки** або створити відповідні **закладки**. Створення гіперпосилань здійснюється у вікні **Додавання гіперпосилання**, для відкриття якого потрібно виконати **Вставлення** ⇒ **Посилання** ⇒ **Посилання**.

У текстовому процесорі **Microsoft Word** для організації спільної роботи з документами використовують режим **рецензування**, який устанавлюється послідовністю дій **Рецензування** ⇒ **Відстеження** ⇒ **Виправлення**. У такому режимі всі внесені зміни відображаються в документі. Під час збереження рецензованого документа всі зміни зберігаються разом з документом.

Після відкриття такого документа автор документа побачить внесені зміни і приймає чи відхиляє ці правки. Опрацювання правок здійснюється елементами керування на вкладці **Рецензування**. Після опрацювання внесених правок слід вимкнути режим рецензування і нові змінення в документі вже не будуть відстежуватись.




Дайте відповіді на запитання

- 1^o. Що таке зміст документа? Для чого він призначений?
- 2^o. Яким чином подається зміст документа? Як ним користуватися?
- 3^o. Як створити зміст документа в текстовому процесорі **Word**?
- 4^o. Що таке гіперпосилання в текстовому документі? Навіщо їх використовують?
- 5^o. Які види гіперпосилання можна створити в текстовому документі?
- 6*. Що спільного і відмінного у створенні внутрішніх та зовнішніх гіперпосилань у документі?
- 7^o. Що таке закладка? Для чого вона використовується?
- 8^o. Що таке режим рецензування документа? Для чого його використовують?
- 9^o. Яким чином користувач може переглянути виправлення в документі? Яким чином вони відображаються в документі?
- 10^o. Які дії в режимі рецензування можна виконувати?
- 11*. Яким чином можна організувати спільну роботу з документом, використовуючи текстовий процесор **Word**, який устанавлено на персональному комп'ютері?



Виконайте завдання

- 1^o. Відкрийте вказаний учителем/вчителькою документ (наприклад, **Розділ 3\Пункт 3.4\зразок 3.4.1.docx**). Відформатуйте назви структурних елементів документа з використанням стилів заголовків відповідних рівнів. Створіть на початку документа його зміст. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.1.docx**.
-  2^o. Відкрийте вказаний учителем/вчителькою документ (наприклад, **Розділ 3\Пункт 3.4\зразок 3.4.2.docx**). Створіть у кінці документа його зміст. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.2.docx**.
- 3^o. Відкрийте вказаний учителем/вчителькою документ (наприклад, **Розділ 3\Пункт 3.4\зразок 3.4.3.docx**). Додайте до назв переможців проєкту «7 чу-

дес України» гіперпосилання на відповідні статті **Вікіпедії** про ці об'єкти. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.3.docx**.



4*. Створіть текстовий документ з переліком ваших друзів, зазначивши *прізвище, ім'я та дату народження*. Додайте до прізвищ кожного з них гіперпосилання на його сторінку в соціальних мережах. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.4.docx**.



5*. Об'єднайтеся в пари. Відкрийте вказаний учителем/вчителькою документ (наприклад, **Розділ 3\Пункт 3.4\зразок 3.4.5.docx**). До кожної назви фізичного приладу додайте гіперпосилання на відповідний малюнок з папки **Розділ 3\Пункт 3.4\фото 3.4.5.docx**. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.5.docx**. Поясніть, чи була перевага в роботі, коли ви виконували роботу вдвох. Чи були непорозуміння під час спільної роботи?



6*. Створіть у текстовому документі таблицю з описом п'яти пам'яток історії вашого регіону, зазначивши назву та місце розміщення. Додайте до назв цих об'єктів гіперпосилання на їх зображення, які знайдіть в Інтернеті та збережіть на власному **Google Диску**. Додайте до місця знаходження гіперпосилання на **Google Карту** цього об'єкта. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.6.docx**.

7*. Відкрийте вказаний учителем/вчителькою документ (наприклад, **Розділ 3\Пункт 3.4\зразок 3.4.7.docx**). Дайте відповіді на запитання в тексті щодо небезпек і ризиків в Інтернеті. Відформатуйте файл на власний розсуд. Перегляньте документ в режимі рецензування. Прийміть усі внесені виправлення. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.7.docx**.



8*. Створіть текстовий документ і сформулюйте в ньому п'ять ваших правил безпечної поведінки в Інтернеті. Відправте документ електронною поштою своєму другові/подрузі на рецензування. Отриманий від друга/подруги документ перегляньте в режимі рецензування. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.8.docx**.



ПРАКТИЧНА РОБОТА № 4

«Стилі. Структура документа. Автоматизоване створення змісту. Гіперпосилання»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Відкрийте вказаний учителем/вчителькою файл (наприклад, **Розділ 3\Пункт 3.4\зразок пр4.docx**).
2. Установіть для всіх нумерованих пунктів тексту *рівень 1*, для маркованих – *рівень 2*.
3. У режимі перегляду **Структура** впорядкуйте структурні елементи документа в порядку номерів пунктів.
4. Відформатуйте таблицю та малюнки з використанням стилю.
5. Відформатуйте назви структурних частин текстового документа з використанням стилів заголовків відповідного рівня.
6. Створіть на початку документа його зміст.
7. До всіх малюнків додайте гіперпосилання на відповідні інтернет-ресурси, які знайдіть самостійно.
8. Збережіть документ у вашій папці у файлі з іменем **практична робота 4_Прізвище.docx**.

9. Розмістіть документ у спільній папці класу, яку вкаже вчитель.
10. Відкрийте із цієї папки файл практичної роботи № 4 сусіда/сусідки праворуч, унесіть в нього такі виправлення: видаліть зміст документа та змініть стилі оформлення таблиці та одного з малюнків.
11. Збережіть файл з вашими виправленнями в спільній папці з іменем **практична робота 4_Прізвище_ред.docx**.
12. Відкрийте файл вашої практичної роботи № 4 після редагування сусідом/сусідкою.
13. Перегляньте виправлення, відхиліть видалення змісту та прийміть зміну форматування об'єктів.
14. Збережіть файл у вашій папці з іменем **практична робота 4_Прізвище_фінал.docx**.
15. Закрийте вікно програми **Word**.



3.5. Створення спеціальних графічних об'єктів у текстовому документі



1. Як можна ввести в текст математичні вирази, наприклад x_3 , x^3 , $\frac{3}{8}$, $\alpha + \beta$?
2. Які графічні зображення можна вставляти в текстовий документ? Як це зробити?
3. Які операції редагування та форматування можна здійснювати над графічними об'єктами в текстовому процесорі **Word**? З використанням яких засобів здійснюються ці операції?




Створення формул у текстовому документі

Під час підготовки в текстовому процесорі реферату для уроку чи наукової статті для учнівської конференції в документ інколи потрібно вставляти формули, рівняння, доведення теорем, що використовують особливу символіку: математичну, фізичну, хімічну та ін.

Наприклад, наведемо нескладні формули та вирази, які записані в одному рядку і які не мають у своєму записі звичайних дробів, знаків коренів та інших нелінійних структур:

$$\text{а) } c_2 m_2 (t_2 - \Theta) = c_1 m_1 (\Theta - t_1); \quad \text{б) } ax^2 + bx + c \geq 0;$$

$$\text{в) } S = v_0 t + a t^2 / 2; \quad \text{г) } \text{Na}_2\text{CO}_3 + 2\text{HCl} \Leftrightarrow 2\text{NaCl} + \text{CO}_2 + \text{H}_2\text{O}.$$

Для створення таких виразів використовують різні спеціальні символи, вставлення яких здійснюється у вікні **Символ (Вставка ⇒ Символи ⇒ Символ  ⇒ Інші символи)**. Уведення символів нижнього і верхнього індексу здійснюється з використанням кнопок **Підрядковий знак ** і **Надрядковий знак **, які розміщено на вкладці **Основне** в групі **Шрифт**.

Але дуже часто в математиці, фізиці трапляються багаторівневі формули та вирази. Наприклад:

$$\text{а) } \frac{h_1}{h_2} = \frac{\rho_1}{\rho_2}; \quad \text{б) } E_k = \frac{mv^2}{2}; \quad \text{в) } \eta = \frac{A_1}{A_2} \cdot 100 \%;$$

$$\text{г) } \frac{3x+7}{4} - \frac{x-3}{2} = \frac{5x+2}{8};$$

$$д) y = \begin{cases} -3x^3 - 2, & \text{якщо } x < 0, \\ \frac{7x}{4} - 2, & \text{якщо } x \geq 0; \end{cases} \quad е) \frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a - 4b)^2}{9a^4 - b^4}.$$

Уставити в текстовий документ такі формули можна, скориставшись спеціальним засобом **Конструктор формул**. Формули в ньому конструюються з окремих символів і структур, використовуючи відповідні шаблони.

Для відкриття **Конструктора формул** потрібно виконати **Вставлення** \Rightarrow **Символи** \Rightarrow **Рівняння** Π . Після цього на **Стрічці** з'явиться тимчасова вкладка **Рівняння** (мал. 3.35, 1), а в документі – спеціальна область для введення формули (мал. 3.35, 2).

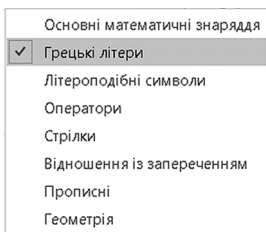


1. Вкладка **Рівняння**
2. Місце для введення формули
3. Кнопки для вставлення символів

Мал. 3.35. Вкладка **Рівняння**

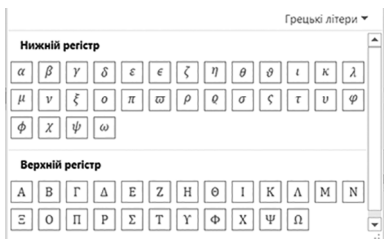
Під час введення формули деякі букви, знаки арифметичних операцій та інші символи можна вводити з клавіатури, а спеціальні – вибирати у відповідних списках групи **Символи** (мал. 3.35, 3).

Усі символи, які можна вставити у формулу, розділено на 8 основних наборів (мал. 3.36), які, у свою чергу, можуть бути розділені по окремих блоках (мал. 3.37).



Мал. 3.36. Список основних наборів символів

На **Стрічці** в групі **Символи** відображаються деякі символи того набору, який використовувався останнім. Для перегляду інших символів цього набору слід скористатися кнопкою **Додатково** ∇ на смузі прокручування в групі **Символи**. Щоб відкрити перелік назв усіх 8 наборів символів, потрібно в поточному списку вибрати його заголовок.

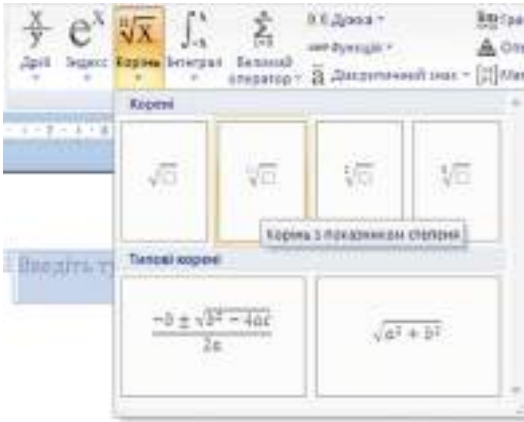


Мал. 3.37. Відкритий список символів

Переглянувши списки і знайшовши потрібний символ, його слід вибрати, і він буде вставлений у поточне місце виразу.

Крім окремих символів, у виразі можуть траплятися звичайні дроби, функції, дужки, знаки коренів тощо. Вони вставляються з використанням елементів керування групи **Структури** вкладки **Рівняння**.

Усі структури розділені за типами, список шаблонів яких відкривається після вибору відповідних кнопок у групі **Структури** (мал. 3.38). Щоб уставити шаблон у формулу, слід відкрити список шаблонів відповідної кнопки і вибрати потрібний.



Мал. 3.38. Група Структури і список шаблонів кнопки Корені



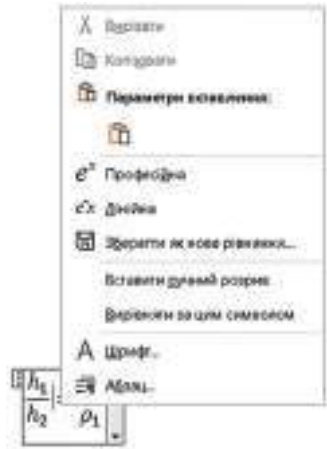
Мал. 3.39. Шаблон дробу

Далі слід заповнити шаблон даними. Місця для введення чисел і символів у шаблоні позначені спеціальними полями, які обведено пунктирною рамкою (мал. 3.39). У поля можна вводити не тільки окремі символи, а й вставляти інші шаблони.

Переміщення в межах області введення формули здійснюється клавішами керування курсором або вказівником. Для виходу з режиму створення формул потрібно вибрати точку за межами області введення формули.

Редагування та форматування створеної формули або її фрагментів здійснюються стандартними засобами **Word**. Для цього використовують елементи керування мініпанелі, вкладки **Основна** та команди контекстного меню формули (мал. 3.40).

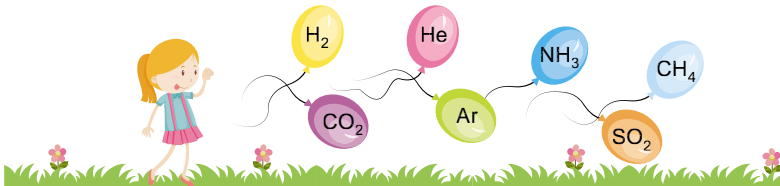
Створену формулу користувач може додати до бібліотеки вбудованих формул і використовувати в подальшому за потреби.



Мал. 3.40. Контекстне меню формули


Створення написів у текстовому документі

Для оздоблення тексту, створення малюнків, підписів і пояснень на малюнках та різних схемах у текстовому документі використовують такі об'єкти, як **написи** (мал. 3.41).



Мал. 3.41. Написи на малюнку

Для розміщення написів у текстовому документі існує кілька способів: текст можна розмістити всередині фігури, яку створено в текстовому документі засобами текстового процесора **Word**, або створити спеціальний об'єкт **WordArt**.

Якщо в текстовому документі потрібно створити текстовий напис, то для цього слід скористатися спеціальним інструментом **Тестове поле** , який розміщено на вкладці **Вставлення** в групі **Ілюстрації** в списку кнопки **Фігури**, або використати інші фігури (прямокутник, стрілка, ромб тощо).

У внутрішній області більшості фігур можна розмістити текст, який буде пояснювати зміст даної фігури на малюнку чи схемі. Для додавання тексту до фігури слід у контекстному меню фігури вибрати команду **Додати текст** і ввести його. За замовчуванням, якщо повернути або дзеркально відобразити фігуру, розміщення тексту в середині фігури буде також змінюватися (мал. 3.42).



Мал. 3.42. Розміщення тексту під час змінення фігури



Мал. 3.43. Напрямки розміщення тексту у фігурі

Бажаємо успіху!


Мал. 3.44. Приклад об'єкта **WordArt**

(написаного текстом) чи з іншою фігурою виконуються в такий самий спосіб, що і з іншими графічними об'єктами.

У текстовому процесорі **Word** можна вставляти в документи так звані *об'єкти WordArt* (мал. 3.44), які використовуються для художнього та яскравого оформлення тексту. Ці об'єкти можна застосовувати для заголовків текстів, для оформлення вітальних листівок, плакатів, малюнків і схем.

Об'єкти **WordArt** мають два види властивостей: *текстові* (шрифт, розмір символів, колір, накреслення) і *графічні* (розмір зображення, розміщення, заливка, колір ліній, наявність тіні та ефектів об'ємних тіл тощо).

Створення об'єкта **WordArt** здійснюється за таким алгоритмом:

1. Вибрати на вкладці **Вставлення** в групі **Текст** кнопку **Додавання об'єкта WordArt** .
2. Вибрати в наведеному списку потрібний стиль оформлення об'єкта.
3. Увести потрібний текст об'єкта **WordArt** у текстове поле.
4. Вибрати точку за межами поля.

Створений об'єкт **WordArt** буде розміщено в поточному місці документа. Редагування та форматування тексту і графічного об'єкта здійснюються з використанням елементів керування тимчасової вкладки **Формат фігури**.







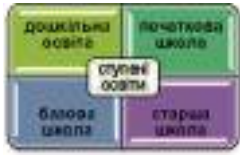
Створення об'єктів **SmartArt** у текстовому документі

Об'єкти **SmartArt** (англ. *smart* – розумний, *art* – мистецтво) – це графічні об'єкти **Microsoft Office**, які призначені для подання в текстовому документі структурованих даних у вигляді різноманітних схем. Використання їх робить документ більш виразним і наочним.


У текстовому процесорі **Word** є готова колекція макетів об'єктів **SmartArt**, які згруповані в кілька категорій (табл. 3.7).

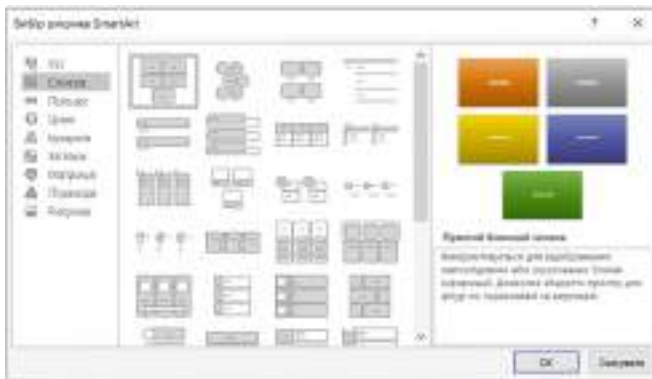
Таблиця 3.7

Приклади об'єктів **SmartArt** різних категорій

			
Список відображає непослідовні дані	Цикл відображає етапи безперервного циклу	Зв'язок відображає відношення між об'єктами	Піраміда відображає відношення пропорційності, взаємозв'язку або підлеглості між об'єктами
			
Процес відображає етапи певного процесу, послідовності дій	Структура відображає ієрархічні зв'язки між об'єктами	Матриця відображає об'єкти як частини єдиного цілого	

Для створення об'єкта **SmartArt** слід:

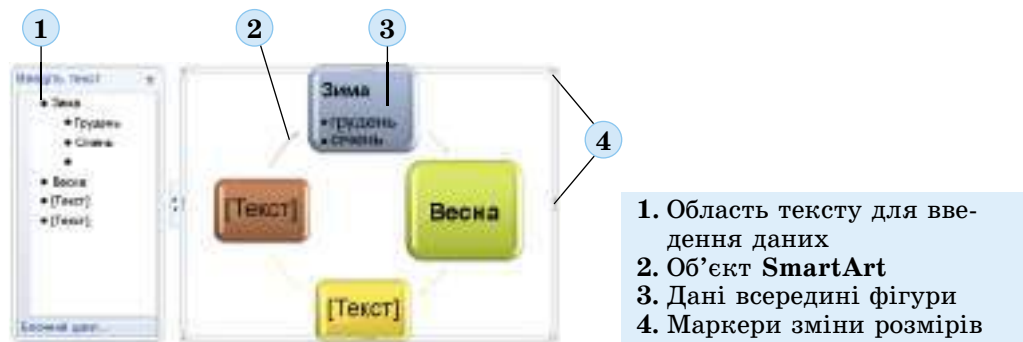
1. Вибрати місце в документі, куди буде вставлятися об'єкт.
2. Виконати **Вставка** ⇒ **Ілюстрації** ⇒ **Додавання рисунка SmartArt** , що відкриває вікно колекції макетів **Вибір рисунка SmartArt** (мал. 3.45).
3. Вибрати в списку зліва вікна **Вибір рисунка SmartArt** потрібну категорію макета.
4. Вибрати в центральній частині вікна **Вибір рисунка SmartArt** відповідний тип макета.
5. Вибрати кнопку **OK**.



Мал. 3.45. Вікно **Вибір рисунка SmartArt**

Вибираючи макет, слід продумати спосіб подання даних з урахуванням призначення макетів, яке описано праворуч у вікні. Також потрібно враховувати розмір тексту та кількість елементів на схемі – їх велика кількість негативно впливатиме на візуальне відображення та сприймання.

Після вставлення в поточному місці документа з'являється макет об'єкта **SmartArt**, обведений рамкою з маркерами зміни розміру (мал. 3.46).



Мал. 3.46. Об'єкт **SmartArt** у документі

Увести потрібний текст у фігуру об'єкта можна безпосередньо в самій фігурі, вибравши її вказівником і ввівши необхідний текст. Також це можна зробити і в області тексту, яка розміщена зліва від уставленого об'єкта, – введені дані автоматично відображаються у відповідній фігурі. Область тексту можна приховати (кнопкою закриття вікна цієї області) та відобразити (вибором зліва на межі об'єкта кнопки < >).





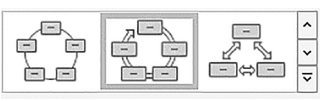






Залежно від макета кожен елемент списку в області тексту представлений на об'єкті **SmartArt** або як окрема фігура, або як елемент списку всередині фігури (мал. 3.45). Додавання чи видалення елемента списку приводить до аналогічних дій над фігурою і навпаки.

Редагування і форматування об'єктів **SmartArt** здійснюються з використанням елементів керування двох тимчасових вкладок **Конструктор SmartArt** і **Формат**, які з'являються на **Стрічці** після вибору об'єкта **SmartArt** (табл. 3.8).

Таблиця 3.8

**Призначення деяких елементів керування тимчасових вкладок
Конструктор SmartArt і Формат**

Елемент керування	Призначення
Група Створення графіки вкладки Конструктор SmartArt	
Додати фігуру	Для додавання до виділеної фігури ще однієї фігури того самого рівня. У списку кнопки можна вибрати інші варіанти додавання фігури
Додати маркер	Для додавання до виділеної фігури ще однієї фігури нижчого рівня
Справа наліво	Для змінення порядку розміщення фігур – справа наліво чи зліва направо



Елемент керування	Призначення
Структура 	Для змінення розміщення фігур на гілках організаційної діаграми
Підвищити рівень 	Для збільшення рівня виділеної фігури
Знизити рівень 	Для зменшення рівня виділеної фігури
Область тексту 	Для відображення або приховування області тексту
Група Макети вкладки Конструктор SmartArt	
	Для вибору іншого макета об'єкта
Група Стили SmartArt вкладки Конструктор SmartArt	
Змінити кольори 	Для зміни кольорової гами макета
	Для вибору стилю оформлення об'єкта
Група Скинуги вкладки Конструктор SmartArt	
Скинуги параметри графіки 	Для скасування всіх змін у форматі макета, які були зроблені після його вставлення в документ
Група Фігури вкладки Формат	
Змінення фігури 	Для відкриття списку графічних примітивів з метою заміни виділеної фігури об'єкта на іншу
Збільшити 	Для збільшення розмірів вибраної фігури
Зменшити 	Для зменшення розмірів вибраної фігури

Працюємо з комп'ютером




Виконайте тренувальні завдання за посиланням <https://cutt.ly/phde6TZ> або QR-кодом.


Найважливіше в цьому пункті

Для створення в текстовому документі нескладних формул лінійного вигляду використовують різні спеціальні символи, уставлення яких здійснюється у вікні **Символ**. Уведення символів нижнього і верхнього індексів здійснюється з використанням кнопок **Підрядковий знак**  і **Надрядковий знак** , які розміщені на вкладці **Основне** в групі **Шрифт**.


Уведення в текстовий документ складних багаторівневих формул здійснюється спеціальним засобом **Конструктор формул**. Формули в ньому конструюються з окремих символів і структур, використовуючи відповідні шаблони, які потрібно заповнити даними.


Літери, знаки арифметичних операцій та інші символи можна вводити з клавіатури, а спеціальні – вибирати у відповідних списках групи **Символи**. Звичайні дробки, функції, дужки, знаки коренів уставляються за допомогою елементів керування групи **Структури** вкладки **Рівняння**. Усі структури розділено за типами, список шаблонів яких відкривається під час вибору відповідних кнопок на вкладці.

Для розміщення в тексті та на малюнках у текстовому документі підписів, пояснень використовують **написи**. Існує кілька способів розміщення написів у текстовому документі: текст можна розмістити всередині фігури або створити спеціальні графічні об'єкти **Текстове поле** чи **WordArt** .

Для додавання тексту до фігури слід у контекстному меню фігури вибрати команду **Додати текст** і ввести його. Якщо потрібно в текстовому документі створити текстовий напис як окремий графічний об'єкт, то для цього слід скористатися спеціальним інструментом **Текстове поле** ,

який розміщено на вкладці **Вставлення** в групі **Ілюстрації** в списку кнопки **Фігури**.

Текстовий процесор **Word** дає змогу також уставляти в документи так звані *об'єкти WordArt*, які використовуються для художнього оформлення документа. Об'єкти **WordArt** мають два види властивостей: *текстові* (шрифт, розмір символів, колір, накреслення) і *графічні* (розмір зображення, розміщення, заливка, колір ліній, наявність тіні та об'ємних ефектів тощо). Створення об'єктів **WordArt** здійснюється на вкладці **Вставлення** в групі **Текст** кнопкою **Додавання об'єкта WordArt** .

Об'єкти **SmartArt** – це графічні об'єкти **Microsoft Office**, призначені для подання в текстовому документі структурованих даних у вигляді різноманітних схем. У текстовому процесорі **Word** є готова колекція шаблонів об'єктів **SmartArt** кількох типів. Для створення об'єкта **SmartArt** потрібно скористатися кнопкою **Додавання рисунка SmartArt** ,

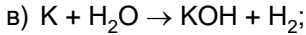
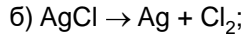
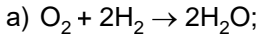


Дайте відповіді на запитання

- 1°. Які набори математичних символів використовують у формулах? Як уставити потрібний символ?
- 2°. Які структури можна вводити у формули? Назвіть кілька з них.
- 3°. Які операції редагування та форматування можна виконувати над об'єктом *формула*? Як вони здійснюються?
- 4°. Як додати текст усередину вставленої фігури? Які операції редагування та форматування можна здійснювати над цим текстом?
- 5°. Для чого використовують об'єкти **WordArt**? Як їх створювати?
- 6°. До якого типу об'єктів належать об'єкти **WordArt**? Які їх властивості?
- 7°. Які види об'єктів **SmartArt** та їх призначення ви знаєте?
- 8°. Які властивості мають об'єкти **SmartArt**? Як їх можна змінювати?
- 9°. Які операції редагування та форматування можна здійснювати в текстовому процесорі **Word** над об'єктами **SmartArt**?
- 10*. Які графічні об'єкти можна створювати в текстовому документі? Чим вони відрізняються? Що в них спільного?

**Виконайте завдання**

- 1*. Створіть новий текстовий документ. Уведіть наведені схеми хімічних реакцій. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.1.docx**.



- 2*. Створіть новий текстовий документ. Уставте в нього наведені фізичні формули. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.2.docx**.

а) Період обертання $T = \frac{t}{N}$;

б) формула сполучних посудин $\frac{h_1}{h_2} = \frac{\rho_2}{\rho_1}$;

в) кінетична енергія $E_k = \frac{mv^2}{2}$;

г) коефіцієнт корисної дії $\eta = \frac{A_1}{A_2} \cdot 100\%$.



- 3*. Створіть новий текстовий документ. Уставте в нього наведені математичні вирази. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.3.docx**.

а) $\frac{3}{4}(12 - x) = \frac{3}{8}$;

б) $\frac{7}{8}(2 - x) = \frac{1}{8}$;

в) $\frac{2}{3}(x - 4) + \frac{1}{3} = 5$;

г) $\frac{3}{7}(1 - 2x) + \frac{1}{7} = -2$.

- 4*. Створіть новий текстовий документ. Уставте в нього наведені математичні функції та рівняння. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.4.docx**.

а) $y = \begin{cases} x^2 + 3, & \text{якщо } x \leq 0, \\ \frac{x}{2} + 3, & \text{якщо } x > 0; \end{cases}$

б) $y = \begin{cases} -3x^3 - 2, & \text{якщо } x < 0, \\ \frac{7x}{4} - 2, & \text{якщо } x \geq 0; \end{cases}$

в) $y = \begin{cases} 3x - y = 1, \\ x + y = 3, \\ 4y = 8; \end{cases}$

г) $y = \begin{cases} x - y = 1, \\ x + y = 3, \\ 2x = 6. \end{cases}$



- 5*. Створіть новий текстовий документ. Уставте в нього наведені математичні вирази та рівняння. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.5.docx**.

а) $\frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a - 4b)^2}{9a^4 - b^4}$;

б) $\frac{x}{7a^2b^3} : \frac{6a^3c^2}{b} = \frac{y}{4c}$;

в) $\frac{a + \frac{25}{a + 10}}{\frac{25}{a} - a}$;

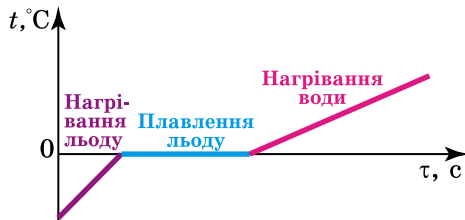
г) $1 - \frac{1}{1 - \frac{1}{1 - \frac{1}{a + 1}}}$.



- 6*. Створіть вітальну листівку до Дня народження вашого друга/подруги, використавши об'єкти **WordArt**, написи та малюнки. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.6.docx**. Обміняйтеся з другом/по-

другою створеними листівками. Який елемент листівки друга/подруги вам сподобався найбільше? Поясніть, як це було зроблено.

- 7*. Створіть у текстовому документі наведений малюнок, використавши створення фігур і написів. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.7.docx**.



- 8*. Створіть у текстовому документі об'єкти **SmartArt** за наведеними зразками. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.8.docx**. Розпитайте в рідних, як би вони використали такі діаграми у своїй роботі.




а)




б)

- 9*. Створіть у текстовому документі об'єкт **SmartArt** для подання алгоритму створення формули з використанням **Конструктора формул**. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.9.docx**.

- 10*. Складіть у текстовому документі схему класифікації чотирикутників, використавши об'єкт **SmartArt** у вигляді організаційної діаграми. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.10.docx**.

- 11*.  Об'єднайтесь у групи по троє. Складіть у текстовому документі радіальну діаграму різних місць розташування козацьких січей в Україні, використавши об'єкт **SmartArt**. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.11.docx**. Як ви розподілили роботу між собою? Чому саме так? Чим ви допомагали один одному?

- 12*.  Об'єднайтесь у пари. Дізнайтесь в Інтернеті, що таке інфографіка та про її призначення. Вам потрібно створити інструкцію до улюбленої вашої гри, візуалізувавши її будь-якими засобами текстового процесора, які ви знаєте. Обговоріть у групі ваші улюблені ігри та варіанти виконання завдання. Оберіть той, який вам обом більше подобається. Виконайте завдання, створивши новий текстовий документ. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.12.docx**. Чому ви обрали саме цю гру? Саме цей спосіб виконання? З якими труднощами ви зіткнулись? Як вам вдалось їх подолати?

Розділ 4.

Створення та публікація вебресурсів

У цьому розділі ви дізнаєтеся про:

- мову гіпертекстової розмітки **HTML**
- використання гіпертекстових, графічних і мультимедійних елементів на вебсторінках
- правила ергономічного розміщення об'єктів на вебсторінці
- етапи створення вебсайту засобами онлайн-системи конструювання сайтів
- способи створення вебсторінок і вставлення на них об'єктів різних типів
- публікацію сайту

4.1. Структура вебсторінок. Поняття про мову HTML

1. Що таке *вебсторінка*? Що таке *вебсайт*?
2. Які об'єкти можуть бути розміщені на вебсторінках?
3. Що таке *дизайн*? Які ви знаєте принципи дизайну презентації?

Елементи структури вебсторінок

Вебсторінки є інформаційними ресурсами служби **World Wide Web**, і ви вже ознайомилися з тим, як здійснювати пошук і перегляд цих ресурсів. Тепер розглянемо правила та способи їх створення.

У структурі вебсторінки можна виділити такі складові (мал. 4.1):

- **контент** (англ. *content* – зміст) – змістове наповнення вебсторінки, доступне користувачу: тексти, зображення, відео, звукові дані та ін.;
- **елементи навігації** – засоби для переходу до інших вебсторінок або до інших об'єктів на тій самій сторінці;
- **дизайн** – особливості розміщення контенту та елементів навігації, їх форматування, оформлення сторінки.

Контент, що розміщується на вебсторінках, має бути цікавим, корисним і, бажано, унікальним, авторським. У першу чергу від якості матеріалів залежить успішність сайту.

Дизайн вебсторінок, як і дизайн комп'ютерних презентацій, потрібно розробляти, спираючись на закони композиції, колористики та ергономіки.

Розглянемо детальніше правила ергономіки для забезпечення зручного перегляду матеріалів сайту.



1. Елементи навігації

2. Елементи дизайну

3. Контент

Мал. 4.1. Складові вебсторінки

Правила ергономічного розміщення об'єктів на вебсторінках

Нагадаємо, що **ергономіка** (грец. έργος – праця, νόμος – закон) – це наука, яка вивчає особливості виробничої діяльності людини з метою забезпечення ефективності, безпеки та зручності цієї діяльності.

Ергономічний сайт – це сайт, який забезпечує для відвідувача необхідні зручності, не викликає фізичної та психологічної втоми, не впливає на здоров'я та працездатність.

Складовою частиною ергономіки сайту є його зручність і простота у використанні для користувача, без необхідності проходження спеціального навчання. Будь-яка людина повинна мати можливість інтуїтивно пов'язувати дії, які їй потрібні, виконати на вебсторінці, з об'єктами, які вона бачить (мал. 4.2).



Мал. 4.2. Приклад ергономічного сайту



Під час розробки структури вебсторінок слід враховувати такі рекомендації:

- розміщувати назву сайту в заголовку всіх сторінок;
- відображати на кожній сторінці, до якого тематичного розділу вона входить;
- пропонувати кілька посилань для переходу на одну й ту саму сторінку, щоб забезпечити кожному користувачу можливість вибрати найбільш зручне або очевидне;
- створити можливість переходу з будь-якої сторінки до перегляду головної (домашньої) сторінки сайту;
- усувати будь-яку двозначність щодо наслідків дій під час вибору елементів керування на вебсторінці;
- у нижньому колонтитулі (підвалі) зазначати авторські права (копірайт) та контактні дані дизайнера та власника ресурсу.

Слід дбати про ергономічне розміщення об'єктів на вебсторінках, а для цього бажано дотримуватися таких правил:

- розміщувати найважливіші дані на початку сторінки;
- заголовки статей виділяти шрифтом збільшеного розміру;
- розміщувати на сторінці не більше трьох колонок з контентом;
- не допускати появи горизонтальних смуг прокручування на вебсторінках;
- виділяти візуально фрагменти тексту та графічні зображення, що використовуються як гіперпосилання;
- створювати гіперпосилання на сторінках з великим обсягом тексту для переходів усередині вебсторінки.

Під час розробки вебсайту потрібно намагатися надати зручності для користувачів з особливими потребами. Для цього варто звертати увагу на таке:

- розмір шрифту повинен бути достатнім, щоб без масштабування сторінок його змогли прочитати без напруження зору більшість користувачів; для людей з вадами зору бажано надавати можливість змінювати розмір шрифту;
- бажано, щоб зображення мали підписи та щоб на зображеннях не було дрібних деталей; надати можливість збільшувати зображення для перегляду;
- за можливості додавати на вебсторінки елементи, що озвучують важливі відомості на сайті.

Поняття про мову HTML

Вебсторінки створюють як текстові документи та зберігають зазвичай у файлах з розширенням імені **html**. Для розробки вебсторінок використовують **мову розмітки гіпертексту HTML** (англ. *Hyper Text Markup Language* – мова розмітки гіпертексту). Команди, що записані цією мовою, виконуються браузером, який розміщує об'єкти на вебсторінці відповідно до задуму розробника. **HTML-код** сторінки складається з даних двох типів:

- **тексту**, який відобразатиметься на сторінці;
- **тегів** (англ. *tag* – ярлик, ознака) – команд, що визначають **розмітку** тексту: його структуру, формат і розташування фрагментів тексту, забезпечують усталення нетекстових об'єктів на сторінку та ін.

Теги записують усередині кутових дужок **<** та **>**. Наприклад, **<hr>** – це тег, призначений для проведення на сторінці горизонтальної лінії.

Деякі теги потребують уточнення дії команди, тоді в них застосовують **атрибути** (властивості) та вказують значення цих атрибутів. Наприклад, якщо на сторінку потрібно вставити зображення, то значення атрибутів визначатимуть, з якого саме файлу буде взято зображення та з якими розмірами його потрібно відобразити на сторінці.

Атрибути записують у кутових дужках після тегу, відділяючи пропусками. Значення атрибутів записують у лапках після знака =.

Наприклад, на сторінку потрібно вставити зображення з файлу *1.jpg*, який розміщено в одній папці з файлом вебсторінки, та встановити для цього зображення ширину та висоту по 200 пікселів. Для вставлення зображення потрібно використати тег **img** (англ. *image* – зображення) з такими атрибутами:

- **src** (англ. *source* – джерело) – для вказівки імені файлу із зображенням;
- **width** (англ. *width* – ширина), **height** (англ. *height* – висота) – для встановлення ширини та висоти зображення.

Тег з атрибутами матиме вигляд:

```
<img src = "1.jpg" width = "200" height = "200">
```

А для вставлення на вебсторінку зображення з файлу, що зберігається в Інтернеті з URL-адресою http://vsviti.com.ua/wp-content/uploads/2017/02/arheol_01.jpg, шириною 200 пікселів і висотою 100 пікселів, може бути використано тег з атрибутами:

```
<img src = "http://vsviti.com.ua/wp-content/uploads/2017/02/arheol_01.jpg"
width = "200" height = "100">
```

Деякі команди визначають призначення або формат лише частини контенту. Наприклад, у деякому реченні потрібно виділити курсивним накресленням лише одне слово. У такому разі використовують **парні теги: відкриваючий і закриваючий**. Дія команди буде стосуватися лише частини контенту між цими тегами. Відкриваючий тег може містити атрибути та їх значення. У закриваючого тегу немає атрибутів, але перед ним поставлено символ /.

Наприклад, якщо в реченні *HTML – це мова розмітки гіпертексту* потрібно виділити курсивним накресленням лише слово *HTML*, то цю команду буде записано так:

```
<i>HTML</i> – це мова розмітки гіпертексту
```

Тут **<i>** – відкриваючий тег, що визначає накреслення символів курсивом, а **</i>** – відповідний йому закриваючий тег.

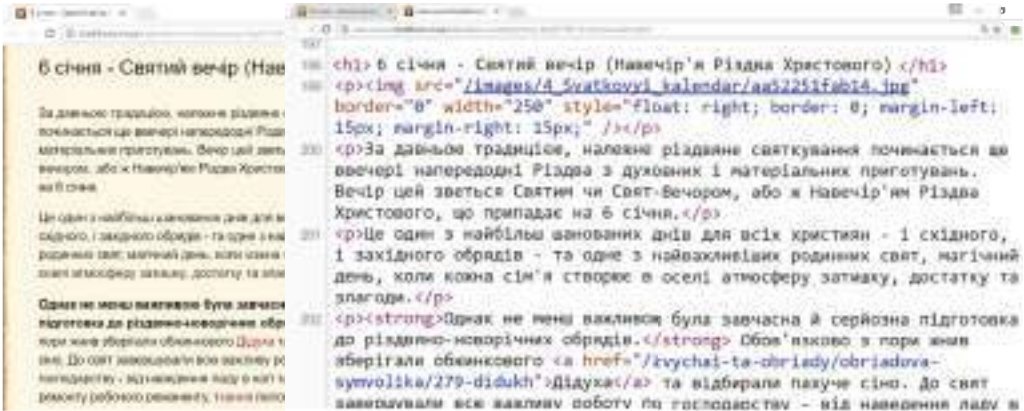
А якщо слова *Українські традиції* потрібно записати символами червоного кольору збільшеного розміру, то може бути використано парний тег **** (англ. *font* – шрифт) з атрибутами **color** і **size** (амер. *color* – колір, англ. *size* – розмір):

```
<font color ="red" size ="5"> Українські традиції </font>
```

Переглянути HTML-код вебсторінки, відкритої у вікні браузера, можна, вибравши в контекстному меню сторінки команду **Переглянути джерело сторінки, Програмний код сторінки** або подібну чи натиснувши сполучення клавіш **Ctrl+U**. На малюнку 4.3 наведено зображення частини вебсторінки та відповідний фрагмент HTML-коду. У коді сторінки теги, їх атрибути та значення атрибутів зазвичай виділяються кольором.

У фрагменті коду, наведеному на малюнку 4.3, присутні теги, що визначають структуру сторінки:

- **<h1> ... </h1>** – тег створення заголовка на вебсторінці. У створеному в прикладі заголовку виводиться текст: *6 січня – Святий вечір (Навечір'я Різдва Христового)*;
- **** – тег вставлення на вебсторінку зображення та встановлення значень його властивостей. У наведеному фрагменті на сторінку вставлено зображення з файлу */images/4_Svatkovyi_kalendar/aa52251fab14.jpg*;
- **<p> ... </p>** – тег створення абзацу на вебсторінці. У створеному абзаці розміщено текст: *За давньою традицією, належне різдвяне святкування починається ще ввечері напередодні Різдва з духовних і матеріальних приготувань. Вечір цей зветься Святим чи Святвечором, або ж Навечір'ям Різдва Христового, що припадає на 6 січня.* Також на сторінці є й інші абзаци;



Мал. 4.3. Фрагмент вебсторінки та відповідний фрагмент HTML-коду

- ** ... ** – теґ для виділення фрагмента тексту напівжирним накресленням. На сторінці в прикладі виділено текст: *Однак не менш важливою була завчасна й серйозна підготовка до різдвяно-новорічних обрядів;*
- **<a> ... ** – теґ створення гіперпосилання. У наведеному фрагменті гіперпосилання пов'язано зі словом *Дідуха* та забезпечує перехід до вебсторінки того самого сайту з адресою [/zvychai-ta-obriady/obriadova-symvolika/279-didukh](https://zvychai-ta-obriady/obriadova-symvolika/279-didukh).



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/chdrqBa> або QR-кодом.



Найважливіше в цьому пункті

У структурі вебсторінки виділяють **контент**, **елементи навігації** та **дизайн**.

Ергономічний сайт – це сайт, що забезпечує необхідні зручності для відвідувача, не викликає фізичної та психологічної втоми, не впливає на здоров'я та працездатність. Слід дбати про ергономічне розміщення об'єктів на вебсторінках.

Вебсторінки створюють як текстові документи та зберігають зазвичай у файлах з розширенням імені **html**. Для розробки вебсторінки використовують **мову розмітки гіпертексту HTML**. **HTML-код** сторінки складається з даних двох типів: **тексту**, який відобразатиметься на сторінці, та **теґів** – команд, що визначають розмітку тексту.

Переглянути HTML-код вебсторінки, відкритої у вікні браузера, можна, вибравши в контекстному меню сторінки команду **Переглянути джерело сторінки**, **Програмний код сторінки** або подібну чи натиснувши сполучення клавіш **Ctrl+U**.



Дайте відповіді на запитання

1. Які складові можна виділити на вебсторінці? Охарактеризуйте кожену групу.
2. Який сайт вважається ергономічним?
3. Які правила потрібно враховувати для забезпечення ергономічності сайту?
4. Яким чином під час створення сайту можна створити зручності для користувачів з особливими потребами?
5. Що таке HTML-код сторінки? Дані яких видів він містить?
6. Як записують теґи; теґи з атрибутами; парні теґи?

- 7°. Який тег призначено для вставлення зображення з файлу на вебсторінку? Для створення гіперпосилання?
- 8°. Як можна переглянути HTML-код сторінки?



Виконайте завдання



- 1°. Перегляньте сторінку сайту **Національної бібліотеки України для дітей** (chl.kiev.ua). Визначте та запишіть у зошит, скільки блоків навігації розміщено на сторінці, якого типу контент, які використано елементи дизайну.
- 2°. Перегляньте сайт **Зелена хвиля** (ecoclubua.com). Запишіть у зошит, які елементи сайту відповідають вимогам ергономіки, а які – ні.
- 3°. Перегляньте сайт **7 чудес України** (7chudes.in.ua). Запишіть у зошит, які елементи сайту відповідають вимогам ергономіки, а які – ні.
- 4*. Перегляньте HTML-код вебсторінки з адресою kievoi.ippo.kubg.edu.ua. Знайдіть у коді теги для вставлення зображень, гіперпосилань і горизонтальних ліній. Зробіть висновок про використання великих і малих літер в HTML-коді. Зробіть припущення про:
 - а) призначення атрибуту **color** тегу `<hr>`;
 - б) можливі значення атрибуту **href** тегу `<a>`.
- 5°. Відкрийте файл **Розділ 4\Пункт 4.1\завдання 4.1.5.txt**. Збережіть файл у вашій папці з розширенням імені **html** і кодуванням **UTF-8**. Упишіть у HTML-код текст і теги для створення на сторінці заголовка першого рівня з назвою вашого населеного пункту, горизонтальної лінії та посилання на сайт вашого населеного пункту в Інтернеті. Збережіть змінений файл і перегляньте його вміст у вікні браузера. Перевірте правильність гіперпосилання.
- 6*. Відкрийте файл **Розділ 4\Пункт 4.1\завдання 4.1.6.txt**. Збережіть файл у вашій папці з розширенням імені **html** і кодуванням **UTF-8**. Перегляньте вміст збереженого файлу у вікні браузера. Змініть HTML-код так, щоб:
 - а) у заголовку браузера та на відповідній кнопці **Панелі завдань** відобразилося ваше прізвище;
 - б) у заголовку першого рівня відобразилося ваше прізвище та ім'я;
 - в) у заголовку другого рівня відобразилася назва вашого закладу освіти;
 - г) горизонтальна лінія стала червоного кольору;
 - д) розмір зображення став 200 × 200 точок;
 - е) гіперпосилання вказувало на головну сторінку **Вікіпедії**.
 Збережіть змінений файл і перегляньте його вміст у вікні браузера. Перевірте правильність гіперпосилання.

4.2. Засоби автоматизованого створення та публікації вебресурсів



1. Які об'єкти можна виділити на вебсторінці?
2. Який сайт вважається ергономічним?
3. Що таке HTML-код сторінки? Дані яких видів він містить?

Засоби автоматизованого створення вебсайтів

Використовуючи мову HTML, можна створити сайт з унікальним дизайном, різноманітними сервісами та можливостями, реалізувати всі творчі задуми розробника.



Для створення простішого сайту можна використати засоби автоматизованого створення та публікації вебресурсів.

Існують **спеціалізовані вебредактори** – програми, які призначено для створення вебсторінок сайтів, наприклад **Adobe Dreamweaver**, **Microsoft FrontPage**, **WYSIWYG Web Builder** та ін. Ці програми мають засоби для візуального конструювання вебсторінок. Розробник розміщує на вебсторінках об'єкти – зображення, таблиці, списки та інше, вибираючи їх заготовки з бібліотеки об'єктів. HTML-код кожної вебсторінки генерується на основі отриманого набору об'єктів. Такі програми називають **WYSIWYG-редакторами** (англ. *What You See Is What You Get* – що ви бачите, то ви й отримуєте).

Іншим засобом розробки сайтів є **системи управління вебконтентом WCMS** (англ. *Web Content Management System* – система управління вебконтентом) – програми, які призначено для створення, редагування та керування вмістом вебсайту. Популярними **WCMS** є системи **Joomla**, **Wordpress**, **MediaWiki** та ін. Вони містять набори вже готових шаблонів оформлення вебсторінок і модулів, що роблять сайт динамічним: форумів, стрічок новин, каталогів файлів тощо. Контент сторінок, створених у цих системах, зберігається в базі даних, а HTML-код генерується автоматично на основі шаблону.

Назване програмне забезпечення можна встановити на локальному комп'ютері. Створені в їх середовищі вебсайти потрібно самостійно розміщувати в Інтернеті.

Існують також онлайн-системи конструювання сайтів, які призначено для розробки сайтів безпосередньо на віддаленому сервері в Інтернеті. Такими є сервіси **Google Сайти**, **Weebly**, **uCoz** та ін. Створені в цих сервісах сайти автоматично розміщуються на серверах відповідних сервісів, і розробник не може безпосередньо редагувати HTML-код вебсторінок.

Створення сайту засобами онлайн-системи конструювання сайтів

Створення вебсайту засобами онлайн-системи конструювання сайтів відбувається в кілька кроків:


1. Реєстрація облікового запису на сервері онлайн-системи конструювання сайтів.
2. Вибір назви сайту та шаблону його оформлення.
3. Створення сторінок сайту та системи навігації.
4. Заповнення сторінок контентом.
5. Публікація сайту.

Розглянемо, як відбувається процес розробки вебсайту засобами, що безкоштовно надає користувачам сервіс **Google Сайти**.

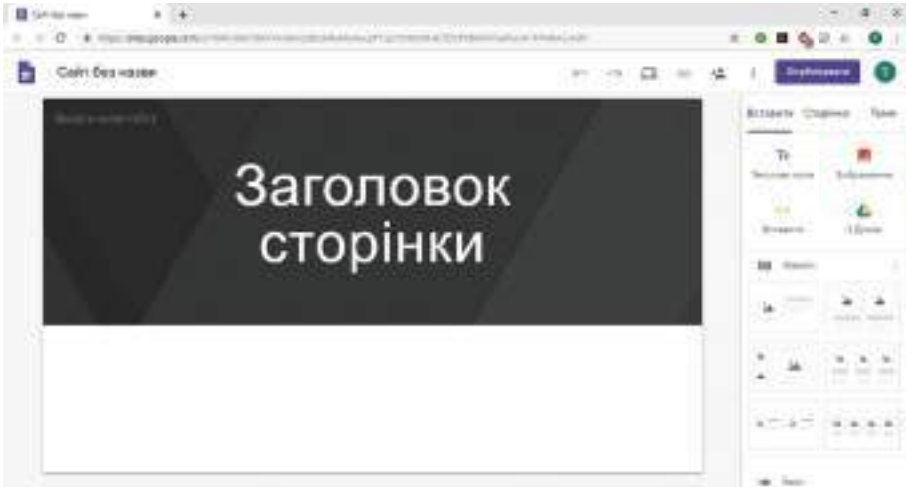
Для використання цього сервісу потрібно мати обліковий запис **Google**, який ви вже створювали, працюючи з електронною поштою та спільними документами **Google**.

Створити сайт можна різними способами:

1. Відкрити вікно **Google Диска**, скориставшись вашим обліковим записом **Google**, та виконати **Створити** ⇒ **Більше** ⇒ **Google Сайти**.

2. Виконати вхід у сервіс **Google**, скориставшись вашим обліковим записом, перейти за адресою sites.google.com і вибрати кнопку **Створити новий сайт**  у нижньому правому куті вікна **Сайти**.


Після виконання цих дій у вікні браузера буде відкрито головну сторінку вашого сайту в режимі редагування (мал. 4.4).



Мал. 4.4. Головна (домашня) сторінка нового сайту в режимі редагування



Мал. 4.5. Ескіз нового сайту

Одночасно у вікні **Google Диска** та у вікні **Сайти** з'явиться ескіз сайту з підписом *Сайт без назви* та значком  (мал. 4.5). Подвійне клацання на цьому ескізі буде відкривати сайт у режимі редагування для подальшої роботи із сайтом.

У правій частині вікна поруч з головною сторінкою сайту розміщено бічну панель з вкладками **Вставити**, **Сторінки**, **Теми**. Інструменти цієї панелі використовуються для додавання, видалення та редагування сторінок, усталення об'єктів на сторінці, вибору та змінення оформлення тощо.

Вибір назви сайту та шаблону оформлення

Кожен сайт має назву, яка відображає його зміст або призначення. Щоб дати назву новому сайту, її потрібно ввести в поле **Введіть назву сайту** у верхній частині заголовка сторінки. Одночасно буде змінено підпис ескізу сайту на **Google Дisku**.

Як і для документів **Google**, усі зміни, виконані під час редагування сайту, будуть зберігатися автоматично.

Для вибору шаблону оформлення сайту слід:

1. Вибрати заголовок вкладки **Теми** в правій частині вікна.
2. Вибрати одну із запропонованих тем оформлення (мал. 4.6).
3. Вибрати кольорову гаму та стиль шрифту із числа тих, які запропоновано для вибраної теми.

Можна також змінити вигляд заголовка на сторінках сайту. Після наведення вказівника на заголовок з'являється панель налаштувань з командами **Змінити зображення** та **Тип верхнього колонтитула**. Вибір першої команди приводить до відкриття списку зображень, які можуть бути використані в заголовку сторінки. За вибору другої команди стає можли-



Мал. 4.6. Темі оформлення сайтів **Google**



вим змінення типу заголовка – *Обкладинка, Великий банер, Банер* або *Лише заголовок*.

На сторінках сайту можна розмістити нижній колонтитул – **підвал**. Для цього потрібно підвести вказівник до нижньої межі сторінки та вибрати кнопку **Додати нижній колонтитул**. У поле, що з'явиться, можна ввести дані розробника сайту, рік його створення або оновлення тощо.

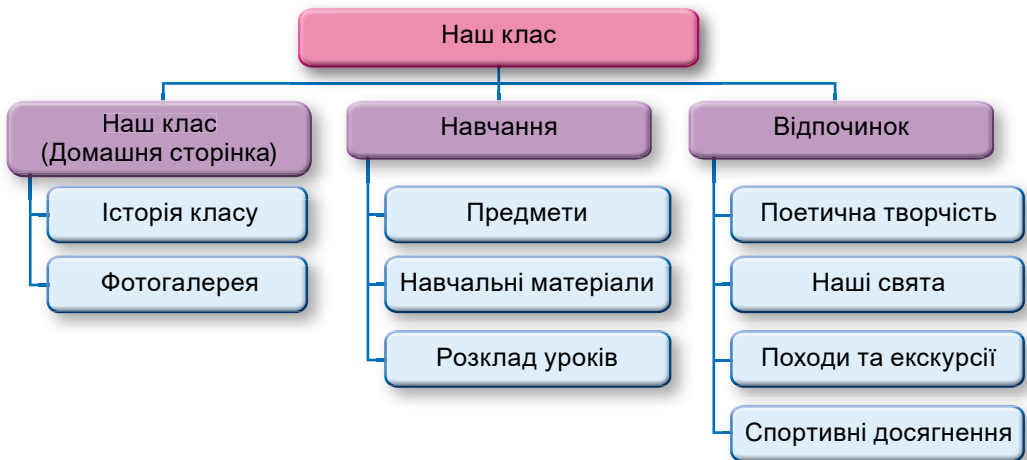
Верхній і нижній колонтитули будуть автоматично повторюватися на всіх сторінках сайту, лише заголовок у верхньому колонтитулі буде таким, як ім'я сторінки.

Створення вебсторінок

Одразу після створення сайт містить лише одну сторінку, яка є **головною (домашньою) сторінкою сайту**. Кожна сторінка сайту має назву. Назву головної сторінки потрібно ввести в поле **Заголовок сторінки**. Часто назва головної сторінки збігається з назвою сайту.

Зазвичай сайт складається з кількох сторінок, з назв яких утворюється так звана **мапа сайту** – схема, що відображає ієрархію сторінок і зв'язки між ними. У мапі сайту на верхньому рівні розміщуються сторінки, які є головними в тематичних розділах, а нижче – сторінки, що входять до цих тематичних розділів. Наприклад, на малюнку 4.7 наведено мапу сайту *Наш клас*.

Ієрархія (грец. *ἱερός* – священний, *ἀρχή* – влада) – розташування частин або елементів цілого в певному порядку від вищого до нижчого.



Мал. 4.7. Мапа сайту *Наш клас*

На наведеній мапі сайту три тематичних розділи – *Наш клас, Навчання, Відпочинок*. До тематичного розділу *Наш клас* входять сторінки *Наш клас* – головна сторінка тематичного розділу, а також *Історія класу* та *Фотогалерея*.

Для створення нової сторінки на сайті слід:

1. Вибрати заголовок вкладки **Сторінки** на бічній панелі в правій частині вікна.
2. Навести вказівник на кнопку і вибрати кнопку **Створити сторінку** .
3. Увести назву сторінки в поле **Ім'я** у вікні **Нова сторінка**.
4. Вибрати посилання **Готово**.



Мал. 4.8. Вкладка **Сторінки**

на **Google Диску**, та ін. Для цього призначено вкладку **Вставити** на бічній панелі.

Усі об'єкти, які додаються на вебсторінку, розміщуються в окремих блоках. Кожен блок можна перемістити, змінити його розміри, використовуючи маркери на межах, або видалити. Для кожного блока, якщо його вибрати, відкривається окрема панель налаштувань.




Призначення окремих елементів керування вкладкою **Вставити** наведено в таблиці 4.1.


Таблиця 4.1

Призначення елементів керування вкладкою **Вставити**


Елемент керування	Призначення
 Текстове поле	Створення блока введення тексту. Текст може бути оформлений як <i>Назва</i> , <i>Заголовок</i> , <i>Підзаголовок</i> , <i>Звичайний текст</i> і <i>Малий</i>
 Зображення	Уставлення зображення з Google Диска , за URL-адресою з Інтернету, з носіїв даних комп'ютера тощо
 Вставити	Уставлення об'єктів різних типів з Інтернету за їх URL-адресою або вставлення фрагментів HTML-коду
 Диска	Уставлення документів різних типів з Google Диска
 Макети	Вибір макета розміщення об'єктів на сторінці
 Зміст	Створення змісту сторінки. Зміст формується автоматично із заголовків, розміщених у текстових полях на сторінці, та призначений для швидкого переходу до вибраного заголовка на поточній сторінці
 Керувати зображеннями	Створення слайд-шоу з вибраних зображень
 Роздільник	Уставлення горизонтальної роздільної лінії




Елемент керування	Призначення
 YouTube	Уставлення відео із сервісу YouTube
 Календар	Уставлення календаря, створеного в сервісі Google Календар
 Карта	Уставлення карти, створеної у сервісі Google Карти

Для видалення вставленого об'єкта потрібно вибрати кнопку **Видалити**  в панелі налаштувань блока цього об'єкта.

Для попереднього перегляду сторінок створеного сайту можна вибрати кнопку

Попередній вигляд  у верхній частині вікна браузера.

У цьому режимі можна побачити, як виглядатиме сайт у мережі Інтернет під час перегляду його на екрані смартфона, планшетного комп'ютера або на великому екрані ноутбука чи стаціонарного комп'ютера. Для цього призначено кнопки на панелі керування попереднім переглядом (мал. 4.9). Вийти з режиму попереднього перегляду можна вибором кнопки .



Мал. 4.9. Панель керування попереднім переглядом сайту

Публікація сайту

Створений вами сайт буде залишатися недоступним для користувачів Інтернету, поки ви його не опублікуєте. Для публікації слід:

1. Вибрати кнопку **Опублікувати** у верхній частині вікна браузера.
2. Увести, використовуючи малі літери латиниці, цифри та тире, останню частину URL-адреси сайту. Перша частина адреси (доменне ім'я сервера та шлях до файлу) для всіх сайтів буде однаковою: sites.google.com/view/. Остання частина повинна бути унікальною для кожного сайту. Якщо введена вами адреса не є унікальною, то про це буде повідомлено і її потрібно змінити.
3. Вибрати кнопку **Опублікувати**.

Опублікований сайт можна переглянути, вибравши команду **Переглянути опублікований сайт** у списку кнопки **Опублікувати**. Адреса, яку потрібно повідомити користувачам Інтернету для перегляду вашого сайту, буде міститися в рядку адреси у вікні перегляду опублікованого сайту.

Після внесення будь-яких змін до вашого сайту його потрібно повторно опублікувати, щоб користувачі Інтернету побачили оновлений сайт. Уведення URL-адреси сайту буде вимагатися лише під час першої публікації сайту.

За потреби можна скасувати публікацію, вибравши в списку кнопки **Опублікувати** команду **Скасувати публікацію**. Сайт буде залишатися недоступним користувачам Інтернету до повторної його публікації.

 **Працюємо з комп'ютером**



Виконайте тренувальні завдання за посиланням <https://cutt.ly/yhdrw8m> або QR-кодом.

**Найважливіше в цьому пункті**

Створення сайту може виконуватися з використанням автоматизованих засобів: спеціалізованих вебредакторів, систем управління вебконтентом **WCMS**, онлайн-систем конструювання сайтів.

Створення вебсайту засобами онлайн-систем конструювання сайтів відбувається в кілька кроків: реєстрація облікового запису на сервері онлайн-системи конструювання сайтів, вибір назви сайту та шаблону його оформлення, створення сторінок сайту та системи навігації, заповнення сторінок контентом, публікація сайту.

**Дайте відповіді на запитання**

- 1°. Які існують засоби автоматизованого створення сайтів? Охарактеризуйте їх.
- 2°. Які етапи створення вебсайтів засобами сервісу **Google Сайти**? Поясніть їх.
- 3°. Як створити сайт і дати йому назву засобами сервісу **Google Сайти**?
- 4°. Як змінити верхній і нижній колонтитули на сторінці?
- 5°. Як вибрати тему оформлення для сайту? Значення яких властивостей можуть бути вибрані разом з темою оформлення?
- 6°. Як створити нову сторінку на сайті? Як перемістити сторінку до тематичного розділу?
- 7°. Які об'єкти можуть бути вставлені на сторінки сайту, створеного засобами сервісу **Google Сайти**?
- 8°. Для чого здійснюється публікація сайтів? Як опублікувати сайт?
- 9°. Яка структура адреси сайтів, створених засобами сервісу **Google Сайти**?
- 10*. Які, на ваш погляд, переваги та недоліки створення сайтів шляхом написання HTML-коду їх сторінок і шляхом використання онлайн-систем конструювання сайтів?

**Виконайте завдання**

- 1°. Доберіть назви сторінок і створіть у текстовому процесорі мапу сайту з теми *Небезпечні тварини*. Збережіть її у вашій папці у файлі з іменем **завдання 4.2.1.docx**.
- 2°. Доберіть назви сторінок і створіть у текстовому процесорі мапу сайту з теми *Колекція рекордів*. Збережіть її у вашій папці у файлі з іменем **завдання 4.2.2.docx**.
- 3°. Створіть сайт з використанням сервісу **Google Сайти** відповідно до мапи сайту, наведеної на малюнку 4.7. Надайте сайту назву *Наш клас*. Виберіть тему оформлення *Дипломат*. Опублікуйте сайт без наповнення. Повідомте вчителю/вчительці URL-адресу створеного вами сайту.
- 4°. Створіть сайт *Наші космічні сусіди* з використанням сервісу **Google Сайти**. Мапа сайту, текст і зображення для розміщення на сторінках містяться в папці **Розділ 4\Пункт 4.2\Завдання 4.2.4**. Виберіть для сайту тему оформлення *Вражаюча*. Опублікуйте сайт. Повідомте вчителю/вчительці URL-адресу створеного вами сайту.
- 5*. Створіть сайт про свій улюблений вид спорту з використанням сервісу **Google Сайти**. Розробіть мапу сайту і доберіть матеріали для наповнення сайту. Опублікуйте сайт. Повідомте вчителю/вчительці URL-адресу створеного вами сайту.



6*. Об'єднайтесь у групу з 2–4 осіб. Обговоріть зміст, розробіть мапу, доберіть матеріали та розробіть сайт на одну з тем:

- Користь і шкода комп'ютерів;
- Українські дослідниці космосу;
- Ми – громадяни;
- Збережемо планету разом;
- Школяр і бізнес.

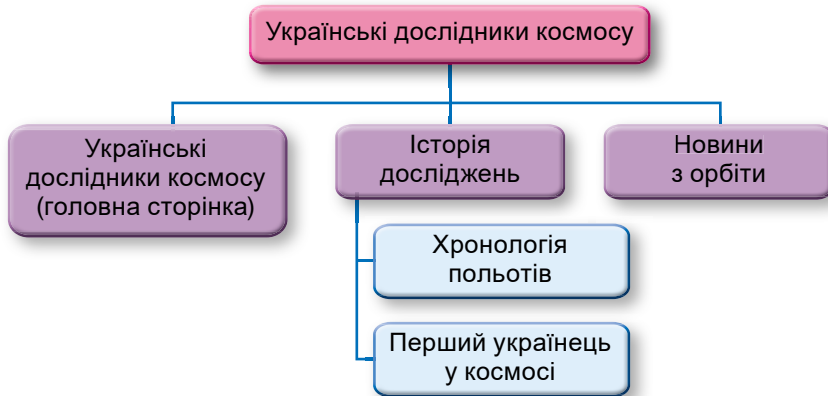
Повідомте вчителю/вчительці URL-адресу сайту, який ви створили.

ПРАКТИЧНА РОБОТА № 5

«Створення сайту з використанням онлайн-системи конструювання сайтів»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

- Створіть сайт *Українські дослідники космосу* з використанням сервісу **Google Сайти**.
- Виберіть тему оформлення *Vision*.
- Створіть і розмістіть вебсторінки на основі такої мапи сайту:



- Заповніть головну сторінку описом матеріалів, які буде розміщено на сторінках сайту.
- Уведіть у нижній колонтитул ваші дані як розробника сайту.
- Заповніть сторінку *Історія досліджень* текстовими матеріалами, наприклад скопійованими з файлу **Розділ 4Практична 5космічні дослідження.docx**.
- Розмістіть на сторінці *Перший українець у космосі* текст з файлу **Павло Попович.docx** і фотографію, що містяться в папці **Розділ 4Практична 5Попович**.
- Завантажте на ваш **Google Диск** файл **Розділ 4Практична 5пльоти.docx**. Уставте файл з диска на сторінку *Хронологія польотів*.
- Розмістіть на сторінці *Новини з орбіти* відео із сервісу **YouTube**, яке знайдете за пошуковим запитом *Mars*.
- Опублікуйте сайт з адресою *kosmos-прізвище*, де *прізвище* – ваше прізвище, записане англійською.
- Перегляньте опублікований сайт.
- Надішліть учителю/вчительці інформатики електронного листа з повідомленням URL-адреси створеного вами сайту.

Розділ 5.

Опрацювання об'єктів мультимедіа

У цьому розділі ви дізнаєтеся про:

- поняття мультимедіа
- принципи кодування аудіо- та відеоданих
- формати аудіо- та відеофайлів
- програмне забезпечення для опрацювання об'єктів мультимедіа
- засоби перетворення аудіо- та відеоформатів
- засоби запису (захоплення) аудіо та відео
- створення аудіо- та відеофрагментів
- сервіси розміщення аудіо- та відеофайлів в Інтернеті
- використання контенту з інтернет-джерел з урахуванням авторських прав

5.1. Опрацювання мультимедійних об'єктів



1. Який спосіб подання повідомлень називають *комбінованим*?

2. Які пристрої для роботи з даними ви знаєте?

3. Які пристрої можуть зберігати й відтворювати для користувача зображення, музику, аудіокнижки, відеофільми?

Поняття мультимедіа

З вивченого раніше ви вже знаєте, що повідомлення можна подавати різними способами: текстом, числами, графікою, звуком, відео, умовними сигналами, спеціальними символами та комбінованим способом – шляхом поєднання кількох різних способів.

Повідомленнями, які подано комбінованим способом, є:

- стаття в часописі зі світлинами, що ілюструють архітектуру міста;
- відеокліп на популярну пісню;
- навчальний відеофільм про **події в історії України**;
- презентація про екскурсію учнів класу до Канева з відповідними світлинами та описом пам'ятних місць;
- відеозапис вашої прогулянки парком тощо.

Повідомлення, подані комбінованим способом, ще називають *мультимедійними*. **Мультимедіа** (лат. *multum* – багато, *medium* – засоби, способи, загальнодоступне) – це поєднання різних способів подання повідомлень. У сприйнятті таких повідомлень задіяні різні органи чуття, а **текст, графічні зображення,**

аудіо та **відео** є об'єктами мультимедійних повідомлень, або **об'єктами мультимедіа**.

Слід зважати на те, що доволі часто, коли говорять про мультимедіа, мають на увазі аудіо- та відеооб'єкти.

Мультимедіа використовуються в різних галузях людської діяльності. Особливо активно вони використовуються в комп'ютерних програмах, зокрема в тих, які призначені для навчання. Ці програми ще називають *електронними підручниками або посібниками* (мал. 5.1). Вони містять, крім текстових даних, ще аудіо- та відеоматеріали, світлини, схеми, графіки, анімацію, репродукції картин, просторові моделі реальних чи уявних об'єктів тощо та надають користувачу можливість навчатися самостійно.



Мал. 5.1. Фрагмент вікна електронного підручника «Мистецтво, 1 клас»



Мал. 5.2. Вікно комп'ютерної гри «Новий рік»

Мультимедіа широко використовуються під час створення кіно, відеофільмів і мультиплікації. Без мультимедійних об'єктів не можна собі уявити сучасні комп'ютерні ігри (мал. 5.2).

Принципи оцифрування звуку та відео

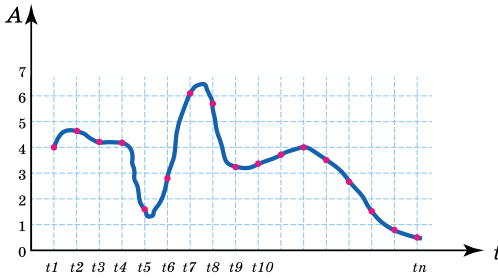
Як ви вже знаєте з першого розділу цього підручника, усі дані, з якими працюють пристрої комп'ютера, повинні бути певним чином закодовані. Кодування здійснюється не тільки для різноманітних символів і графічних зображень, але й для звукових і відеоданих.

Звук – це хвилі, що розповсюджуються в різних середовищах (газі, рідині, твердих тілах) (мал. 5.3). Органи чуття людини, як вам відомо з курсу біології, налаштовані на сприйняття звуку з повітря, хоча за певних умов можуть сприймати звук і у воді або під час контакту з твердими тілами. Перетворення звукового сигналу в дані, що можуть бути опрацьовані пристроями комп'ютера, відбувається у два етапи:

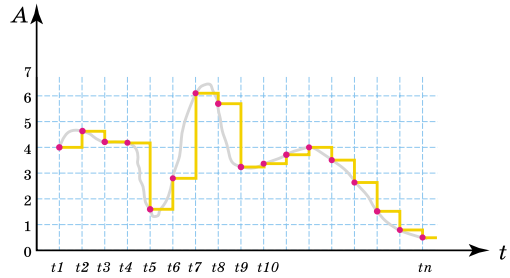
Аналоговий (грец. *análogos* – співрозмірний, відповідний) **сигнал** – сигнал, неперервний на всьому інтервалі часу його відтворення.

Дискретний (лат. *diskretus* – перервний, той, що складається з окремих частин) **сигнал** – сигнал, що змінюється із часом на кратні значення певної величини.

- перетворення звуку в неперервний (аналоговий) електричний сигнал – здійснюється з використанням мікрофонів, звукознімачів електронних інструментів тощо;
- перетворення (оцифрування) отриманого аналогового електричного сигналу в дискретний – здійснюється шляхом вимірювання через певні інтервали часу значень основних властивостей аналогового сигналу та запису цих



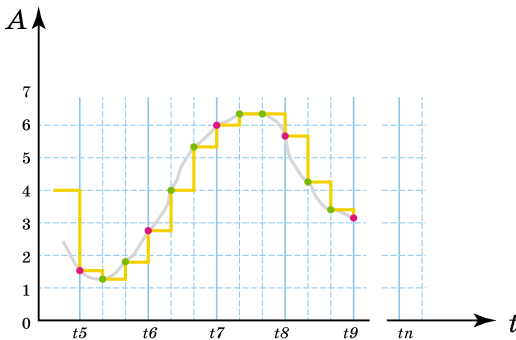
Мал. 5.3. Графік залежності амплітуди аналогового сигналу від часу



Мал. 5.4. Графік залежності амплітуди оцифрованого звукового сигналу від часу (помаранчева лінія)

значень у вигляді послідовності чисел. Основним пристроєм для таких перетворень є звукова карта комп'ютера. Спрощений графік такого дискретного сигналу подано на малюнку 5.4.

Помітно, що наведений графік є ламаною лінією. Точки зламу – це точки, у яких відбувається вимірювання значення властивості сигналу, і це значення на графіку залишається незмінним до наступного вимірювання. При незначних змінах значення властивості графік дискретного сигналу близький за формою до аналогового (ділянки графіка, що відповідають часу від t_1 до t_4 та від t_{10} до t_n). Кажуть, що оцифрування проведено на цих ділянках без значних спотворень. На деяких інших ділянках (наприклад, від t_6 до t_9) форма лінії графіка дискретного сигналу суттєво відрізняється від аналогового. Відбувається спотворення початкового сигналу.



Мал. 5.5. Графік залежності амплітуди оцифрованого звукового сигналу від часу при збільшенні кількості вимірювань

Щоб уникнути спотворень і забезпечити високу якість оцифрованого звуку, збільшують кількість вимірювань, тобто зменшують інтервали часу між вимірюваннями. На малюнку 5.5 подано графік дискретного сигналу на ділянці від t_6 до t_9 при збільшенні втричі кількості вимірювань. Як бачимо, графік дискретного сигналу став більш наближений до графіка аналогового сигналу. Негативним наслідком такого підходу до покращення якості оцифрування звуку є збільшення розмірів звукових файлів.

Серед основних властивостей оцифрування звукових даних виділяють частоту дискретизації – кількість вимірювань значень властивостей звукового сигналу за одну секунду. Вимірюється вона в герцах (Гц). Значення частоти змінюють залежно від особливостей звуку, що оцифровується. Так, для розмови по телефону достатньою є частота у 8 КГц, а для запису якісних музичних творів – не менше від 44 КГц.

Кодування відеоданих є поєднанням кодування графічних і звукових даних. Оскільки кожний кадр відеофільму є малюнком або світлиною, то використовується кодування графіки, а окремо кодується звукова доріжка за правилами кодування звукових даних, які описано вище.

Формати аудіо- та відеофайлів

Раніше ви вже вивчали типи (формати) файлів. Тип (формат) файлу визначає структуру даних у файлі. Ви ознайомилися з різними форматами текстових і графічних файлів, файлів презентацій і електронних таблиць. Аудіо- та відеодані також зберігаються у файлах певних форматів. Ці типи файлів ще називають *мультимедійними*. Приклади форматів аудіо- та відеофайлів з описом їхніх особливостей подано в таблиці 5.1.

Таблиця 5.1

Типи (формати) мультимедійних файлів

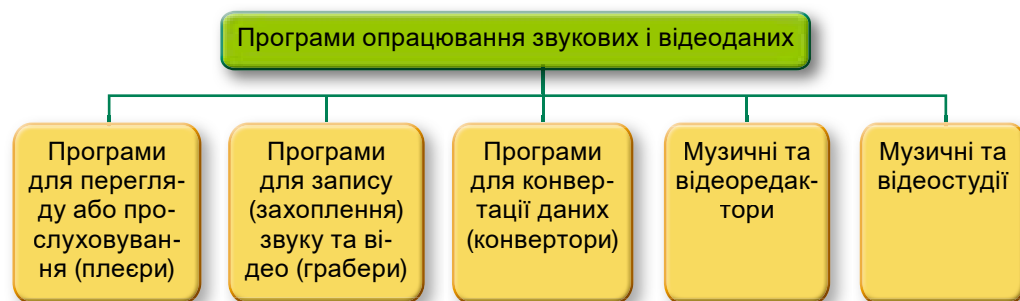
Тип (формат)	Опис	Розширення імені
Формати файлів, у яких не використовується стиснення або використовується стиснення без втрати даних		
<i>Формати аудіофайлів</i>		
WAV (або WAVE) (англ. <i>waveform audio format</i> – аудіо-формат типу хвиля)	Формат звукових файлів, який використовується як основний для нестиснутих аудіоданих в операційних системах сімейства Windows	wav
MIDI (англ. <i>Musical Instrument Digital Interface</i> – цифровий інтерфейс музичних інструментів)	Формат файлів, які містять команди для відтворення звуку спеціальним пристроєм або програмою-синтезатором	mid, midi
FLAC (англ. <i>Free Lossless Audio Codec</i> – вільний аудіо-кодек без втрат)	Формат для стиснення аудіоданих без втрат, який використовується серед авторів і користувачів вільного програмного забезпечення	flac
Формати файлів, у яких використовується стиснення із частковою втратою даних		
<i>Формати аудіофайлів</i>		
MP3 (точніше MPEG-1/2/2.5 Layer 3) (англ. <i>Motion Picture Experts Group</i> – експертна група з питань рухомих зображень, Layer 3 – третій рівень)	Один з найрозповсюдженіших форматів аудіофайлів. Використовується у глобальних мережах, файлообмінних системах	mp3
<i>Формати відеофайлів</i>		
MP4 (або MPEG-4 Part 14)	Файли цього формату зазвичай використовують для цифрового телебачення, зберігання відеофільмів тощо	mp4
WMA/WMV (англ. <i>Windows Media Audio/Video</i>)	Формат файлів для аудіо-/відеоданих, який розроблено корпорацією Microsoft і стандартно використовується в ОС Windows (тут і далі ОС – операційна система)	wma, wmv
QuickTime (англ. <i>Quick Time</i> – швидкий час)	Формат відеофайлів, що базується на технології корпорації Apple	mov

У зв'язку з тим, що у більшості форматів аудіо- та відеофайлів дані для зменшення обсягів стискаються (кодуються) з використанням різноманітних алгоритмів, то для їхнього відтворення потрібно здійснювати декодування. Щоб на різних комп'ютерах можна було відтворювати аудіо- та відеофайли будь-яких форматів, слід мати програми, що містять алгоритми **кодування** і **декодування** мультимедійних даних, які називають **кодеками**. Кодеки входять до складу програм, що опрацьовують аудіо- чи відеофайли, а також можуть випускатися як додатковий набір програм, наприклад **K-Lite Codec**.

Програмне забезпечення для опрацювання об'єктів мультимедіа

Вам вже, мабуть, траплялися певні програми для опрацювання мультимедійних даних – аудіо- та відеопрогравачі, оскільки ви відтворювали на своїх комп'ютерах музичні композиції або відеофільми. З використанням смартфонів або планшетних комп'ютерів, можливо, здійснювали записи відео або звуку. Для цього також використовувалось певне програмне забезпечення. Більш загальну класифікацію за призначенням програм опрацювання аудіо- та відеоданих подано на малюнку 5.6.

Розглянемо детальніше призначення цих програм та алгоритми роботи з ними.



Мал. 5.6. Схема класифікації програм для опрацювання аудіо- та відеоданих



Для тих, хто хоче знати більше

Програвачі аудіо- та відеофайлів

Існує багато різноманітних мультимедійних програвачів. Більшість з них може відтворювати і відео, і звукові дані, окремі спеціалізуються тільки на одному з видів даних, і тоді їх називають або аудіо-, або відео-плеєрами. Розповсюдженими є мультимедійні плеєри **Winamp**, **Media Player Classic**, **QuickTime Player**, **Flv Player** та ін.

Зазвичай у складі операційної системи є один або кілька мультимедійних плеєрів. Так, у **Windows 10** це **Програвач Windows Media**. З їх використанням можна відтворювати аудіо- та відеофайли, створювати власні бібліотеки для впорядкування медіафайлів.

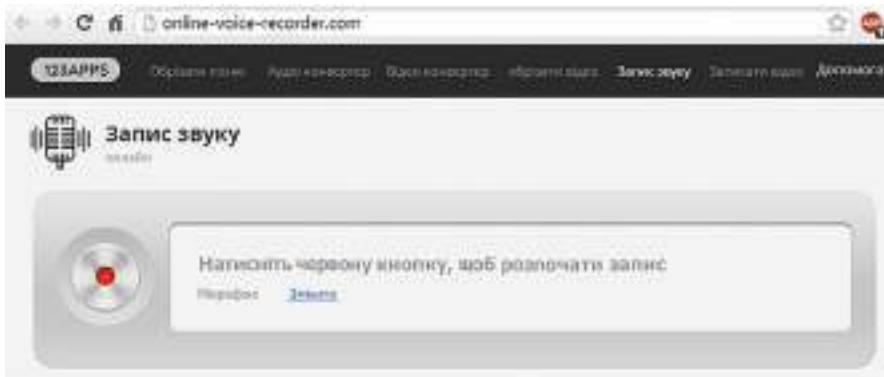
Запис (захоплення) аудіо та відео

Для збереження аудіо- та відеоданих, що можуть бути отримані з використанням пристроїв введення/виведення мультимедійних даних, призначені спеціальні *програми запису (захоплення) звуку та відео*, які ще називають **граберами** (англ. *grabber* – той, хто захоплює, хапуга, грабіжник), або **рекордерами** (англ. *record* – запис). Після опрацювання цими програмами мультимедійні дані будуть збережені у відповідних файлах.





Використовуючи грабери, можна записати звуковий коментар до презентації, відеоконференцію в **Skype** або **Zoom**, вебтрансляцію спортивного змагання тощо.

Створення аудіозапису. Прикладами програм для захоплення звуку є стандартна програма ОС **Windows 10 Диктофон**, а також **Audacity**, **Audiograbber**, **Streamripper**, **AML Easy Audio Recorder**, **Free Sound Recorder** та ін. Можна також використати інтернет-сервіси для запису та збереження звуку, наприклад <http://vocaroo.com>, <http://online-voice-recorder.com> (мал. 5.7).








Мал. 5.7. Сервіс Інтернету 123APPS для запису звуку

Щоб записати звукове повідомлення з використанням мікрофона та програми **Диктофон**, слід:

1. Підключити мікрофон або скористатися вбудованим (у мобільних комп'ютерах).
2. Запустити програму запису звуку, наприклад **Пуск** ⇒ **Диктофон**.
3. Розпочати запис звуку вибором кнопки **Записувати**  (мал. 5.8).
4. Проговорити в мікрофон потрібний текст.
5. Зупинити запис вибором кнопки, наприклад **Зупинити записування** .

Програма автоматично збереже створений запис в аудіофайлі з іменем **Запис (n)** (де *n* – порядковий номер запису) у папці **Звукові записи (Цей ПК ⇒ Документи ⇒ Звукові записи)**. Для прослуховування створеного аудіозапису та його редагування слід вибрати його ім'я у вікні програми. Над створеним записом можна здійснити такі операції, вибравши відповідні кнопки (мал. 5.9):


- прослухати – кнопка **Відтворити**  ;
- додати позначку для швидкого переходу до певного фрагмента запису або для його обтинання – кнопка **Додати маркер**  ;
- поділитися створеним аудіозаписом з іншими користувачами – кнопка **Надіслати**  ;
- обітнути аудіозапис – кнопка **Обітнути**  ;
- змінити назву – кнопка **Перейменувати** .



Мал. 5.8. Вікно програми **Диктофон**



Мал. 5.9. Вікно програми **Диктофон** у режимі прослуховування

Обтинання запису можливо здійснити тільки на початку або в кінці запису. Для цього слід після вибору кнопки **Обітнути**  (мал. 5.10) розмістити лівий чи правий нижній маркер (або обидва разом) у положення, до якого або з якого запис буде обітнено.



Мал. 5.10. Фрагмент вікна програми **Диктофон** у режимі обтинання



Мал. 5.11. Вікно програми **Bandicam**


у відео з використанням програми **Bandicam** (мал. 5.11), яка вільно розповсюджується в Інтернеті (<https://www.bandicam.com/ua/>), слід:

1. Відкрити вікно програми, відео з якої потрібно записати, або відповідний сайт, наприклад сайт, що містить відеозапис уроку фізики.
2. Запустити програму **Bandicam**.
3. Вибрати кнопку, що відповідає потрібному режиму запису, наприклад кнопку режиму запису прямокутної ділянки екрана.
4. Указати, використовуючи спеціальний вказівник, прямокутну область, у якій буде виконано захоплення відео.
5. Увімкнути запис відео вибором кнопки .
6. Розпочати програвання відео в програмі або на сайті-джерелі (мал. 5.12).
7. Для завершення запису відео вибрати кнопку **Стоп** . Відео автоматично збережеться відповідно до параметрів, що встановлені за замовчуванням або які встановив користувач.

Переглянути створений відеозапис можна в цій самій програмі. Для цього слід:

1. Вибрати вкладку **Відео**.
2. Обрати в списку ім'я файлу, що був збережений.
3. Вибрати кнопку **Переглянути**.

Зазначений алгоритм з невеликими змінами може бути використаний під час запису відео з використанням й інших програм.

Для збереження відредагованого запису слід вибрати кнопку **Зберегти**  та обрати один з варіантів збереження – *замінити існуючий запис* або *записати в новий файл*.

У подальшому записи, які ви зберегли, можна відкрити, прослухати та відредагувати в середовищі цієї самої програми, іншого програвача або редактора аудіофайлів.

Створення відеозапису. Під час записування відео може виконуватися запис відео і звуку або тільки відео. Відповідно до комп'ютера слід під'єднати пристрої введення відео- і звукових даних чи тільки відеоданих або використати вбудовані.

Запис (захоплення) відео можна виконувати з вебкамери, з вікна програвача відеофайлів чи іншої програми, з вебсторінки. В усіх випадках для запису слід використати відповідну програму, наприклад **OBS Studio**, **FlashBack Express**, **ApowerREC**, **XSplif Broadcaster**, **VLC Media Player**, **Bandicam** та ін. Для запису



Мал. 5.12. Вікно програми **Bandicam** у режимі запису прямокутної області

Виконуючи записи аудіо- та відеоматеріалів, слід урахувати вимоги законодавства до захисту авторських прав. Значна частина матеріалів, розміщених на інтернет-сервісах, таких як **YouTube**, уже передбачає, що автор/авторка надає доступ до їх перегляду. Однак ці матеріали зазвичай не передбачають розповсюдження в комерційних цілях або розповсюдження без посилання на канал автора/авторки.


Засоби перетворення аудіо- та відеоформатів

Залежно від потреб користувача щодо використання аудіо- та відеоданих часто виникає потреба змінити формат мультимедійного файлу. Наприклад, якщо ваш мультимедійний програвач не може відтворити потрібний файл або в презентацію не можете вставити наявний файл відео. Для таких випадків використовують спеціальні програми – **конвертори**. Для конвертації аудіофайлів використовують такі програми: **HAMSTER Free Audio Converter**, **SoundConverter XRECODE II**, **TAudio Converter**, **AudioConverter Studio** та ін.

Для конвертації аудіофайлів, наприклад у програмі **SoundConverter** (<https://soundconverter.org>) (мал. 5.13), після запуску програми слід:



Мал. 5.13. Вікно програми **SoundConverter** з підготовленим до конвертації файлом

1. Вибрати кнопку **Add File**  (англ. *Add File* – додати файл) і у вікні, що відкрилося, вибрати файл, який потрібно конвертувати.
2. Вибрати кнопку **Параметри** та вказати в списку **Format** формат файлу, у який потрібно конвертувати файл.
3. Почати процес конвертації вибором кнопки **Перетворити**.

Конвертований файл за замовчуванням буде записано в ту саму папку, де знаходився початковий файл. Установити іншу папку для запису можна у вікні, що відкривається після вибору кнопки **Параметри**.



Мал. 5.14. Вікно програми **HAMSTER Free Video Converter** на етапі встановлення властивостей формату

Для конвертації відео існує багато різноманітних програм, що розповсюджуються за різними видами ліцензій. Наприклад, **HAMSTER Free Video Converter**, **AVS Video Converter**, **Any Video Converter Free**, **SUPER** та ін.

Розглянемо послідовність конвертації відеофайлів у програмі **HAMSTER Free Video Converter**. Після запуску програми слід:

1. Додати файли для конвертації, вибравши кнопку **Додати файли** та вибравши у вікні, що відкрилося, потрібні файли.
2. Вибрати кнопку **Далі**.
3. Вибрати формат файлу, у який потрібно конвертувати, та за потреби значення властивостей цього формату (мал. 5.14).
4. Вибрати кнопку **Конвертувати**.
5. Указати папку для запису конвертованого файлу.

Також існують вебверсії конверторів як аудіо, так і відео, наприклад на сайтах <http://audio.online-convert.com>, <https://convert-video-online.com/> та ін.



Для тих, хто працює з Linux

Для встановлення програм опрацювання аудіо- та відеоданих, якщо їх не було встановлено під час інсталяції операційної системи, можна використати **Центр програмного забезпечення Ubuntu**. Для цього у списку **Всі програми** слід установити відображення програм роботи зі звуком і відео, вибрати потрібну програму та кнопку **Встановити** (мал. 5.15).



Мал. 5.15. Вікно **Центра програмного забезпечення Ubuntu**



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/XhdreNg> або QR-кодом.



Найважливіше в цьому пункті

Повідомлення, подане комбінованим способом, називають *мультимедійним*. **Мультимедіа** – це поєднання різних способів подання повідомлень. У сприйнятті таких повідомлень задіяні різні органи чуття, а **текст, графічні зображення, аудіо та відео** є об'єктами мультимедійних повідомлень, або **об'єктами мультимедіа**.

Слід зважати на те, що доволі часто до об'єктів мультимедіа відносять тільки аудіо- та відеооб'єкти.

Перетворення звукового сигналу в дані, що можуть бути опрацьовані пристроями комп'ютера, здійснюється у два етапи:

- перетворення звуку в неперервний (аналоговий) електричний сигнал – здійснюється з використанням мікрофонів, звукознімачів електронних інструментів тощо;
- перетворення (оцифрування) отриманого аналогового електричного сигналу в дискретний – здійснюється шляхом вимірювання через певні інтервали часу значень основних властивостей аналогового сигналу та запису цих значень у вигляді послідовності чисел.

Серед основних властивостей оцифрування звукових даних виділяють **частоту дискретизації** – кількість вимірювань значень звукового сигналу за одну секунду. Вимірюється вона в герцах (Гц).

Кодування відеоданих є поєднанням кодування графічних і звукових даних. Оскільки кожний кадр відеофільму є малюнком або світлиною, то використовується кодування графіки, а окремо кодується звукова доріжка за правилами кодування звукових даних, описаними вище.

Аудіо- та відеодані зберігаються у файлах різних форматів. Файли аудіо мають такі розширення імені: **wav, mid, flac, mp3** та ін. Файли відео мають такі розширення імені: **mp4, wmv, mov, flv** та ін.

Програми, що містять алгоритми кодування і декодування мультимедійних даних, називають **кодеками**.

Програми, що опрацьовують відео та аудіо, залежно від призначення поділяють на програми для перегляду або прослуховування (**плеєри**), програми для запису (захоплення) звуку і відео (**грабери, рекордери**), програми для конвертації даних (**конвертори**), музичні та відеоредактори, музичні та відеостудії.

Виконуючи записи аудіо- та відеоматеріалів, слід ураховувати вимоги законодавства до захисту авторських прав.



Дайте відповіді на запитання

- 1•. Що таке мультимедіа? У яких галузях їх застосовують? Наведіть приклади.
- 2•. Яка послідовність оцифрування звуку? Які пристрої при цьому використовують?
- 3•. Що таке частота дискретизації? У яких одиницях вона вимірюється?
- 4•. Які особливості кодування відеоданих?
- 5◦. Які типи (формати) аудіофайлів ви знаєте?
- 6◦. Які типи (формати) відеофайлів ви знаєте?
- 7◦. Які види програм для опрацювання мультимедійних даних ви знаєте?
- 8◦. Для чого призначені грабери? Наведіть приклади таких програм.
- 9•. З яких джерел можна виконати захоплення відео? Як не порушувати при цьому авторські права?
- 10•. Яка послідовність дій під час конвертації аудіо- та відеофайлів?

**Виконайте завдання**

1*. Знайдіть в Інтернеті відомості про вчених, які сформулювали закон про мінімальні значення частоти дискретизації для якісного оцифрування різного виду звукових повідомлень. Підготуйте із цього питання презентацію.

2°. Запишіть у власному виконанні декламацію вірша Тараса Шевченка «Думи мої, думи...» (Розділ 5\Пункт 5.1\думи мої.doc) з використанням програми **Диктофон**. Збережіть запис у вашій папці у файлі з іменем **завдання 5.1.2.wav**.



3°. Використовуючи інтернет-ресурс **123APPS** для запису звуку <http://online-voice-recorder.com>, запишіть у власному виконанні декламацію вірша Володимира Сосюри «Осінь» («Облітають квіти») (Розділ 5\ Пункт 5.1\облітають квіти.doc). Збережіть запис у папці **Документи** у файлі з іменем **завдання 5.1.3.mp3**.



4°. Запишіть відеопривітання для вашого друга (подруги) з вебкамери вашого комп'ютера з використанням програми **Bandicam** тривалістю 30 с. Збережіть запис у папці **Документи** у файлі з іменем **завдання 5.1.4.mp3**.



5°. Запишіть відео з вікна відеопрогравача під час відтворення файлу **відео 2.avi** (Розділ 5\Пункт 5.1\відео 2.avi) з використанням програми захоплення відео. Збережіть запис у вашій папці у файлі з іменем **завдання 5.1.5.wmv**.

6*. Виконайте конвертування двох перших файлів формату **MP3** з папки, вказаної вчителем/вчителькою (наприклад, **Розділ 5\Пункт 5.1\Аудіо**), у формат **WMA**, використовуючи один з конверторів звукових файлів. Збережіть ці файли в папці **Завдання 5.1.6**, яку створить у вашій папці. Порівняйте розміри початкових та отриманих файлів. Поясніть, за рахунок чого досягнуто різницю в розмірах.

7*. Виконайте конвертування двох перших файлів формату **MOV** з папки **Розділ 5\Пункт 5.1\Відео** у формат **WMV**, використовуючи один з конверторів відеофайлів. Збережіть ці файли в папці **Завдання 5.1.7**, яку створить у вашій папці **Документи**. Порівняйте розміри початкових та отриманих файлів. Поясніть, за рахунок чого досягнуто різницю в розмірах.

5.2. Редагування аудіо- та відеоданих

1. Які формати аудіо- та відеофайлів ви знаєте?

2. Які види програм для роботи з мультимедійними даними ви знаєте?

3. Які операції над мультимедійними даними можна виконати з використанням плеєрів, граберів, конверторів?

Програми для редагування аудіо- та відеоданих

У попередньому пункті серед програм для роботи з мультимедійними даними не було детально розглянуто два види програм – *музичні та відеоредактори*, а також *музичні та відеостудії*. Це програми, призначені для редагування звукових і відеоданих. Використовуючи їх, можна вставляти, видаляти, копіювати, змінювати тривалість аудіо- чи відеофрагментів, об'єднувати фрагменти, уставляти у відео текстові та графічні об'єкти тощо.

Різниця між названими програмами в тому, що мультимедійні студії – це багатифункціональні програми. Вони мають розширений набір функцій інструментів

і використовуються в професійній діяльності на студіях звукозапису, кіно- й анімаційних студіях для створення звукозаписів, аудіо- і відеоконпозицій, анімації, кіно- і відеофільмів, рекламних роликів, відеокліпів тощо. Прикладами програм музичних студій є: **Linux MultiMedia Studio, Psycladelics Psydle, Steinberg Cubase, Cakewalk Sonar, Adobe Audition** та ін.; програмами відеостудій є: **Pinnacle STUDIO, Kino, Adobe Premiere Pro, Avid Media Composer** та ін. Зазвичай програми-студії є комерційними програмами.

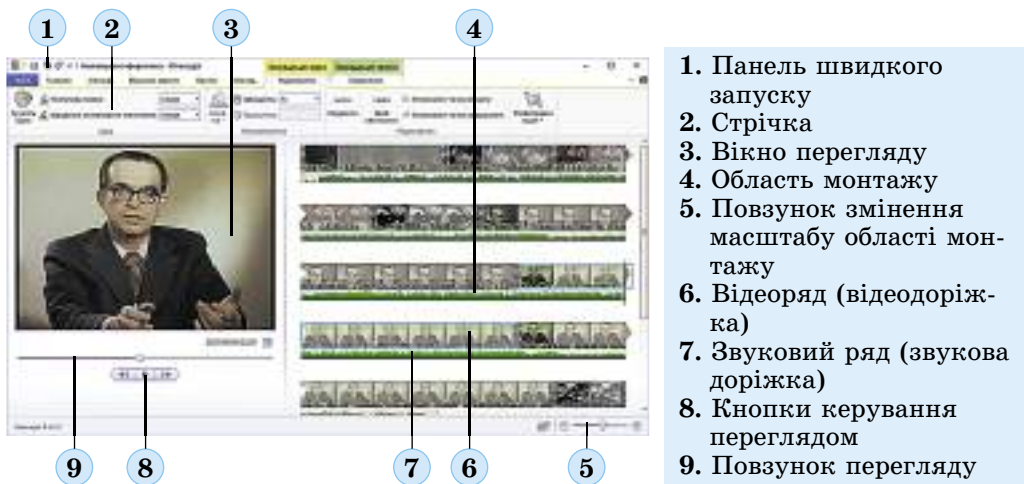
Для створення та редагування простих звукових і відеофайлів використовують різноманітні мультимедійні редактори. Наприклад, для редагування аудіоданих можна використати **Audacity, Free Audio Editor, Wave Editor**, а відеоданих – **Кіностудію Windows, Відеоредактор, Open Shot, Shotcut, Virtual Dub, ZS4 Video Editor** тощо.

Кіностудія Windows

Кіностудія Windows є версією редактора відеоданих **Windows Movie Maker**. Безкоштовну версію програми користувачі **Windows 7** і **Windows 10** можуть завантажити з Інтернету. Для попередніх версій ОС **Windows** можна використати **Windows Movie Maker**, що за замовчуванням включається до складу стандартних програм під час інсталяції.

Кіностудія Windows призначена для створення відеофільмів, що можуть містити відеофрагменти, фото, звуковий супровід (дикторський текст), різноманітні написи (титри). Створене відео можна опублікувати в Інтернеті або зберегти у файлі одного з можливих форматів.

Загальний вигляд вікна програми **Кіностудія Windows** у режимі редагування відео подано на малюнку 5.16.



1. Панель швидкого запуску
2. Стрічка
3. Вікно перегляду
4. Область монтажу
5. Повзунок змінення масштабу області монтажу
6. Відеоряд (відеодоріжка)
7. Звуковий ряд (звукова доріжка)
8. Кнопки керування переглядом
9. Повзунок перегляду

Мал. 5.16. Вікно програми **Кіностудія Windows**



Для тих, хто працює з програмою **Відеоредактор**

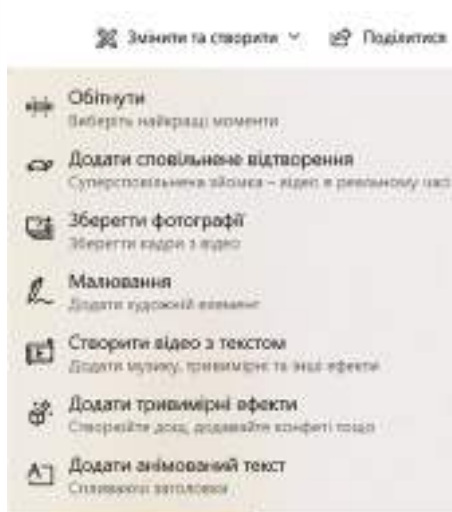
В останніх версіях **Windows 10** до складу програм, що інсталюються з операційною системою за замовчуванням, входить програма **Відеоредактор**, яка інтегрована з іншою стандартною програмою для перегляду графічних зображень і відео – **Фотографії**. Відкрити вікно редактора можна, виконавши **Пуск** ⇒ **Відео-**

редактор. Загальний вигляд вікна програми **Відеоредактор** у режимі редагування відеофільму подано на малюнку 5.17.



1. Назва проєкту
2. Кнопка редагування назви проєкту
3. Бібліотека фото й відео
4. Вікно попереднього перегляду
5. Команди редагування відеофільму
6. Порожній кадр для розміщення фото або відео
7. Команди редагування відеофільму
8. Індикатор часу показу фрагмента
9. Уставлений фрагмент відеофільму

Мал. 5.17. Вікно програми **Відеоредактор**



Мал. 5.18. Кнопка зі списком **Змінити та створити**

Щоб внести окремі зміни в наявні відеофайли, можна в режимі перегляду відео в програмі **Фотографії** вибрати в списку кнопки **Змінити та створити** одну з команд редагування, наприклад команду **Додати анімований текст** (мал. 5.18).

Загальні поняття про відеофільм

Під **відеофільмом**, як і **кінофільмом** (англ. *cine-film* – кіноплівка, кінофільм), здебільшого розуміємо твори кіномистецтва. На сьогодні істотної відмінності між ними немає. Термін «відеофільм» з'явився як відображення широкого розповсюдження відеомагнітофонів і відеокамер, що використовували як носій даних магнітну плівку, на відміну від кінофільмів, для яких використовувалася кіноплівка зі спеціального прозорого матеріалу з нанесеним світлочутливим шаром.

На сьогодні під час створення і професійного кіно, і приватного відео використовують одну і ту саму технологію – запис відеоданих на комп'ютерні носії – жорсткі магнітні диски або електронні мікросхеми. Тому й терміни «кінофільм» і «відеофільм» стали практично синонімами. Хоча найчастіше під відеофільмом розуміють відео, створене непрофесіоналами, або відео, не призначене для перегляду в кінотеатрах.

Для невеликих за розміром кіно- та відеофільмів використовують термін «**кліп**» (англ. *clip* – стискувати, обрізати), або «**відеокліп**».

Відеофільм складається з **кадрів** – окремих фрагментів рухомого (відео) або нерухомого (фото) зображення. Відеофільм зазвичай також містить такі об'єкти:

- **заголовок** – один або кілька текстових об'єктів, що з'являються на початку відеофільму на окремих кадрах;
- **титри** – текстові об'єкти, що з'являються або поверх відео та використовуються для пояснення зображення, або в кінці відео та містять відомості про авторів відеофільму;
- **звуковий супровід** може бути кількох видів:
 - записаний разом з відео;
 - дикторський – коментар або пояснення до відео;
 - музичний супровід – музичний твір, що використовується як фоновий під час перегляду відео;
- **візуальні ефекти** – використовуються для зміни зовнішнього вигляду зображення, наприклад переходу до чорно-білого зображення або досягнення ефекту старого кіно;
- **ефекти переходу** – ефекти, що використовуються для переходу від одного кадру до іншого.

Під час створення відеофільму у відеоредакторі кадри утворюють відеоряд і розміщуються у відеоредакторі на спеціальній **відеодоріжці**. Візуальні ефекти та ефекти переходу відображаються зазвичай на відеодоріжці. Звуковий супровід може розміщуватися на кількох окремих **звукових доріжках**. У **Кіностудії Windows** звук, записаний разом з відео, відображається на відеодоріжці.

Під час створення відеофільмів варто дотримуватися певної послідовності:

- **на першому етапі** потрібно розробити сценарій – детально розроблений план послідовності окремих кадрів фільму і розміщення окремих об'єктів. Що детальніше буде сценарій, то простіше буде створювати відеофільм;
- **на другому етапі** здійснюється підготовка потрібних матеріалів – відбувається зйомка всіх епізодів відповідно до сценарію, добирається музика, фото, малюнки, записується звуковий супровід;
- **на третьому етапі** здійснюється **монтаж** – опрацювання та розміщення у відповідній послідовності окремих кадрів для отримання цілісного твору – відеофільму. У процесі монтажу відзнятий матеріал редагується, уточнюється сценарій, накладаються звук і титри – текстові коментарі до кадрів, тощо.

Розробка сценарію відеофільму

Розглянемо процес розробки сценарію відеофільму на прикладі створення відеофільму «Безпаперова інформатика» про реалізацію ідей В. М. Глушкова відносно використання комп'ютерних засобів у наші дні. Він говорив, що на початок XXI ст. основна маса інформації буде зберігатися в безпаперовому вигляді – у пам'яті ЕОМ.

Для реалізації цих ідей під керівництвом В. М. Глушкова була розроблена серія ЕОМ **МИР** (рос. *Машина для Инженерных Расчётов* – машина для інженерних розрахунків) – прообрази персональних комп'ютерів. Основна ідея їх створення – можливість їх широкого застосування всіма верствами населення.

Ураховуючи мету відеофільму – ознайомлення учнів з ідеями академіка В. М. Глушкова, розробимо сценарій. Сценарій складається зі сценарних кадрів – опису змістових фрагментів відеофільму та тих подій, які в них відбуваються.

1. Кадр 1. Титульна сторінка:

- заголовок – *Безпаперова інформатика*;
- колір тла – *світло-зелений*, колір літер – *темно-зелений*;

- тривалість показу – 7 с;
 - звук – загальний фоновий звук з файлу **Physical.wma**;
 - візуальний ефект – Кінематографічні згасання, зліва;
 - ефект переходу на наступний кадр – розчинення, мозаїка.
2. Кадр 2. **Фотографія В. М. Глушкова:**
- фото з файлу **Glushkov.jpg**;
 - тривалість показу – 6 с;
 - звук – дикторський текст, що починається із цього кадру і закінчується в наступному; тривалість звучання – визначається швидкістю відтворення тексту;
 - дикторський текст: «Директор Інституту кібернетики Академії наук України академік В. М. Глушков однією із задач використання комп'ютерної техніки вважав перехід до безпаперової інформатики. Він говорив, що на початок двадцять першого століття основні об'єми інформації будуть зберігатися в безпаперовому вигляді – у пам'яті ЕОМ. І людина, яка не буде вміти працювати з комп'ютером і використовувати цю інформацію, буде подібна до того, хто на початку ХХ ст. не вмів ні писати, ні читати»;
 - візуальний ефект – відсутній;
 - ефект переходу на наступний кадр – розчинення, мозаїка.
3. Кадр 3. **Відео за участі В. М. Глушкова:**
- відео з файлу **Глушков відео 0.wmv**;
 - тривалість показу – завершується одночасно з дикторським текстом;
 - звук – дикторський текст, що починається з попереднього кадру і закінчується в цьому; тривалість звучання – визначається швидкістю відтворення тексту;
 - візуальний ефект – відсутній.
4. Кадр 4. **Заголовок:**
- текст – *Послухаємо самого Віктора Михайловича Глушкова;*
 - колір тла – світло-зелений, колір літер – темно-зелений;
 - тривалість показу – 7 с;
 - звук – загальний фоновий звук з файлу **Physical.wma**;
 - візуальний ефект – Кінематографічні згасання, зліва;
 - ефект переходу на наступний кадр – розчинення, мозаїка.
5. Кадр 5. **Відео розповіді В. М. Глушкова про безпаперову інформатику:**
- відео з файлу **Глушков відео 1.wmv**;
 - тривалість показу – 48 с;
 - звук – звук з файлу відео;
 - візуальний ефект – відсутній;
 - ефект переходу на наступний кадр – розчинення, мозаїка.
6. Кадр 6. **Фотографія В. М. Глушкова разом з розробниками комп'ютера МІР-1:**
- фото з файлу **Glushkov_MIR-1.jpg**;
 - тривалість показу – 6 с;
 - звук – загальний фоновий звук з файлу **Physical.wma**;
 - підпис – *Для того щоб кожна людина могла працювати з ЕОМ, потрібно перейти до використання малих комп'ютерів, і В. М. Глушков висуває ідею створення комп'ютерів МІР;*
 - колір літер – помаранчевий, колір фону – темно-зелений;
 - візуальний ефект – відсутній;
 - ефект переходу на наступний кадр – розчинення, мозаїка.



7. Кадр 7. **Фотографія комп'ютера МИР-1:**
 - фото з файлу **MIR-1.jpg**;
 - підпис – *Комп'ютер МИР мав малі розміри і зручний механізм введення-виведення даних – електричну друкарську машинку;*
 - усі інші значення властивостей аналогічні до кадру 6.
8. Кадр 8. **Фотографія комп'ютера МИР-2:**
 - фото з файлу **MIR-2.jpg**;
 - підпис – *Комп'ютер МИР-2 мав додаткову систему введення даних – світлове перо;*
 - усі інші значення властивостей аналогічні до кадру 6.
9. Кадр 9. **Фотографія комп'ютера МИР-3:**
 - фото з файлу **MIR-3.jpg**;
 - підпис – *Комп'ютер МИР-3 – швидкодія порівнянно з МИР-2 збільшена у 20 разів;*
 - усі інші значення властивостей аналогічні до кадру 6.
10. Кадр 10. **Заключні титри:**
 - титри – *Шкільна відеостудія «Майбутнє»:*
 - оператор – *Степан Михайленко;*
 - автор тексту – *Оксана Коваленко;*
 - монтаж – *Микола Іваненко;*
 - колір тла – *жовто-зелений*, колір літер – *білий*;
 - тривалість показу – *10 с.*

Добірка матеріалів до фільму

Використовуючи матеріали Інтернету та друковані матеріали, доберемо файли для використання у відеофільмі. Їх перелік подано в таблиці 5.2.

Таблиця 5.2

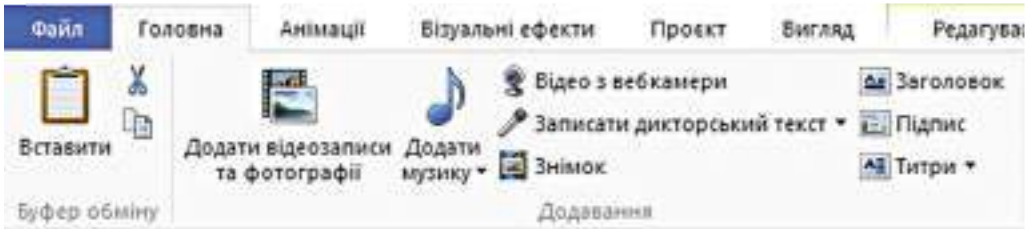
Файли для включення до відеофільму

Ім'я файлу	Папка розміщення	Тривалість, с	Номер кадру
Physical.wma	Розділ 5\Пункт 5.2\Аудіо	220	1–9, крім 5
Глушков відео 0.wmv	Розділ 5\Пункт 5.2\Відео	53	3
Глушков відео 1.wmv	Розділ 5\Пункт 5.2\Відео	48	5
Glushkov.jpg	Розділ 5\Пункт 5.2\Фото		2
Glushkov_MIR-1.jpg	Розділ 5\Пункт 5.2\Фото		6
MIR-1.jpg	Розділ 5\Пункт 5.2\Фото		7
MIR-2.jpg	Розділ 5\Пункт 5.2\Фото		8
MIR-3.jpg	Розділ 5\Пункт 5.2\Фото		9

Створення відеофільму

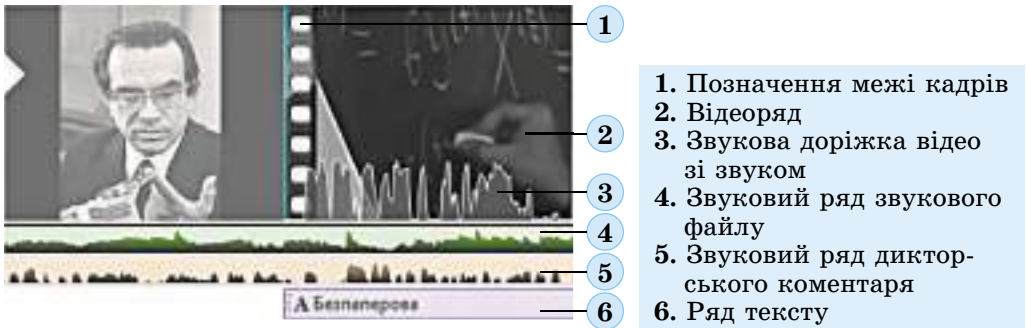
Для створення відеофільму з використанням **Кіностудії Windows** слід:

1. Запустити програму.
2. Зберегти проєкт відеофільму (**Файл** ⇒ **Зберегти проєкт**). За замовчуванням програма зберігає проєкт у файлі з розширенням імені **wlmp**.
3. Уставити зображення, відео та аудіоматеріали з файлів у послідовності, визначеній сценарієм, використавши для цього елементи керування групи **Додавання** вкладки **Головна** (мал. 5.19).



Мал. 5.19. Елементи керування групи **Додавання**

4. Додати заголовки фільму, титри, підписи відповідно до сценарію, використавши для цього елементи керування групи **Додавання** вкладки **Головна**.
5. Відредагувати об'єкти фільму, використовуючи елементи керування тимчасових вкладок **Знаряддя для відео**, **Знаряддя для музики**, **Знаряддя для дикторських текстів**, **Знаряддя для текстів**. При цьому слід зважати на те, що в області монтажу відео- і фотоматеріали вставляються у відеоряд (мал. 5.20, 2), звук із файлу – у звуковий ряд (мал. 5.20, 4), звук дикторських коментарів – ще в один звуковий ряд (мал. 5.20, 5), а титри, заголовки і підписи – в окремий ряд з текстом (мал. 5.20, 6). Якщо відео має звук, то звукова доріжка відображається в нижній частині відеоряду (мал. 5.20, 3).
6. Узгодити відеоряд з іншими рядами по часу перегляду, по гучності звуку тощо, використовуючи переміщення об'єктів по доріжці, перетягуванням лівої або правої межі кадру тощо.
7. За потреби вставити ефекти, що будуть накладатися на окремі кадри (елементи керування вкладки **Візуальні ефекти**).
8. Уставити ефекти переходу між окремими кадрами відеофільму (елементи керування вкладки **Анімація**).
9. Зберегти файл відеофільму – **Файл** ⇒ **Зберегти фільм**.



1. Позначення межі кадрів
2. Відеоряд
3. Звукова доріжка відео зі звуком
4. Звуковий ряд звукового файлу
5. Звуковий ряд дикторського коментаря
6. Ряд тексту

Мал. 5.20. Об'єкти **Області монтажу**

Час від часу в ході монтажу відеофільму слід зберігати проєкт. Якщо редагування відеофільму не завершено, то для його продовження потрібно відкрити збережений файл проєкту. Майте на увазі, що всі файли з мультимедійними даними повинні розміщуватись у тих папках, звідки вони були вставлені в проєкт, бажано в окремій папці для кожного проєкту. Наявність збереженого файлу проєкту також дає змогу користувачеві створити кілька варіантів відеофільму з різними ефектами і налаштуваннями.



Для тих, хто працює з програмою Відеоредактор

Основними відмінностями програми **Відеоредактор** від програми **Кіностудія Windows 10** є:

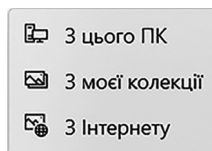
- відсутність накладення ефектів переходу від одного фрагмента фільму до іншого;
- відсутність можливості вибору значень властивостей шрифтів у написах. Оформлення тексту написів вибирається з набору стилів;
- для додавання авторського аудіокоментаря (дикторського тексту) потрібно його попередньо записати (наприклад, у програмі **Диктофон**) і зберегти у файлі.

Після запуску програми для створення нового відеофільму слід у початковому вікні вибрати кнопку **Новий відеопроєкт** і в діалоговому вікні (мал. 5.21) увести назву вашого відеофільму. Для продовження редагування вже існуючого проєкту потрібно вибрати його в списку **Мої відеопроєкти**.

Наведемо алгоритми основних операцій зі створення відеофільму в програмі **Відеоредактор**.



Мал. 5.21. Вікно Назвіть своє відео



Мал. 5.22. Джерело отримання

Додавання відео та графічних об'єктів до бібліотеки проєкту:


1. В області **Бібліотека проєктів** вибрати кнопку **+ Додати**.
2. Вибрати в списку кнопки джерело отримання потрібних відео чи зображень (мал. 5.22).
3. Зробити поточним джерело (наприклад, папку на носії даних комп'ютера), у якому розміщено потрібні файли з відео та зображеннями.
4. Вибрати файли, які потрібно включити до відео, та вибрати кнопку **Відкрити**.
5. Повторити команди 2–4 алгоритму для включення до проєкту всіх потрібних об'єктів.

Включення графічного або відеооб'єкта до відеофільму:


1. Вибрати в області **Бібліотека проєктів** потрібний об'єкт.
2. Перетягнути вибраний об'єкт в **Область історій** в один із вільних прямокутників або між уже заповненими прямокутниками.

Додавання текстових написів на окремих кадрах перед відео або графічним об'єктом:


1. Вибрати потрібний об'єкт в **Області історій**.
2. Вибрати в **Області історій** кнопку Додати картку назви.
3. Вибрати в **Області історій** кнопку Текст.
4. Увести в поле **Назва** потрібний текст напису.
5. Вибрати в списку **Стиль анімованого тексту** один із стилів оформлення тексту.
6. Вибрати в списку **Макет** один із запропонованих способів розміщення тексту у вікні.

7. Вибрати над вікном попереднього перегляду кнопку  Фон .
8. Вибрати в палітрі **Фон** потрібний колір фону.
9. Вибрати кнопку **Готово**.

Додавання текстових написів (титрів) на кадрах відео або графічних об'єктів:




1. Вибрати потрібний об'єкт в **Області історій**.
2. Вибрати в **Області історій** кнопку  Текст .
3. Увести в поле **Введіть текст** потрібний текст напису.
4. Вибрати в списку **Стиль анімованого тексту** один зі стилів оформлення тексту.
5. Вибрати в списку **Макет** один із запропонованих способів розміщення тексту у вікні.
6. Вибрати кнопку **Готово**.

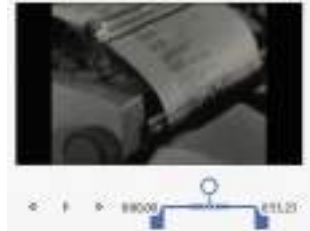
Змінення тривалості показу текстових і графічних об'єктів:

1. Вибрати потрібний об'єкт в **Області історій**.
2. Вибрати в **Області історій** кнопку  Тривалість .
3. Вибрати в списку потрібне значення або ввести у відповідне поле значення тривалості показу в секундах.

Редагування відеооб'єктів:



1. Обітнути відеофрагмент:

1. Вибрати об'єкт в **Області історій**.
2. Вибрати в **Області історій** кнопку  Обітнути .
3. Використовуючи вікно попереднього перегляду, встановити маркери початку  та кінця обтинання  в потрібне положення (мал. 5.23).



Мал. 5.23. Фрагмент вікна програми в режимі **Обтинання**

2. Розділити відеофрагмент:



1. Вибрати об'єкт в **Області історій**.
2. Вибрати в **Області історій** кнопку  Розділити .
3. Використовуючи вікно попереднього перегляду, встановити маркер точного місця відтворення  в потрібне положення (мал. 5.24).
4. Вибрати кнопку **Готово**.




Мал. 5.24. Фрагмент вікна програми в режимі **Розділити**

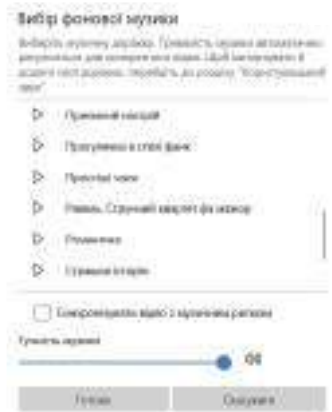
Додавання звукових об'єктів:

1. Додавання дикторського супроводу:

1. Вибрати над вікном перегляду кнопку  Користувачський звук .
2. Установити у вікні перегляду позицію , з якої почнеться відтворення звукового файлу.




3. Вибрати в області **Користувацький звук** кнопку **+** Додати аудіофайл.
 4. Вибрати потрібний файл у вікні **Відкриття файлу** та вибрати кнопку **Відкрити**.
2. **Додавання фонового звуку:**
1. Вибрати над вікном перегляду кнопку  Фонова музика .
 2. Вибрати в списку потрібну мелодію (мал. 5.25).
 3. Установити, використовуючи повзунок, рівень гучності музики та вибрати кнопку **Готово**.



Мал. 5.25. Вікно Вибір фонові музики

Завершення редагування відеофільму:

1. Вибрати над вікном перегляду кнопку  Завершити відео .
2. Відкрити в діалоговому вікні **Завершення відео** список **Якість відео** (мал. 5.26) та встановити потрібне значення якості.
3. Вибрати кнопку **Експорт**.
4. Указати у вікні **Збереження файлу** місце збереження відеофайлу.



Мал. 5.26. Вікно Завершення відео

Після завершення експорту відеофільм буде запущено для перегляду в окремому вікні.



Для тих, хто працює з Linux

У **Linux** для редагування відео можна використати програму **OpenShot** (мал. 5.27). За своїм інтерфейсом та основними операціями вона багато в чому подібна до програми **Кіностудія Windows**, яка описана в цьому пункті.



Мал. 5.27. Вікно програми OpenShot



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/7hdrzX> або QR-кодом.



Найважливіше в цьому пункті

Музичні та відеоредактори, а також музичні та відеостудії – це програми, призначені для редагування звукових і відеоданих. Використовуючи їх, можна вставляти, видаляти, копіювати, змінювати тривалість аудіо- чи відеофрагментів, об'єднувати фрагменти, вставляти у відео текстові та графічні об'єкти тощо.

Різниця між названими програмами в тому, що мультимедійні студії мають розширений набір інструментів і використовуються в професійній діяльності на студіях звукозапису, кіно- й анімаційних студіях.

Кіностудія Windows призначена для створення відеофільмів, що можуть містити відеофрагменти, фотографії, звуковий супровід (дикторський текст), різноманітні написи (титри). Створене відео можна опублікувати в Інтернеті або зберегти у файлі одного з відеоформатів.

Під час створення відеофільмів варто дотримуватися певної послідовності:

- **на першому етапі** потрібно розробити сценарій;
- **на другому етапі** здійснюється підготовка аудіо-, фото- та відеоматеріалів;
- **на третьому етапі** здійснюється монтаж відеофільму.



Дайте відповіді на запитання

- 1°. Для чого призначені музичні та відеоредактори?
- 2°. У чому відмінність між відеоредакторами і відеостудіями?
- 3°. Які музичні редактори ви знаєте? Які відеоредактори ви знаєте?
- 4°. Для чого призначена **Кіностудія Windows**?
- 5°. Для чого призначена **Область монтажу** в програмі **Кіностудія Windows**?
- 6°. Які об'єкти можуть бути вставлені у відеофільм? Їх властивості?
- 7°. Яка послідовність створення відеофільму?
- 8°. Що таке сценарій відеофільму? Для чого його створюють?
- 9°. Як уставити до проекту відеофільму відео або фото з файлу?
- 10°. Як записати дикторський текст?
- 11*. У чому сутність узгодження відео та звуку? Як це здійснити в **Кіностудії Windows**?



Виконайте завдання

- 1°. Складіть сценарій і створіть за ним в обраному вами відеоредакторі відеофільм про Лесю Українку до уроків української літератури. У відеофільмі використайте фрагмент з фільму «Великі українці. Леся Українка» і фотографії Лесі Українки та пам'яток їй з папки **Розділ 5\Пункт 5.2\Завдання 5.2.1\Українка**. Підготуйте і додайте до відеофільму титульний слайд і титри супроводу. Самостійно доберіть музику, яка повинна звучати протягом усього відеофільму. Збережіть відеофільм у вашій папці у файлі з іменем **завдання 5.2.1.wmv**.
- 2°. Розробіть сценарій і створіть в обраному вами відеоредакторі відеофільм про визначного мандрівника Христофора Колумба, використовуючи матеріал підручника з всесвітньої історії 8-го класу та матеріали, розміщені в папці **Розділ 5\Пункт 5.2\Завдання 5.2.2\Колумб**. Дикторський текст і музичний супровід доберіть самостійно. Збережіть відеофільм у вашій папці у файлі з іменем **завдання 5.2.2.wmv**.
- 3°. Розробіть сценарій і створіть в обраному вами відеоредакторі відеофільм про визначного українського письменника і кінорежисера Олександра До-



вженка для повідомлення про нього на уроці української літератури. Використайте матеріали підручника 8-го класу з української літератури та матеріали, розміщені в папці **Розділ 5\Пункт 5.2\Завдання 5.2.3\Довженко**. Дикторський текст і музичний супровід доберіть самостійно. Збережіть відеофільм у папці **Документи** у файлі з іменем **завдання 5.2.3.wmv**.



4*. Розробіть сценарій і створіть в обраному вами відеоредакторі відеофільм, що буде ілюструвати один з фізичних процесів з курсу фізики 8-го класу. Матеріали доберіть самостійно. Збережіть відеофільм у папці **Документи** у файлі з іменем **завдання 5.2.4.wmv**.



5*. Створіть в обраному вами відеоредакторі відеофільм про ваш клас або визначну шкільну подію. Складіть сценарій і доберіть матеріали самостійно. Збережіть відеофільм у вашій папці у файлі з іменем **завдання 5.2.5.wmv**.

ПРАКТИЧНА РОБОТА № 6

«Створення відеофільму»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

Створіть, використовуючи програму відеоредактора, відеофільм з розповіддю про термометри тривалістю до 2,5 хв. Для цього:

1. Розробіть сценарій відеофільму та реалізуйте його, передбачивши:
 - титульну сторінку з назвою відеокліпу, наприклад **Термометри**;
 - огляд різних видів термометрів, використавши для цього фотографії (4–6 фото) з папки **Розділ 5\Практична 6\Фото**. До кожного фото додайте титри з назвою типу термометра, що відповідає імені файлу. Тривалість демонстрації кожного фото – не більше 5 с;
 - розповідь про термометр Галілея, використавши для цього 2–3 фотографії з папки **Розділ 5\Практична 6\Фото** та відео з папки **Розділ 5\Практична 6\Відео**. До фотографій додайте титри, а до відеофрагмента – дикторський текст. Для підготовки титрів і дикторського тексту використовуйте матеріал з файлу **Розділ 5\Практична 6\Термометр Галілея.doc**;
 - кінцеві титри з вашим прізвищем як сценариста та режисера фільму;
 - музичний супровід усіх частин тексту, крім тих, де використовується дикторський текст. Для цього використовуйте файли музичних творів з папки **Розділ 5\Практична 6\Музика**;
 - ефекти переходів між кадрами доберіть самостійно (не більше двох різних).
2. Збережіть відеокліп у вашій папці у файлі з іменем **Практична робота 6**.


5.3. Сервіси для роботи з аудіо- і відеоданими та публікування їх в Інтернеті



1. Що таке *обліковий запис*? Як створити обліковий запис на поштово-му сервері?
2. Як надати доступ до матеріалів на **Google Диск**у?
3. Як додати музику до фільму в відеоредакторі?

Щоб розповсюдити створені вами відео та спростити доступ до них вашим знайомим, їх можна розмістити в Інтернеті. Це можна зробити з використанням різноманітних «хмарних» сервісів зберігання даних, а також можна використати такі

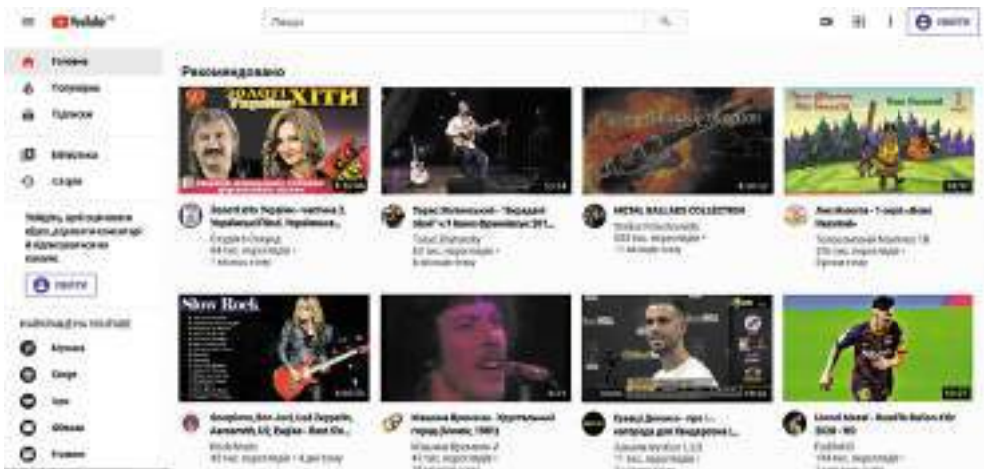
сервіси, як **Vimeo** (<https://vimeo.com>), **dailymotion** (<http://www.dailymotion.com>), **Відео ukrhomenet** (<http://video.ukrhome.net>), або соціальні мережі, наприклад **Facebook**.

Одним з розповсюджених сервісів для розміщення відеофайлів в Інтернеті є сервіс **YouTube**  (англ. *You Tube* – твоя труба (канал)). Розміщувати матеріали та надавати доступ до них на **YouTube** може будь-який користувач, який має обліковий запис **Google**.

Створення власного каналу на сайті YouTube для розміщення відеофайлів

Для розміщення мультимедійних матеріалів на **YouTube** потрібно створити **власний канал**. Для цього слід:

1. Відкрити в браузері головну сторінку сайту **YouTube** – <https://www.youtube.com> (мал. 5.28).



Мал. 5.28. Головна сторінка сайту **YouTube**

2. Вибрати кнопку **Увійти** та ввести дані логіна та пароля облікового запису **Google**.
3. Вибрати в правому верхньому куті вікна сайту **YouTube** кнопку власного облікового запису, наприклад таку , і вибрати команду **Створити канал**.
4. Вибрати у вікні **Розкрийте свій авторський потенціал** кнопку **Почати**.
5. Вибрати у вікні **Виберіть, як створити канал** один з двох варіантів назви каналу – ваше власне ім'я або інше ім'я (назву), під яким ваші відеоматеріали будуть розміщуватися на сервісі **YouTube**.
6. На сторінці завершення налаштування каналу слід:
 1. Завантажити зображення вашого профілю.
 2. Увести в поле **Опис каналу** відомості про ваш канал – про що і для кого будуть ваші відеоматеріали.
 3. Додати за потреби посилання на ваш сайт, електронну пошту та сторінки в соціальних мережах.
 4. Вибрати кнопку **Зберегти та продовжити**, якщо ви ввели потрібні дані, або кнопку **Налаштувати пізніше**, якщо ви зараз не бажаєте вводити зазначені дані.

Вигляд сторінки створеного власного каналу **YouTube** подано на малюнку 5.29. У подальшому можна:

- продовжити налаштування параметрів каналу (кнопка **Персоналізувати канал**);
- ознайомитися з різноманітними відеоматеріалами на допомогу користувачу по роботі з відео та перейти до **Інформаційної панелі каналу** (кнопка **Творча студія YouTube**);
- завантажити відеофайл на власний канал (кнопка **Завантажити відео**).



Мал. 5.29. Початкове вікно створеного власного каналу **YouTube**

Творча студія YouTube в основному призначена для опрацювання ваших відео та для ознайомлення з різноманітними аналітичними матеріалами, новинами сайту тощо.


На **Інформаційній панелі Творчої студії YouTube** відображаються відомості про ваш канал, що включають статистичні дані про підписки на канал, його відвідування та перегляд відео за певний період часу тощо (мал. 5.30).



Мал. 5.30. Інформаційна панель створеного каналу **YouTube**

Завантаження відео на канал **YouTube**

Для завантаження відеофайлу на ваш канал слід:

1. Вибрати кнопку **Завантажити відео** (на **Інформаційній панелі каналу** або на головній сторінці каналу).
2. Вибрати у вікні **Завантажити відео** кнопку **Вибрати файли** або кнопку .
3. Вибрати потрібні файли в пам'яті вашого комп'ютера або на диску в хмарі та вибрати кнопку **Відкрити**.
4. Здійснити три кроки налаштування параметрів відео, що завантажуються (мал. 5.31):



Мал. 5.31. Вікно першого кроку налаштування параметрів відео – **Деталі**

1. **Перший крок – Деталі:**
 1. Увести в поле **Назва** ім'я відео.
 2. Увести в поле **Опис** відомості про зміст відео.
 3. Указати в розділі **Значок відео**, яке зображення буде значком відео (можна вибрати кадр із запропонованих програмою або вставити зображення з файлу (кнопка **Завантажити значок відео**)).
 4. Додати в поле **Списки відтворення** назву списку, який буде, по суті, іменем групи тематично пов'язаних відео.
 5. Вибрати перемикач відповідності змісту відео певній віковій категорії (призначений чи ні цей контент для дітей). За потреби встановити інші вікові обмеження.
2. **Другий крок – Елементи відео** вказуються тільки у відео, яке не призначене для дітей. Для дорослих є можливість додати рекламу інших ваших власних відео.
3. **Третій крок – Видимість:**
 1. Вибрати в групі **Зберегти і опублікувати** перемикач, що визначає обмеження на доступ до вашого відео: **Приватне** (доступне тільки для вас), **Не для всіх** (доступне обмеженому колу глядачів, яким ви надали посилання на відео), **Для всіх**.
 2. Запланувати за потреби терміни зміни доступності вашого відео на **Для всіх**.
5. Скопіювати за потреби адресу вашого відео, яка відображається під вікном попереднього перегляду.
6. Вибрати кнопку **Зберегти**.

Редагування відео, розміщеного на каналі YouTube

Після завантаження відеофайлу його можна переглянути, а також здійснити редагування. Для цього слід:

1. Відкрити список **Відео** на лівій панелі вікна **Творчої студії YouTube**.
2. Двічі клацнути на значку потрібного відео.

У вікні, що відкрилося (мал. 5.32), можна переглянути та змінити введені раніше дані про ваше відео, а вибравши на лівій панелі посилання **Редактор**, перейти до редагування відео.



Мал. 5.32. Вікно YouTube Studio в режимі редагування Інформації про відео

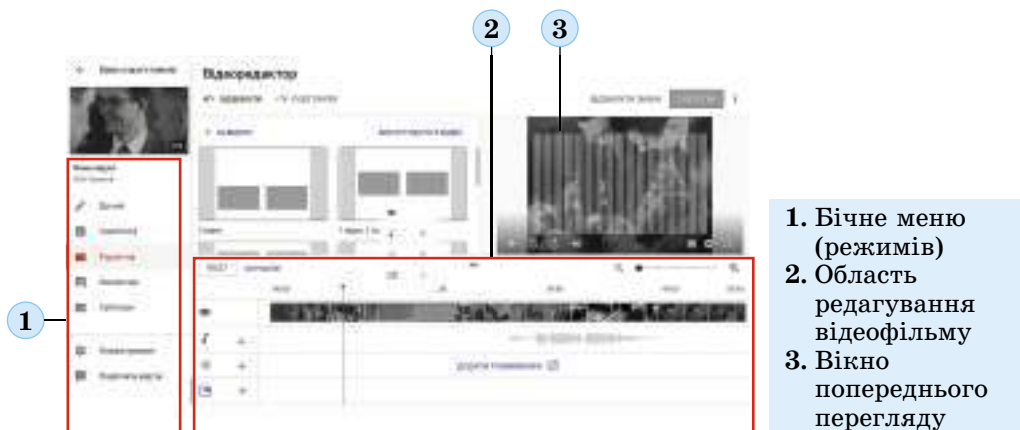


Відеоредактор сервісу **YouTube** (мал. 5.33) надає користувачу можливість здійснити такі основні операції редагування:

- обітнути відео, видалити довільні фрагменти відеофільму в будь-якій його частині;
- додати до відео фонову музику;
- додати до відео посилання на свій канал або окремі відео;
- додати розмивання певних ділянок відео.

Для **обтинання відео** слід:

1. Вибрати в області редагування посилання **Обрізати**.
2. Установити поточним положення, у якому буде здійснено розрив відео.
3. Вибрати команду **Розділити**, що з'явилася в нижній частині екрана редактора.



1. Бічне меню (режимів)
2. Область редагування відеофільму
3. Вікно попереднього перегляду

Мал. 5.33. Вікно **YouTube Studio** в режимі редагування відео

4. Підвести вказівник миші до вертикальної лінії, що утворилася після виконання команди **Розділити** (вказівник повинен набути вигляду двонаправленої стрілки ↔).
5. Перетягнути вертикальну лінію вліво або вправо до моменту завершення обтинання.

У результаті в **Області редагування** повинен відобразитися виділений темно-сірим кольором фрагмент відео, який у подальшому буде видалено (мал. 5.34). Щоб відмінити виділення, потрібно скористатися командою **Очистити все** або вибрати кнопку ⊗ над виділеним фрагментом.




Мал. 5.34. Фрагмент вікна **YouTube Studio** в режимі редагування обтинання



У режимі обтинання можна виділити кілька фрагментів для вирізання в різних частинах відео. Після того як всі ділянки для видалення обрано, слід вибрати команду **Переглянути**, щоб вийти з режиму обтинання та переглянути, як буде відтворюватися відео без вибраних фрагментів. За потреби можна повернутися до режиму обтинання, вибравши команду **Редагувати обтинання**.

Остаточне видалення обраних фрагментів відбудеться після вибору команди **Зберегти**. Процес унесення змін може тривати доволі довго, і на цей час користувачам буде доступна версія вашого відео без внесених змін.

У **YouTube Studio** передбачено лише один варіант **додавання звукових даних** – фонової музики з набору композицій, що включені до **Творчої студії YouTube**. Для вставлення фонової музики слід:

1. Вибрати в області редагування відеофільму, зліва від звукової доріжки, кнопку +.
2. Вибрати у фонотеці композицій **YouTube** потрібну мелодію (за потреби прослухати її, вибравши кнопку **Відтворити композицію** .
3. Вибрати кнопку **Додати до відео**.
4. Установити (якщо відео вже мало власний звук) співвідношення між рівнем оригінального звуку відео і мелодією, що була додана (мал. 5.35).
5. Вибрати кнопку **Зберегти зміни**.

В окремих випадках, наприклад, коли ви розповсюджуєте через канал **YouTube** відео, на якому відображені приватні особи, які не дали згоди на розповсюдження цього відео, або яке містить сцени жорстокості, варто скористатися функцією розмивання фрагментів зображення. Для цього слід:

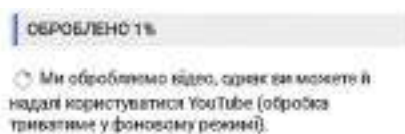
1. Вибрати в області редагування відеофільму на доріжці **Розмити**  послання **ДОДАТИ РОЗМИВАННЯ** .
2. Вибрати у вікні, що відкрилося (мал. 5.36), кнопку **Редагувати** біля назви одного з режимів розмивання, наприклад **Розмити обличчя (Розмиття вручну)**.



Мал. 5.35. Елемент керування встановлення рівнів звучання звуку і фонової музики



Мал. 5.36. Фрагмент вікна **YouTube Studio** в режимі редагування розмивання



Мал. 5.37. Індикатор ходу опрацювання відео в режимі розмивання

3. Дочекатися, поки програма здійснить опрацювання відео (пошук облич) (мал. 5.37) і запропонує варіанти розмивання.
4. Вибрати на панелі **Розмити обличчя** зображення, які потрібно на відео розмити (мал. 5.38).
5. Вибрати кнопку **Зберегти**.

Під час розмиття фрагментів вручну слід зупинити перегляд на моменті появи на екрані зображення, яке потрібно розмити, та виділити прямокутну область розмивання. Програма здійснить опрацювання відео і запропонує уточнені параметри часу розмивання ділянки екрана. Користувач може самостійно, використовуючи перетягування лівої та правої меж, змінити час початку та завершення розмивання.

Використання Кіностудії Windows для розміщення відео в Інтернеті

Після розробки проєкту відеофільму в програмі **Кіностудія Windows** можна відразу розмістити його в Інтернеті. Для цього слід відкрити проєкт створеного відеофільму та виконати такий алгоритм:

1. Виконати **Файл** ⇒ **Опублікувати фільм**.
2. Вибрати зі списку сервісів (мал. 5.39) сервіс, на сайті якого будете розміщувати відеофайл, наприклад **YouTube**.
3. Увести в поля діалогового вікна логін і пароль доступу до вашого відеоканалу на **YouTube**.
4. Увести в поля діалогового вікна значення властивостей відеофільму: назву, опис, теги, вид дозволу на використання, категорію відеофільму.
5. Вибрати кнопку **Опублікувати**.
6. Дочекатися завершення збереження та опублікування фільму.

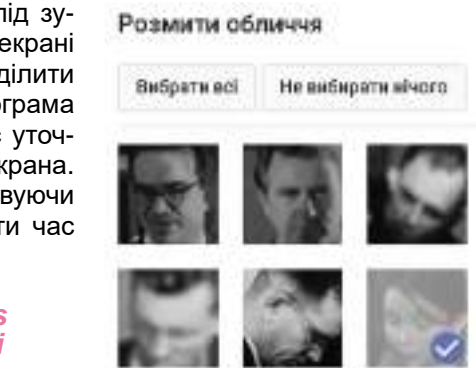
За подібним алгоритмом відбувається й опублікування фільму на інших сайтах. Обов'язковою умовою розміщення мультимедійних даних є створення облікового запису на сайті відповідного сервісу.

Розміщення аудіофайлів в Інтернеті

Переважна більшість сервісів для розміщення аудіофайлів призначена для тих, хто створює власні файли з різноманітними музичними творами. Ще одним напрямком розміщення аудіофайлів є створення аудіокнижок і різноманітних оглядів з певних питань: політичних, економічних, технічних тощо.

Зазначені аудіофайли називають **подкастами**. У подальшому термін «подкаст» поширився і на відеофайли, що розміщують користувачі в Інтернеті.

Для розміщення аудіофайлів в Інтернеті існують різні сервіси, такі як **Канал громадського подкастингу**



Мал. 5.38. Зображення облич на панелі **Розмити обличчя**



Мал. 5.39. Список варіантів опублікування фільму

Подкаст (англ. *ipod* – медіаплеєр корпорації Apple; *broadcasting* – ефірне мовлення) – медіафайл або серія таких файлів, що створюються користувачами (*подкастерами*) і розповсюджуються з використанням Інтернету.

(<http://cpod.co/>), Український подкаст-термінал (<http://podcaster.org.ua>), Громадське радіо (<https://soundcloud.com/hromadske-radio>), сторінка медіакомпанії BBC (<https://www.bbc.com/ukrainian/learning-english>) тощо.

Для обміну аудіофайлами з друзями та знайомими можна використати сервіс **Google Диск**, роботу з яким ви опанували в 7-му класі.



Для тих, хто працює з Linux

У **Linux** для розміщення відео в Інтернеті можна використати редактор відеофільмів **OpenShot**. Після запуску програми слід:

1. Виконати **Файл** ⇒ **Завантажити відео**.
2. Вибрати відеофайл, який буде опубліковано в Інтернеті. Для цього скористатися елементом керування **Файл відео** (мал. 5.40).
3. Вибрати сервіс, на якому буде розміщено відеофільм (елемент керування **Відвантажити до**), наприклад **YouTube**.
4. Увести у відповідні поля вікна **Завантаження відео** значення властивостей відеофільму: *назву* (заголовок) та *опис*.
5. Увести логін і пароль доступу до сервісу.
6. Вибрати кнопку **Завантажити відео**.

Усі інші дії з розміщеним відеофайлом аналогічні до розглянутих вище.



Мал. 5.40. Вікно **Завантажити відео**



Працюємо з комп'ютером

Виконайте тренувальні завдання за посиланням <https://cutt.ly/AhdryjR> або QR-кодом.



Найважливіше в цьому пункті

В Інтернеті є значна кількість сайтів, що надають послуги з розміщення мультимедійних файлів. Зазвичай для розміщення файлів на сайтах потрібно мати відповідний обліковий запис.

Одним з розповсюджених сервісів для розміщення відеофільмів є **YouTube** (<https://www.youtube.com>). Для завантаження фільмів на сайт **YouTube** слід створити власний канал, який у подальшому буде містити добірку відеофайлів користувача. Користувач може керувати доступом до перегляду кожного з відеофайлів, установлюючи певний рівень доступу.

Значна частина відеоредакторів має засоби для розміщення відеофільмів в Інтернеті, наприклад такі програми, як **Кіностудія Windows** та **OpenShot**.

Мультимедійні файли, призначені для розміщення в Інтернеті, називаються **подкастами**, а особи, що створюють такі файли, – **подкастерами**.



Дайте відповіді на запитання

- 1°. Які сервіси для розміщення мультимедійних файлів в Інтернеті ви знаєте?
- 2°. Як створити канал для розміщення відеофайлів на сайті **YouTube**?

- 3°. Які значення властивостей відео можуть бути встановлені користувачем під час завантаження файлу на сайт **YouTube**?
- 4°. Які рівні конфіденційності можна встановити для перегляду завантаженого на **YouTube** файлу?
- 5°. Які дії слід здійснити для додавання музичного супроводу до завантаженого на **YouTube** файлу?
- 6°. Які дії потрібно виконати для розміщення відеофайлу на одному із сервісів Інтернету безпосередньо з програми відеоредактора, наприклад з **Кіностудії Windows**?
- 7°. Яка послідовність дій для розміщення аудіофайлів на **Диску Google**?
- 8°. Як надати доступ до ваших відео на каналі **YouTube** тільки окремим особам? Як їх повідомити про можливість такого перегляду?



Виконайте завдання

- 1°. Розмістіть на **YouTube** запропонований учителем/вчителькою файл, наприклад **Розділ 5\Пункт 5.3\Відео\Сейм, старе річище.mov**. Додайте опис, теги та музичний супровід. Надішліть учителю/вчительці адресу розміщеного файлу.
- 2°. Використовуючи проєкт відеофільму **Довженко**, розроблений на минулих заняттях, розмістіть на **YouTube** цей фільм, безпосередньо з програми **Кіностудія Windows**.
- 3°. Розмістіть на **YouTube** запропонований учителем/вчителькою файл, наприклад **Розділ 5\Пункт 5.3\Відео\Равлик.mov**. Знайдіть в Інтернеті відомості про равлика та додайте їх до відеофільму в описі та тегах.
- 4°. Розмістіть на **Google Диску** запропонований учителем/вчителькою файл, наприклад **Розділ 5\Пункт 5.3\Аудіо\Щедрівочка.wma**. Надайте доступ до цього файлу тим користувачам, що мають посилання. Надішліть електронною поштою адресу файлу в Інтернеті учителю/вчительці та одному/одній з ваших друзів/подруг.

ПРАКТИЧНА РОБОТА № 7

«Розміщення аудіо- та відеоматеріалів в Інтернеті»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Розмістіть на **YouTube** запропонований учителем/вчителькою файл, наприклад **Розділ 5\Практична 7\Відео\комахи.wmv**.
2. Установіть такі значення його властивостей:
 - назва – *Комахи*;
 - опис – *Комахи, що трапляються в центральній частині України*;
 - теги – *природа, комахи, жук-олень*;
 - рівень доступу – *Не для всіх*;
 - ескіз відео – *жук-олень*;
 - музика – *вид Кантрі та фольк, назва As We Go*.
3. Надішліть на електронну адресу вчителя/вчительки посилання на ваше відео.

Розділ 6.

Алгоритми та програми

У цьому розділі ви дізнаєтеся про:

- мови програмування та їх складові елементи
- поняття програми і проекту
- середовища розробки проектів **Lazarus** та **IDLE**
- складання проектів мовами **Pascal** і **Python**, їх редагування та виконання
- налагодження проектів
- об'єкти та їх властивості
- події та методи, які є обробниками подій
- компоненти **форма, кнопка, напис, вікно повідомлень, поле, прапорець, перемикач**, їх властивості, події, що можуть відбуватися з ними, їх використання в проектах
- змінення значень властивостей компонентів перед виконанням проекту і під час його виконання
- змінні та постійні величини, їх властивості
- лінійні проекти, проекти з розгалуженнями і циклами

6.1. Комп'ютерні програми і мови програмування



1. Що таке *алгоритм*? Що таке *система команд виконавця алгоритму*? У чому полягає формальність виконання алгоритму виконавцем?
2. Що таке *комп'ютерна програма*?
3. Яку мову програмування ви вивчали в попередніх класах? Пригадайте кілька команд цієї мови програмування.

Комп'ютерні програми

Ви вже знаєте, що комп'ютер працює під керуванням створеного людиною програмного забезпечення, яке складається з комп'ютерних програм різноманітного призначення. Працюючи з комп'ютером у школі та вдома, ви використовували текстовий процесор, графічний редактор, табличний процесор, редактор комп'ютерних презентацій, навчальні та контролювальні програми, ігрові програми та багато інших.

Ви також знаєте, що **комп'ютерна програма** – це алгоритм, записаний спеціальною мовою та призначений для виконання комп'ютером.

У процесі своєї роботи програма може опрацьовувати **дані**.

Дані, які потрапляють до програми від певних пристроїв (наприклад, з клавіатури або від датчика температури), або від іншої програми (наприклад, від табличного процесора), або з іншого джерела (наприклад, з текстового файлу), називаються **вхідними (початковими) даними**. Деякі програми працюють без вхідних даних.

Під час виконання програми можуть утворюватися та опрацьовуватися й інші дані, які називаються **проміжними даними**.



Якщо метою виконання програми є отримання певних даних, то ці дані називаються **вихідними (результуючими) даними**.

У багатьох сучасних програмах використовуються засоби керування (кнопки, меню та ін.), засоби введення даних (поля, лічильники та ін.), засоби виведення результатів (вікно повідомлень, написи, текстові поля та ін.) тощо. Сукупність таких засобів, а також методів їх використання утворюють **інтерфейс користувача**.

Сучасні комп'ютерні програми часто називають **комп'ютерними проєктами**.

Мови програмування

Складаючи алгоритми, призначені для виконання людиною, користуються мовою спілкування людей: українською, англійською, німецькою, російською тощо.

Але для запису алгоритмів, які повинен виконувати автоматичний пристрій (зокрема, комп'ютер), мова спілкування людей складна, має неоднозначні трактування (наприклад, слова-омоніми). Тому для запису алгоритмів, які призначені для виконання автоматичними пристроями, розробляють і використовують спеціальні мови.

Мова, яка використовується для запису комп'ютерних програм, називається **мовою програмування**.

За останні 70 років створено приблизно 3000 різних мов програмування. Деякі з них уже вийшли з користування, для деяких постійно з'являються нові версії, які зручніші для складання програм. Постійно створюються нові мови програмування.

Деякі мови програмування є **універсальними** і використовуються для складання програм для розв'язування задач з різних галузей науки, техніки, виробництва, сфери побуту та ін. Деякі мови програмування є **спеціальними** і створені для складання програм для розв'язування специфічного кола задач.

У 2020 році найбільше програм створювалося мовами програмування **Java, JavaScript, C# (C Sharp), PHP, Python, C++**.

Мови програмування є прикладами так званих **штучних мов**. Іншими прикладами штучних мов є, наприклад, мова запису хімічних реакцій, мова запису розв'язання математичних рівнянь і нерівностей, мова запису поштових індексів на конвертах і листівках, мова запису ходів шахової партії, мова запису музичного твору та багато інших.

Кожна мова програмування має такі складові:

- **Алфавіт** – множина символів, з яких можна утворювати слова й речення цієї мови;
- **Словник** – набір слів, які використовуються в цій мові;
- **Синтаксис** – правила складання і запису мовних конструкцій: несловникових слів і речень;
- **Семантика** – встановлене однозначне тлумачення мовних конструкцій, правил їх виконання.

У 5–7-му класах ви вивчали мову програмування **Scratch 2**. Пригадаємо, що до алфавіту цієї мови програмування входять цифри, літери української абетки, англійського алфавіту, символи **:**, **=**, **+**, *****, **<** та ін. До її словника входять, наприклад, слова **повторити**, **перемістити**, **якщо**, **відбити**, **коли**. Відповідно до синтаксичних правил цієї мови програмування команда циклу з передумовою має містити слова **повторити**, **поки не** і саме в цьому порядку. А відповідно до семантики цієї мови програмування комп'ютер однозначно виконає команду **перемістити на 10 кроків** і перемістить виконавця на 10 кроків у напрямку його руху.

Використання символів, що не входять до алфавіту мови програмування, неправильне написання словникових слів, порушення синтаксичних правил призводить до неможливості виконання комп'ютером відповідної команди. Такі порушення називаються **синтаксичними помилками**.



Для тих, хто вивчає Pascal

У 8-му і 9-му класах ви будете вивчати мову програмування **Pascal** (англ. *Pascal* – Паскаль).

Цю мову використовують як для навчання учнів і студентів основних принципів програмування, так і професійні програмісти для створення великих і складних проєктів.



Цікаві факти з історії



Мал. 6.1.
Ніклаус Вірт

Мову програмування **Pascal** створила в 1970 році група співробітників Інституту інформатики Швейцарської вищої політехнічної школи під керівництвом професора **Ніклауса Вірта** (нар. у 1934 р., мал. 6.1), та її було названо на честь видатного французького математика, фізика, інженера, літератора, філософа Блеза Паскаля (1623–1662).

Створювалася мова програмування **Pascal** для навчання студентів основних принципів і методів програмування. Згодом цю мову програмування почали використовувати не тільки для навчання, а й для створення програм для розв'язування практичних завдань з використанням комп'ютера. З'явилися програми-компілятори для автоматичного перекладання програм, написаних мовою програмування **Pascal**, у програми, які може виконати процесор комп'ютера, а також середовище розробки програм **Turbo Pascal**, які із часом здобули широку популярність.

З роками мова програмування **Pascal** розвивалася й удосконалювалася. У середині 80-х років ХХ ст. одна з версій цієї мови одержала назву **Object Pascal**. А на початку 2000-х років чергова нова версія **Object Pascal** одержала назву **Delphi**.

Середовище розробки проєктів

Для створення, редагування, налагодження та виконання проєктів мовою програмування **Pascal** будемо використовувати **середовище розробки проєктів Lazarus**.

Це середовище складається з:

- **текстового редактора** для введення і редагування тексту проєкту;
- **компілятора** для перекладу команд проєкту з мови програмування **Pascal** на мову команд, які може виконати процесор комп'ютера;
- **засобів налагодження проєкту** для пошуку в ньому помилок;
- **довідкової системи**


та інших компонентів.

Це середовище розробки надає можливість використовувати велику бібліотеку **візуальних компонентів** (англ. *Lazarus Component Library (LCL)* – бібліотека компонентів **Lazarus**). Це вже знайомі вам компоненти (елементи керування), які ви використовували під час роботи з різноманітними програмами: кнопки, поля, написи, прапорці, перемикачі, лічильники, списки, смуги прокручування, а також багато інших.

При використанні в проєкті компонентів певні команди мовою програмування **Pascal** включаються до проєкту автоматично, їх не потрібно вводити вручну, що значно зменшує, спрощує і пришвидшує роботу зі створення проєкту. Такий спосіб розробки проєкту називають **візуальним**.

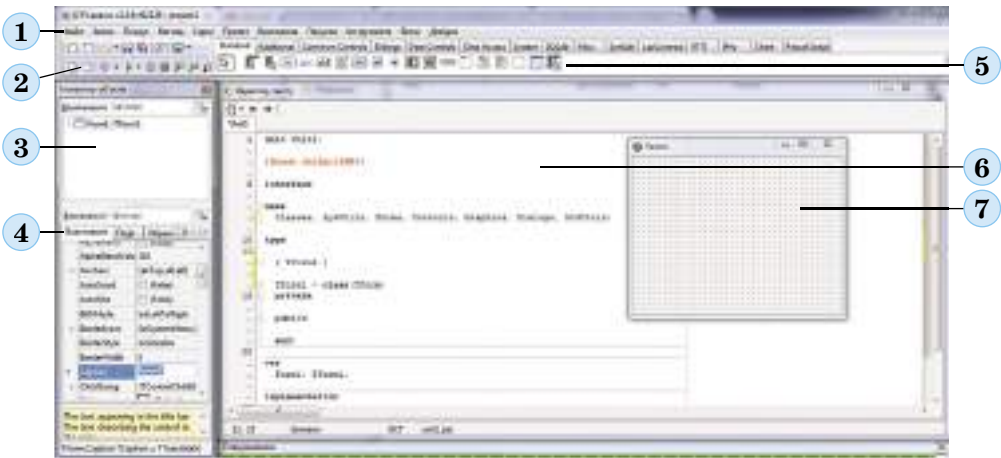


Відкрити вікно середовища **Lazarus** можна:

- використовуючи команду **Lazarus** меню **Пуск**;
- використовуючи значок  на **Робочому столі**.

У результаті відкривається вікно середовища **Lazarus** (мал. 6.2), у якому розміщено **Головне меню** (мал. 6.2, 1), Панелі інструментів (мал. 6.2, 2), вкладки з компонентами (елементами керування) (мал. 6.2, 5), а також кілька підпорядкованих вікон, серед яких:


- вікно **Інспектор об'єктів** (мал. 6.2, 3) з вкладками (мал. 6.2, 4), у якому можна для кожного компонента переглядати і змінювати значення його властивостей, події, які можуть відбуватися із цим компонентом, та ін.;
- вікно **Редактор тексту** з вкладками (мал. 6.2, 6), на яких розміщуються фрагменти проекту мовою програмування **Pascal**;
- вікно **форми** (мал. 6.2, 7), на якому конструюється інтерфейс користувача майбутнього проекту.



- | | |
|--------------------------------------|----------------------------|
| 1 – Головне меню | 5 – Вкладки з компонентами |
| 2 – Панелі інструментів | 6 – Вікно Редактор тексту |
| 3 – Вікно Інспектор об'єктів | 7 – Вікно форми |
| 4 – Вкладки вікна Інспектор об'єктів | |

Мал. 6.2. Вікно середовища **Lazarus** і підпорядковані вікна

Збереження проєкту

Проєкти, створені в середовищі **Lazarus**, складаються з кількох файлів. Тому доцільно для кожного нового проєкту створювати окрему папку. Зберегти проєкт можна, використовуючи команду **Зберегти всі** меню **Файл**, або команду **Зберегти проєкт** меню **Проєкт**, або кнопку **Зберегти всі**  на панелі інструментів.

Система послідовно пропонує зберегти два файли проєкту:

- файл з іменем **Project1** (англ. *project* – проєкт) і з розширенням імені **lpi**;
- файл з іменем **Unit1** (англ. *unit* – блок) і розширенням імені **pas**.

Розширення імен файлів змінювати не можна, а імена файлів можна змінити. Усі інші файли проєкту автоматично зберігаються у створеній папці (мал. 6.3).


Якщо вже збережений проєкт змінити, то під час збереження його нової версії в ту саму папку без змінення імен його файлів ніякі додаткові вікна відкриватися не будуть.

Уже збережений проєкт можна зберегти в іншій папці, імена його файлів можна змінити. Для цього потрібно виконати команду **Зберегти проєкт як меню Проєкт** і після цього – команду **Зберегти як меню Файл**.

backup	18.03.2020 23:19	Папка с файлами	
project1	18.03.2020 23:18	Значок	134 КБ
project1	18.03.2020 23:19	Lazarus Project Inf...	2 КБ
project1	18.03.2020 23:19	Lazarus Project M...	1 КБ
project1.lps	18.03.2020 23:19	Файл "LPS"	2 КБ
project1.res	18.03.2020 23:19	Файл "RES"	136 КБ
unit1	18.03.2020 23:19	Lazarus Form	1 КБ
unit1	18.03.2020 23:19	Pascal Source Code	1 КБ

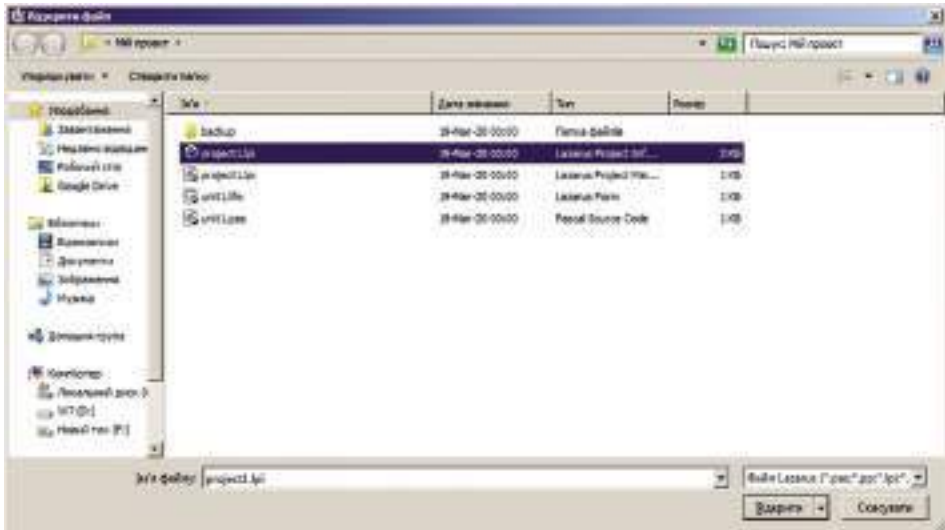
Мал. 6.3. Файли проєкту

Відкриття збереженого проєкту

Щоб відкрити проєкт, збережений раніше на носії, можна виконати команду **Відкрити** з меню **Файл** або вибрати кнопку **Відкрити**  на панелі інструментів, після цього у вікні **Відкрити файл**, що відкриється, вибрати файл з розширенням імені **lps**, а далі вибрати кнопку **Відкрити** (мал. 6.4).

Можна також виконати команду **Відкрити проєкт** з меню **Проєкт**.

Ще один спосіб відкриття проєкту – відкрити папку з файлами проєкту і двічі клацнути на файлі з розширенням імені **lps**. Після цього автоматично відкриється середовище **Lazarus** і в ньому відкриються файли проєкту.




Мал. 6.4. Вікно **Відкрити файл** з файлами проєкту

Виконання проєкту

Як вже зазначали, сучасні середовища візуальної розробки проєктів, одним з яких є **Lazarus**, частково автоматизують процес їх створення, тим самим полег-




шуючи і пришвидшуючи роботу як професіоналам, так і початківцям. Уже навіть під час відкриття нового проекту, коли ще не було введено жодного рядка тексту проекту, **Lazarus** автоматично створює (генерує) певний початковий текст проекту. Цього тексту проекту достатньо, щоб запустити його на виконання.

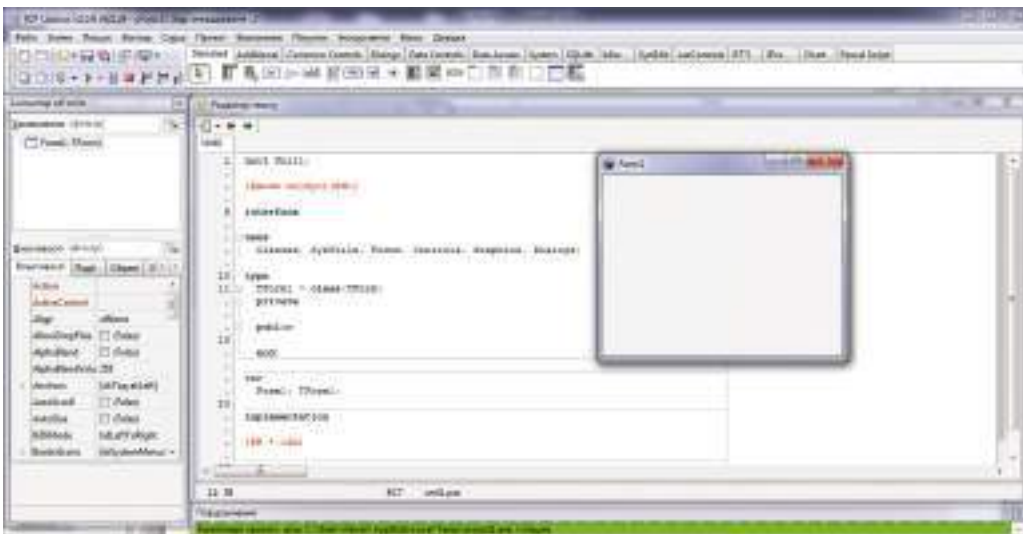
Розпочати виконання проекту можна, виконавши команду **Виконати** з меню **Виконання**, або вибравши кнопку **Виконати**  на панелі інструментів, або натиснувши клавішу **F9**.

Після запуску проекту на виконання розпочинає свою роботу **програма-компілятор**. Перш за все ця програма аналізує текст проекту на наявність синтаксичних помилок. За наявності таких помилок компілятор перериває свою роботу і вказує на місце першої з них. Потрібно виправити помилку і знову запустити проект на виконання. Якщо текст проекту не містить синтаксичних помилок, то компілятор перекладає текст проекту з мови **Pascal** на мову команд, які може виконати процесор комп'ютера, і передає вже перекладений проект процесору на виконання.

Компілятор (англ. *compiler* – той, хто збирає в ціле) – комп'ютерна програма, що перетворює (компілює) програму, написану певною мовою програмування, на еквівалентний код іншою мовою програмування, який може виконати комп'ютер.

При виконанні проекту, який автоматично створюється після відкриття нового проекту, на екрані відкривається вікно, значення властивостей якого (колір, розміри, положення на екрані, текст у рядку заголовка та ін.) встановлено за замовчуванням (мал. 6.5).

Для завершення виконання проекту можна закрити вікно його виконання, або вибрати кнопку **Закінчити**  на панелі інструментів, або натиснути сполучення клавіш **Alt+F4**.



Мал. 6.5. Проект, що створюється за замовчуванням, і результат його виконання

Щоб закрити вікно середовища **Lazarus**, можна виконати команду **Вихід** з меню **Файл** або вибрати кнопку закриття вікна середовища в рядку заголовка вікна **Lazarus**.

Проекти з вікнами. Форма та її властивості

Форма – це об'єкт, на якому під час створення проекту можна розмістити різні компоненти (елементи керування), зокрема кнопки, поля, написи, меню, смуги прокручування та ін. Під час виконання проекту форма відкривається як **вікно прикладної програми**, на якому будуть відображатися всі об'єкти, розміщені на формі.

Форма має **властивості**, перелік яких відображається на вкладці **Властивості** вікна **Інспектор об'єктів** (мал. 6.2, 4). У лівій частині цієї вкладки розміщено список властивостей, а у правій – їх значення. Значення деяких властивостей форми потрібно вводити у відповідні поля, а деяких – можна як вводити в поле, так і вибирати зі списку.

Наведемо приклади деяких властивостей форми:

- **Name** (англ. *name* – ім'я) – її значення визначає ім'я форми як об'єкта;
- **Caption** (англ. *caption* – заголовок) – її значення визначає текст у рядку заголовка форми;
- **Color** (амер. *color* – колір) – її значення визначає колір фону форми;
- **Height** (англ. *height* – висота) – її значення визначає висоту форми в пікселях (**піксель** – це точка графічного зображення на екрані);
- **Width** (англ. *width* – ширина) – її значення визначає ширину форми в пікселях;
- **Top** (англ. *top* – верх) – її значення визначає відстань (у пікселях) від верхньої межі форми до верхньої межі екрана;
- **Left** (англ. *left* – лівий) – її значення визначає відстань (у пікселях) від лівої межі форми до лівої межі екрана.

Зазначимо, що значення властивостей **Top** і **Left** можна також змінити перетягуванням форми по екрану, а значення властивостей **Height** і **Width** – зміненням розмірів форми, використовуючи маркери, розташовані на її межах.

До початкового проекту, як і до будь-якого іншого, можна вносити зміни. Наприклад, на вкладці **Властивості** вікна **Інспектор об'єктів** можна встановити такі значення властивостей форми:

Caption – *Перше вікно*;

Color – *c/Red* (*color* – колір, *red* – червоний);

Height – *350*;

Width – *600*;

Left – *550*;

Top – *200*.

Якщо зберегти нову версію проекту і знову виконати його, теж відкриється вікно, але іншого кольору, інших розмірів, розташоване в іншому місці екрана, з іншим текстом у рядку заголовка (мал. 6.6).



Мал. 6.6. Результат виконання зміненого проекту



Для тих, хто вивчає Python

У 8-му і 9-му класах ви будете вивчати мову програмування **Python** (англ. *Python* – пітон). Ця мова має широке застосування в різних сферах: для навчання учнів і студентів основних принципів програмування, наукових розрахунків, графічної візуалізації даних, опрацювання тривимірних зображень, адміністрування мереж та операційних систем, розширення можливостей готових програм, створення власних додатків тощо.



Цікаві факти з історії

Мова програмування **Python** була вперше описана в 1991 році нідерландським програмістом Гвідо ван Россумом (нар. у 1956 р., мал. 6.7), на той час співробітником нідерландського Національного дослідницького інституту математики та інформатики.

Назва мови виникла не від виду плазунів. Автор назвав її на честь героя популярного британського комедійного серіалу 70-х років XX ст. «Повітряний цирк Монті Пайтона». Назва **IDLE** середовища розробки проєктів, у якому можна створювати проєкти мовою **Python**, також походить від імені одного з героїв цього серіалу Ерика Айдла (англ. *Eric Idle*).



Мал. 6.7. Гвідо ван Россум

Середовище розробки проєктів


Для створення, редагування і налагодження проєктів мовою програмування **Python** будемо використовувати **середовище розробки проєктів IDLE 3.7** (англ. *Integrated Development and Learning Environment* – інтегроване середовище розробки та навчання). Завантажити на свій комп'ютер останню версію **Python** для встановлення середовища розробки **IDLE** можна безкоштовно з офіційного сайту **Python Software Foundation** (англ. *Python Software Foundation* – фонд програмного забезпечення **Python**) (python.org).

Середовище розробки проєктів **IDLE** складається з:

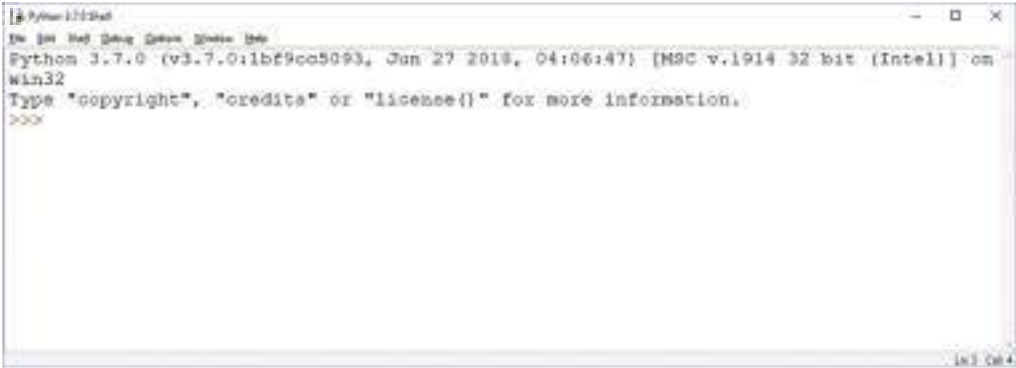
- **текстового редактора** для введення та редагування тексту проєкту;
- **інтерпретатора** для перекладу команд з мови програмування **Python** на мову команд, які може виконати процесор комп'ютера;
- **засобів налагодження проєкту** для пошуку в ньому помилок;
- **довідкової системи** та інших компонентів.

Це середовище розробки надає можливість використовувати велику бібліотеку **модулів** – готових фрагментів програмного коду, які можуть бути приєднані до нових проєктів, що спрощує та пришвидшує їх створення.

Відкрити вікно середовища **IDLE** можна:

- використовуючи команду **IDLE (Python 3.7)** меню **Пуск**;
- використовуючи значок  на **Робочому столі**.

У результаті відкривається вікно **Python 3.7.0 Shell** середовища розробки та виконання проєктів **IDLE** (мал. 6.8), у якому розміщено головне меню та робоче поле.



Мал. 6.8. Вікно **Python 3.7.0 Shell** середовища розробки та виконання проєктів **IDLE**

У робочому полі символи `>>>` позначають позицію, у якій можна вводити команди мовою програмування **Python**. Уведення команди завершується натисканням клавіші **Enter**, після чого команда виконується і результат її виконання відображається в наступному після команди рядку. Такий режим уведення і виконання команд називається **інтерактивним**.

Інтерактивний режим зручний для проведення обчислень, перевірки правильності написання команд і реалізації найпростіших проєктів. Але для розробки більш складних проєктів доцільно записувати команди проєкту в окремому файлі, що дасть змогу неодноразово виконувати проєкт, не вводячи повторно кожен команду.

Створити нове вікно для введення тексту проєкту можна командою **File** ⇒ **New File**. Після цього відкривається вікно текстового редактора. Ім'я вікна **Untitled** (англ. *untitled* – без назви) буде замінено на ім'я файлу, у якому текст проєкту збережеться на носії після виконання команди **File** ⇒ **Save**. У вікні збереження файлу потрібно вибрати папку для зберігання та ввести ім'я файлу. Файл зберігається з розширенням імені **py**. Під час повторного збереження раніше створеного проєкту ім'я файлу вводити не потрібно.

Команди, введені у вікні текстового редактора, не будуть одразу виконуватися, а натискання клавіші **Enter** переводитиме курсор на новий рядок, де можна вводити наступну команду.

Для виконання проєкту потрібно зберегти файл на носії, після чого виконати команду **Run** ⇒ **Run Module** або натиснути клавішу **F5**.

Інтерпретатор (англ. *interpreter* – перекладач) – комп'ютерна програма, що здійснює пооператорне (покомандне) перетворення в машинний код і виконання команд програми, яку записано деякою мовою програмування.


Після запуску проєкту на виконання розпочинає свою роботу **програма-інтерпретатор**. Ця програма по чергово аналізує команди проєкту на наявність синтаксичних помилок. Якщо команда не має помилок, то інтерпретатор перекладає її з мови **Python** на мову команд, які може виконати процесор комп'ютера, і передає перекладену команду на виконання процесору.

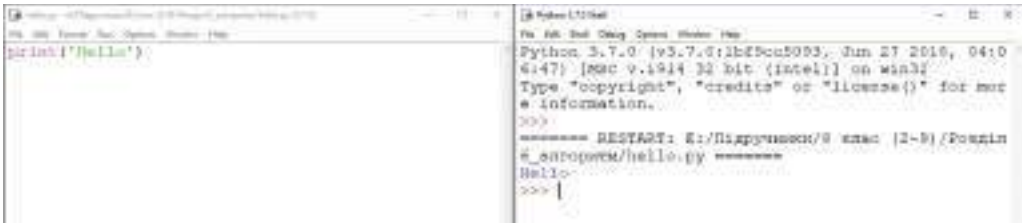
Після цього здійснюється аналіз і виконання наступної команди. За наявності синтаксичних помилок інтерпретатор перериває виконання проєкту і вказує на місце першої з них. Потрібно виправити помилку і знову запуснути проєкт на виконання.



Результати виконання проєкту та повідомлення про помилки відображаються у вікні середовища розробки та виконання проєктів **Python 3.7.0 Shell**.

Під час розробки проєкту вікно текстового редактора та вікно середовища розробки **Python 3.7.0 Shell** зручно розташовувати на екрані поруч (мал. 6.9). Для цього слід:

1. Зробити поточним вікно текстового редактора.
2. Натиснути сполучення клавіш  + ←.
3. Вибрати у правій половині екрана вікно середовища розробки **Python 3.7.0 Shell**.



Мал. 6.9. Вікно з текстом проєкту – ліворуч, вікно середовища розробки **Python 3.7.0 Shell** – праворуч

Створений раніше та збережений у файлі з розширенням імені **py** проєкт можна відкрити для редагування та виконання. Для відкриття файлу з текстом проєкту слід у вікні середовища **IDLE** виконати команду **File** ⇒ **Open** та вибрати потрібний файл.

Проекти з вікнами

Для створення мовою **Python** проєкту з вікнами потрібно приєднати до проєкту модуль **tkinter**, що входить до стандартної бібліотеки модулів **Python**. Модуль приєднується до проєкту виконанням команди: **from tkinter import *** (англ. *from* – з, *import* – імпортувати, ввозити).

Для створення вікна потрібно вибрати для нього ім'я та виконати команду: **<ім'я вікна> = Tk()**. Головному вікну проєкту часто надають ім'я *root* (англ. *root* – корінь). Далі в командах будемо використовувати саме це ім'я для вікна.

Команда створення вікна з іменем *root* повинна бути такою: **root = Tk()**.

Після виконання цієї команди відкривається вікно, вигляд якого наведено на малюнку 6.10. За замовчуванням вікно відкривається у верхньому лівому куті екрана, має розміри 200 × 200 пікселів (**піксель** – точка графічного зображення на екрані), рядок заголовка вікна містить напис *tk*, колір фону – світло-сірий.



Мал. 6.10. Вікно зі значеннями властивостей, установленими за замовчуванням

Значення всіх властивостей вікна можна змінити.

Для змінення напису в рядку заголовка вікна з іменем *root* потрібно виконати команду **root.title(<текст>)** (англ. *title* – заголовок, назва). Наприклад, команда **root.title('Мій перший проєкт')** виводить у рядок заголовка вікна напис *Мій перший проєкт*.

Змінення розмірів і положення вікна виконує команда **`root.geometry('<срядок параметрів>')`** (англ. *geometry* – геометрія). У рядку параметрів можна вказати: ширину та висоту вікна, між якими стоїть символ «**x**», відстань лівої межі вікна від лівої межі екрана і відстань верхньої межі вікна від верхньої межі екрана, перед якими стоять символи «**+**». Так команда **`root.geometry('600x400+200+100')`** установить ширину вікна 600 пікселів, висоту вікна 400 пікселів, відстань лівої межі вікна від лівої межі екрана – 200 пікселів, відстань верхньої межі вікна від верхньої межі екрана – 100 пікселів.

Зверніть увагу, у рядку параметрів, що визначає розміри та положення вікна, відсутні пропуски.

Команда **`root.geometry('400x250')`** установить ширину вікна – 400 пікселів і висоту вікна – 250 пікселів, а положення вікна не змінить. А якщо виконати команду **`root.geometry('+300+200')`**, то розміри вікна не зміняться, а зміняться лише відстані вікна від лівої та верхньої меж екрана.

Для змінення кольору фону вікна потрібно вказати назву кольору в команді **`root['bg']='<колір>'`** (англ. *background* – фон, тло). Наприклад, після виконання команди **`root['bg']='yellow'`** (англ. *yellow* – жовтий) колір фону вікна стане жовтим.

На малюнку 6.11 наведено вигляд вікна після виконання команд:

```
from tkinter import *
root=Tk()
root.title('Мій перший проєкт')
root.geometry('600x400+200+100')
root['bg']='yellow'
```



Мал. 6.11. Вигляд вікна після змінення напису в рядку заголовка, розмірів, положення та кольору фону



Працюємо з комп'ютером

Виконайте тренувальні завдання за посиланням <https://cutt.ly/hhdrubZ> або QR-кодом.



Найважливіше в цьому пункті

Комп'ютерна програма – це алгоритм, записаний спеціальною мовою та призначений для виконання комп'ютером.

У процесі своєї роботи програма може опрацьовувати **дані**.

Дані, які потрапляють до програми від певних пристроїв (наприклад, з клавіатури або від датчика температури), або від іншої програми, або з іншого джерела (наприклад, з текстового файлу), називають **вхідними (початковими) даними**. Деякі програми працюють без вхідних даних.

Під час виконання програми утворюються та опрацьовуються й інші дані, які називають **проміжними даними**.

Якщо метою виконання програми є отримання певних даних, то ці дані називають **вихідними (результуючими) даними**.

Мова, яка використовується для запису комп'ютерних програм, називається **мовою програмування**.

Кожна мова програмування має такі складові:



- **Алфавіт** – множина символів, з яких можна утворювати слова і речення цієї мови;
- **Словник** – набір слів, які використовуються в цій мові;
- **Синтаксис** – правила складання і запису мовних конструкцій: несловникових слів і речень;
- **Семантика** – установлене однозначне тлумачення мовних конструкцій і правил їх виконання.

Середовище розробки програм **Lazarus** складається з:

- **текстового редактора** для введення і редагування тексту проекту;
- **компілятора** для перекладу програми з мови програмування **Pascal** на мову команд, які може виконати процесор комп'ютера;
- **засобів налагодження програми** для пошуку в ній помилок;
- **довідкової системи**

та інших елементів керування.

Середовище розробки програм **IDLE** складається з:

- **текстового редактора** для введення та редагування тексту проєктів;
- **інтерпретатора** для перекладу команд з мови програмування **Python** на мову команд, які може виконати процесор комп'ютера;
- **засобів налагодження програми** для пошуку в ній помилок;
- **довідкової системи**

та інших компонентів.

Форма – це об'єкт, на якому під час створення проєкту можна розмістити різні компоненти (елементи керування), зокрема кнопки, поля, написи, меню, смуги прокручування та ін. Під час виконання проєкту форма відкривається як вікно прикладної програми, на якому будуть відображатися всі об'єкти, розміщені на формі.

Приклади деяких з властивостей форми:

- **Name** – визначає ім'я форми як об'єкта;
- **Caption** – визначає текст у рядку заголовка форми;
- **Color** – визначає колір фону форми;
- **Height** – визначає висоту форми в пікселях (**піксель** – це точка графічного зображення на екрані);
- **Width** – визначає ширину форми в пікселях;
- **Top** – визначає відстань (у пікселях) від верхньої межі форми до верхньої межі екрана;
- **Left** – визначає відстань (у пікселях) від лівої межі форми до лівої межі екрана.

У мові **Python** для створення проєкту з вікном потрібно приєднати до проєкту модуль **tkinter**, використовуючи команду **from tkinter import ***. Команда створення вікна з іменем *root* є такою: **root = Tk()**.

Для змінення напису в рядку заголовка вікна з іменем *root* потрібно виконати команду **root.title(' <текст>')**. Змінення розмірів і положення вікна виконує команда **root.geometry(' <рядок параметрів>')**. Для змінення кольору фону вікна потрібно вказати назву кольору в команді **root['bg']=' <колір>'**.




Дайте відповіді на запитання

- 1°. Що таке комп'ютерна програма?
- 2°. Які дані, що опрацьовує програма, називають вхідними, проміжними, вихідними?
- 3°. Що таке мова програмування?
- 4°. Які складові частини мови програмування? Опишіть кожну з них.
- 5°. Що таке синтаксична помилка?

- 6°. Для чого призначено середовище **Lazarus (IDLE)**?
- 7°. Які основні складові частини середовища **Lazarus (IDLE)** і яке їх призначення?
- 8°. Для чого призначено програму-компілятор (програму-інтерпретатор)? Які основні етапи її виконання?
- 9°. Що таке форма? Назвіть кілька її властивостей. Як змінити їх значення? Як створити проєкт з вікном мовою **Python** і встановити значення властивостей вікна?
- 10°. Як відкрити проєкт у середовищі **Lazarus (IDLE)**? Як зберегти проєкт? Як запустити проєкт на виконання?



Виконайте завдання

- 1°. Створіть проєкт, виконання якого відкриє вікно червоного кольору з розмірами 400 × 350 пікселів і змініть текст у рядку заголовка вікна на ваше прізвище. Збережіть проєкт у папці з іменем **Завдання 6.1.1**, створеній у вашій папці.
-  2°. Створіть проєкт, виконання якого відкриє вікно синього кольору з розмірами 200 × 450 пікселів і змініть текст у рядку заголовка вікна на ваше ім'я. Збережіть проєкт у папці з іменем **Завдання 6.1.2**, створеній у вашій папці.
- 3°. Створіть проєкт, виконання якого відкриє вікно синього кольору з розмірами 300 × 300 пікселів, ліва межа якого розміщена на відстані 300 пікселів від лівої межі екрана і верхня межа якого знаходиться на відстані 400 пікселів від верхньої межі екрана. Збережіть проєкт у папці з іменем **Завдання 6.1.3**, створеній у вашій папці.
-  4°. Створіть проєкт, виконання якого відкриє вікно сірого кольору з розмірами 200 × 100 пікселів, ліва межа якого розміщена на відстані 150 пікселів від лівої межі екрана і верхня межа якого знаходиться на відстані 100 пікселів від верхньої межі екрана. Збережіть проєкт у папці з іменем **Завдання 6.1.4**, створеній у вашій папці.

6.2. Події. Обробники подій



1. Що таке форма? Назвіть кілька її властивостей.
2. Як можна змінити значення властивостей форми?
3. Яке призначення програми-компілятора? Які основні етапи її роботи?
4. Як створити вікно проєкту та задати значення його властивостей?
5. Яке призначення програми-інтерпретатора? Які основні етапи її роботи?

Події та їх обробники

Під час виконання проєкту з об'єктами можуть відбуватися **події**.

Так, наприклад, з вікном можуть відбуватися такі події:

- **Click** (англ. *click* – клацання) – вибір будь-якої точки вікна;
- **DbiClick** (англ. *double* – подвійний) – установлення вказівника в довільну точку вікна і подвійне клацання лівої кнопки миші;
- **Close** (англ. *close* – закрити) – закривання вікна;
- **Create** (англ. *create* – створення) – створення (відкриття) вікна;
- **MouseMove** (англ. *mouse move* – переміщення миші) – переміщення вказівника у вікні виконання проєкту

та ін.



Події можуть відбуватися з клавіатурою. Такою подією, наприклад, може бути подія **KeyPress** (англ. *key press* – натиснення клавіші) – натиснення будь-якої клавіші на клавіатурі. Події можуть відбуватися й з іншими об'єктами: кнопка, напис, поле та ін. Ми їх вивчатимемо в наступних пунктах.

З будь-якою подією, яка може відбутися з вікном, можна пов'язати **фрагмент проєкту**, який буде виконуватися одразу після настання цієї події. Такий фрагмент проєкту називається **обробник події**.

Фрагмент проєкту, який пов'язаний з певним об'єктом, називається **методом цього об'єкта**.



Для тих, хто вивчає Pascal

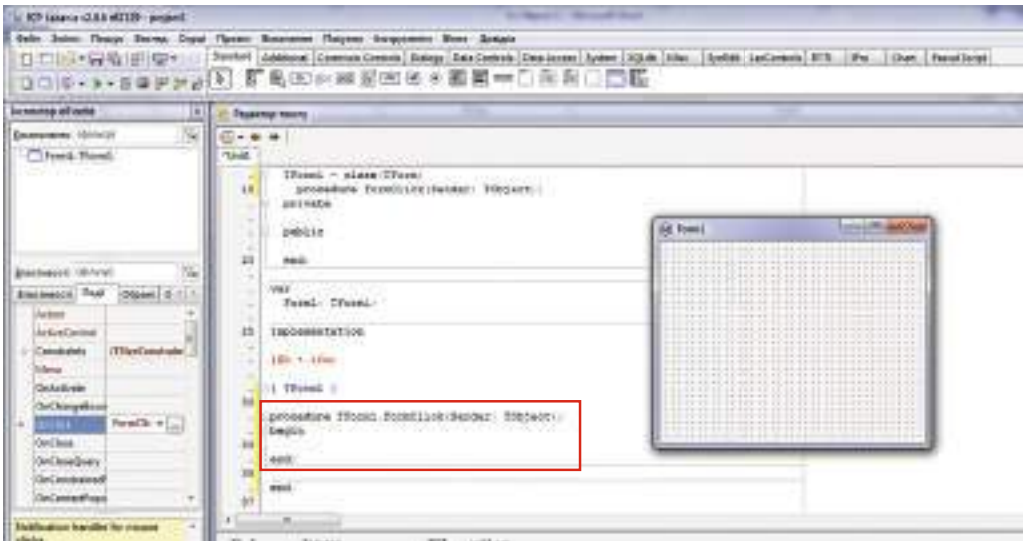
Середовище **Lazarus** за замовчуванням створює обробник події як **процедуру**. **Процедура** – це окремо записана частина проєкту, яка має своє ім'я і яку можна за цим іменем запуснути на виконання.

Виконання процедури також є подією. Якщо процедура виконується в результаті настання певної події з вікном, наприклад події **Click**, то подію самого виконання цієї процедури називають **OnClick** (англ. *on click* – на клацання).

Наприклад, якщо потрібно пов'язати деяку процедуру з подією **Click** для вікна, то слід:

1. Вибрати форму.
2. Відкрити вкладку **Події** у вікні **Інспектор об'єктів**.
3. Вибрати подію **OnClick**. Поле справа від тексту **OnClick** стане поточним, і в ньому з'явиться курсор.
4. Клацнути двічі у цьому поточному полі.

Далі в полі **OnClick** на вкладці **Події** з'являється текст **FormClick** (англ. *form click* – клацання форми), а у вікні редактора тексту проєкту на вкладці **Unit1** (англ. *unit* – блок, частина цілого) з'являється заготовка для тексту процедури з іменем **TForm1.FormClick**. Команди цієї процедури будуть виконуватися під час настання події **Click** з вікном (мал. 6.12), і у їх виконанні полягатиме подія **OnClick**.



Мал. 6.12. Заготовка для тексту процедури **TForm1.FormClick**

Заготовка процедури розпочинається з ключового слова **procedure** (англ. *procedure* – процедура) і завершується горизонтальною лінією.

Команди процедури записуються між двома словами: **begin** (англ. *begin* – початок) і **end** (англ. *end* – кінець). *Звертаємо вашу увагу, що begin і end – це не команди, ці слова відіграють роль відповідно відкриваючої та закриваючої дужок, що визначають початок і закінчення процедури. Ці слова часто називають операторними дужками.*

Після слова **end** у кінці процедури повинна стояти крапка з комою (;). Кожна команда процедури також повинна закінчуватися крапкою з комою (;). *Зверніть увагу, що в кінці заготовки тексту всього проєкту стоїть ще одна операторна дужка end, після якої поставлено крапку.*

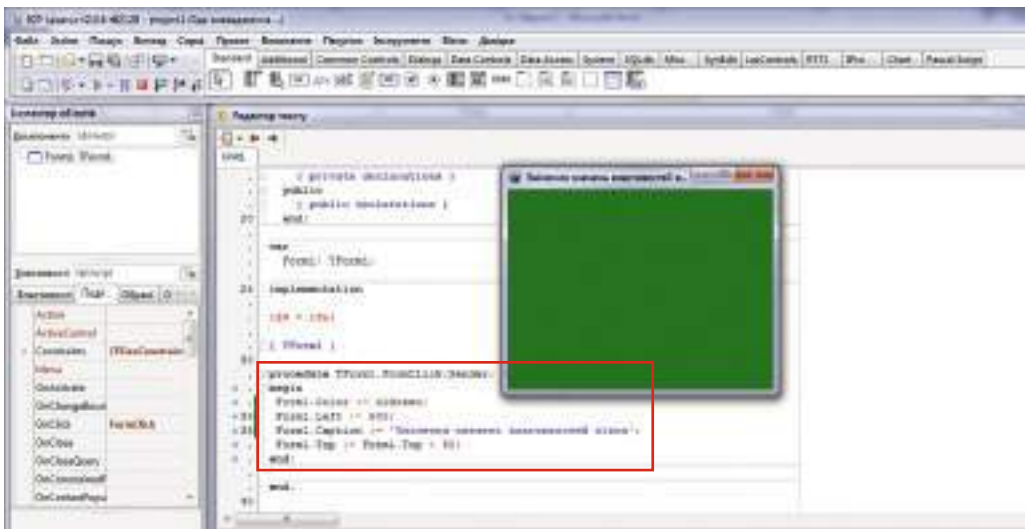
Якщо потрібно, щоб у результаті виконання процедури змінилося значення певної властивості вікна (а в подальшому й інших об'єктів), процедура повинна містити команду **присвоювання**. Її загальний вигляд такий:

<ім'я об'єкта>.<назва властивості> := <значення або вираз>;

Знак **:=** називається **знаком присвоювання**. Він вводиться з клавіатури послідовним уведенням двох символів: **двокрапка (:)** і **дорівнює (=)** без пропуску між ними.

Наприклад, виконання команди присвоювання **Form1.Color := clGreen** установить зелений колір фону вікна. Виконання команди присвоювання **Form1.Left := 600** установить відступ 600 пікселів лівої межі вікна від лівої межі екрана. Командою присвоювання **Form1.Caption := 'Змінення значень властивостей вікна'** можна вивести вказаний у лапках текст у рядку заголовка вікна. А виконання команди **Form1.Top := Form1.Top + 50** збільшить поточний відступ верхньої межі вікна від верхньої межі екрана на 50 пікселів. Під час виконання останньої команди спочатку до поточного значення властивості **Top** вікна буде додано число 50, після чого отримане число стане новим значенням цієї властивості вікна.

На малюнку 6.13 наведено приклад процедури **TForm1.FormClick**, що складається з вищеописаних команд, у результаті виконання якої зміняться значення властивостей **Caption, Color, Left** і **Top** вікна, і результат її виконання.



Мал. 6.13. Процедура змінення значення властивостей **Caption, Color, Left** і **Top** вікна та результат її виконання



Після запуску проекту на виконання потрібно дочекатися відкриття вікна виконання проекту і вибрати будь-яку його точку. Тим самим відбудеться подія **Click** для вікна і як наслідок відбудеться подія **OnClick**, яка полягатиме у виконанні процедури **TForm1.FormClick** – обробника події **Click** для вікна.

Звертаємо вашу увагу, що в одному проекті можуть бути створені процедури опрацювання кількох різних подій з одним і тим самим або з різними об'єктами.



Для тих, хто вивчає Python

У мові **Python** обробник події для вікна створюють як окрему **підпрограму** та пов'язують її з вікном, використовуючи **метод bind** (англ. *bind* – в'язати, пов'язувати).

Підпрограма – це окремо виділена частина проекту, яка має своє ім'я та яку можна за цим іменем запустити на виконання. У **Python** розрізняють підпрограми двох видів: **процедури** і **функції**.

Процедура – це підпрограма, результатом виконання якої є створення компонента (вікна, кнопки та ін.) або змінення значень властивостей існуючого, виведення певного тексту, створення зображення тощо. Зазвичай обробники подій є процедурами.

Функція – це підпрограма, результатом виконання якої є одне або кілька значень: числа, рядки тексту та ін.

Метод – це процедура або функція, що пов'язана з певним об'єктом.

У загальному вигляді команда пов'язування обробника події з вікном, що має ім'я *root*, має такий вигляд:

```
root.bind('<подія>', <ім'я_обробника_події>)
```

Позначення в мові **Python** деяких подій, які можуть відбутися з вікном, наведено в таблиці 6.1.

Таблиця 6.1

Позначення подій у мові Python

Подія	Позначення в мові Python
Click – клацання лівої кнопки миші	<Button-1> або <1>
DbClick – подвійне клацання лівої кнопки миші	<Double-Button-1> або <Double-1>
MouseMove – рух вказівника по вікну	<Motion> (англ. <i>motion</i> – рух)
KeyPress – натискання будь-якої клавіші на клавіатурі	<KeyPress>
Create – створення вікна	<Expose> (англ. <i>expose</i> – розкривати)
Close – закривання вікна	<FocusOut> (англ. <i>focus out</i> – вихід фокуса)

Наприклад, якщо обробнику події **Click** (клацання лівої кнопки миші) надати ім'я *click*, то команда пов'язування цієї процедури з вікном з іменем *root* має такий вигляд:

```
root.bind('<1>', click)
```

Загальний вигляд обробника події такий:

```
def <ім'я_обробника_події> (event):
```

```
    <команди, які будуть виконуватися під час настання події>
```

(англ. *define* – визначати, *event* – подія).

Перший рядок обробника події називають **рядком заголовка**. Наприклад, для обробника події з іменем *click* рядок заголовка буде таким:

```
def click (event):
```

Команди обробника події потрібно записувати з відступом праворуч від лівої межі вікна редактора коду. Відступ утворюється автоматично, якщо після символу «:» натиснути клавішу **Enter**, або його можна зробити самостійно, натиснувши один раз клавішу **Tab**.

Якщо потрібно, щоб у результаті настання події змінилися значення певних властивостей вікна, обробник події повинен містити команди змінення значень цих властивостей. Це можуть бути команди застосування до вікна деяких методів, наприклад **geometry()**, **title()**, або команда **присвоювання**. Загальний вигляд команди присвоювання такий:

<ім'я об'єкта>[<властивість>] = <значення>

Знак = називають **знаком присвоювання**.

Наприклад, виконання команди присвоювання **root['bg'] = 'green'** установить зелений колір вікна з іменем *root*. Застосувавши метод **root.title('Змінення значень властивостей вікна')**, можна вивести вказаний у лапках текст у рядку заголовка вікна. А для того щоб розміри вікна стали **400 × 200** пікселів, відстань від лівої межі екрана – **500** пікселів, від верхньої – **70** пікселів, потрібно застосувати метод **root.geometry('400x200+500+70')**.

Обробник події записують перед його використанням, найчастіше на початку тексту проекту. Запис команд після обробника події повинен розпочинатися від лівої межі вікна. Для переміщення курсора в цю позицію потрібно використати клавішу **Backspace**. Так, після обробника події слід записати команди створення вікна та пов'язування з ним обробника події:

root = Tk()

root.bind('<1>', click)

На малюнку 6.14 наведено приклад процедури *click* – обробника події **Click** для вікна, що складається з вищеописаних команд, під час виконання якої зміняться значення властивості **bg**, текст заголовка, розміри і положення вікна, і результат її виконання.



Мал. 6.14. Проект і результат його виконання після настання події **Click** для вікна

Іноді треба під час опрацювання події встановити нове значення властивості об'єкта, що обчислюється на основі поточного значення цієї самої властивості. Наприклад, потрібно, щоб після клацання лівої кнопки миші вікно опустилося на 50 пікселів, тобто слід збільшити відстань вікна від верхньої межі екрана на 50 пікселів. У такому разі треба отримати поточне значення відстані, змінити його на 50 і запам'ятати результат в окремій змінній, щоб установити нове положення вікна.



Для отримання поточних значень розмірів і положення вікна використовують такі методи вікна:

- **winfo_width()** – (англ. *window information* – інформація про вікно) ширина вікна;
- **winfo_height()** – висота вікна;
- **winfo_x()** – відстань лівої межі вікна від лівого краю екрана;
- **winfo_y()** – відстань верхньої межі вікна від верхнього краю екрана.

Наприклад, щоб дізнатися поточну відстань вікна з іменем *root* від верхньої межі екрана та збільшити її на 50, потрібно виконати команду:

t = root.winfo_y() + 50

Щоб установити нове значення, потрібно в рядок визначення положення та розмірів вікна методу **geometry()** замість числа підставити значення змінної. Для цього на місці, де потрібно вставити значення змінної, ввести символи **{}** і до рядка застосувати метод підстановки **format()** з іменем змінної, відокремивши його крапкою:

root.geometry('300x200+500+{}'.format(t))

Якщо потрібно підставити в рядок значення кількох змінних, то на всіх відповідних місцях потрібно ввести символи **{}**, а у методі **format()** перерахувати імена змінних через кому.

Наприклад, для того щоб після настання події *кляцання лівої кнопки миші* вікно опустилося на 50 пікселів униз, а його ширина збільшилася на 40 пікселів, можна виконати проєкт, наведений на малюнку 6.15.

```
from tkinter import *
def click (event):
    t = root.winfo_y()+50
    w = root.winfo_width()+40
    root.geometry('{}x{}+{}+{}'.format(w, t))
root = Tk()
root.geometry('480x320+200+200')
root.bind('<1>', click)
```

Мал. 6.15. Проєкт для змінення положення та розмірів вікна залежно від попередніх значень положення та розмірів

Після запуску проєкту на виконання потрібно дочекатися відкриття вікна і вібрати будь-яку його точку. Тим самим відбудеться подія **Click** для вікна та буде виконана процедура – обробник події, результатом виконання якої буде змінення положення та розмірів вікна.

Звертаємо вашу увагу, що в одному проєкті можуть бути створені процедури опрацювання кількох різних подій з одним і тим самим або з різними об'єктами.

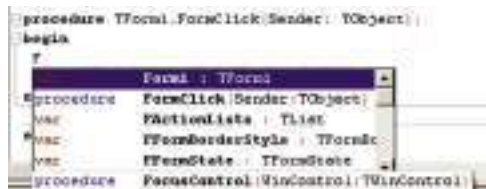
Засоби створення і редагування проєкту



Для тих, хто вивчає Pascal

Середовище **Lazarus** має зручні засоби допомоги під час написання тексту проєкту:

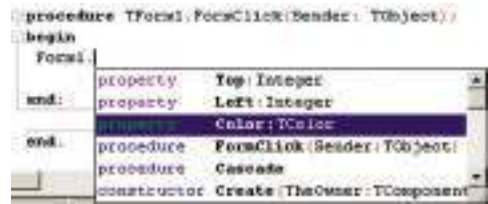
- після введення першої або кількох перших літер імені об'єкта можна натиснути сполучення клавіш **Ctrl+Пропуск**, і відкриється список, з якого можна вибрати потрібне ім'я об'єкта (мал. 6.16);
- після введення крапки, що розділяє ім'я об'єкта і назву його властивості, автоматично відкривається вікно зі



Мал. 6.16. Список для вибору імені об'єкта

списком, який містить назви властивостей даного об'єкта (мал. 6.17). Далі назву властивості можна не вводити з клавіатури, а знайти її в списку і натиснути клавішу **Enter** або двічі клацнути на відповідному рядку.

Можна після крапки ввести кілька перших літер з назви властивості. Тоді цей список скорочуватиметься та в ньому залишатимуться лише ті назви, які починаються з уведених літер. Такий спосіб пришвидшує введення назв властивостей об'єктів, а також запобігає допущенню синтаксичних помилок, особливо тими, хто не знає англійської мови.



Мал. 6.17. Список для вибору властивостей об'єкта

- У багатьох випадках система автоматично розставляє в потрібних місцях тексту проекту операторні дужки, знак присвоювання, крапку з комою в кінці команди тощо.

Зазначимо також, що система не розрізняє великі та малі літери. Однак, якщо слово утворено з кількох слів, прийнято (і зручно для кращого розуміння) писати першу літеру кожного слова великою.

Ви вже знаєте, що на початку виконання проекту програма-компілятор перевіряє текст проекту на наявність синтаксичних помилок. Якщо такі помилки наявні, то виконання проекту переривається та у вікні **Повідомлення**, яке відкривається під вікном **Редактора тексту**, з'являються повідомлення-підказки про номери рядків і номери позицій у них, де допущено помилки, а також коментарі до кожної із цих помилок. Крім того, кольором виділяється фон першого з тих рядків тексту проекту, що містять помилки, або рядка, наступного за рядком з помилкою.

На малюнку 6.18 виділено кольором рядок 34, бо саме в ньому містяться перші синтаксичні помилки проекту – неправильно написано слова **Color** і **Green**. Крім того, у кінці команди, що знаходиться в рядку 35, пропущено крапку з комою. Для такої помилки у вікні **Повідомлення** з'являється повідомлення про неправильний символ, з якого починається наступний, 36-й рядок. І якщо б саме ця помилка була в проекті першою, то кольором було б виділено саме наступний, 36-й рядок.

Якщо програма-компілятор знайшла помилки, потрібно виправити їх у тексті проекту і знову запустити проект на виконання.



Мал. 6.18. Позначення синтаксичних помилок у середовищі **Lazarus**



Для тих, хто вивчає Python

Середовище **IDLE** має деякі допоміжні засоби для написання тексту проєкту:

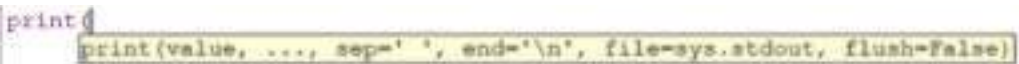
- після введення першої або кількох перших літер ключового слова можна натиснути сполучення клавіш **Ctrl+Пропуск**, і відкриється список, з якого можна вибрати потрібне ключове слово (мал. 6.19);
- після введення імені функції та відкриваючої дужки автоматично виводиться підказка, які аргументи можуть бути в цієї функції (мал. 6.20);
- система автоматично розставляє в потрібних місцях тексту проєкту відступи після рядків, що завершуються символом двокрапка (:)

та ін.

Система вважає різними великі та малі літери. Якщо слово утворено з кількох слів, прийнято (і зручно для кращого розуміння) писати першу літеру кожного слова великою.



Мал. 6.19. Список для вибору ключового слова

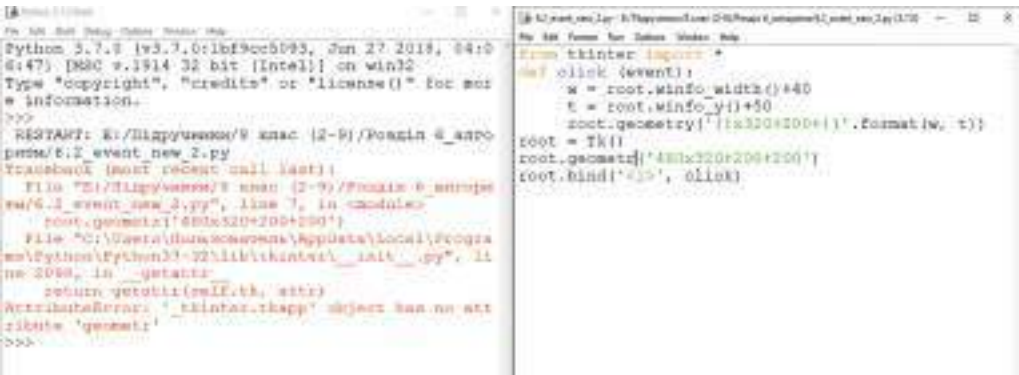


Мал. 6.20. Підказка з можливими аргументами функції

Ви вже знаєте, що в процесі виконання проєкту програма-інтерпретатор перевіряє кожен рядок тексту проєкту на наявність синтаксичних помилок. Якщо такі помилки присутні, то виконання проєкту переривається й у вікні середовища **IDLE** з'являються повідомлення-підказки про номер рядка, де допущено помилку, а також коментар до цієї помилки.

Так, на малюнку 6.21 виведено червоним кольором повідомлення про помилку в рядку 7 та її опис. В останньому рядку повідомлення про помилку вказано ім'я методу *'geometr'*, який викликав зупинку виконання проєкту.

Потрібно виправити в тексті проєкту знайдені синтаксичні помилки і знову запустити проєкт на виконання.



Мал. 6.21. Повідомлення про синтаксичні помилки в середовищі **IDLE**

Вікно повідомлень і його використання

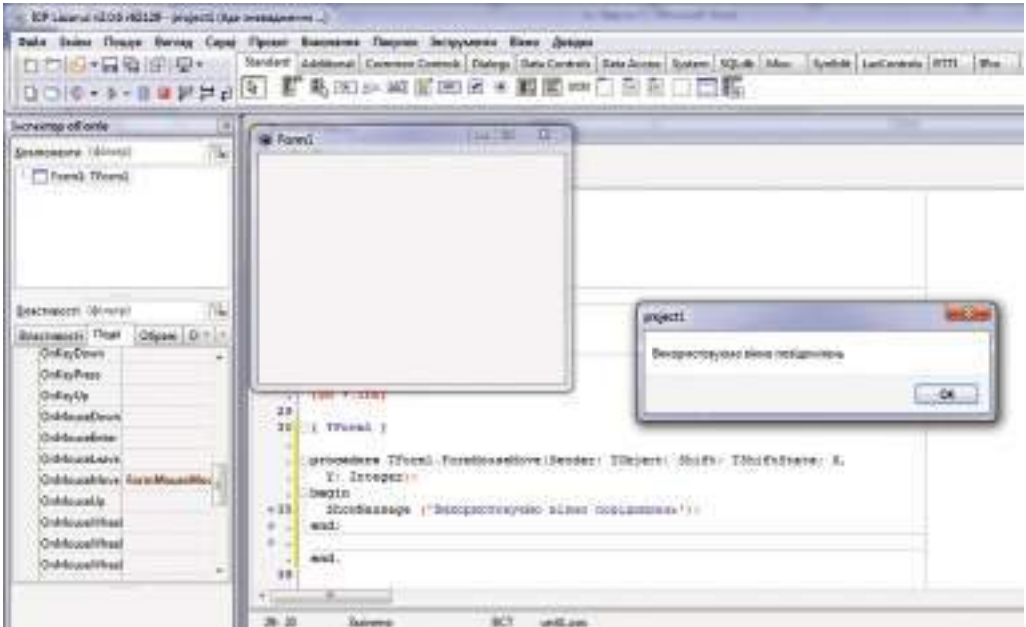
Для виведення на екран у ході виконання проекту текстових повідомлень можна використати спеціальне вікно – **вікно повідомлень**.



Для тих, хто вивчає Pascal

Для відкриття такого вікна і виведення в ньому потрібного текстового повідомлення використовується команда **ShowMessage** ('<текст>') (англ. *show message* – показувати повідомлення).

На малюнку 6.22 наведено приклад процедури – обробника події **MouseMove** для форми, яка містить команду відкриття вікна повідомлень і виведення в ньому повідомлення.



Мал. 6.22. Проект з використанням вікна повідомлень




Після ознайомлення з повідомленням потрібно закрити це вікно, вибравши кнопку **OK** або кнопку закриття вікна. Тільки після цього можна закрити вікно виконання проекту.

*Звертаємо вашу увагу, що в цьому прикладі використано подію **MouseMove** для вікна. Цікавим є те, що в цьому випадку вікно виконання проекту не вдасться закрити, використовуючи мишу. Адже, як тільки вказівник буде переміщуватися по вікну, наступатиме подія **MouseMove** і як результат – подія **OnMouseMove**, тобто запускатиметься на виконання обробник події **MouseMove** – процедура **TForm1.FormMouseMove**, яка знову відкриватиме вікно повідомлень. Тому закінчити виконання цього проекту можна, вибравши кнопку **Закінчити** на панелі інструментів або натиснувши сполучення клавіш **Alt+F4**.*



Для тих, хто вивчає Python

У мові **Python** можна створити вікна повідомлень кількох видів. Для цього призначено команди:

- **showinfo**(‘<текст у рядку заголовка>’, ‘<текст повідомлення>’) (англ. *show info* – показувати інформацію) – створення інформаційного вікна зі значком  і текстовим повідомленням;
- **showwarning**(‘<текст у рядку заголовка>’, ‘<текст повідомлення>’) (англ. *warning* – увага) – створення вікна попередження зі значком  і текстом попередження;
- **showerror**(‘<текст у рядку заголовка>’, ‘<текст повідомлення>’) (англ. *error* – помилка) – створення вікна оповіщення про помилку зі значком  і текстом повідомлення про помилку

та інші.

Для відкриття вікна повідомлення із заданим текстом заголовка й виведення в ньому заданого тексту повідомлення потрібно імпортувати до проекту модуль **tkinter.messagebox** і використати команду відповідно до призначення вікна.

Наприклад, потрібно під час руху вказівника у вікні виконання проекту відкрити вікно повідомлень з текстом у рядку заголовка *Події* та текстом повідомлення *Я використовую вікно повідомлень*. Приклад тексту обробника події **MouseMove** для вікна проекту з командою виведення вікна повідомлень і результат виконання відповідного проекту наведено на малюнку 6.23.



Мал. 6.23. Використання вікна повідомлень

Після ознайомлення з повідомленням потрібно закрити це вікно, вибравши кнопку **OK** або кнопку закриття вікна. Тільки після цього можна закрити вікно виконання проекту.

Звертаємо вашу увагу, що в цьому прикладі використано подію **MouseMove** (позначення цієї події в мові **Python** – **<Motion>**). Цікавим є те, що в цьому разі, якщо вікно повідомлення відкривається поверху вікна виконання проекту, проект буде складно завершити, використовуючи мишу. Адже, як тільки вказівник буде переміщуватися по вікну проекту, наступатиме подія **MouseMove** і як результат запускатиметься на виконання обробник цієї події, який знову відкриватиме вікно повідомлень. Тому закінчити виконання проекту можна, натиснувши сполучення клавіш **Alt+F4**. Це сполучення клавіш можна використовувати для закриття вікна виконання будь-якого проекту.



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/hhdriEE> або QR-кодом.

Найважливіше в цьому пункті

З будь-якою подією, яка може відбутися з вікном, можна пов'язати фрагмент проекту, який буде виконуватися одразу після настання цієї події. Такий фрагмент проекту називається **обробник події**. Обробник події, який пов'язаний з певним об'єктом, називається **методом цього об'єкта**.



Для тих, хто вивчає Pascal

Середовище **Lazarus** за замовчуванням оформлює обробник події як процедуру. **Процедура** – це окремо описана частина проекту, яка має своє ім'я і яку можна за цим іменем запустити на виконання.

Команди процедури записуються між двома словами: **begin** і **end**. *Звертаємо вашу увагу, що **begin** і **end** – це не команди, ці слова відіграють роль відповідно відкриваючої та закриваючої дужок, які визначають початок і закінчення процедури.* Ці слова часто називають **операторними дужками**.

Після слова **end** у кінці процедури повинна стояти крапка з комою (;). Кожна команда процедури також має закінчуватися крапкою з комою (;).

Якщо потрібно, щоб у результаті виконання процедури змінилося значення певної властивості форми, процедура повинна містити команду **присвоювання**. Її загальний вигляд такий:

<ім'я об'єкта>.<назва властивості> := <значення або вираз>;

Знак **:=** називається **знаком присвоювання**.

Для відкриття в ході виконання проекту вікна повідомлень і виведення в ньому потрібного текстового повідомлення використовують команду:

ShowMessage ('<текст>');



Для тих, хто вивчає Python

У мові **Python** обробник події для вікна створюють як окрему **процедуру** та пов'язують її з вікном, використовуючи метод **bind()**.

Процедура – це окремо виділена частина програми, яка має своє ім'я та яку можна за цим іменем запустити на виконання.

У загальному вигляді команда пов'язування обробника події з вікном, що має ім'я **root**, має такий вигляд:

root.bind('<подія>', <ім'я_обробника_події>)

Позначення подій у мові Python

Подія	Позначення в мові Python
Click – клацання лівої кнопки миші	<Button-1> або <1>
DbClick – подвійне клацання лівої кнопки миші	<Double-Button-1> або <Double-1>
MouseMove – рух вказівника по вікну	<Motion> (англ. <i>motion</i> – рух)
KeyPress – натискання будь-якої клавіші на клавіатурі	<KeyPress>
Create – створення вікна	<Expose> (англ. <i>expose</i> – розкривати)
Close – закривання вікна	<FocusOut> (англ. <i>focus out</i> – вихід фокуса)



Обробник події у загальному вигляді записують так:

```
def <ім'я_обробника_події> (event):
```

<команди, які будуть виконуватися під час настання події>

Якщо потрібно, щоб у результаті настання події змінилося значення певної властивості вікна, обробник події повинен містити команду змінення значення цієї властивості – команду присвоювання чи відповідний метод.




Для отримання поточного значення розмірів і положення вікна використовують методи:

- **winfo_width()** – ширина вікна;
- **winfo_height()** – висота вікна;
- **winfo_x()** – відстань лівої межі вікна від лівого краю екрана;
- **winfo_y()** – відстань верхньої межі вікна від верхнього краю екрана.

Щоб установити нове значення властивості, потрібно в рядок визначення розмірів і положення вікна функції **geometry** замість числа підставити значення змінної. Для цього на місці, де потрібно вставити значення змінної, увести символи {}, а після рядка після крапки вказати метод підстановки **format** з іменем змінної. Наприклад, **root.geometry('300x200+500+{}'.format(t))**

Середовище **IDLE** має деякі допоміжні засоби для прискорення написання тексту проекту. Для системи великі та малі літери вважаються різними.

Для виведення на екран у ході виконання проекту текстових повідомлень можна використати спеціальне вікно – **вікно повідомлень**. У мові **Python** можна створити вікна повідомлень кількох видів. Для цього призначено команди:

- **showinfo('<текст у рядку заголовка>', '<текст повідомлення>')** – створення інформаційного вікна зі значком  і текстовим повідомленням;
- **showwarning('<текст у рядку заголовка>', '<текст повідомлення>')** – створення вікна попередження зі значком  і текстом попередження;
- **showerror('<текст у рядку заголовка>', '<текст повідомлення>')** – створення вікна оповіщення про помилку зі значком  і текстом повідомлення про помилку та ін.

Для відкриття вікна повідомлень із заданим текстом заголовка та виведення в ньому заданого тексту повідомлення потрібно імпортувати до проекту модуль **tkinter.messagebox** і використати команду відповідно до призначення вікна.



Дайте відповіді на запитання

- 1°. Які події можуть відбуватися з вікном?
- 2°. Який зв'язок між подіями **Click** і **OnClick**?
- 3°. Що називають обробником події?
- 4°. Як пов'язати з подією її обробник?
- 5°. Що називають методом об'єкта?
- 6°. Що називають процедурою?
- 7°. Що називають операторними дужками в **Pascal**? Для чого вони використовуються?
- 8°. Як оформити обробник події в **Python**?
- 9°. Якими командами можна змінити значення властивостей об'єкта? Який їх загальний вигляд? Наведіть приклади таких команд.
- 10°. У яких випадках виникають повідомлення про синтаксичні помилки? Що потрібно робити в таких випадках?
- 11°. Якою командою можна відкрити вікно повідомлень?



Виконайте завдання

- 1°. Запишіть у зошиті команду, яка:
- установить висоту вікна 200 пікселів;
 - збільшить висоту вікна на 200 пікселів;
 - зменшить ширину вікна на 100 пікселів;
 - установить відступ вікна від лівої межі екрана 100 пікселів;
 - перемістить вікно на 100 пікселів управо;
 - перемістить вікно на 100 пікселів уліво;
 - перемістить вікно на 200 пікселів угору;
 - перемістить вікно на 100 пікселів униз.



- 2°. Запишіть у зошиті команду, яка:
- установить ширину вікна 150 пікселів;
 - збільшить ширину вікна на 150 пікселів;
 - зменшить висоту вікна на 100 пікселів;
 - установить відступ вікна від верхньої межі екрана 100 пікселів;
 - перемістить вікно на 200 пікселів управо;
 - перемістить вікно на 200 пікселів уліво;
 - перемістить вікно на 100 пікселів угору;
 - перемістить вікно на 200 пікселів униз.

- 3°. Створіть проєкт, у якому в результаті настання події **Click** для вікна встановляться розміри вікна 400 × 300 пікселів, синій колір фону і зміниться текст заголовка на назву вашого населеного пункту. Збережіть проєкт у папці з іменем **Завдання 6.2.3**, створеній у вашій папці.



- 4°. Створіть проєкт, у якому в результаті настання події **KeyPress** встановляться розміри вікна 300 × 200 пікселів, червоний колір фону і зміниться текст заголовка на ваше прізвище. Збережіть проєкт у папці з іменем **Завдання 6.2.4**, створеній у вашій папці.

- 5°. Створіть проєкт, у якому в результаті настання події **DbClick** для вікна зміниться колір фону вікна на вибраний вами, текст заголовка на вибраний вами і збільшиться на 200 пікселів відступ верхньої межі вікна від верхньої межі екрана. Збережіть проєкт у папці з іменем **Завдання 6.2.5**, створеній у вашій папці.



- 6°. Створіть проєкт, у якому в результаті настання події **Create (<Expose>)** зміняться розміри вікна, колір фону і зменшиться на 100 пікселів відступ його лівої межі від лівої межі екрана. Збережіть проєкт у папці з іменем **Завдання 6.2.6**, створеній у вашій папці.

- 7°. Створіть проєкт, у якому в результаті настання події **Click** для вікна встановляться розміри вікна 500 × 500 пікселів, коричневий колір фону і зміниться текст заголовка на *Проект з двома процедурами*, а в результаті настання події **MouseMove** відкриється вікно повідомлень з текстом *Я – громадянин України!*. Збережіть проєкт у папці з іменем **Завдання 6.2.7**, створеній у вашій папці.

- 8°. Створіть проєкт, у якому в результаті настання події **Close (<FocusOut>)** для вікна відкриється вікно повідомлень з текстом *Виконання проєкту закінчено! До побачення!*. Збережіть проєкт у папці з іменем **Завдання 6.2.8**, створеній у вашій папці.

- 9°. Створіть проєкт, у якому в результаті настання події **Click** для вікна відкриється вікно повідомлень, у яке буде виведено текст заголовка вікна. Збережіть проєкт у папці з іменем **Завдання 6.2.9**, створеній у вашій папці.



ПРАКТИЧНА РОБОТА № 8

«Проекти, що змінюють значення властивостей вікна та використовують вікно повідомлень»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

- Створіть проєкт, під час виконання якого буде відкриватися вікно з такими значеннями властивостей:
 - колір фону – *сірий*;
 - ширина – *200 пікселів*;
 - висота – *100 пікселів*;
 - відступ лівої межі – *150 пікселів*;
 - відступ верхньої межі – *100 пікселів*;
 - текст у рядку заголовка – *Практична робота*.
- Збережіть проєкт у папці з іменем **Практична 8**, створеній у вашій папці.
- Додайте до проєкту обробник події **Click** для вікна, виконання якого установить червоний колір фону вікна, відступ верхньої межі вікна 200 пікселів від верхньої межі екрана, збільшить його ширину на 300 пікселів, зменшить на 50 пікселів відступ лівої межі вікна від лівої межі екрана, відкриє вікно повідомлень з текстом *Ми вивчаємо програмування!*
- Збережіть проєкт із тим самим іменем у тому самому місці.
- Додайте до проєкту обробник події **KeyPress**, виконання якого змінить колір фону вікна на синій, збільшить на 300 пікселів відступ верхньої межі вікна від верхньої межі екрана, збільшить його ширину на 200 пікселів.
- Збережіть проєкт із тим самим іменем у тому самому місці.

6.3. Кнопка




- Які властивості вікна ви знаєте? Як змінити значення його властивостей до початку виконання проєкту й під час його виконання?
- Які події можуть відбуватися з вікном? У чому полягає кожна з них?
- Що таке *обробник події*? Як його створити? Як його виконати?

Властивості кнопки

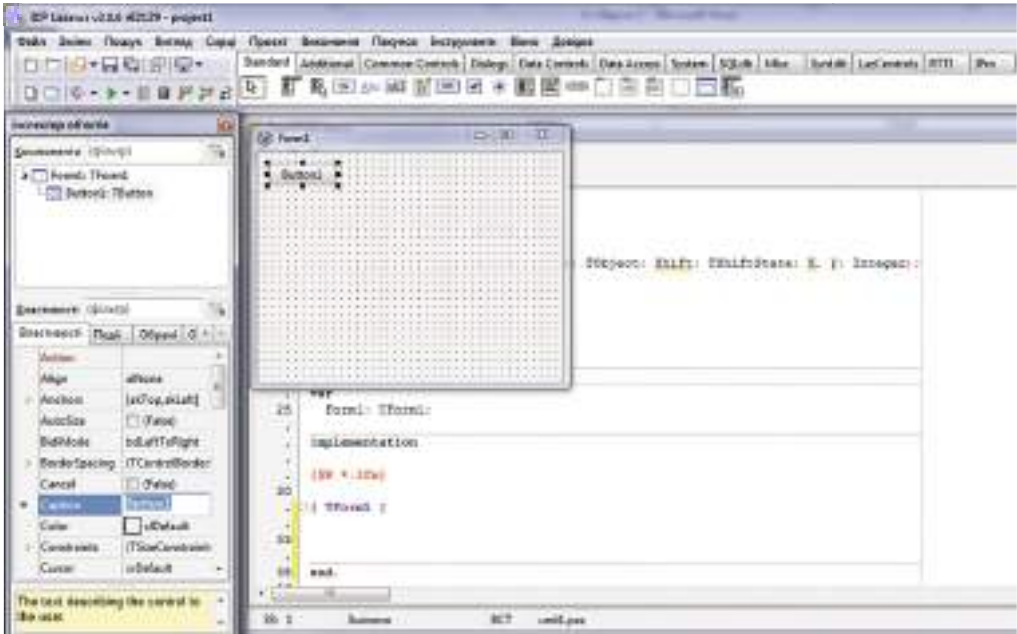
Одним з компонентів (елементів керування), який можна використати в проєктах, є **кнопка**.



Для тих, хто вивчає Pascal

Щоб розмістити кнопку на формі, потрібно підвести вказівник до її зображення  на вкладці **Standard** (англ. *standard* – стандарт) вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші. Після цього кнопка з'явиться у верхній лівій частині форми (мал. 6.24), а її ім'я додасться до списку компонентів проєкту в полі **Компоненти** вікна **Інспектор об'єктів**. За потреби кнопку можна перетягнути в будь-яке інше місце форми.

Якщо вибрати кнопку на формі або вибрати її ім'я в полі **Компоненти** вікна **Інспектор об'єктів**, то на вкладці **Властивості** цього вікна відобразиться список властивостей кнопки та їх значень (мал. 6.24). Як і для форми, значення деяких властивостей кнопки встановлені за замовчуванням і їх можна змінювати. Для



Мал. 6.24. Кнопка на формі

деяких властивостей їх значення потрібно вводити у відповідні поля, для деяких – можна вибирати зі списку, для деяких – можна змінювати перетягуванням кнопки по формі або зміненням її розмірів, використовуючи маркери на її межах.

У кнопки, як і у форми, є властивості **Name**, **Caption**, **Height**, **Width**, **Top**, **Left**, **Color**.

Значення властивості **Name** визначає ім'я кнопки як об'єкта. Значення властивості **Caption** визначає текст на кнопці. Значення властивостей **Height** і **Width** визначають висоту й ширину кнопки. Значення властивостей **Top** і **Left** визначають відстані верхньої та лівої меж кнопки від відповідних меж форми, на якій вона розміщена, а не від відповідних меж екрана, як для форми.

Значення всіх цих властивостей кнопки можна змінити на вкладці **Властивості** вікна **Інспектор об'єктів** до запуску проекту на виконання.

Значення всіх цих властивостей також можна змінити командами присвоювання під час виконання проекту. Наприклад, **Button1.Width := 100**, **Button1.Left := Button1.Left + 200**, **Button1.Caption := 'Виконати'**.

Значення властивості **Color** для кнопки, яке встановлюється за замовчуванням, змінити не можна.

Розглянемо ще дві властивості, які має як кнопка, так і форма:

- **Enabled** (англ. *enabled* – доступний) – її значення визначає, чи доступний об'єкт для операцій над ним. Ця властивість може набувати лише одне з двох значень: **True** (англ. *true* – істина, правда) – об'єкт доступний або **False** (англ. *false* – хиба, неправда) – об'єкт недоступний; недоступна кнопка виглядає так: .
- **Visible** (англ. *visible* – видимий) – її значення визначає, чи видимий даний об'єкт у вікні. Ця властивість може набувати теж лише одне з двох значень: **True** – об'єкт видимий або **False** – об'єкт невидимий.



Для тих, хто вивчає Python

Щоб розмістити кнопку у вікні, потрібно:

1. Створити новий об'єкт типу **Button**, пов'язати його зі змінною, яка визначатиме ім'я об'єкта.
2. Установити значення властивостей кнопки або залишити їх за замовчуванням.
3. Розмістити створений об'єкт у вікні.

Для створення кнопки та пов'язування її зі змінною використовують команду вигляду:

```
<ім'я_кнопки> = Button()
```

або

```
<ім'я_кнопки> = Button(<набір властивостей та їх значень>)
```

У першому випадку буде створено кнопку зі значеннями властивостей **висота**, **ширина** і **колір**, які встановлено за замовчуванням, без напису на ній. У другому випадку буде створено кнопку, значення властивостей якої задано в цій команді. У дужках через кому вказують пари **<властивість> = <значення>**. Для кнопки можуть бути встановлені значення таких властивостей:

- **text** – текст на кнопці;
- **width** – ширина кнопки (кількість символів, які можуть бути розміщені поруч по горизонталі);
- **height** – висота кнопки (кількість символів, які можуть бути розміщені один під одним по вертикалі);
- **bg** – колір фону (значенням може бути назва кольору англійською);
- **fg** – колір символів тексту;
- **font** – шрифт символів тексту на кнопці (значення може містити назву шрифту, розмір символів і накреслення);
- **state** – стан кнопки (**normal** – звичайний, **disable** – недоступний)

та інші.

Наприклад, кнопка з іменем *button* з текстом *Змінити* завширшки 15 символів може бути створена командою:

```
button = Button(text = 'Змінити', width = 15).
```

Для розміщення кнопки у вікні можуть бути використані такі методи:

- **pack()** – кнопка автоматично розміщується під рядком заголовка вікна з однаковими відступами від лівої та правої меж вікна. Якщо у вікні розміщується кілька кнопок, то вони будуть відображатися одна під одною без відступів між ними. Якщо потрібно розмістити кнопку з деяким відступом від рядка заголовка або від попередньої кнопки, то в дужках можна вказати цей відступ у пікселях як значення властивості **pady**. Наприклад, після виконання команди **button.pack(pady = 20)** кнопку буде розміщено з відступом 20 пікселів від верхньої межі вікна або від попередньої кнопки;
- **place(x = <відступ від лівої межі вікна>, y = <відступ від верхньої межі вікна>)** – верхня ліва вершина кнопки розміщується в точці вікна із заданими координатами **x** та **y**. Наприклад, після виконання команди **button.place(x = 200, y = 100)** верхня ліва вершина кнопки буде розміщена в точці вікна з координатами (200, 100)

та інші.

Під час виконання проєкту може виникнути потреба зробити кнопку невидимою. Для цього, залежно від методу розміщення кнопки, можна використати методи **pack_forget()** або **place_forget()**. Щоб зробити кнопку знову видимою, слід застосувати метод **pack()** або **place()**.

На малюнку 6.25 наведено текст проєкту зі створення та розміщення кнопки в вікні та результат його виконання.



Мал. 6.25. Кнопка у вікні

Обробники подій для кнопки

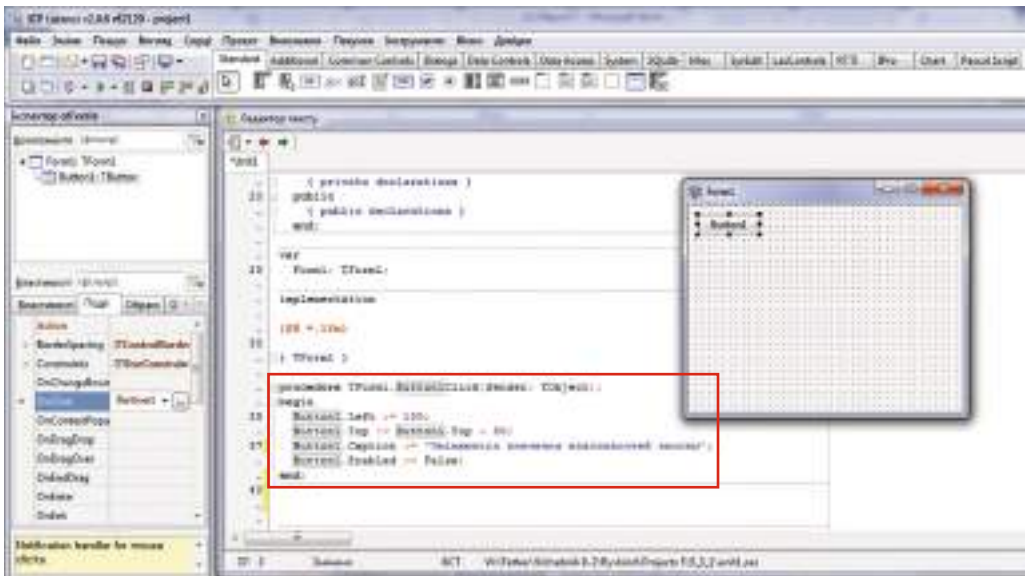


Для тих, хто вивчає Pascal

Якщо вибрати кнопку, то на вкладці **Події** вікна **Інспектор об'єктів** можна, як і для форми, вибрати подію і створити для неї в **Редакторі тексту** заготовку процедури – обробника цієї події. До цієї процедури можна ввести команди змінення значень властивостей як кнопки, так і форми.

На малюнку 6.26 наведено приклад обробника події **Click** для кнопки – процедури з іменем **TForm1.Button1Click**, який містить команди:

- установлення відступу 150 пікселів лівої межі кнопки від лівої межі вікна;
- збільшення на 50 пікселів відступу верхньої межі кнопки від верхньої межі вікна;
- виведення на кнопці вказаного тексту;
- установлення для кнопки режиму *Недоступна*.



Мал. 6.26. Обробник події Click для кнопки



Для тих, хто вивчає Python

Пов'язати з кнопкою обробник будь-якої події можна так само, як з вікном, використовуючи метод **bind()**. Наприклад, на малюнку 6.27 наведено приклад обробника події **Click** для кнопки, яка містить команду встановлення жовтого кольору вікна.

```
from tkinter import *
def click(event):
    root['bg'] = 'yellow'
root = Tk()
root.geometry('600x400+350+200')
button = Button(text = 'Змінити', width = 15)
button.pack(pady = 30)
button.bind('<l>', click)
```

Мал. 6.27. Обробник події **Click** для кнопки

Для кнопки подією за замовчуванням є *клацання лівої кнопки миші*. Тому пов'язати з кнопкою метод опрацювання саме цієї події можна простіше:

- під час створення кнопки вказати ім'я відповідного обробника події як значення атрибуту **command**;
- обробник події з указаним іменем описати перед створенням кнопки; у заголовку не потрібно вказувати посилання на подію, що відбулася, – **event**.

На малюнку 6.28 наведено приклад обробника події **Click** для кнопки – процедури з іменем *click()*, яка містить команди:

- встановлення для кнопки *червоного* кольору фону;
- встановлення шрифту *Arial* розміром *14* пікселів;
- виведення на кнопці вказаного тексту;
- встановлення для кнопки стану *недоступна*.

```
from tkinter import *
def click():
    button['bg'] = 'red'
    button['font'] = 'Arial 14'
    button['text'] = 'Змінюється значення властивостей кнопки'
    button['state'] = 'disable'
root = Tk()
root.geometry('600x400+350+200')
button = Button(text = 'Вибери мене', command = click)
button.pack(pady = 30)
```

Мал. 6.28. Обробник події **Click** для кнопки




Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/ghdrpzl> або QR-кодом.

**Найважливіше в цьому пункті****Для тих, хто вивчає Pascal**

Щоб розмістити кнопку на формі, потрібно підвести вказівник до її зображення  на вкладці **Standard** вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші.

У кнопки, як і у форми, є властивості **Name**, **Caption**, **Height**, **Width**, **Top**, **Left**, **Color**.

Значення властивості **Name** визначає ім'я кнопки як об'єкта. Значення властивості **Caption** визначає текст на кнопці. Значення властивостей **Height** і **Width** визначають висоту й ширину кнопки. Значення властивостей **Top** і **Left** визначають відстані верхньої і лівої меж кнопки від відповідних меж форми, на якій вона розміщена, а не від відповідних меж екрана, як для форми.

Значення всіх цих властивостей кнопки можна змінити до запуску проекту на виконання на вкладці **Властивості** вікна **Інспектор об'єктів**. Значення деяких з них можна змінити перетягуванням кнопки або зміненням її розмірів, використовуючи маркери на її межах.

Значення всіх цих властивостей також можна змінити командами присвоювання під час виконання проекту. Наприклад, **Button1.Width := 100**, **Button1.Left := Button1.Left + 200**, **Button1.Caption := 'Виконати'**.

Значення властивості **Color** для кнопки, яке встановлюється за замовчуванням, змінити не можна.

Розглянемо ще дві властивості, які має як кнопка, так і форма:

- **Enabled** – її значення визначає, чи доступний об'єкт для операцій над ним. Ця властивість може набувати лише одне з двох значень: **True** – об'єкт доступний або **False** – об'єкт недоступний; недоступна кнопка виглядає так:



- **Visible** – її значення визначає, чи видимий даний об'єкт у вікні. Ця властивість може набувати теж лише одне з двох значень: **True** – об'єкт видимий або **False** – об'єкт невидимий.

**Для тих, хто вивчає Python**

Щоб розмістити кнопку у вікні, потрібно створити новий об'єкт типу **Button**, пов'язати його зі змінною, що визначатиме ім'я кнопки, установити значення властивостей кнопки, розмістити створений об'єкт у вікні.

Дія створення кнопки та пов'язування її зі змінною використовують команду вигляду **<ім'я_кнопки> = Button()**.

Під час створення кнопки можуть бути встановлені значення її властивостей. Для цього потрібно ввести в дужках через кому пари **<властивість> = <значення>**. Для кнопки можуть бути встановлені значення властивостей **text**, **width**, **height**, **bg**, **fg**, **font**, **state** та ін.

Для розміщення кнопки у вікні можуть бути використані методи **pack()** або **place()**.

Пов'язати з кнопкою обробник будь-якої події можна так само, як з вікном, використовуючи метод **bind()**.

Щоб пов'язати з кнопкою метод опрацювання події **Click**, можна під час створення кнопки вказати ім'я відповідної процедури – обробника події як значення атрибуту **command**. У заголовку цієї процедури не потрібно вводити ключове слово **event**.





**Дайте відповіді на запитання****Для тих, хто вивчає Pascal**

- 1°. Як розмістити кнопку на формі?
- 2°. Які властивості кнопки ви знаєте? Що означає кожна з них?
- 3°. Чим відрізняється властивість **Color** для кнопки і для форми?
- 4°. Чим відрізняється властивість **Caption** для кнопки і для форми?
- 5°. Чим відрізняються властивості **Top** і **Left** для кнопки і для форми?
- 6°. Які значення можуть набувати властивості **Enabled** і **Visible**? Що означає кожне із цих значень для кожної із цих властивостей?

**Для тих, хто вивчає Python**

- 1°. Як створити кнопку?
- 2°. Які властивості кнопки ви знаєте? Що означає кожна з них?
- 3°. Як задати значення властивостей кнопки?
- 4°. Як розмістити кнопку у вікні?
- 5°. Якими способами можна призначити для кнопки обробники подій?
- 6°. Як призначити для кнопки обробник події **Click**?

**Виконайте завдання**

- 1°. Створіть проект, у якому в результаті настання події **Click** для кнопки її розміри стануть 40×30 пікселів ( ширина – 30 символів, висота – 3 символи) і текст на кнопці зміниться на назву вашого населеного пункту. Збережіть проект у папці з іменем **Завдання 6.3.1**, створеній у вашій папці.
- 2°. Створіть проект, у якому в результаті настання події **KeyPress** для кнопки її розміри стануть 20×40 пікселів ( ширина – 20 символів, висота – 4 символи) і текст на кнопці зміниться на ваше прізвище. Збережіть проект у папці з іменем **Завдання 6.3.2**, створеній у вашій папці.
- 3°. Створіть проект, у якому в результаті настання події **MouseMove** для кнопки зміниться текст на кнопці та збільшиться її відступ верхньої межі на 20 пікселів. Збережіть проект у папці з іменем **Завдання 6.3.3**, створеній у вашій папці.
- 4°. Створіть проект, у якому в результаті настання події **Click** для кнопки зміняться її розміри і зменшиться відступ її лівої межі на 20 пікселів. Збережіть проект у папці з іменем **Завдання 6.3.4**, створеній у вашій папці.
- 5°. Створіть проект, у якому в результаті настання події **Click** для кнопки її розміри стануть 50×50 пікселів ( ширина – 50 символів, висота – 5 символи) і текст на кнопці зміниться на *Мій проект*, а в результаті настання події **MouseMove** для кнопки кнопка переміститься на 30 пікселів управо та на 40 пікселів униз і стане недоступною ( установиться жовтий колір фону кнопки та її стан – *недоступний*). Збережіть проект у папці з іменем **Завдання 6.3.5**, створеній у вашій папці.

6*. Створіть проєкт розміщення у вікні трьох кнопок. Установіть такі значення їх властивостей, щоб на початку виконання проєкту перша кнопка була *видима і доступна*, друга – *видима і недоступна*, третя – *невидима і недоступна*. Для кожної кнопки створіть обробник події **Click**, щоб:

- 1) вибір першої кнопки робив її невидимою і недоступною, другу кнопку – видимою і доступною, третю кнопку – видимою і недоступною;
 - 2) вибір другої кнопки робив її невидимою і недоступною, третю кнопку – видимою і доступною, першу кнопку – видимою і недоступною;
 - 3) вибір третьої кнопки робив її невидимою і недоступною, першу кнопку – видимою і доступною, другу кнопку – видимою і недоступною.
- Збережіть проєкт у папці з іменем **Завдання 6.3.6**, створеній у вашій папці.

6.4. Напис



1. Які властивості кнопки ви знаєте? Як змінити значення її властивостей до початку виконання проєкту й під час його виконання?
2. Які події можуть відбуватися з кнопкою? У чому полягає кожна з них?
3. Які відмінності властивостей кнопки й форми ви знаєте?
4. Які відмінності у створенні обробника події для вікна та для кнопки?

Властивості напису

Ще одним компонентом (елементом керування), який можна використати в проєкті, є *напис*. **Напис** – це компонент, який використовується для виведення текстових повідомлень.



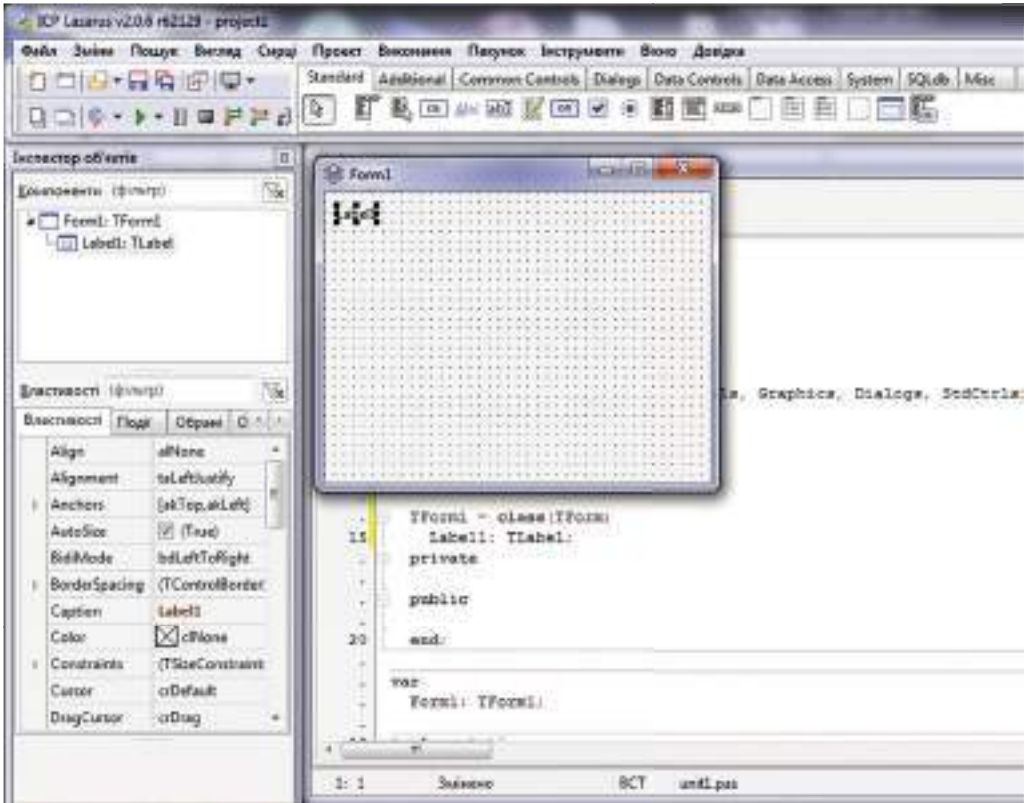
Для тих, хто вивчає Pascal

Щоб розмістити напис на формі, потрібно підвести вказівник до кнопки **TLabel** (англ. *label* – напис, етикетка, позначка) на вкладці **Standard** вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші. Після цього напис з'явиться у верхній лівій частині форми (мал. 6.29), а його ім'я додасться до списку компонентів проєкту в полі **Компоненти** вікна **Інспектор об'єктів**. За потреби напис можна перетягнути в будь-яке інше місце форми.

Виділити напис можна, вибравши його на формі або вибравши його ім'я в полі **Компоненти** вікна **Інспектор об'єктів**. При виділеному написі на вкладці **Властивості** цього вікна відображається список властивостей напису та їх значень. Значення властивостей напису можна змінювати. Для деяких властивостей їх значення потрібно вводити у відповідні поля, для деяких – можна вибирати зі списку, для деяких – можна змінювати перетягуванням напису по формі.

Властивості напису **Top**, **Left**, **Name**, **Enabled**, **Visible** аналогічні до відповідних властивостей кнопки. Значення властивості **Color** напису визначає колір його фону. Значення цих властивостей можна змінювати на вкладці **Властивості** перед запуском проєкту на виконання або командами присвоювання під час виконання проєкту, наприклад, **Label1.Top := 150**, **Label1.Left := Label1.Left + 150**, **Label1.Enabled := True**.


За замовчуванням значення властивостей **Height** і **Width** для напису, на відміну від форми і кнопки, не можна змінити на вкладці **Властивості** або перетягуванням. Ширина й висота напису визначаються текстом, розміщеним у цьому написі. Якщо змінити текст у полі значення властивості **Caption** або виконати команду



Мал. 6.29. Напис на формі

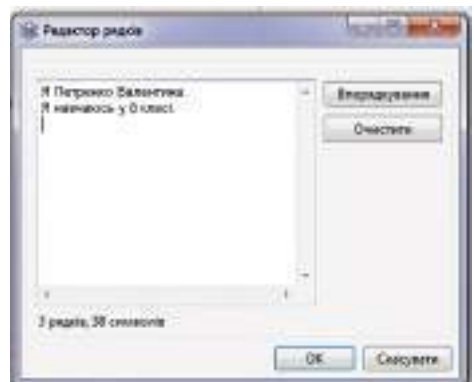
присвоювання, наприклад `Label1.Caption := 'Я навчаюсь у 8 класі'`, то відповідно до його довжини зміниться ширина напису. Але якщо встановити значення **False** властивості **AutoSize** (англ. **Automatic Size** – автоматичний розмір) напису, то його ширину й висоту можна змінювати незалежно від наявного в ньому тексту.

До запуску проекту на виконання текст напису можна вводити і редагувати як у полі **Caption**, так і у вікні **Редактор рядків** (мал. 6.30). Щоб відкрити це вікно,

потрібно зробити поле **Caption** поточним і вибрати кнопку  у його правій частині.

У цьому вікні можна задавати також і багаторядкові тексти в написі.



Для встановлення і змінювання значень властивостей символів, якими виводиться текст у написі, потрібно використати властивість **Font** (англ. *font* – шрифт). На відміну від усіх попередніх властивостей, ця властивість є **комплексною**. Комплексною вона називається тому, що складається з кількох властивостей символів: **Color**, **Height**, **Name**, **Size** (англ. *size* – розмір) та ін. Значення кожної з них можна



Мал. 6.30. Задання тексту напису у вікні **Редактор рядків**



Мал. 6.31. Вікно Шрифт для встановлення значень властивостей символів

змінити, якщо відкрити їх список вибором кнопки  ліворуч від назви властивості **Font** або у вікні **Шрифт** (мал. 6.31), яке відкривається вибором кнопки  у правій крайній частині рядка **Font**.

Вибираючи шрифт, потрібно звертати увагу на список **Сценарій**. Якщо в цьому списку відсутній елемент **Кирилиця**, то текст напису не буде коректно відображатися українською.

Під час виконання проекту змінити значення складових цієї комплексної властивості можна, використовуючи команду присвоювання. Так, колір символів можна змінити, наприклад, командою: **Label1.Font.Color := c1Blue**.



Для тих, хто вивчає Python

У вікні проекту, створеного мовою **Python**, можна також розміщувати **написи**. Для цього потрібно в тексті проекту:

1. Увести команду створення напису.
2. Установити значення його властивостей.
3. Розмістити напис у вікні.

Команда створення напису має вигляд:

```
<ім'я напису> = Label(text = '<текст напису>')
```

(англ. *label* – етикетка, позначка). Наприклад, для створення напису з іменем *label* і текстом *Я навчаюсь у 8 класі* потрібно виконати команду:

```
label = Label(text = 'Я навчаюсь у 8 класі')
```

Одночасно зі створенням напису можна також установити значення деяких його властивостей, наприклад, **bg** – колір фону напису, **fg** – колір символів, **font** – вид шрифту, розмір символів і накреслення (**bold** – напівжирне, **italic** – курсивне, **underline** – підкреслене), **height** – висота напису, **width** – ширина напису та ін. Для цього в команді створення напису в круглих дужках потрібно ввести через кому пари **<властивість> = <значення>**. Наприклад, після виконання команди:

```
label = Label(text = 'Я навчаюсь у 8 класі', bg = 'yellow', fg = 'green', font = 'Arial 14')
```

буде створено напис зеленими літерами на жовтому фоні із текстом *Я навчаюсь у 8 класі*, шрифт символів *Arial*, розмір символів *14*.

Значення властивостей також можна задавати окремими командами виду:

```
<ім'я напису>[<назва властивості>] = <значення властивості>
```

Наприклад, установити для напису з іменем *label* ширину 25 символів можна командою **label['width'] = 25**, установити шрифт *Times New Roman* з висотою символів 28 – командою **label['font'] = ('Times New Roman', 28)**, задати синій колір символів – командою **label['fg'] = 'blue'**. Такого виду команди можна використовувати і для змінення значень властивостей напису під час виконання проекту.



Для розміщення напису у вікні проекту використовуються вже відомі вам методи **pack()** і **place()**. Наприклад, для розміщення у вікні напису з іменем *label* з відступом 20 пікселів від рядка заголовка вікна можна використати команду **label.pack(pady = 20)**.

На малюнку 6.32 наведено вигляд вікна проекту зі створеним написом.



Мал. 6.32. Вікно проекту з написом

Обробники подій для напису

Хоча написи призначені переважно для виведення текстів, але з ними, як і з вікнами і кнопками, можна пов'язувати обробники різних подій.

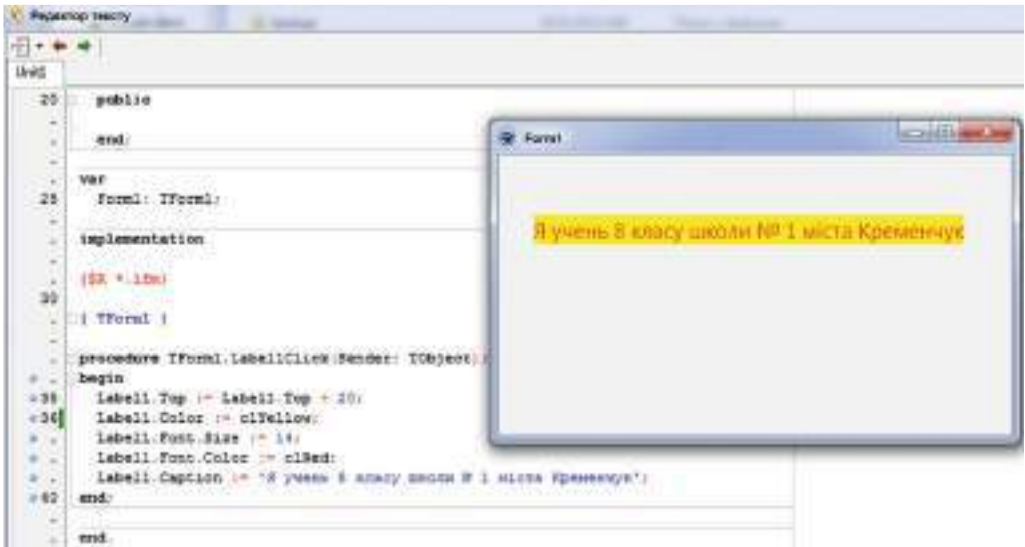


Для тих, хто вивчає Pascal

Якщо виділити напис, то на вкладці **Події** вікна **Інспектор об'єктів** можна, як і для форми та кнопки, вибрати подію і створити заготовку процедури – обробника цієї події (мал. 6.33). Як і для форми та кнопки, можна ввести до цієї процедури команди змінення значень властивостей напису, а також команди змінення властивостей форми і кнопки, команди для виконання обчислень та ін.

Наведена на малюнку 6.33 процедура – обробник події **Click** для напису – містить команди:

- збільшення на 20 пікселів відступу верхньої межі напису від верхньої межі вікна;
- установлення жовтого фону напису;
- установлення розміру 14 для символів тексту в написі;
- установлення червоного кольору символів тексту в написі;
- виведення в напис наведеного тексту.



Мал. 6.33. Проект з обробником події **Click** для напису і результат його виконання

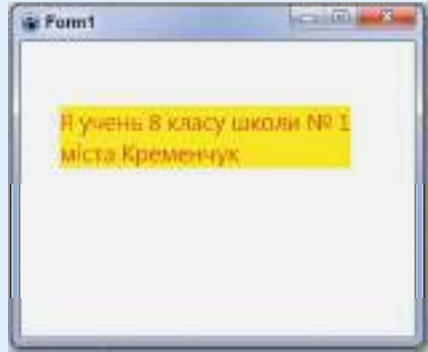


Для тих, хто хоче знати більше

Щоб у команді виведення тексту в напис текст у потрібному місці переносився на новий рядок, потрібно в ній використати символ переходу на новий рядок (#13). Наприклад, якщо в наведеному на малюнку 6.33 проєкті команду виведення тексту в напис змінити на таку:

Label1.Caption := 'Я учень 8 класу школи № 1' + #13 + 'міста Кременчук',
результат виконання цього проєкту виглядатиме так, як на малюнку 6.34.

Можна також задати перенесення тексту на наступний рядок, установивши значення **True** властивості **WordWrap** (англ. *word wrap* – перенесення слів), значення **False** властивості **AutoSize** і задавши потрібні розміри напису. Текст буде переноситися на новий рядок, як у текстовому процесорі.



Мал. 6.34. Результат виконання проєкту з перенесенням тексту



Для тих, хто вивчає Python

З написом, як і з вікном проєкту або кнопкою, можна пов'язати процедуру – обробник деякої події, використовуючи метод **bind()**:

<ім'я напису>.bind(' <подія>', <ім'я процедури>)

Окремо потрібно ввести текст процедури – обробника події із зазначеним іменем. Послідовність дій може бути такою:

1. Створити напис та установити значення його властивостей.
2. Пов'язати напис з обробником події, указавши подію та ім'я процедури.
3. Створити процедуру – обробник події, надавши їй вказане ім'я.
4. Увести в процедуру команди для встановлення потрібних значень властивостей об'єктів.

Наприклад, потрібно, щоб після настання події **Click** з написом з іменем *label* колір фону напису ставав *жовтим*, колір символів – *червоним*, розмір символів – 14, у напис виводився заданий текст. На малюнку 6.35 наведено текст відповідного проєкту та результат виконання процедури – обробника події **Click** для напису.

```
from tkinter import *
def label_click(event):
    label['text'] = 'Я навчаюсь у 8 класі'
    label['font'] = 14
    label['fg'] = 'red'
    label['bg'] = 'yellow'
root=Tk()
root.geometry('300x200')
label = Label(text = 'Це напис')
label.bind('<1>', label_click)
label.pack(pady = 20)
```



Мал. 6.35. Обробник події **Click** для напису та результат його виконання



Для тих, хто хоче знати більше

Текст у написі можна вивести в кілька рядків. Для цього на місці, де повинен розпочинатися новий рядок, потрібно ввести символ `\n`. Наприклад, у написі, створеному командою `Label = Label(text = 'Я навчаюсь \n у 8 класі')`, у першому рядку буде відображатися текст *Я навчаюсь*, а у другому – *у 8 класі*. Результат виконання цієї команди наведено на малюнку 6.36.



Мал. 6.36. Виведення тексту в написі у два рядки



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/mhdraDg> або QR-кодом.



Найважливіше в цьому пункті

Напис – це компонент (елемент керування), який використовується для виведення текстових повідомлень.



Для тих, хто вивчає Pascal

Щоб розмістити напис на формі, потрібно підвести вказівник до кнопки **TLabel** на вкладці **Standard** вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші.

Властивості напису **Top**, **Left**, **Name**, **Enabled**, **Visible** аналогічні до відповідних властивостей кнопки. Значення властивості **Color** напису визначає колір його фону. Значення цих властивостей можна змінювати на вкладці **Властивості** вікна **Інспектор об'єктів** перед запуском проекту на виконання або командами присвоювання під час виконання проекту, наприклад `Label1.Top := 150`, `Label1.Left := Label1.Left + 150`, `Label1.Enabled := True`.

За замовчуванням значення властивостей **Height** і **Width** для напису, на відміну від форми і кнопки, не можна змінити на вкладці **Властивості** або перетягуванням. Ширина й висота напису визначаються текстом, який розміщено в цьому написі. Якщо змінити текст у полі значення властивості **Caption** або виконати команду присвоювання, наприклад `Label1.Caption := 'Я навчаюсь у 8 класі'`, то відповідно до його довжини зміниться ширина напису. Але якщо встановити значення **False** властивості **AutoSize** (англ. *Automatic Size* – автоматичний розмір) напису, то його ширину й висоту можна змінювати незалежно від наявного в ньому тексту.

Текст напису можна вводити і редагувати як у полі **Caption**, так і у вікні **Редактор рядків**.

Властивість **Font** визначає значення властивостей символів, якими буде виводитися текст у написі. Ця властивість є **комплексною**, оскільки складається з кількох властивостей символів: **Color**, **Height**, **Name**, **Size** та ін. Під час виконання проекту значення цих властивостей можна змінити командою присвоювання, наприклад командою: `Label1.Font.Color := clBlue`.

Виділити напис можна, вибравши його на формі або вибравши його ім'я в полі **Компоненти** вікна **Інспектор об'єктів**.

Якщо виділити напис, то на вкладці Події вікна **Інспектор об'єктів** можна вибрати подію і для неї у вікні **Редактор тексту** створити обробник цієї події, виконання якого може змінити значення властивостей як напису, так і вікна, кнопки та інших компонентів.



Для тих, хто вивчає Python

Команда створення напису мовою **Python** має вигляд:

```
<ім'я напису> = Label(text = '<текст напису>')
```

Одночасно зі створенням напису можна також установити значення деяких властивостей, наприклад **bg** – колір фону напису, **fg** – колір символів, **font** – вид шрифту, розмір символів і накреслення (**bold** – напівжирне, **italic** – курсивне, **underline** – підкреслене), **height** – висота напису, **width** – ширина напису та ін. Для цього в команді створення напису в круглих дужках потрібно ввести через кому пари **властивість = значення**.

Значення властивостей також можна задавати та змінювати окремими командами виду:

```
<ім'я напису>['<назва властивості>'] = <значення властивості>
```

Для розміщення напису у вікні проєкту використовують методи **pack()** і **place()**.

З написом можна пов'язати процедуру – обробник деякої події, використовуючи метод **bind()**:

```
<ім'я напису>.bind('<подія>', <ім'я процедури>)
```

Окремо потрібно ввести код процедури – обробника події із зазначеним іменем.



Дайте відповіді на запитання

1°. Для чого використовується напис?



Для тих, хто вивчає Pascal

2°. Як розмістити напис на формі?

3°. Що визначають значення властивостей **Top**, **Left**, **Name** для напису?

4°. Що визначають значення властивостей **Enabled** і **Visible** для напису?

5°. Що визначають значення комплексної властивості **Font** для напису?

6°. Чим відрізняється властивість **Color** для напису і для кнопки?

7°. Чим відрізняються властивості **Top** і **Left** для напису і для форми?

8°. Чим відрізняються властивості **Height** і **Width** для напису і для кнопки?

9°. Як можна змінити текст у написі?

10°. Як створити обробник події для напису?



Для тих, хто вивчає Python

2°. Як створити напис?

3°. Якими способами можна встановити значення властивостей напису?

4°. Які властивості напису ви знаєте? Поясніть їх.

5°. Як розмістити напис у вікні?

6°. Як пов'язати з написом процедуру – обробник події?

7°. Чим відрізняються властивості напису і вікна?

8°. Чим відрізняються властивості напису і кнопки?



Виконайте завдання

1^о. Створіть проєкт, у якому в результаті настання події **Click** для напису буде встановлено:

- відступ його верхньої межі від верхньої межі вікна *120 пікселів* і текст у написі зміниться на назву вулиці, на якій розміщено ваш заклад освіти;
- ширину напису *30 символів* і текст у написі зміниться на назву вулиці, на якій розміщено ваш заклад освіти.

Збережіть проєкт у папці з іменем **Завдання 6.4.1**, створеній у вашій папці.



2^о. Створіть проєкт, у якому в результаті настання події **DbClick** для напису буде встановлено:

- відступ його лівої межі від лівої межі вікна *80 пікселів* і текст у написі зміниться на ваше ім'я;
- висоту напису *5 символів* і текст у написі зміниться на ваше ім'я.

Збережіть проєкт у папці з іменем **Завдання 6.4.2**, створеній у вашій папці.

3^о. Створіть проєкт, у якому в результаті настання події **MouseMove** для напису буде встановлено:

- новий текст на ньому, червоний колір тексту і збільшиться відступ напису від верхньої межі вікна на *20 пікселів*;
- новий текст на ньому, червоний колір тексту і збільшиться ширина напису на *5 символів*.

Збережіть проєкт у папці з іменем **Завдання 6.4.3**, створеній у вашій папці.



4^о. Створіть проєкт, у якому в результаті настання події **Click** для напису буде встановлено:

- зменшення його відступу від лівої межі вікна на *20 пікселів*, розмір символів *12*, колір символів *синій* і виведеться в напис назва вашого закладу освіти;
- зменшення ширини напису на *2 символи*, розмір символів *12*, колір символів *синій* і виведеться в напис назва вашого закладу освіти.

Збережіть проєкт у папці з іменем **Завдання 6.4.4**, створеній у вашій папці.

5^о. Створіть проєкт, у якому в результаті настання події **Click** для напису буде встановлено коричневий колір фону і текст у написі зміниться на *Мій проєкт*, а в результаті настання події **MouseMove** для напису:

- він переміститься на *30 пікселів* управо і на *40 пікселів* униз;
- збільшиться ширина та висота напису на *3 символи*.

Збережіть проєкт у папці з іменем **Завдання 6.4.5**, створеній у вашій папці.

6^о. Створіть проєкт, у якому в результаті настання події **Click** для кнопки:

- кнопка переміститься на *60 пікселів* управо і на *10 пікселів* уверх, а в напис буде виведено ваше прізвище та ім'я зеленим кольором на сірому фоні;
- ширина напису зменшиться на *2 символи*, висота напису збільшиться на *3 символи*, у напис буде виведено ваше прізвище та ім'я зеленим кольором на сірому фоні.

Збережіть проєкт у папці з іменем **Завдання 6.4.6**, створеній у вашій папці.



7^о. Створіть проєкт, у якому в результаті настання події **Click** для кнопки вікно проєкту переміститься на *60 пікселів* уліво і на *80 пікселів* униз і в напис виведеться назва вашого закладу освіти синім кольором на жовтому фоні. Збережіть проєкт у папці з іменем **Завдання 6.4.7**, створеній у вашій папці.

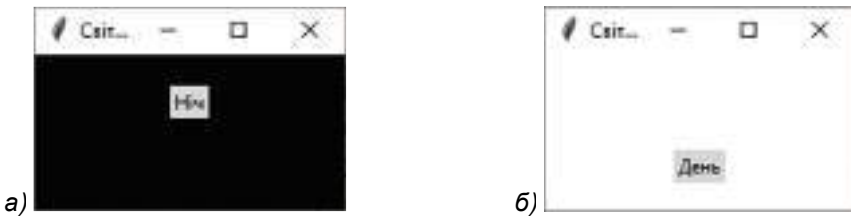
8^о. Створіть проєкт, у якому розташуйте у вікні два написи *Моя країна* та *Україна*. Установіть заголовок вікна – *Україна*. Після настання події **Click** з першим написом колір фону першого напису повинен стати *синім*, другого напису – *жовтим*. Після настання події **Click** з другим написом колір

символів першого напису повинен стати *жовтим*, другого напису – *синім*, шрифт символів в обох написах – *Arial, 14, напівжирний*. Вигляд вікна проєкту наведено на малюнку 6.37. Збережіть проєкт у папці з іменем **Завдання 6.4.8**, створеній у вашій папці.



Мал. 6.37. Вигляд вікна проєкту: а) після запуску; б) після настання події **Click** для першого напису; в) після настання події **Click** для другого напису

- 9*. Створіть проєкт, у якому розташуйте у вікні два написи *Ніч* і *День*, задайте заголовок вікна – *Світло*. Після запуску проєкту колір фону вікна – *чорний*, колір фону першого напису – *світло-сірий*, другий напис – *не відображається*. Після настання події **Click** з першим написом колір фону вікна стає *білим*, перший напис *зникає*, другий напис відображається на *світло-сірому* фоні. Після настання події **Click** з другим написом вікно проєкту повертається до початкового стану. Вигляд вікна проєкту наведено на малюнку 6.38. Збережіть проєкт у папці з іменем **Завдання 6.4.9**, створеній у вашій папці.



Мал. 6.38. Вікно проєкту: а) початковий стан і після настання події **Click** для другого напису; б) після настання події **Click** для першого напису

ПРАКТИЧНА РОБОТА № 9

«Проєкти з кнопками та написами»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.



Для тих, хто вивчає Pascal

1. Відкрийте середовище **Lazarus**.
2. Розмістіть на формі дві кнопки та напис.
3. Установіть перед початком виконання проєкту такі значення властивостей першої кнопки:
 - ширина – *60 пікселів*;
 - висота – *20 пікселів*;
 - відступ від лівої межі вікна – *120 пікселів*;
 - відступ від верхньої межі вікна – *100 пікселів*;
 - текст на кнопці – *Вікно проєкту*.



4. Установіть перед початком виконання проєкту такі значення властивостей другої кнопки:
 - ширина – 100 пікселів;
 - висота – 30 пікселів;
 - відступ від лівої межі форми – 300 пікселів;
 - відступ від верхньої межі форми – 100 пікселів;
 - текст на кнопці – *Напис*.
5. Установіть перед початком виконання проєкту такі значення властивостей напису:
 - відступ від лівої межі форми – 150 пікселів;
 - відступ від верхньої межі форми – 200 пікселів;
 - текст у написі – назва вашого класу.
6. Збережіть проєкт у папці з іменем **Практична 9**, створеній у вашій папці.
7. Створіть обробник події **Click** для першої кнопки, виконання якого встановить ширину вікна проєкту – 800 пікселів, висоту вікна проєкту – 400 пікселів, колір фону вікна – *зелений*, ширину цієї кнопки – 200 пікселів, перемістить її на 50 пікселів уліво і на 30 пікселів уверх, зробить її недоступною, збільшить ширину другої кнопки на 10 пікселів.
8. Створіть обробник події **MouseMove** для другої кнопки, виконання якого встановить жовтий колір фону напису, відступ напису від верхньої межі вікна на 200 пікселів, зменшить відступ напису від лівої межі вікна на 50 пікселів, установить колір символів тексту напису червоний і виведе в напис текст *Ми вивчаємо мову програмування Pascal!*.
9. Збережіть нову версію проєкту з тим самим іменем у тому самому місці.



Для тих, хто вивчає Python

1. Створіть проєкт з вікном розміра 400 × 300 пікселів, у якому розмістіть дві кнопки та напис.
2. Установіть такі значення властивостей першої кнопки:
 - ширина – 20 символів;
 - висота – 2 символи;
 - колір фону – *сірий*;
 - колір символів – *жовтий*;
 - текст на кнопці – *Вікно проєкту*.
3. Установіть такі значення властивостей другої кнопки:
 - ширина – 10 символів;
 - висота – 3 символи;
 - колір фону – *синій*;
 - колір символів – *білий*;
 - текст на кнопці – *Напис*.
4. Установіть такі значення властивостей напису:
 - колір символів – *синій*;
 - текст у написі – назва вашого класу;
 - шрифт символів – *Arial 14*.
5. Створіть обробник події **Click** для першої кнопки, виконання якого встановить ширину вікна проєкту – 800 пікселів, висоту вікна проєкту – 400 пікселів, колір фону вікна – *зелений*, ширину першої кнопки – 30 символів, збільшить ширину другої кнопки на 5 символів, змінить колір першої кнопки на *синій*, а колір символів на ній – на *білий*, зробить першу кнопку *недоступною*.
6. Створіть обробник події **MouseMove** для другої кнопки, виконання якого встановить *червоний* колір фону напису, ширину напису – 40 символів, збільшить висоту напису на 5 символів, установить колір символів тексту напису *жовтий* і виведе у напис текст *Ми вивчаємо мову програмування Python!*.
7. Збережіть проєкт у вашій папці у файлі з іменем **Практична 9.py**.

6.5. Поле. Проекти з вхідними даними і кінцевими результатами

1. Які властивості має компонент *напис*? Що визначають значення цих властивостей?
2. Для чого використовується команда присвоювання? Який вигляд має ця команда?
3. Які константи (постійні величини) ви використовували на уроках математики, фізики, хімії? Чим відрізняється константа від змінної?

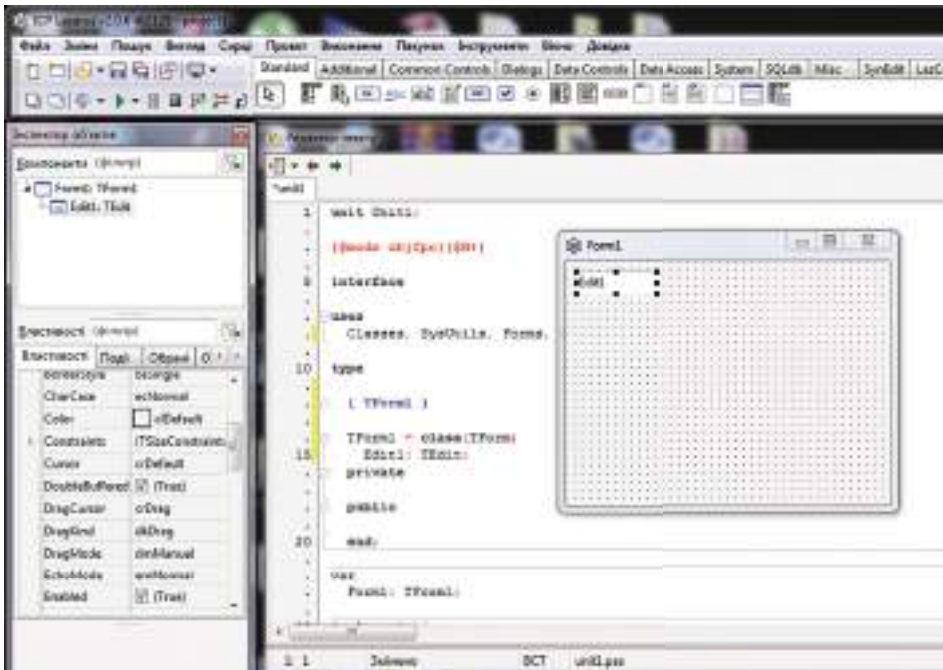
Властивості поля

Ще одним компонентом (елементом керування), який можна використати у вікні проекту, є *поле*. **Поле** – це елемент керування, який використовується в основному для введення даних. Поле можна використовувати також для виведення результатів.



Для тих, хто вивчає Pascal

Щоб розмістити поле на формі, потрібно підвести вказівник до кнопки **TEdit** (англ. *edit* – редагувати) на вкладці **Standard** вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші. Після цього поле з'явиться у верхній лівій частині форми (мал. 6.39), а його ім'я додасться до списку елементів керування проекту в полі **Компоненти** вікна **Інспектор об'єктів**. За потреби поле можна перетягнути в будь-яке інше місце форми.



Мал. 6.39. Поле на формі



Виділити поле можна, вибравши його на формі або вибравши його ім'я в полі **Компоненти** вікна **Інспектор об'єктів**. При вибраному полі список його властивостей та їх значень відображається на вкладці **Властивості** цього вікна. Значення властивостей поля можна змінювати. Для деяких властивостей їх значення потрібно вводити у відповідні поля на вкладці **Властивості**, для деяких – можна вибирати зі списку, а для деяких – можна змінювати перетягуванням поля по формі.

Властивості поля **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font** аналогічні до відповідних властивостей напису. Значення властивості **Width**, аналогічно до форми і кнопки, можна змінити або на вкладці **Властивості**, або перетягуванням спеціальних маркерів на межах поля. Значення властивостей поля **Height** не можна змінити, якщо значення властивості **AutoSize** дорівнює **True**.

Крім того, значення властивостей поля можна встановлювати під час виконання проекту командами присвоювання, наприклад **Edit1.Top := 200**, **Edit1.Left := Edit1.Left - 50**, **Edit1.Enabled := False**.

Текст у полі є значенням його властивості **Text** (англ. *text* – текст). Його можна вводити й редагувати у відповідному полі вкладки **Властивості**. Часто перед початком виконання проекту значення цієї властивості доцільно зробити порожнім. Значення цієї властивості також можна встановити під час виконання проекту командою присвоювання, наприклад **Edit1.Text := 'Текстове поле'**.

Можна, наприклад, вивести в напис текст, який є в полі. Для цього потрібно виконати команду: **Label1.Caption := Edit1.Text**.

Звертаємо вашу увагу: якщо ширина поля буде менша від довжини тексту, який в нього вводиться або виводиться, то, на відміну від напису, незалежно від значення властивості **AutoSize**, ширина поля автоматично не збільшиться і частина тексту буде знаходитися за межами поля і невидима для користувача.



Для тих, хто вивчає Python

Команда для створення поля в мові **Python** має вигляд:

<ім'я поля> = Entry(<набір властивостей та їх значень>)

(англ. *entry* – запис, вхід).

Властивості поля **width**, **bg**, **fg**, **font** аналогічні до відповідних властивостей напису та кнопки, а властивість **height** відсутня – висота поля визначається висотою символів у його вмісті. Для поля можна задавати товщину рамки як значення властивості **bd**.

Для розміщення поля у вікні проекту можна використовувати методи **pack()** і **place()**.

На малюнку 6.40 наведено вигляд вікна проекту з двома полями. Верхнє поле створено командою **entry1 = Entry()** зі значеннями властивостей, установленими за замовчуванням, нижнє – створене командою:

entry2 = Entry(width = 15, bg = 'yellow', fg = 'green', font = 'Arial 20', bd = 3)



Мал. 6.40. Вікно проекту з двома полями

Щоб вивести в поле деякий текст, використовують команду виду:

<ім'я поля>.insert(<позиція>, <текст>),

де *позиція* – це позиція в полі, починаючи з якої буде вставлено текст. Якщо поле порожнє, то текст вставляється в позицію 0, навіть якщо в команді зазначено іншу позицію. Якщо ж у полі вже є деякий вміст, то текст можна вставити, починаючи з будь-якої позиції в цьому вмісті. Для вставлення тексту в кінець існуючого вмісту вказують позицію **END**.

Наприклад, щоб вивести в порожнє поле з іменем *entry* текст *Я навчаюсь у 8 класі*, потрібно виконати команду:

entry.insert(0, 'Я навчаюсь у 8 класі')

Для змінення вмісту цього поля на *Я навчаюсь у 8-А класі* можна після попередньої команди додати команду:

entry.insert(14, '-А')

А щоб отримати в полі текст *Я навчаюсь у 8-А класі школи № 1*, можна після попередніх команд додати команду:

entry.insert(END, ' школи № 1')

Отримати з поля його вміст можна, використовуючи метод **get()**. Наприклад, щоб у напис з іменем *label* вивести текст, що введено в поле з іменем *entry*, потрібно виконати команду:

label['text'] = entry.get()

Для очищення вмісту поля з іменем *entry* використовують команду:

entry.delete(0, END)

Під час виконання проекту буває зручно, щоб одразу після запуску курсор було встановлено в середині поля. У такому разі кажуть, що поле знаходиться у **фокусі**. Для цього використовують метод **focus_set()**. Наприклад, виконання команди **entry.focus_set()** приводить до встановлення курсора в середині поля з іменем *entry*.

Обробники подій для поля

З полем, як і з іншими компонентами, можна пов'язати обробники різних подій, які можуть відбуватися з ним.



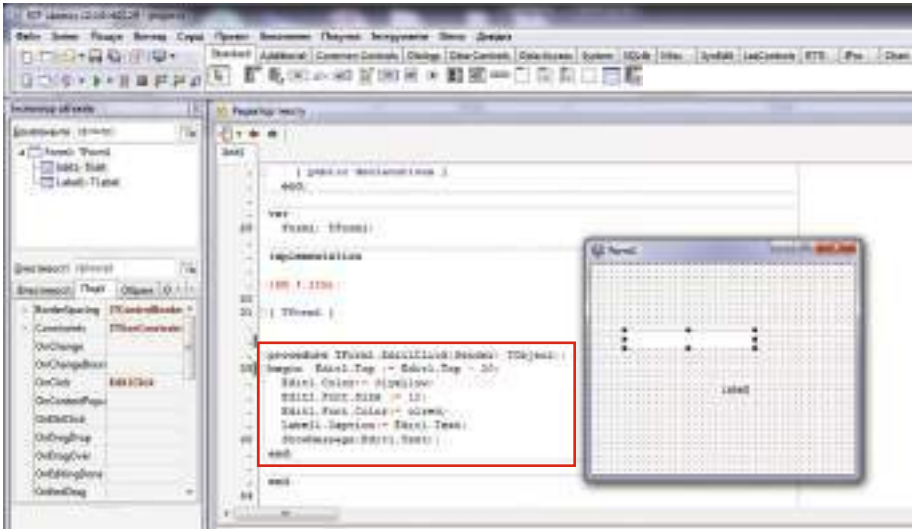
Для тих, хто вивчає Pascal

Якщо виділити поле, то на вкладці **Події** вікна **Інспектор об'єктів** можна, як і для розглянутих раніше компонентів, вибрати подію, для якої, аналогічно до форми, кнопки й напису, створити процедуру – обробник цієї події.

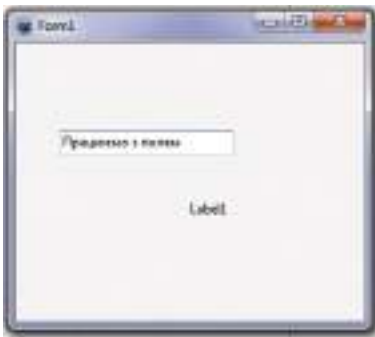
Наприклад, наведена на малюнку 6.41 процедура – обробник події **Click** для поля – містить команди:

- зменшення на 20 пікселів відступу верхньої межі поля від верхньої межі вікна;
- встановлення *жовтого* кольору фону поля;
- встановлення розміру 12 для символів тексту в полі;
- встановлення *червоного* кольору символів для тексту в полі;
- виведення тексту, який уведено в полі, у напис і у вікно повідомлень.

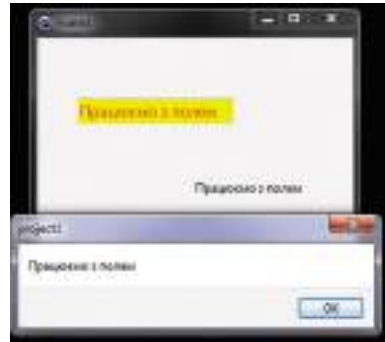
Після запуску проекту на виконання можна ввести в поле довільний текст (на малюнку 6.42 у поле введено текст *Працюємо з полем*), установити вказівник на полі та клацнути ліву кнопку миші. Результат виконання процедури – обробника події **Click** для поля наведено на малюнку 6.43. Колір фону поля стає жовтим, колір символів – червоним, розмір символів – 12, текст, уведений у поле, виводиться в напис і у вікно повідомлень.



Мал. 6.41. Обробник події Click для поля



Мал. 6.42. Вікно виконання проекту після введення тексту в поле й до настання події Click для поля



Мал. 6.43. Вікно виконання проекту після настання події Click для поля



Для тих, хто вивчає Python

Як і з іншими об'єктами вікна, з полем можна пов'язати процедури – обробники подій, використовуючи метод **bind()**.

Наприклад, на малюнку 6.44 наведено текст проекту з процедурою – обробником події **Click** для поля. Ця процедура містить команди:

- установлення жовтого кольору фону поля;
- установлення червоного кольору для символів тексту в полі;
- установлення розміру 12 для символів тексту в полі;
- виведення тексту, уведеного в полі, у напис;
- виведення тексту, уведеного в полі, у вікно повідомлень.

Після запуску проекту на виконання можна ввести в поле довільний текст (ліворуч на малюнку 6.45 у поле введено текст *Працюємо з полем*), установити вказівник на полі та клацнути ліву кнопку миші. Результат виконання обробника події **Click** для поля наведено на малюнку 6.45 праворуч.

```

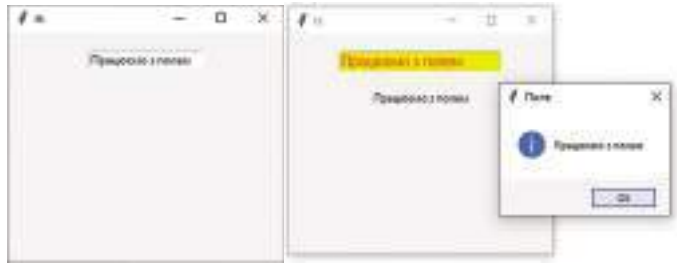
from tkinter import *
from tkinter.messagebox import *

def click(event):
    entry['bg'] = 'yellow'
    entry['fg'] = 'red'
    entry['text'] = z
    label['text'] = entry.get()
    showinfo('Info', entry.get())

root = Tk()
root.geometry("300x230")

entry = Entry()
entry.bind('<Return>', click)
entry.pack(pady = 20)
entry.focus_set()

label = Label()
label.pack()
    
```



Мал. 6.44. Текст проекту з обробником події **Click** для поля

Мал. 6.45. Вікно виконання проекту після введення тексту в поле й до настання події **Click** для поля (ліворуч) та після настання події (праворуч)

Проекти з введенням даних і виведенням результатів

У ході виконання проектів можна вводити дані та виводити результати.

Для введення даних використовується поле, а для виведення результатів можна використовувати поле, напис, вікно повідомлень.

Створимо для прикладу проект, у якому обчислюватиметься сума двох довільних чисел.

Побудуємо математичну модель для цього проекту:

вхідні дані: два довільних числа x і y ;

кінцеві результати: їх сума z ;

формули: $z = x + y$.

Тому використаємо в цьому проекті два поля, у які користувач під час виконання проекту вводитиме доданки (вони можуть бути як цілі, так і дробові), напис для виведення значення суми і кнопку, після вибору якої відбуватиметься додавання введених у поля чисел і виведення суми в напис.



Для тих, хто вивчає Pascal

Розмістимо на формі (мал. 6.46):

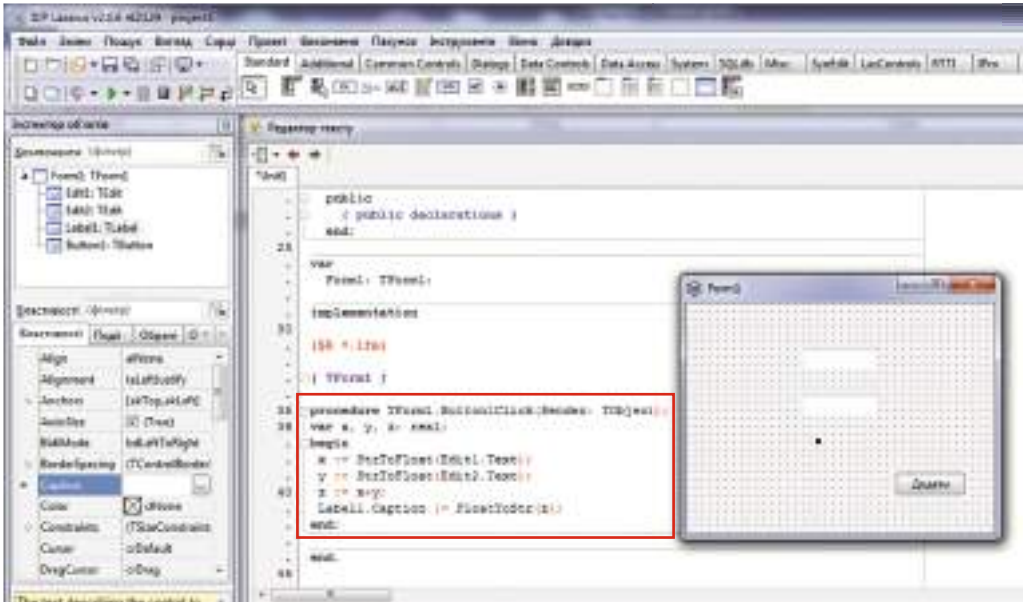
- два поля; зробимо їх порожніми, для чого на вкладці **Властивості** вікна **Інспектор об'єктів** видалимо текст у полі властивості **Text**;
- напис; зробимо його порожнім, для чого на вкладці **Властивості** видалимо текст у полі властивості **Caption**; у такому разі на формі напис виглядатиме, як чорний квадратик;
- кнопку; змінимо текст на ній на *Додати*, використовуючи властивість **Caption** на вкладці **Властивості**.

На малюнку 6.46 наведено процедуру – обробник події **Click** для кнопки, у якій обчислюється сума двох чисел, уведених у поля. Розглянемо наведену процедуру.

Хоча користувач під час виконання проекту вводить у поля числа, система тим не менше сприймає їх як тексти. Ці тексти є значеннями властивості **Text** полів з іменами **Edit1** і **Edit2**. Тому перші дві команди процедури

$x := \text{StrToFloat}(\text{Edit1.Text})$ і $y := \text{StrToFloat}(\text{Edit2.Text})$ (англ. *string to float* – рядок у число з плаваючою десятковою комою) переводять текстове подання чисел безпосередньо в числа і присвоюють ці числа змінним x і y відповідно.

Третя команда додає ці числа і присвоює суму змінній z .



Мал. 6.46. Форма з двома полями, написом і кнопкою та процедура для додавання двох чисел

У четвертій команді **Label1.Caption := FloatToStr (z)** (англ. *float to string* – число з плаваючою десятковою комою у рядок) обчислена сума переводиться із числа в його текстове подання, і цей текст присвоюється значенню властивості **Caption** напису з іменем **Label1**.

У мові програмування **Pascal** для будь-якої змінної повинен бути вказаний її **тип**.

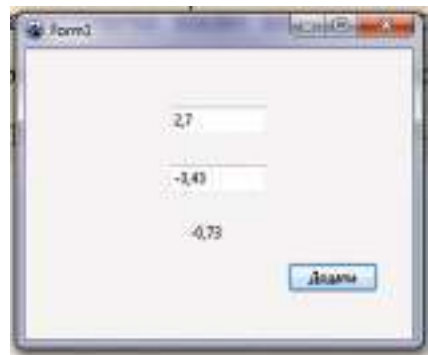
Тип змінної визначає:

- які значення може набувати ця змінна;
- які операції над нею можна виконувати;
- який обсяг оперативної пам'яті комп'ютера буде виділено для зберігання значення цієї змінної.

Імена змінних та їх типи вказуються в проєкті після ключового слова **var** (англ. *variable* – змінна). Цей рядок потрібно розмістити між рядком заголовка процедури і рядком зі словом **begin**.

Значення змінних **x**, **y**, **z**, які використані в процедурі, за умовою задачі можуть бути довільними числами, тобто і цілими, і дробовими. Для таких змінних можна використати тип **real** (англ. *real* – дійсний). Змінні цього типу можуть набувати цілі й дробові значення, модуль яких не перевищує $1,7 \cdot 10^{38}$. Для кожного із цих значень буде виділено 8 Б пам'яті.

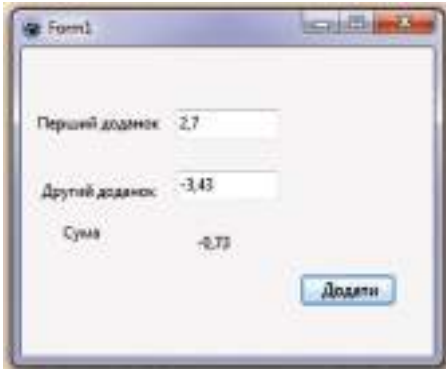
Під час виконання створеного проєкту користувач увводить у поля два числа (на малюнку 6.47 це числа 2,7 і $-3,43$), вибирає кнопку із заголовком **Додати**, і результат додавання ($-0,73$) виводиться в напис. Не закри-



Мал. 6.47. Вікно виконання проєкту додавання двох чисел

ваючи вікно виконання проєкту, користувач може ввести в поля інші числа, знову вибрати кнопку із заголовком *Додати* й одержати новий результат. І так можна повторювати довільну кількість разів.

Щоб під час виконання проєкту для користувача було зрозуміліше, у які поля що саме потрібно вводити і де виводитиметься результат, доцільно на формі розмістити зліва від кожного поля і від напису, призначеного для виведення результату, написи з пояснювальними текстами, наприклад *Перший доданок*, *Другий доданок*, *Сума* (мал. 6.48).



Мал. 6.48. Вікно виконання проєкту додавання двох чисел з пояснювальними текстами

Зазначимо, що виведення результату можна було б здійснити в поле, якщо розмістити на формі замість напису третє поле і виконати команду **Edit3.Text := FloatToStr(z)**, або у вікно повідомлень командою **ShowMessage (FloatToStr(z))**.

Якщо змінні можуть набувати тільки цілі значення, то для них можна використовувати тип **integer** (англ. *integer* – цілий) або тип **longint** (англ. *long integer* – довге ціле). Змінні першого із цих типів можуть набувати цілі значення від **-32 768** до **32 767**, а другого – цілі значення від **-2 147 483 648** до **2 147 483 647**. Для кожного із цих значень буде виділено 4 Б пам'яті.

Якщо в проєкті використовуватимуться змінні будь-якого цілого типу, для читання їх значень з поля можна використати, наприклад, команду **x := StrToInt (Edit1.Text)** (англ. *string to integer* – рядок у ціле число), а для виведення цілих чисел у напис – команду **Label1.Caption := IntToStr(z)** (англ. *integer to string* – ціле число в рядок).

У рядочку **var** тип таких змінних має бути описаний як **integer** або **longint**.

Зауважимо: якщо в рядочку **var** описуються змінні різних типів, то він виглядатиме, наприклад, так: **var x, y: real; t: integer;**. Запис може бути зроблений у кілька рядків, але слово **var** використовується лише один раз:

```
var x, y: real;
    t: integer;
```



Для тих, хто вивчає Python

Розмістимо у вікні проєкту два поля, напис і кнопку з текстом *Сума*. Проєкт, що створює вікно з розміщеними в ньому об'єктами, процедура для додавання двох чисел, а також вікно виконання проєкту наведені на малюнку 6.49.

Розглянемо наведену процедуру.

Хоча користувач під час виконання проєкту вводить у поля числа, система тим не менше сприймає їх як тексти. Ці тексти є вмістом полів з іменами **entry1** і **entry2**. Тому перші дві команди процедури **x = float(entry1.get())** і **y = float(entry2.get())** (англ. *float* – число з плаваючою десятковою комою) переводять текстове подання чисел безпосередньо в числа і присвоюють ці значення змінним **x** і **y** відповідно.

Третя команда додає ці числа та присвоює суму змінній **z**.

Четверта команда **label['text'] = str(z)** (англ. *string* – рядок) переводить обчислену суму із числа в його текстове подання та присвоює цей текст значенню властивості **text** напису з іменем **label**.



```

from tkinter import *

def click():
    x = float(entry1.get())
    y = float(entry2.get())
    z = x + y
    label['text'] = str(z)

root = Tk()
root.geometry("300x250")

entry1 = Entry()
entry1.pack(pady = 10)
entry2 = Entry()
entry2.pack(pady = 10)

label = Label()
label.pack(pady = 10)

button = Button(text='Сума', command=click)
button.pack(pady = 10)

```



Мал. 6.49. Вікно з двома полями, написом і кнопкою та текст проєкту з процедурою для додавання двох чисел

У мові програмування **Python** змінним відповідають значення певного типу.

Тип визначає:

- які значення може набувати відповідна змінна;
- які операції над значеннями можна виконувати;
- який обсяг оперативної пам'яті комп'ютера буде виділено для зберігання даних цього типу.

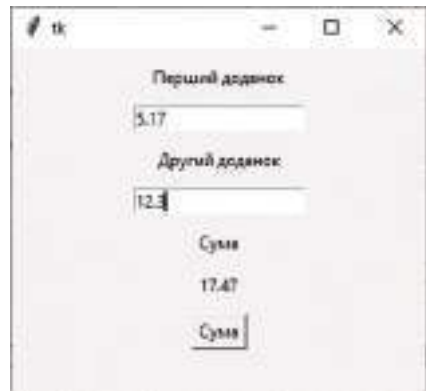
Значення змінних **x**, **y**, **z**, які використані в процедурі, за умовою задачі можуть бути довільними числами, тобто і цілими, і дробовими. Для таких значень можна використати тип **float**.

Під час виконання створеного проєкту користувач уводить у поля два числа (на малюнку 6.49 це числа $-2,7$ і $15,38$), вибирає кнопку із заголовком *Сума*, і результат ($12,68$) виводиться в напис. Не закриваючи вікно виконання проєкту, користувач може ввести в поля інші числа, знову вибрати кнопку із заголовком *Сума* й одержати новий результат. І так можна повторювати довільну кількість разів.

Щоб під час виконання проєкту для користувача було зрозуміліше, у які поля що саме потрібно вводити і де виводитиметься результат, доцільно у вікні розмістити поруч з кожним полем і написом, призначеним для виведення результату, написи з пояснювальними текстами, наприклад *Перший доданок*, *Другий доданок*, *Сума* (мал. 6.50).

Якщо змінні можуть набувати тільки цілі значення, то для них можна використовувати тип **int** (англ. *integer* – цілий).

Якщо в проєкті використовуватимуться змінні типу **int**, для читання їх значень з поля можна використати, наприклад, команду `x = int(entry1.get())`.



Мал. 6.50. Вікно виконання проєкту додавання двох чисел з пояснювальними текстами

У мові **Python** задачі з введенням даних і виведенням результатів можуть бути розв'язані без створення графічного інтерфейсу користувача.

Для введення даних під час виконання проекту в команді присвоювання використовують функцію `input(<текст>)`. Наприклад, після виконання команди:

```
x = float(input('Уведіть перший доданок: '))
```

- у вікно середовища **IDLE** буде виведено текст, що міститься в дужках у лапках;
- користувачу потрібно ввести значення першого доданка з клавіатури та натиснути клавішу **Enter**;
- уведене значення першого доданка (цілого або дробового) буде прийнято як текст, функцією `float` буде переведено в число, і це число присвоюється змінній `x`.

Якщо число, яке потрібно ввести, може бути тільки цілим, то потрібно замість `float` використати `int`.

Для виведення значень використовують команду `print(<значення>)`. Наприклад, команда:

- `print(z)` виведе значення змінної `z`;
- `print('Сума дорівнює')` виведе текст *Сума дорівнює*;
- `print('Сума дорівнює', z)` виведе текст *Сума дорівнює*, а після нього через пропуск значення змінної `z`.

Проект для розв'язування вищенаведеної задачі та результат його виконання в середовищі **IDLE** наведені на малюнку 6.51.

```
x = float(input('Уведіть перший доданок: '))
y = float(input('Уведіть другий доданок: '))
z = x + y
print('Сума дорівнює', z)
```

```
Python 3.7.0 (v3.7.0:1bf9ccc5093
[MSC v.1914 32 bit (Intel)] on
Type "copyright", "credits" or
imulation.
>>>
RESTART: E:\Підручники\8 клас
.py input.py
Уведіть перший доданок: 12.3
Уведіть другий доданок: 7.25
Сума дорівнює 19.55
>>> |
```

Мал. 6.51. Проект для знаходження суми двох чисел (ліворуч) і результат його виконання (праворуч)

Тип результатів арифметичних операцій над числами

Ви вже знаєте, що в проєктах в арифметичних виразах використовуються такі знаки арифметичних операцій:

+ – додавання; – – віднімання; * – множення; / – ділення.



Для тих, хто вивчає Pascal

Якщо в командах `x := a + b`, `x := a - b`, `x := a * b` і `x := a / b` змінні `a` і `b` є змінними типу `real`, то і тип змінної `x` має бути визначений як `real`.

Якщо в командах `x := a + b`, `x := a - b`, `x := a * b` змінні `a` і `b` є змінними типу `integer (longint)`, то тип змінної `x` може бути визначений і як `integer (longint)`, і як `real`.

А от якщо в команді `x := a / b` змінні `a` і `b` є змінними типу `integer (longint)`, то тип змінної `x` має бути визначений як `real`.



Для тих, хто вивчає Python

Тип результату обчислення значень арифметичних виразів у мові **Python** залежить від типу величин, що є операндами в цьому виразі.

Так, якщо обидва операнди є величинами цілого типу, то в результаті їх додавання, віднімання та множення отримаємо значення цілого типу.

Якщо хоча б один з операндів дійсного типу, то і результат виконання названих операцій буде також дійсного типу.

Результат ділення величин як цілого, так і дійсного типу завжди є дійсним числом.

Використання в проєкті констант (постійних величин)

У проєкті, крім змінних, можна використовувати **константи** (постійні величини).

Створимо проєкт, у якому потрібно обчислити масу сосни, якщо відомий її об'єм. Він може бути важливим під час вирішення питання, яка вантажівка потрібна для перевезення сосен на деревообробний завод.

Побудуємо математичну модель для цього проєкту:

вхідні дані: об'єм сосни V ;

кінцеві результати: маса сосни m ;

формули: $m = \rho \cdot V$, де ρ – густина сосни – константа, що дорівнює 520 кг/м^3 .



Для тих, хто вивчає Pascal

Вважатимемо, що в проєкті значення об'єму вводиться в кубічних метрах у поле з іменем **Edit1**, а значення маси виводиться в кілограмах у напис з іменем **Label1**.

```
const p = 520;
var v, m: real;
begin
  v := StrToFloat(Edit1.Text);
  m := p*v;
  Label1.Caption := FloatToStr(m);
end;
```



Для тих, хто вивчає Python

Вважатимемо, що в проєкті значення об'єму вводиться в кубічних метрах у поле з іменем *entry*, а значення маси виводиться в кілограмах у напис з іменем *label*.

У мові **Python** не можна створити сталі величини (константи), як в інших мовах програмування, щоб для них було заборонено змінювати значення величини. Натомість серед програмістів є домовленість записувати імена величин, значення яких не рекомендується змінювати, великими літерами.

Тоді відповідний фрагмент проєкту виглядатиме так:

```
P = 520
v = float(entry.get())
m = P * v
label['text'] = str(m)
```



Працюємо з комп'ютером




Виконайте тренувальні завдання за посиланням <https://cutt.ly/YhdrsFE> або QR-кодом.

**Найважливіше в цьому пункті**

Поле – це компонент (елемент керування), який використовується для введення даних і виведення результатів.

**Для тих, хто вивчає Pascal**

Щоб розмістити поле на формі, потрібно підвести вказівник до кнопки **TEdit**  на вкладці **Standard** вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші.

Властивості поля **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font** аналогічні до відповідних властивостей напису. Значення властивості **Width**, аналогічно до форми і кнопки, можна змінити або на вкладці **Властивості**, або перетягуванням спеціальних маркерів на межах поля. Значення властивостей поля **Height** не можна змінити, якщо значення властивості **AutoSize** дорівнює **True**.

Крім того, значення властивостей поля можна встановлювати під час виконання проекту командами присвоювання, наприклад,

Edit1.Top := 200, Edit1.Left := Edit1.Left - 50, Edit1.Enabled := False.

Текст у полі є значенням його властивості **Text** (англ. *text* – текст). Його можна вводити й редагувати у відповідному полі вкладки **Властивості**. Часто перед початком виконання проекту значення цієї властивості доцільно зробити порожнім. Значення цієї властивості також можна встановити під час виконання проекту командою присвоювання, наприклад **Edit1.Text := 'Текстове поле'**.

У мові програмування **Pascal** для будь-якої змінної повинен бути вказаний її **тип**. **Тип змінної** визначає:

- які значення може набувати ця змінна;
- які операції над нею можна виконувати;
- який обсяг оперативної пам'яті буде виділено для зберігання значення цієї змінної.

Якщо значення змінної можуть бути і цілими, і дробовими числами, то ця змінна повинна мати тип **real**. Якщо значення змінної можуть бути тільки цілими числами, то ця змінна повинна мати тип **integer** або **longint**. Імена змінних з їх типами записуються в процедурі в рядку **var**, який потрібно розмістити між рядком заголовка і словом **begin**.

Прочитати число з поля та присвоїти його змінній можна командою **x := StrToFloat (Edit1.Text)** або командою **x := StrToInt (Edit1.Text)**.

Вивести значення змінної в напис можна командою **Label1.Caption := FloatToStr (z)**, а в поле – командою **Edit1.Caption := FloatToStr (z)**.

У проекті можна використовувати **константи** (постійні величини). Імена констант та їх значення записуються в рядочку **const**, який потрібно розмістити після рядка заголовка процедури.

**Для тих, хто вивчає Python**

Команда для створення поля в мові **Python** має вигляд:

<ім'я поля> = Entry(<набір властивостей та їх значень>)

Властивості поля **width**, **bg**, **fg**, **font** аналогічні до відповідних властивостей напису та кнопки, а властивість **height** відсутня – висота поля визначається висотою символів у його вмісті. Для поля можна задавати товщину рамки як значення властивості **bd**.



Для розміщення поля у вікні проєкту можна використовувати методи `pack()` і `place()`.

Щоб вивести в поле деякий текст, використовують команду виду:

`<ім'я поля>.insert(<позиція>, <текст>)`.

Отримати з поля його вміст можна, використовуючи метод `get()`, наприклад `label['text'] = entry.get()`.

Для очищення вмісту поля з іменем `entry` використовують команду:

`entry.delete(0, END)`.

Як і з іншими об'єктами вікна, з полем можна пов'язати процедури – обробники подій, використовуючи метод `bind()`.

Прочитати число з поля та присвоїти його змінній можна командою `x = float(entry1.get())` або `x = int(entry1.get())`.

Вивести значення змінної в напис можна командою `label['text'] = str(z)`, а в поле – командою `entry.insert(0, z)`.

У мові **Python** задачі з уведенням даних і виведенням результатів можуть бути розв'язані без створення графічного інтерфейсу користувача.

Для введення даних під час виконання проєкту в команді присвоювання використовують функцію `input('<текст>')`, наприклад, `x = float(input('Уведіть перший доданок: '))` або `x = int(input('Уведіть перший доданок: '))`.

Для виведення значень використовують команду `print(<значення>)`, наприклад `print('Сума дорівнює', z)`.




Дайте відповіді на запитання

- 1°. Для чого може використовуватися поле?
- 2°. Як розмістити поле на формі?
- 3°. Що визначають значення властивостей **Top**, **Left**, **Name**, **Color**, **Text**, **Enabled** і **Visible** для поля?
- 4°. Чим відрізняються властивості **Top** і **Left** для поля і для форми?
- 5°. Чим відрізняються властивості **Height** і **Width** для поля і для напису?
- 6°. Як створити поле в мові **Python**?
- 7°. Які властивості поля ви знаєте? Охарактеризуйте їх.
- 8°. Як отримати значення з поля; записати в поле; видалити дані з поля?
- 9°. Як можна змінити текст у полі до виконання проєкту й під час його виконання?
- 10°. Якими командами можна перевести текстове подання числа безпосередньо в число й навпаки?
- 11°. Що визначає тип змінної? Які типи змінних ви знаєте? Чим вони відрізняються один від одного?
- 12°. Де та як вказуються імена змінних та їх типи?
- 13°. Чим константа відрізняється від змінної? Як вказується ім'я та значення константи?
- 14°. Як увести та вивести дані під час виконання проєкту без графічного інтерфейсу користувача?




Виконайте завдання


- 1°. Створіть проєкт, у якому в результаті настання події **Click** для поля буде встановлено відступ його верхньої межі від верхньої межі вікна 100 пікселів ( ширину поля – 15 символів) і змінено текст у полі на слово *Інформа-*

тика. Збережіть проєкт у папці з іменем **Завдання 6.5.1**, створеній у вашій папці.




- 2°. Створіть проєкт, у якому в результаті настання події **KeyPress** для поля буде встановлено відступ його лівої межі від лівої межі вікна 120 пікселів ( товщину рамки поля – 3 пікселя) і змінено текст у полі на ваше ім'я.

Збережіть проєкт у папці з іменем **Завдання 6.5.2**, створеній у вашій папці.

- 3°. Створіть проєкт, у якому в результаті настання події **MouseMove** для поля буде змінено текст у ньому, встановлено червоний колір тексту, сірий колір фону і поле переміститься на 100 пікселів управо і на 80 пікселів униз ( ширина поля збільшиться на 5 символів). Збережіть проєкт у папці з іменем **Завдання 6.5.3**, створеній у вашій папці.



- 4°. Створіть проєкт, у якому в результаті настання події **DbClick** для поля воно переміститься на 80 пікселів управо і на 20 пікселів уверх ( товщина рамки поля збільшиться на 5 пікселів), виведеться в поле ваше прізвище та ім'я зеленим кольором на жовтому тлі. Збережіть проєкт у папці з іменем **Завдання 6.5.4**, створеній у вашій папці.

- 5°. Створіть проєкт, у якому поле зафарбується в сірий колір, переміститься на 50 пікселів управо і стане недоступним, якщо під час виконання проєкту виправляється текст у ньому. Збережіть проєкт у папці з іменем **Завдання 6.5.5**, створеній у вашій папці.

- 6°. Побудуйте математичну модель і створіть проєкт для обчислення значення виразу $(2 * a + b) * (a + c)$. Збережіть проєкт у папці з іменем **Завдання 6.5.6**, створеній у вашій папці.



- 7°. Побудуйте математичну модель і створіть проєкт для обчислення значення виразу $(3 * a - b) * (a + 2 * b)$. Збережіть проєкт у папці з іменем **Завдання 6.5.7**, створеній у вашій папці.

- 8°. Побудуйте математичну модель і створіть проєкт для обчислення довжини кола, використовуючи константу. Збережіть проєкт у папці з іменем **Завдання 6.5.8**, створеній у вашій папці.

- 9°. Побудуйте математичну модель і створіть проєкт для обчислення шляху, який проходить світло за заданий час. Збережіть проєкт у папці з іменем **Завдання 6.5.9**, створеній у вашій папці.

- 10°. Побудуйте математичну модель і створіть проєкт для визначення, яку суму грошей має заплатити споживач за спожиту електроенергію. Збережіть проєкт у папці з іменем **Завдання 6.5.10**, створеній у вашій папці.

- 11°. Вкладник поклав у банк x грн під p % річних. Після закінчення року він забирає прибуток. Побудуйте математичну модель і створіть проєкт для визначення, який прибуток отримає вкладник через 3 роки. Збережіть проєкт у папці з іменем **Завдання 6.5.11**, створеній у вашій папці.

- 12°. Клієнт поклав у банк x грн під p % річних. За умовою банку прибуток за рік додається до вкладу. Побудуйте математичну модель і створіть проєкт для визначення, який прибуток він отримає через 2 роки. Збережіть проєкт у папці з іменем **Завдання 6.5.12**, створеній у вашій папці.

- 13°. Автомобіль проїхав t годин зі швидкістю v км/год. На кожні 100 км шляху він витрачав k літрів пального. Ціна пального складає p грн за літр. Побудуйте математичну модель і створіть проєкт для визначення витрат на пальне для цієї поїздки. Збережіть проєкт у папці з іменем **Завдання 6.5.13**, створеній у вашій папці.



ПРАКТИЧНА РОБОТА № 10

«Проекти з полем, з уведенням даних і виведенням результатів»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєвості та санітарно-гігієнічних норм.

Задача 1. Створіть проєкт з використанням поля і кнопки. Установіть такі початкові значення властивостей поля:

**Для тих, хто вивчає Pascal**

- ширина – 400 пікселів;
- відступ від лівої межі форми – 70 пікселів;
- відступ від верхньої межі форми – 100 пікселів;
- текст у полі – назва вашого класу.

Створіть обробник події **Click** для кнопки, виконання якого встановить сірий колір тла поля, відступ поля від верхньої межі вікна – 150 пікселів, зменшить відступ поля від лівої межі вікна на 50 пікселів, установить колір тексту синій, виведе в поле текст *Ми використовуємо властивості поля!*

**Для тих, хто вивчає Python**

- ширина – 15 символів;
- товщина рамки – 5 пікселів;
- відступ від верхньої межі вікна – 30 пікселів;
- текст у полі – назва вашого класу.

Створіть обробник події **Click** для кнопки, виконання якого встановить червоний колір фону поля, шрифт символів *Arial*, розмір символів 14, збільшить ширину поля на 6 символів, установить колір тексту білий і виведе в поле текст *Ми використовуємо властивості поля!*

Збережіть проєкт у папці з іменем **Практична 10_1**, створеній у вашій папці.




Задача 2. Складіть проєкт для обчислення значення виразу $a * (4 * b - a)$.

1. Побудуйте математичну модель для розв'язування задачі. Визначте вхідні дані, кінцеві результати та формули. Запишіть її в зошит.
2. Визначте, які компоненти потрібно розмістити на формі для введення вхідних даних і виведення результатів. Розмістіть їх на формі. Підпишіть їх.
3. Створіть обробник події **Click** для кнопки для обчислення значення виразу і виведення результату.
4. Збережіть проєкт у папці з іменем **Практична 10_2**, створеній у вашій папці.


Задача 3. Мати дала Марійці певну суму грошей і попросила купити 400 г печива, 2 пакети молока і хліб. Складіть проєкт для обчислення суми грошей, яка залишиться в Марійки.

1. Побудуйте математичну модель для розв'язування задачі. Визначте вхідні дані, кінцеві результати та формули. Запишіть її в зошит.
2. Визначте, які компоненти потрібно розмістити на формі для введення вхідних даних і виведення результатів. Розмістіть їх на формі. Підпишіть їх.
3. Створіть обробник події **Click** для кнопки для обчислення результату й виведення його.
4. Збережіть проєкт у папці з іменем **Практична 10_3**, створеній у вашій папці.

6.6. Налаштування проекту

1. Що таке *компіляція* ( *інтерпретація*) проекту?
2. Яке призначення програми-компілятора ( програми-інтерпретатора)? Які особливості її роботи?
3. Які помилки називаються синтаксичними? Як вони виявляються на етапі компіляції ( інтерпретації) проекту?

Тестування проекту

Ви вже знаєте, що під час створення проекту автор/авторка може допускати синтаксичні помилки. Синтаксичні помилки, які не помітили автор чи авторка проекту, знаходить програма-компілятор на етапі компіляції під час запуску проекту на виконання ( інтерпретатор на етапі інтерпретації команди). Потрібно виправити всі такі помилки і знову запустити на виконання нову версію проекту.

Але інколи помилки проявляються на етапі виконання проекту або після закінчення його виконання. Наприклад, виконання проекту може за якоїсь причини несподівано перерватися. Або результат його виконання може виявитися неправильним. У таких і в аналогічних випадках говорять, що проект містить **логічні помилки**.



Одним зі способів виявлення логічної помилки є створення і використання **тестових наборів** вхідних даних. Для кожного із цих наборів даних визначають результат до початку виконання проекту та порівнюють його з результатом виконання проекту. У разі розбіжностей цих результатів або в разі переривання виконання проекту потрібно знайти команди, які є причиною цих розбіжностей або є причиною переривання виконання проекту.

Процес пошуку логічних помилок у проекті з використанням тестових наборів вхідних даних називають **тестуванням проекту**.

Процес виправлення помилок у проекті називають **налагодженням проекту**.

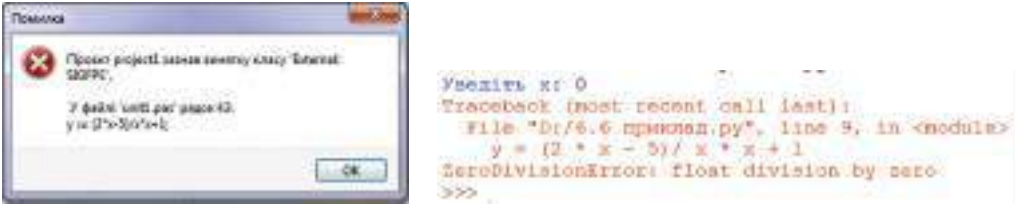
Тестування лінійних фрагментів проекту

Розглянемо приклад тестування лінійного фрагмента проекту. Нехай потрібно скласти проект для обчислення значень функції $y = \frac{2x - 5}{x^2 + 1}$ при різних значеннях змінної x . Фрагмент проекту для обчислення значень цієї функції виглядатиме так:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre> procedure TForm1.Button1Click(Sender: TObject); var x, y: real; begin x := StrToFloat(Edit1.Text); y := (2*x-5)/(x*x+1); Label1.Caption := FloatToStr(y); end; </pre>	<pre> x = float(input('Уведіть x: ')) y = (2*x-5)/(x*x+1) print(y) </pre>



Часто учні, записуючи команду присвоювання для обчислення значення відповідного виразу $y := (2 * x - 5) / (x * x + 1)$, забувають узяти знаменник (а то й чисельник) у дужки, тобто вводять, наприклад, команду $y := (2 * x - 5) / x * x + 1$ або $y := 2 * x - 5 / x * x + 1$. Таку логічну помилку нескладно помітити, застосувавши вдалу добірку тестів. Так, під час тестування для $x = 0$ стане очевидним, що проєкт не видає очікуваного правильного результату -5 , а несподівано перериває своє виконання, повідомляючи про помилку: номер і вміст рядка з помилкою (мал. 6.52).



Мал. 6.52. Повідомлення про помилку




Для тих, хто вивчає Pascal

Після вибору кнопки **ОК** вікно повідомлень закривається, а рядок з помилкою в тексті проєкту стає зафарбованим (мал. 6.53).

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text);
  y := (2*x-5)/x*x+1;
  Label1.Caption := FloatToStr(y);
end;
```

Мал. 6.53. Текст проєкту із зафарбованим рядком, що містить помилку

У такому разі, щоб повернутися до режиму редагування проєкту, потрібно натиснути сполучення клавіш **Ctrl+F2** або вибрати кнопку  **Завершити**.

Далі потрібно зрозуміти, у чому полягає помилка, виправити її і знову запустити проєкт на виконання.

У розглянутому випадку це відбувається тому, що в ході виконання проєкту зі значенням $x = 0$ має виконатися операція ділення на 0, яку виконати неможливо. Якщо як тестовий приклад буде взято інше значення x , наприклад -1 , то виконання проєкту не переривається, повідомлення про помилку не виводиться, але як результат буде виведено число -6 , хоча очікуваним правильним результатом, який потрібно визначити до виконання проєкту, є число $-3,5$. У такому разі також потрібно уважно переглянути текст проєкту, виявити помилку та виправити її.

Коментарі

Досить часто для полегшення пошуку логічних помилок і надання пояснень у тексті програми для її користувачів до проєкту додають *коментарі*. **Коментар** – це текст, який ігнорується компілятором (інтерпретатором) і включається до тексту проєкту з метою полегшення його розуміння та пошуку логічних помилок.



Для тих, хто вивчає Pascal

Для позначення коментарів їх потрібно брати у фігурні дужки, наприклад {Команда обчислення значення y }, або в круглі дужки із зірочками, наприклад (*Команда виведення значення y *) (мал. 6.54).

Якщо текст коментаря короткий і не займає більше одного рядка, то можна замість дужок поставити перед ним дві похилі риски, наприклад //Команда введення значення x (мал. 6.54).

Якщо коментар займає не більше одного рядка, то він може знаходитися після команди, а може розпочинатися з початку рядка. Будь-який коментар не може знаходитися всередині команди.

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text); //Команда введення значення x
  {Команда обчислення значення y}
  y := (2*x-5)/x*x+1;
  (*Команда виведення значення y*)
  Label1.Caption := FloatToStr(y);

end;
```

Мал. 6.54. Текст проєкту з коментарями

Під час тестування проєкту іноді можна тимчасово не виконувати певні команди. У такому разі їх потрібно перетворити на коментар.

Так, наприклад, можна закоментувати рядок з командою $y := (2*x-5)/x*x+1$. А замість неї до тексту проєкту включити команди виведення, наприклад у вікно повідомлень значень чисельника і знаменника дробу.

Після виправлення логічної помилки потрібно проєкт повернути до початкового вигляду.



Для тих, хто вивчає Python

Коментарем у проєкті мовою **Python** є частина рядка від символу **#** до кінця рядка. Він може розміщуватися після команди, а може міститися в окремому рядку (мал. 6.55). Коментар не може знаходитися всередині команди.

У мові **Python** не передбачено коментарі, що займають кілька рядків. Але якщо потрібно додати до проєкту опис, що займає кілька рядків, то його можна подати як текст, розмістивши на початку проєкту. Перед текстом і після нього потрібно поставити по три символи лапок (мал. 6.55).

Під час тестування проєкту іноді можна тимчасово не виконувати певні команди.

```
''' Проєкт для обчислення значення виразу
2*x - 5
y = (2*x - 5) / (x*x + 1)
'''
# Команда введення значення x
x = float(input('Введіть x: '))
# Команда обчислення значення y
y = (2 * x - 5) / (x * x + 1)
print(y) # Виведення значення y
```

Мал. 6.55. Текст проєкту з коментарями

У такому разі їх потрібно перетворити на коментар. Для цього слід виділити рядки з такими командами та натиснути **Alt+3**. На початку виділених рядків з'являться символи **##** і рядки вважатимуться коментарями. Коли потрібно буде знов виконувати ці команди, їх слід виділити та натиснути **Alt+4**. Символи **##** у виділених рядках зникнуть.



Працюємо з комп'ютером

Виконайте тренувальні завдання за посиланням <https://cutt.ly/Ehdrd8p> або QR-кодом.




Найважливіше в цьому пункті

Логічна помилка – це помилка в проєкті, що призводить до несподіваного переривання його виконання або до неправильного результату його виконання.

Тестовий набір вхідних даних – це набір значень вхідних даних, для яких можна заздалегідь визначити правильний результат виконання проєкту, після чого потрібно порівняти його з реальним результатом виконання проєкту для цих значень вхідних даних.

Тестування проєкту – це процес пошуку логічних помилок у проєкті з використанням тестових наборів вхідних даних.


Налагодження проєкту – це процес виправлення помилок у проєкті.

Коментар – це текст, який ігнорується компілятором ( інтерпретатором) і включається до тексту проєкту з метою полегшення його розуміння та пошуку логічних помилок.



Для тих, хто вивчає Pascal

Коментар потрібно взяти у фігурні дужки `{}`, або в круглі дужки із зірочкою `(* *)`, або, якщо коментар не довше одного рядка, відокремити від команд двома похилими рисками `//`.

Виконання проєкту можна перервати, натиснувши сполучення клавіш **Ctrl+F2** або вибравши кнопку  **Завершити**.



Для тих, хто вивчає Python

Коментарем у мові **Python** є частина рядка від символу `#` до кінця рядка. Він може розміщуватися після команди, а може міститися в окремому рядку. Коментар не може знаходитися всередині команди. Якщо потрібно додати до проєкту опис, що займає кілька рядків, то його можна подати як текст, розмістивши на початку тексту та після його завершення три символи лапок.

Щоб перетворити рядки з командами на коментар, потрібно виділити рядки з такими командами та натиснути **Alt+3**. Щоб видалити коментарі з деяких рядків, потрібно ці рядки виділити та натиснути **Alt+4**.




Дайте відповіді на запитання

- 1•. Які помилки в проєкті називаються логічними?
- 2◦. У чому полягає процес тестування проєкту?
- 3•. Для чого потрібно тестувати проєкт?
- 4•. Що таке налагодження проєкту?
- 5◦. Що таке коментар у проєкті? Яким чином можна коментар включити до тексту проєкту?
- 6•. Для чого до тексту проєкту включають коментарі?



Виконайте завдання

- 1*. Відкрийте проект, наприклад з папки **Розділ 6\Пункт 6.6\Проект 6.6.1**, призначений для розв'язування такої задачі: «*Два поїзди виїхали одночасно назустріч один одному. Визначити, через який час вони зустрінуться, якщо відомо відстань між ними в момент початку руху та швидкості кожного поїзда*». Для заданих вхідних даних обчисліть результати та запишіть у зошит:
 - а) відстань – 1000 км, швидкості – 120 км/год і 130 км/год;
 - б) відстань – 600 км, швидкості – 80 км/год і 120 км/год;
 - в) відстань – 1200 км, швидкості – 115 км/год і 85 км/год.
 Виконайте проект для першого з наведених наборів вхідних даних. Порівняйте результат виконання проекту з очікуваним. Протестуйте проект і знайдіть логічну помилку. Виправте її. Виконайте проект для інших двох наведених наборів вхідних даних. Порівняйте результати виконання проекту з очікуваними. Збережіть виправлений проект у папці з іменем **Завдання 6.6.1**, створеній у вашій папці.
- 2*.  Складіть три тестових набори вхідних даних для тестування проекту, призначеного для розв'язування такої задачі: «*Два поїзди виїхали одночасно з однієї станції у протилежних напрямках. Знайти відстань між поїздами через заданий час, якщо відомо швидкості кожного з них*». Обчисліть очікувані результати за складеними тестами. Відкрийте проект, наприклад з папки **Розділ 6\Пункт 6.6\Проект 6.6.2**. Протестуйте його на ваших наборах тестових даних. Зробіть висновок, чи містить проект логічні помилки. Якщо так, виправте їх. Збережіть виправлений проект у папці з іменем **Завдання 6.6.2**, створеній у вашій папці.
- 3*. Учні для свята купили цукерки та печиво. Відома ціна цукерок і печива та маса куплених солодоців кожного виду. Створіть проект для знаходження повної вартості покупки. Збережіть проект у папці з іменем **Завдання 6.6.3**, створеній у вашій папці. Запишіть набори вхідних даних для тестування проекту.
- 4*. Відкрийте проект, наприклад з папки **Розділ 6\Пункт 6.6\Проект 6.6.4**. Перегляньте текст проекту і визначте, яку задачу розв'язує цей проект. Складіть три тестових набори вхідних даних для тестування проекту. Обчисліть очікувані результати. Протестуйте проект на ваших тестових даних. Зробіть висновок, чи містить проект логічні помилки. Якщо так, виправте їх. Збережіть виправлений проект у папці з іменем **Завдання 6.6.4**, створеній у вашій папці.

6.7. Логічні вирази. Змінні логічного типу. Логічні операції



1. Що таке *висловлювання*? Яке висловлювання вважається *істинним*, а яке – *хибним*?
2. Які властивості змінної ви знаєте?
3. Що визначає тип змінної? Які типи числових змінних ви знаєте?

Логічні вирази

Ви вже знаєте, що **висловлювання** – це речення, яке містить твердження про певний об'єкт або про зв'язки між об'єктами і про яке можна однозначно сказати, **істинне** воно чи **хибне**. Приклади **істинних висловлювань**: «*Київ – столиця*



України», $2 * 5 - 4 = 6$, $4 < 12$. Приклади **хибних висловлювань**: «У січні 30 днів», $2 * 7 + 3 = 12$, $3 > 5$. Приклади речень, які не є висловлюваннями: «Закрий, будь ласка, вікно», «Тобі сподобався цей фільм?».

Висловлювання можна розглядати як **логічний вираз**.

Логічним виразом називається вираз, який може набувати одне з двох значень: **true** (англ. *true* – істина, істинність, правда) або **false** (англ. *false* – хиба, хиби́ність, неправда).



Логіка (грец. λόγος – слово, смисл, думка, мова) – наука про форми, методи й закони правильного мислення, про способи міркування.

Якщо висловлювання істинне, то вважають, що значення відповідного логічного виразу дорівнює **true**. Тобто «Київ – столиця України» = **true**, $(2 * 5 - 4 = 6)$ = **true**, $(4 < 12)$ = **true**.

Якщо висловлювання хибне, то вважають, що значення відповідного логічного виразу дорівнює **false**. Тобто «Цей підручник для учнів сьомого класу» = **false**, $(2 * 7 + 3 = 12)$ = **false**, $(3 > 5)$ = **false**.


Звертаємо вашу увагу, що логічні вирази використовують у розгалуженнях і циклах. Розглянемо таке використання в наступних пунктах.

Змінні логічного типу

У попередніх пунктах ми використовували змінні, значеннями яких є числа. Нагадаємо, що ми використовували змінні типу **real** ( **float**), значеннями яких можуть бути будь-які числа (цілі та дробові), а також змінні типу **integer** ( **int**), значеннями яких можуть бути тільки цілі числа.



В інформатиці використовують також змінні, значеннями яких можуть бути **true** або **false**.


Змінні, які можуть набувати одного з двох значень **true** або **false**, називаються **змінними логічного типу**.

Логічний тип змінних позначається **boolean** ( **bool**).



Звертаємо вашу увагу, що змінні логічного типу є одним з видів логічних виразів.

Надавати змінним логічного типу відповідні значення можна командами присвоювання. Наприклад:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<code>x := true</code> або <code>x := false</code>	<code>x = True</code> або <code>x = False</code>

Змінній логічного типу можна присвоювати значення іншої логічної змінної, наприклад `x := y` ( `x = y`).

Змінній логічного типу можна також присвоювати значення логічного виразу. Тобто можна використовувати такі команди присвоювання:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<code>x := 2 + 12 / 3 = 6</code> <code>y := 32 < 13</code>	<code>x = 2 + 12 / 3 == 6</code> <code>y = 32 < 13</code>

Тоді змінна **x** матиме значення **true**, а змінна **y** – **false**.

**Для тих, хто вивчає Pascal**

Змінні логічного типу також повинні бути описані в рядку **var**. Наприклад, **var x, y: boolean**.

**Для тих, хто вивчає Python**

Для перевірки, чи є рівними значення двох виразів, у мові **Python** використовують знак **==**.

Значення **True** та **False** потрібно писати з великої літери.

Логічні операції

Розглянемо три логічні операції: **заперечення**, **кон'юнкція** (лат. *conjunctio* – об'єднання) і **диз'юнкція** (лат. *disjunctio* – роз'єднання).

Запереченням логічного виразу **x** називається логічна операція, результат якої дорівнює **true**, якщо значення логічного виразу **x** дорівнює **false**, і дорівнює **false**, якщо значення логічного виразу **x** дорівнює **true**.

Заперечення логічного виразу **x** позначають **not x**. Інколи заперечення логічного виразу **x** позначається \bar{x} або $\neg x$.

Тобто, якщо **x = true**, то **not x = false**, і якщо **x = false**, то **not x = true**. Можна скласти таку таблицю, яку називають **таблицею істинності** для операції заперечення:

x	not x
true	false
false	true

Для заперечення висловлювання часто використовують частку **не**. Наприклад, результатом заперечення висловлювання **A** = «*Існує найбільше просте число*» є висловлювання **not A** = «*Не існує найбільшого простого числа*».

Кон'юнкцією двох логічних виразів **x** і **y** називається логічна операція, результатом якої є **true**, якщо значення кожного з логічних виразів **x** і **y** є **true**, і є **false**, якщо значення хоча б одного з логічних виразів **x** або **y** є **false**.

Кон'юнкція двох логічних виразів **x** і **y** позначається **x and y** (англ. *and* – і, та). Інколи кон'юнкція двох логічних виразів **x** і **y** позначається **x ∧ y**.

Таблиця істинності для кон'юнкції виглядає так:

x	y	x and y
true	true	true
true	false	false
false	true	false
false	false	false

Для утворення кон'юнкції двох висловлювань часто використовують сполучник **і**. Наприклад, висловлювання «*Число 36 кратне 3 і число 36 кратне 2*» є кон'юнкцією двох висловлювань: «*Число 36 кратне 3*» і «*Число 36 кратне 2*».

Диз'юнкцією двох логічних виразів **x** і **y** називається логічна операція, результатом якої є **true**, якщо значення хоча б одного з логічних виразів **x** або **y** є **true**, і є **false**, якщо значення кожного з логічних виразів **x** і **y** є **false**.



Диз'юнкція двох логічних виразів x і y позначається $x \text{ or } y$ (англ. *or* – або). Інкою диз'юнкція двох логічних виразів x і y позначається $x \vee y$.

Таблиця істинності для диз'юнкції виглядає так:

x	y	$x \text{ or } y$
true	true	true
true	false	true
false	true	true
false	false	false

Для утворення диз'юнкції висловлювань часто використовують сполучник **або**. Наприклад, висловлювання $11 \leq 21$ (11 менше або дорівнює 21) є диз'юнкцією двох висловлювань: $(11 < 21)$ **або** $(11 = 21)$.

Часто використовують **логічні вирази**, що містять кілька логічних операцій. Наприклад, **$x \text{ or } y \text{ and not } z$** .

У таких виразах для логічних операцій, які розглянули вище, визначено такий **пріоритет виконання операцій**: спочатку виконуються операції заперечення, потім – операції кон'юнкції, а за ними – операції диз'юнкції.

Для обчислення значень логічного виразу можна використати таблицю істинності. У ній потрібно розглянути всі можливі значення змінних логічного типу, що входять до цього виразу.

x	y	z	1	2	3
			<i>not z</i>	<i>y and not z</i>	<i>x or y and not z</i>
true	true	true	false	false	true
true	true	false	true	true	true
true	false	true	false	false	true
false	true	true	false	false	false
true	false	false	true	false	true
false	true	false	true	true	true
false	false	true	false	false	false
false	false	false	true	false	false

Для змінення порядку виконання логічних операцій використовують дужки. Так, наприклад, у логічному виразі **$(x \text{ or } y) \text{ and not } z$** спочатку буде виконано операцію диз'юнкції в дужках, потім операцію заперечення, а потім операцію кон'юнкції.



Цікаві факти з історії

Логіка як наука зародилася в Стародавній Греції. Засновником цієї науки вважається давньогрецький філософ **Арістотель** (384–322 до н. е.) (мал. 6.56).

Точні означення логічних операцій і формулювання їх властивостей були сформульовані англійським математиком і філософом **Джорджем Булем** (1815–1864) (мал. 6.57). У 1854 році вийшла його основна робота «Дослідження законів думки», на цих законах засновано математичні теорії логіки й імовірності. У ній досліджується система, яку сьогодні називають **Алгеброю висловлювань**, або **Булевою алгеброю**.

Булева алгебра стала фундаментальною науковою основою для створення комп'ютерів і використовується в усіх мовах програмування. Саме іменем цього вченого названо логічний тип даних у програмуванні.



Мал. 6.56.
Арістотель



Мал. 6.57.
Джордж Буль



Мал. 6.58.
Ірина Хоменко

Потрібно також відзначити значний внесок українських учених у розвиток логіки як науки: Яна Лукасевича (1878–1956), Павла Копніна (1922–1971), Клавдії Руденко (1919–1974), **Ірини Хоменко** (нар. 1961 р.) (мал. 6.58).



Найважливіше в цьому пункті

Висловлювання – це речення, яке містить твердження про певний об'єкт або про зв'язки між об'єктами та про яке можна однозначно сказати, **істинне** воно чи **хибне**.

Логічним виразом називається вираз, який може набувати одне з двох значень: **true** або **false**.

Висловлювання можна розглядати як **логічний вираз**. Якщо висловлювання істинне, то вважають, що значення відповідного логічного виразу дорівнює **true**. Якщо висловлювання хибне, то вважають, що значення відповідного логічного виразу дорівнює **false**.

Змінні, які можуть набувати одного з двох значень, **true** або **false**, називаються **змінними логічного типу**. Логічний тип змінних позначається **boolean** (🔗 **bool**). Змінні логічного типу також вважаються логічними виразами.

Над логічними виразами можна виконувати **логічні операції**: **заперечення**, **кон'юнкцію**, **диз'юнкцію**.

Запереченням логічного виразу **x** називається логічна операція, результатом якої дорівнює **true**, якщо значення логічного виразу **x** дорівнює **false**, і дорівнює **false**, якщо значення логічного виразу **x** дорівнює **true**.

Кон'юнкцією двох логічних виразів **x** і **y** називається логічна операція, результатом якої є **true**, якщо значення кожного з логічних виразів **x** і **y** є **true**, і є **false**, якщо значення хоча б одного з логічних виразів **x** або **y** є **false**.

Диз'юнкцією двох логічних виразів **x** і **y** називається логічна операція, результатом якої є **true**, якщо значення хоча б одного з логічних виразів **x** або **y** є **true**, і є **false**, якщо значення кожного з логічних виразів **x** і **y** є **false**.


У логічних виразах для логічних операцій, розглянутих вище, визначено такий **пріоритет виконання операцій**: спочатку виконуються операції заперечення, потім – операції кон'юнкції, а далі – операції диз'юнкції. Для змінення цього порядку виконання логічних операцій використовують дужки.

Для обчислення значень логічного виразу можна використати таблицю істинності. У ній потрібно розглянути всі можливі значення змінних логічного типу, що входять до цього виразу.

**Дайте відповіді на запитання**

- 1^о. Що таке висловлювання? Наведіть приклади істинних висловлювань, хибних висловлювань, невисловлювань.
- 2^о. Який вираз називається логічним? Наведіть приклади.
- 3^о. Які змінні називаються змінними логічного типу?
- 4^о. Як описуються змінні логічного типу?
- 5^о. Що таке заперечення логічного виразу?
- 6^о. Що таке кон'юнкція двох логічних виразів?
- 7^о. Що таке диз'юнкція двох логічних виразів?
- 8^о. Яким є пріоритет виконання логічних операцій?
- 9*. Що вивчає наука логіка?

**Виконайте завдання**

- 1^о. Наведіть приклади використання логічних виразів і логічних операцій у математиці, українській мові, інших предметах, у життєвих ситуаціях.
- 2^о. Сформулюйте заперечення висловлювань і з'ясуйте, істинні вони чи хибні:
 - а) Число 3 є дільником числа 545.
 - б) Автомобіль не має права їхати на червоне світло.
 - в) Існують паралелограми з прямими кутами.
 - г) Рівняння $2x - 3 = 0$ має цілий корінь.
 - д) Не існує натурального числа, що ділиться на 2.
 - е) Існує ціле число, яке ділиться на всі цілі числа.
- 3^о. Серед нижченаведених висловлювань знайдіть кон'юнкції та диз'юнкції та визначте, істинні вони чи хибні:
 - а) Число 27 кратне 3 і кратне 9.
 - б) $17 < 42 < 18$.
 - в) Число 2 просте або парне.
 - г) $\triangle ABC$ є гострокутним, прямокутним або тупокутним.
 - д) Діагоналі будь-якого паралелограма взаємно перпендикулярні та точкою перетину діляться навпіл.
 - е) $7^2 = 49$ і $(-7)^2 = 49$.
 - ж) $21 \leq 21$.
 - з) $21 \leq 18$.
- 4^о.  Нехай $A = \text{«Сергієнко вивчає англійську мову»}$, $B = \text{«Сергієнко має оцінку 8 з інформатики»}$. Сформулюйте висловлювання:
 - а) **A and not B** ;
 - б) **$\text{not } A$ or B** ;
 - в) **A or not A and B** ;
 - г) **$\text{not } A$ and not B and A** .
- 5^о. Укажіть порядок виконання логічних операцій у таких логічних виразах:
 - а) **a or not a and b** ;
 - б) **$\text{not } a$ or not b and a** ;
 - в) **$(a$ or not $b)$ and (not a or not $b)$** ;
 - г) **$\text{not } a$ and not b or c and d** .
- 6^о. Складіть таблиці істинності логічних виразів зі змінними логічного типу a і b :
 - а) **a or a and b** ;
 - б) **a or not a and b** ;
 - в) **a or b or not b** ;



- г) *(a or not b) and (a or not b)*;
- д) *not a or not b*;
- е) *a and (not b or a) and b*;
- ж) *a and (b or c) not a*;
- з) *a or b and c or not b*.

- 7*. Складіть таблиці істинності для таких висловлювань:
- а) Даринка любить кататися на скейті або Петрик займається легкою атлетикою і Марійка навчається в музичній школі.
 - б) Сьогодні дощ або вітер, і потрібно одягнути плащ.
 - в) Катруся має оцінки 8 з інформатики і української мови або оцінку 9 з історії України.
- 8*. Визначте істинність або хибність висловлювань для наведених значень змінних:
- а) $x > 0$ і $y > 0$ або $x < 0$ і $y < 0$; б) $x > 0$ і не $y < 0$ або $x < 0$ та $y > 0$.
- Значення змінних:
- 1) $x = 5, y = 8$; 2) $x = 5, y = -8$;
 - 3) $x = -5, y = 8$; 4) $x = -5, y = -8$.
- 9*. На виробництві є три установки. На кожній з них встановлено датчики. Якщо одночасно будуть перевищені граничні показники датчиків на першій установці і на будь-якій іншій або на всіх трьох установках, то виникає надзвичайна ситуація. У такому разі повинен пролунати попереджувальний сигнал. Для створення автоматизованої системи подавання цього сигналу складіть логічний вираз і для нього таблицю істинності, на основі якої буде працювати ця система.
- 10*. Мишко може погратися на комп'ютері, якщо він виконав домашні завдання або навів порядок у своїй кімнаті і завершила свою роботу на комп'ютері його старша сестра. У системі «Розумного будинку» можна фіксувати виконання кожного виду роботи. Це робить можливим автоматично отримувати повідомлення про надання доступу до комп'ютера. Складіть логічний вираз і для нього таблицю істинності, на основі якої працює система надсилання повідомлень.



6.8. Розгалуження

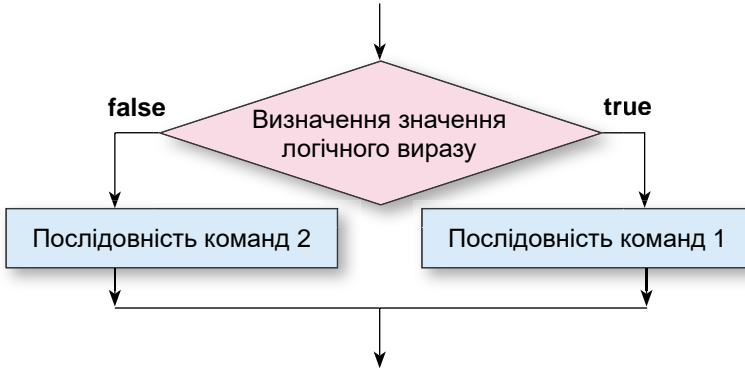


1. Які вирази називаються *логічними*?
2. Назвіть логічні операції, які ви знаєте. Що буде результатом кожної з них?
3. Що таке розгалуження в алгоритмі? Які види розгалуження ви знаєте? Чим вони відрізняються між собою?

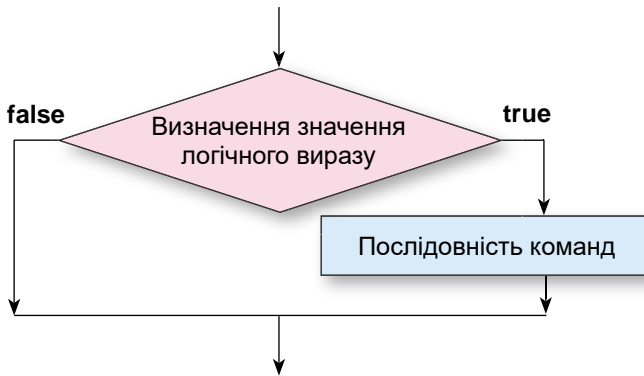
Алгоритми з розгалуженням

Ви вже знаєте, що алгоритми можуть містити розгалуження. Також відомо, що виконання розгалуження як фрагмента алгоритму починається з перевірки умови, результатом якої може бути або **істина** (**так**, **true**), або **хиба** (**ні**, **false**). І залежно від результату перевірки умови будуть виконуватися ті чи інші команди алгоритму. Також вам уже відомо, що умова є одним з прикладів логічного виразу.

Тому в загальному випадку виконання будь-якого розгалуження як фрагмента алгоритму розпочинається з визначення значення певного логічного виразу, яке може бути або **true**, або **false**. Блок-схеми повного і неповного розгалуження наведено на малюнках 6.59 і 6.60.





Мал. 6.59. Блок-схема повного розгалуження



Мал. 6.60. Блок-схема неповного розгалуження

Розгалуження в мовах програмування



Команда повного розгалуження в мовах програмування має такий загальний вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre> if <логічний вираз> then begin <послідовність команд 1> end else begin <послідовність команд 2> end;</pre>	<pre> if <логічний вираз>: <послідовність команд 1> else: <послідовність команд 2></pre>

(англ. *if* – якщо, *then* – тоді, *else* – інакше).

Виконання цієї команди відбувається так: визначається значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням; якщо це значення **false**, то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням.

Команда неповного розгалуження в мовах програмування має такий загальний вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>if <логічний вираз> then begin <послідовність команд> end;</pre>	<pre>if <логічний вираз>: <послідовність команд></pre>

Виконання цієї команди відбувається так: визначається значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд** і після цього виконується команда, наступна за розгалуженням; якщо це значення **false**, одразу виконується команда, наступна за розгалуженням.



Для тих, хто вивчає Pascal

Звертаємо вашу увагу:

- якщо послідовності команд містять тільки по одній команді, то операторні дужки **begin** і **end** можна не використовувати;
- перед **else** ставити крапку з комою не можна.



Для тих, хто вивчає Python

Ще однією формою команди розгалуження в мові **Python** є команда багатоальтернативного розгалуження:

```
if <логічний вираз 1>:
    <послідовність команд 1>
elif <логічний вираз 2>:
    <послідовність команд 2>
elif <логічний вираз 3>:
    <послідовність команд 3>
...
else:
    <послідовність команд N>
```

Виконання цієї команди відбувається так:

- ✓ визначається значення **логічного виразу 1**;
- ✓ якщо це значення **true**, то виконується **послідовність команд 1** й після цього виконується команда, наступна за розгалуженням;
- ✓ якщо це значення **false**, то обчислюється значення **логічного виразу 2**;
- ✓ якщо це значення **true**, то виконується **послідовність команд 2** й після цього виконується команда, наступна за розгалуженням;
- ✓ якщо це значення **false**, то обчислюється значення наступного логічного виразу і т. д.
- ✓ якщо всі логічні вирази в команді мають значення **false**, то виконується **послідовність команд N**, розміщена після **else**, і після цього виконується команда, наступна за розгалуженням.

Звертаємо вашу увагу:

- Послідовності команд, виконання яких залежить від значень логічних виразів, записують з відступом вправо відносно першої літери слів **if**, **elif** та **else**. Розмір відступу для кожної з команд має бути однаковим.



- Якщо послідовності команд містять тільки по одній команді, то цю команду можна записувати без переходу на новий рядок після двокрапки.

Розв'язування задач з використанням розгалуження

Розглянемо задачу, алгоритм розв'язування якої містить розгалуження.

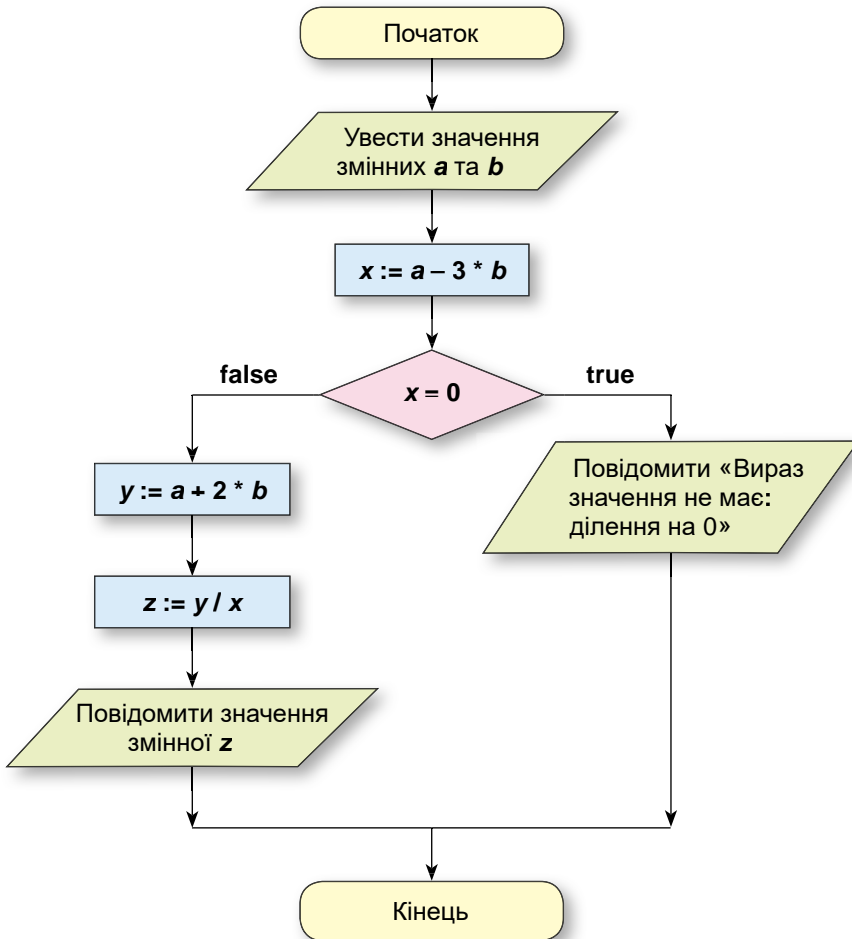
Задача 1. Обчислити значення виразу $z = \frac{a + 2b}{a - 3b}$, де a, b – довільні числа.

Побудуємо математичну модель для цієї задачі:

- **вхідні дані:** два довільних числа a та b ;
- **кінцеві результати:** значення виразу z ;
- **формули:** $z = (a + 2 * b) / (a - 3 * b)$.



Цей вираз містить дію ділення на вираз $a - 3 * b$. Дільник $a - 3 * b$ може дорівнювати нулю (наприклад, при $a = 3$ і $b = 1$), а може й не дорівнювати нулю (наприклад, при $a = 5$ і $b = 2$). Тому алгоритм розв'язування цієї задачі повинен містити розгалуження, у якому буде враховано два можливих випадки: значення дільника $a - 3 * b$ дорівнює 0 і значення дільника $a - 3 * b$ не дорівнює 0.

Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку 6.61.





Мал. 6.61. Блок-схема алгоритму розв'язування задачі 1

Алгоритм, наведений на малюнку 6.61, мовами програмування можна записати так:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre> a := StrToFloat(Edit1.Text); b := StrToFloat(Edit2.Text); x := a - 3 * b; if x = 0 then Label1.Caption := 'Вираз значення не має: ділення на 0' else begin y := a + 2 * b; z := y / x; Label1.Caption := FloatToStr(z); end;</pre>	<pre> a = float(input('Уведіть a: ')) b = float(input('Уведіть b: ')) x = a - 3 * b if x == 0: print('Вираз значення не має: ділен- ня на 0') else: y = a + 2 * b z = y / x print(z)</pre>

Звертаємо вашу увагу, що обчислення значення дільника $a - 3 * b$ можна було б не здійснювати в окремій команді і не запам'ятовувати як значення змінної x , а здійснити безпосередньо в команді розгалуження. Також можна обчислення значення виразу $(a + 2 * b) / (a - 3 * b)$ записати не двома командами, а однією. Тоді цей фрагмент проєкту матиме такий вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre> a := StrToFloat(Edit1.Text); b := StrToFloat(Edit2.Text); if a - 3 * b = 0 then Label1.Caption := 'Вираз значення не має: ділення на 0' else begin z := (a + 2 * b) / (a - 3 * b); Label1.Caption := FloatToStr(z); end;</pre>	<pre> a = float(input('Уведіть a: ')) b = float(input('Уведіть b: ')) if a - 3 * b == 0: print('Вираз значення не має: ділен- ня на 0') else: z = (a + 2 * b) / (a - 3 * b) print(z)</pre>

Для перевірки правильності проєкту можна виконати його, наприклад, для таких наборів вхідних даних:

- 1) $a = 6, b = 2$;
- 2) $a = 2, b = 3$.

Розглянемо задачу, під час розв'язування якої в команді розгалуження використовуються кон'юнкція.



Задача 2. Обчислити значення функції $y = \begin{cases} 2x - 12, & \text{якщо } -3 < x < 5; \\ 7 - 8x & \text{для всіх інших } x. \end{cases}$

Побудуємо математичну модель для цієї задачі:


- **вхідні дані:** довільне число x ;
- **кінцеві результати:** значення функції y ;
- **формули:** $y = \begin{cases} 2x - 12, & \text{якщо } -3 < x < 5; \\ 7 - 8x & \text{для всіх інших } x. \end{cases}$



Фрагмент проєкту для розв'язування цієї задачі має такий вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>x := StrToFloat(Edit1.Text); if (x > -3) and (x < 5) then y := 2 * x - 12 else y := 7 - 8 * x; Label1.Caption := FloatToStr(y);</pre>	<pre>x = float(input('Уведіть x: ')) if x > -3 and x < 5: y = 2 * x - 12 else: y = 7 - 8 * x print(y)</pre>

Для перевірки правильності проєкту, можна виконати його, наприклад, для таких значень: 1) $x = -4$; 2) $x = 0$; 3) $x = 7$.

Для розв'язування багатьох задач потрібно використати вкладені розгалуження ( багатоальтернативне розгалуження). Розглянемо одну з таких задач.

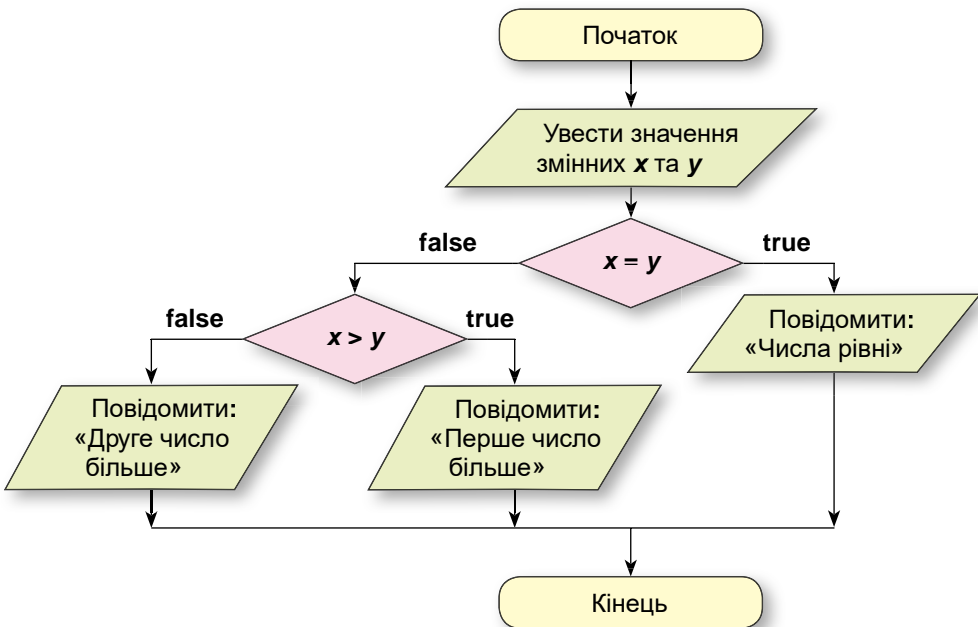
Задача 3. Дано два числа. Визначити, чи рівні вони. Якщо ні, то яке з них більше. Побудуємо математичну модель для цього проєкту:

- **вхідні дані:** два довільних числа x та y ;
- **кінцеві результати:** висновок: числа рівні, або перше число більше, або друге число більше;

- **формули:**



$$\begin{cases} \text{Числа рівні, якщо } x = y; \\ \text{Перше число більше, якщо } x > y; \\ \text{Друге число більше, якщо } x < y. \end{cases}$$

Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку 6.62.



Мал. 6.62. Блок-схема алгоритму розв'язування задачі 3

Ось як виглядає фрагмент відповідного проекту:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre> x := StrToFloat(Edit1.Text); y := StrToFloat(Edit2.Text); if x = y then Label1.Caption := 'Числа рівні' else if x > y then Label1.Caption := 'Перше число більше' else Label1.Caption := 'Друге чис- ло більше'; </pre>	<pre> x = float(input('Уведіть x: ')) y = float(input('Уведіть y: ')) if x == y: print('Числа рівні') elif x > y: print('Перше число більше') else: print('Друге число більше') </pre>

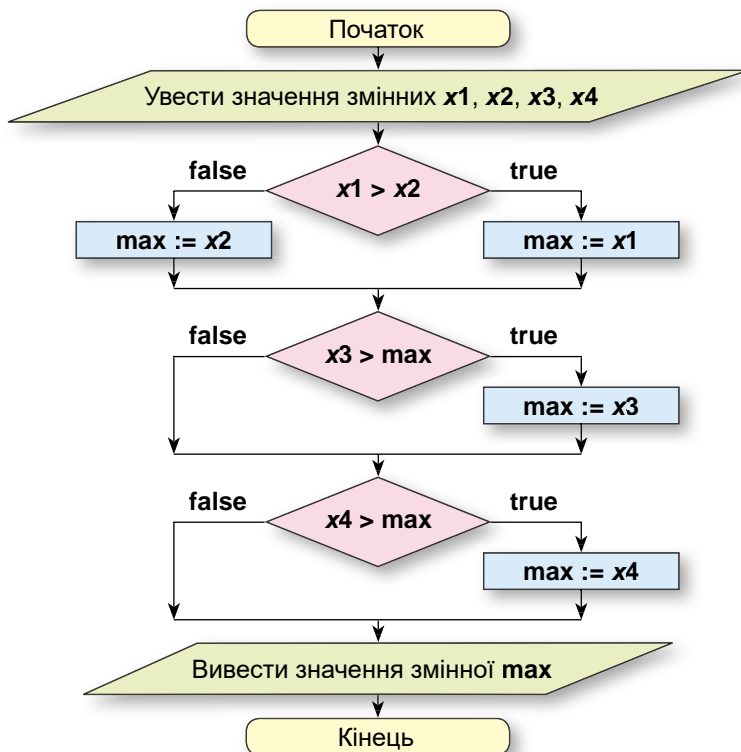
Для перевірки правильності проекту можна виконати його, наприклад, для таких наборів вхідних даних: 1) $x = 3, y = 3$; 2) $x = 5, y = 2$; 3) $x = 4, y = 10$.

Задача 4. Дано чотири нерівних між собою числа. Визначити найбільше з них. Побудуємо математичну модель для цієї задачі:

- **вхідні дані:** чотири довільних числа x_1, x_2, x_3, x_4 ;
- **кінцеві результати:** найбільше із цих чисел.

Для розв'язування цієї задачі визначимо спочатку більше з перших двох чисел і присвоїмо це значення деякій змінній, наприклад змінній *max*. А далі порівняємо кожне наступне число (третє і четверте) зі значенням цієї змінної. І якщо якесь число виявиться більше, ніж значення змінної *max*, замінимо значення цієї змінної на це число.



Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку 6.63.



Мал. 6.63. Блок-схема алгоритму розв'язування задачі 4



Відповідний проєкт має такий вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>x1 := StrToFloat(Edit1.Text); x2 := StrToFloat(Edit2.Text); x3 := StrToFloat(Edit3.Text); x4 := StrToFloat(Edit4.Text); if x1 > x2 then max := x1 else max := x2; if x3 > max then max := x3; if x4 > max then max := x4; Label1.Caption := FloatToStr(max);</pre>	<pre>x1 = float(input('Уведіть x1: ')) x2 = float(input('Уведіть x2: ')) x3 = float(input('Уведіть x3: ')) x4 = float(input('Уведіть x4: ')) if x1 > x2: max = x1 else: max = x2 if x3 > max: max = x3 if x4 > max: max = x4 print(max)</pre>

Створений проєкт працюватиме правильно і в тому разі, коли серед цих чотирьох чисел будуть рівні між собою числа. Перевірте це для чисел: 4, 4, 8, 2.

Тестування проєкту з розгалуженням

Тестувати проєкт з розгалуженням потрібно для таких наборів вхідних даних, щоб перевірити правильність роботи при всіх можливих варіантах його виконання.

Так, наприклад, проєкт для обчислення значення виразу $y = \frac{x-5}{x^2-9}$ містить команду розгалуження, бо його виконання залежить від того, дорівнює знаменник нулю чи не дорівнює. Тому його потрібно тестувати для значень x , при яких знаменник дорівнює нулю, тобто 3 і -3 , і при яких знаменник не дорівнює нулю, наприклад 5 і 2.

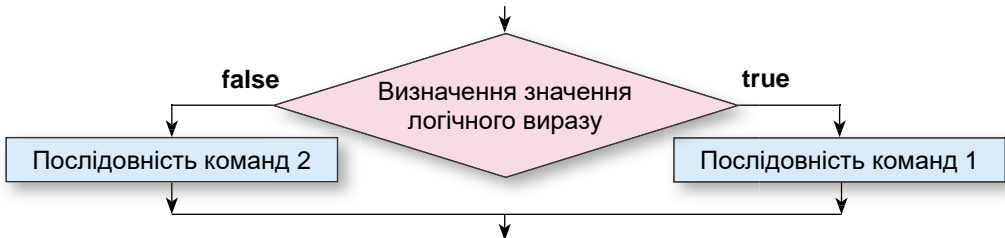
Працюємо з комп'ютером



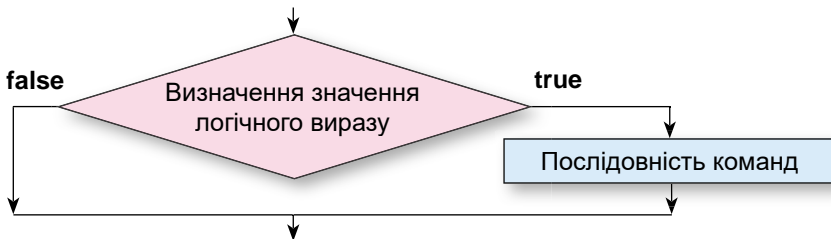
Виконайте тренувальні завдання за посиланням <https://cutt.ly/ehdrhey> або QR-кодом.

Найважливіше в цьому пункті

Блок-схеми повного і неповного розгалужень наведено на малюнках 6.64 і 6.65:





Мал. 6.64. Блок-схема повного розгалуження





Мал. 6.65. Блок-схема неповного розгалуження

Команда повного розгалуження в мовах програмування має такий вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>if <логічний вираз> then begin <послідовність команд 1> end else begin <послідовність команд 2> end;</pre>	<pre>if <логічний вираз>: <послідовність команд 1> else: <послідовність команд 2></pre>

Виконання цієї команди відбувається так: визначається значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням; якщо це значення **false**, то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням.

Команда неповного розгалуження в мовах програмування має такий вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>if <логічний вираз> then begin <послідовність команд> end;</pre>	<pre>if <логічний вираз>: <послідовність команд 1></pre>

Виконання цієї команди відбувається так: визначається значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд** і після цього виконується команда, наступна за розгалуженням; якщо це значення **false**, одразу виконується команда, наступна за розгалуженням.



Дайте відповіді на запитання

- 1°. Який вигляд має блок-схема повного розгалуження? Як виконується такий фрагмент алгоритму?
- 2°. Який вигляд має блок-схема неповного розгалуження? Як виконується такий фрагмент алгоритму?
- 3°. У чому полягають відмінності у виконанні повного і неповного розгалужень?
- 4°. Який вигляд має команда повного розгалуження в мові програмування? Як вона виконується?
- 5°. Який вигляд має команда неповного розгалуження в мові програмування? Як вона виконується?
- 6°. У чому полягають відмінності виконання команд повного і неповного розгалужень у мові програмування?
- 7*. У яких випадках в алгоритмах потрібно використовувати повне розгалуження, а в яких неповне?
- 8°. Як потрібно підбирати тестові приклади для тестування проєкту з розгалуженням?
- 9°. Який вигляд має команда багатоальтернативного розгалуження в Python? Як вона виконується?





Виконайте завдання

1*. Складіть блок-схему та поясніть виконання команди:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>if <логічний вираз 1> then begin <команди 1> end else if <логічний вираз 2> then begin <команди 2> end else begin <команди 3> end;</pre>	<pre>if <логічний вираз>: <команди 1> else: if <логічний вираз 2>: <команди 2> else: <команди 3></pre>

2*. Складіть блок-схему та поясніть виконання послідовності команд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>if <логічний вираз 1> then begin <команди 1> end; if <логічний вираз 2> then begin <команди 2> end;</pre>	<pre>if <логічний вираз>: <команди 1> if <логічний вираз 2>: <команди 2></pre>

3*. Поясніть, чим виконання команди в завданні 1 відрізняється від виконання послідовності команд у завданні 2.

4*. Наведіть приклади життєвих ситуацій, які можна описати алгоритмом з розгалуженням. Зобразіть ці приклади у вигляді блок-схем.




5*. Наведіть приклади правил з української мови, математики, інших предметів, що містять розгалуження. Зобразіть ці правила у вигляді блок-схем.

6*. Побудуйте математичну модель і створіть проєкт для обчислення значення виразу: $(a + b) - c : (a - b)$. Збережіть проєкт у папці з іменем **Завдання 6.8.6**, створеній у вашій папці. Запишіть у зошит набори вхідних даних, для яких ви тестували цей проєкт.



7*. Побудуйте математичну модель і створіть проєкт для обчислення значення виразу: $a + b - c : (a - 2b)$. Збережіть проєкт у папці з іменем **Завдання 6.8.7**, створеній у вашій папці. Запишіть у зошит набори вхідних даних, для яких ви тестували цей проєкт.

8*. Побудуйте математичну модель і створіть проєкт, який визначає менше з двох довільних дійсних чисел або виводить повідомлення, що числа рівні. Збережіть проєкт у папці з іменем **Завдання 6.8.8**, створеній у вашій папці. Запишіть у зошит набори вхідних даних, для яких ви тестували цей проєкт.

-  9°. Автоматизована система продажу авіаквитків запропонувала 5 варіантів перельоту з міста А до міста В. Побудуйте математичну модель і створіть проєкт, який визначає найдешевший із запропонованих перельотів. Збережіть проєкт у папці з іменем **Завдання 6.8.9**, створеній у вашій папці.
- 10°. Побудуйте математичну модель і створіть проєкт, який визначає найменше і найбільше із шести нерівних дійсних чисел. Збережіть проєкт у папці з іменем **Завдання 6.8.10**, створеній у вашій папці.
-  11°. Побудуйте математичну модель і створіть проєкт, який обчислює модуль довільного числа. Збережіть проєкт у папці з іменем **Завдання 6.8.11**, створеній у вашій папці. Запишіть у зошит вхідні дані, для яких ви тестували цей проєкт.
- 12°. Створіть проєкт, у якому кнопка буде переміщуватися на 30 пікселів праворуч за її вибору. Після того як кнопка зникне за правою межею вікна, вона повинна з'явитися біля його лівої межі. Збережіть проєкт у папці з іменем **Завдання 6.8.12**, створеній у вашій папці.
- 13°. Побудуйте математичну модель і створіть проєкт, у якому можна обчислити значення функції $y = \begin{cases} 2x - 4, & \text{якщо } x < 2; \\ x^2 + 5, & \text{якщо } 2 \leq x \leq 7; \\ 3 - 5x, & \text{якщо } x > 7. \end{cases}$ Збережіть проєкт у папці з іменем **Завдання 6.8.13**, створеній у вашій папці. Запишіть у зошит вхідні дані, для яких ви тестували цей проєкт.
-  14°. Побудуйте математичну модель і створіть проєкт, у якому можна обчислити значення функції $y = \begin{cases} 15 - x, & \text{якщо } -1 < x < 4; \\ 6 + 4x & \text{для всіх інших } x. \end{cases}$ Збережіть проєкт у папці з іменем **Завдання 6.8.14**, створеній у вашій папці. Запишіть у зошит вхідні дані, для яких ви тестували цей проєкт.
- 15°. Побудуйте математичну модель і створіть проєкт для визначення, чи можна з трьох відрізків із заданими довжинами утворити трикутник. Якщо так, то чи буде цей трикутник рівнобедреним, рівностороннім? Збережіть проєкт у папці з іменем **Завдання 6.8.15**, створеній у вашій папці. Запишіть у зошит вхідні дані, для яких ви тестували цей проєкт.
- 16°. Якщо споживач за місяць спожив не більше як 100 кВт · год електроенергії, то за кожну кВт · год потрібно заплатити 0,9 грн, а за кожну кВт · год спожитої електроенергії, що понад 100, потрібно заплатити 1,68 грн. Побудуйте математичну модель і створіть проєкт для визначення, яку суму грошей має заплатити споживач за спожиту електроенергію. Збережіть проєкт у папці з іменем **Завдання 6.8.16**, створеній у вашій папці.
- 17°. За 1 м³ спожитого газу в Україні потрібно заплатити 5,45 грн, а в Польщі – 1,61 злотих і ще й 9 злотих на місяць абонплати. Побудуйте математичну модель і створіть проєкт для порівняння оплати за один і той самий об'єм спожитого газу в Україні та в Польщі. Збережіть проєкт у папці з іменем **Завдання 6.8.17**, створеній у вашій папці.
- 18°. Банк пропонує вкладнику два види депозитів: 11 % прибутку за рік і 5 % прибутку за півроку. В обох депозитах прибуток за вказаний час додається до вкладу. Побудуйте математичну модель і створіть проєкт для порівняння прибутку за 1 рік за цими депозитами, якщо на обидва покласти одну й ту саму суму грошей. Збережіть проєкт у папці з іменем **Завдання 6.8.18**, створеній у вашій папці. Запишіть у зошит вхідні дані, для яких ви тестували цей проєкт.



6.9. Прапорці. Перемикачі



1. Який вигляд мають команди повного і неповного розгалуження в мовах програмування і як вони виконуються?
2. Для чого використовують написи і поля? Які їх властивості ви знаєте? Що визначає їх значення? Як їх можна змінити?
3. Для чого використовувалися в програмах, з якими ви працювали, прапорці та перемикачі? Наведіть приклади.

Прапорці та їх властивості

Розглянемо компонент **прапорець**. Прапорці використовують, щоб вибрати або один варіант з набору можливих, або кілька варіантів, або не вибрати жодного (мал. 6.66).

Якщо у вікні розміщено кілька прапорців, то серед них можна вибрати один або кілька, не вибрати жодного або вибрати всі. Якщо прапорець вибрано, то в нього всередині з'являється позначка .



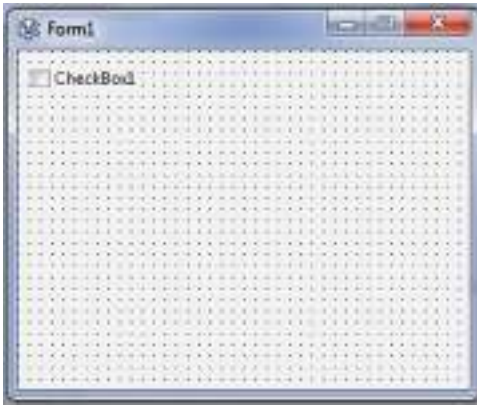
Мал. 6.66. Приклад використання прапорців



Для тих, хто вивчає Pascal

Для розміщення на формі прапорця потрібно двічі клацнути на кнопці **TCheckBox** (англ. *checkbox* – коробка для перевірки) на вкладці з компонентами **Standard**. Після цього прапорець з'явиться у верхній лівій частині форми

(мал. 6.67), а його ім'я буде додано до списку компонентів проекту в полі **Компоненти** вікна **Інспектор об'єктів**. За потреби прапорець можна перетягнути в будь-яке інше місце форми.



Мал. 6.67. Прапорець на формі

визначається значенням його властивості **Checked** (англ. *checked* – перевірено): вибраний – **true**, невибраний – **false**. За замовчуванням ця властивість має значення **false**.

Властивості прапорця **Caption, Top, Left, Name, Color, Enabled, Visible, Font, Width, Height** аналогічні до відповідних властивостей напису. Зокрема, значення властивостей **Width** і **Height** прапорця визначаються довжиною тексту праворуч від прапорця та розміром символів, якими виведено цей текст. Сам цей текст є значенням властивості **Caption** прапорця. Цей текст, як і для напису, можна вводити безпосередньо на вкладці **Властивості** в полі **Caption**, а можна вводити у вікні **Редактор рядків**, яке відкривається вибором кнопки  в правій частині поля **Caption**.

Вибрано прапорець чи не вибрано

Для тих, хто вивчає Python

Команда для створення прапорця в мові **Python** має вигляд:

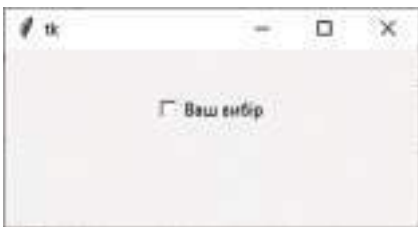
```
<ім'я прапорця> = Checkbutton(text = '<текст підпису>')
```

Після розміщення у вікні прапорця, створеного в такий спосіб, він матиме вказаний підпис. Наприклад, після виконання команд:

```
flag = Checkbutton(text = 'Ваш вибір')
```

```
flag.pack()
```

буде створено та розміщено у вікні прапорець з іменем *flag* і підписом *Ваш вибір* (мал. 6.68).



Мал. 6.68. Прапорець у вікні

Властивості прапорця **width, height, bg, fg, font** аналогічні до відповідних властивостей напису, поля та кнопки.

Щоб мати можливість дізнаватися під час виконання проекту стан прапорця (вибраний або невибраний), потрібно пов'язати з ним спеціальну змінну логічного типу. Для створення такої змінної використовують команду виду:

```
<ім'я змінної> = BooleanVar()
```

Пов'язують цю змінну з прапорцем під час його створення командою, що має вигляд:

```
<ім'я прапорця> = Checkbutton(text = '<текст підпису>', variable = <ім'я змінної>)
```

Наприклад, після виконання команд:

```
choice = BooleanVar()
```

```
flag = Checkbutton(text = 'Ваш вибір', variable = choice)
```

```
flag.pack()
```



буде створено та розміщено у вікні прапорець з іменем *flag* і підписом *Ваш вибір*, а змінна з іменем *choice* зберігатиме стан цього прапорця: **True** – якщо прапорець вибрано, **False** – якщо прапорець не вибрано.

Значення цієї змінної за замовчуванням – **False**, і після запуску проекту на виконання позначка прапорця буде відсутня. Якщо під час виконання проекту буде вибрано прапорець, то значення змінної стане **True**. За повторного вибору прапорця його стан зміниться на протилежний і також зміниться значення змінної.

Щоб одразу після запуску проекту прапорець був у стані *вибраний*, потрібно використати команду виду **<ім'я змінної>.set(True)** (англ. *set* – установити), наприклад **choice.set(True)**.

Щоб визначити значення цієї змінної під час виконання проекту, потрібно виконати команду виду **<ім'я змінної>.get()**, наприклад **choice.get()**. Значення цієї змінної можна використовувати в проектах у команді розгалуження.

Якщо в проекті використовується кілька прапорців, то для кожного з них повинна бути створена окрема змінна, наприклад:

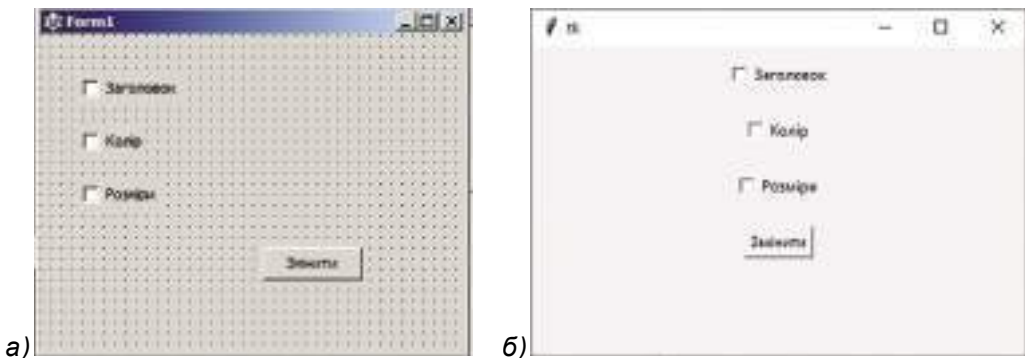
```
choice1 = BooleanVar()
flag1 = Checkbutton(text = 'Заголовок', variable = choice1)
choice2 = BooleanVar()
flag2 = Checkbutton(text = 'Колір', variable = choice2)
choice3 = BooleanVar()
flag3 = Checkbutton(text = 'Розміри', variable = choice3)
```

Використання прапорців

Розглянемо, як можна використати прапорці для реалізації розгалуження в проектах.



Нехай, наприклад, нам потрібно в ході виконання проекту змінювати значення або тільки однієї з властивостей вікна (або заголовок, або колір, або розміри), або кількох з них, або не змінювати значення жодної з них.

Для цього використаємо кнопку та три прапорці (мал. 6.69). Оскільки під час виконання проекту жодного прапорця може бути не вибрано, то доцільно для кожного з них залишити значення **false** властивості **Checked** (Python значення логічної змінної), яке встановлюється за замовчуванням.

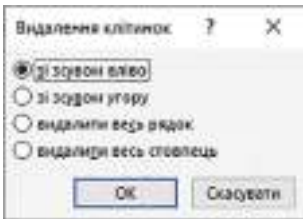


Мал. 6.69. Використання прапорців: а) у **Pascal**; б) у **Python**

Створимо процедуру для опрацювання події **Click** для кнопки та до її тексту помістимо такі три послідовні (не вкладені одна в одну) команди розгалуження:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre> if CheckBox1.Checked then Form1.Caption := 'Заголовок змі- нений'; if CheckBox2.Checked then Form1.Color := clRed; if CheckBox3.Checked then begin Form1.Width := 500; Form1.Height := 200; end; </pre>	<pre> if choice1.get(): root.title('Заголовок змінений') if choice2.get(): root['bg'] = 'red' if choice3.get(): root.geometry('500x200') </pre>


У кожній із цих команд незалежно одна від одної буде визначатися – вибрано відповідний прапорець чи ні. І якщо відповідний прапорець вибрано, то зміняться значення відповідних властивостей вікна. А якщо певний прапорець не вибрано, значення відповідних властивостей вікна не зміняться.




Мал. 6.70. Приклад використання перемикачів

Перемикачі та їх властивості

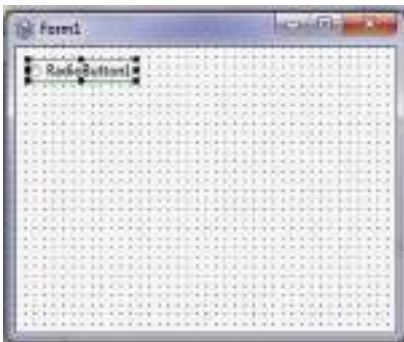
Розглянемо компонент **перемикач**. Перемикачі використовують для вибору тільки одного варіанта з кількох можливих (мал. 6.70).

Якщо у вікні розміщено кілька перемикачів, то серед них можна вибрати тільки один. Якщо перемикач вибрано, то в нього всередині з'являється позначка .

Для тих, хто вивчає Pascal

Для розміщення на формі перемикача потрібно двічі клацнути на кнопці **TRadioButton**  (англ. *radiobutton* – кругла кнопка) на вкладці з компонентами **Standard** (мал. 6.71).

Усі ті властивості, які ми розглядали для прапорця, має і перемикач, у тому числі й властивість **Checked**, значення якої можна використати в логічному виразі в команді розгалуження.




Мал. 6.71. Перемикач на формі



Мал. 6.72. Дві групи перемикачів на формі



На формі можна також розмістити кілька незалежних одна від одної груп перемикачів. Для цього потрібно двічі клацнути на кнопці **TRadioGroup**  (англ. *radiogroup* – кругла група) на вкладці з компонентами **Standard**. Тоді в кожній групі можна вибрати по одному перемикачу (мал. 6.72).



Для тих, хто вивчає Python

На відміну від прапорців, з однієї групи перемикачів можна вибрати тільки один. Для цього всі перемикачі з однієї групи повинні бути пов'язані з однією змінною цілого типу. Команда створення такої змінної має вигляд:

<ім'я змінної> = IntVar()

Команда створення кожного з перемикачів групи має такий вигляд:

<ім'я перемикача> = Radiobutton(text = '<тест підпису>', variable = <ім'я змінної>, value = <значення змінної>), де:

- властивість **text** визначає текст підпису перемикача;
- властивість **variable** визначає ім'я змінної, яка пов'язана з перемикачем. Повинна бути одна й та сама змінна для всіх перемикачів однієї групи;
- властивість **value** визначає значення, яке набуде змінна, якщо буде вибрано саме цей перемикач. Значення повинні бути різними для кожного з перемикачів однієї групи.

На малюнку 6.73 наведено фрагмент проєкту, у якому створено три перемикачі однієї групи *switch1*, *switch2* та *switch3*, для яких змінна *choice* набуває відповідно значень 1, 2 та 3.

```
choice = IntVar()
switch1 = Radiobutton(text = "Заголовок", variable = choice, value = 1)
switch1.pack(pady = 10)
switch2 = Radiobutton(text = "Колір", variable = choice, value = 2)
switch2.pack(pady = 10)
switch3 = Radiobutton(text = "Розмір", variable = choice, value = 3)
switch3.pack(pady = 10)
```

Мал. 6.73. Фрагмент проєкту зі створення трьох перемикачів однієї групи

Властивості перемикача **width**, **height**, **bg**, **fg**, **font** аналогічні до відповідних властивостей прапорця.

За замовчуванням змінна **choice** має значення 0. Оскільки це значення не відповідає жодному перемикачу, то після запуску проєкту на виконання жоден перемикач не буде вибрано. Щоб один з перемикачів одразу було вибрано, то потрібно встановити для змінної відповідне значення командою виду:

<ім'я змінної>.set(<значення змінної>)

Наприклад, після виконання команди **choice.set(1)** стане вибраним перший перемикач.

Щоб дізнатися в ході виконання проєкту, який з перемикачів вибрано, потрібно виконати команду виду **<ім'я змінної>.get()**, наприклад **choice.get()**.

Якщо потрібно розмістити у вікні кілька незалежних груп перемикачів, то кожную групу перемикачів потрібно пов'язати з окремою змінною.

Використання перемикачів

Розглянемо, як можна, використовуючи перемикачі, змінювати значення тільки однієї з властивостей вікна: або заголовок, або колір, або розміри.



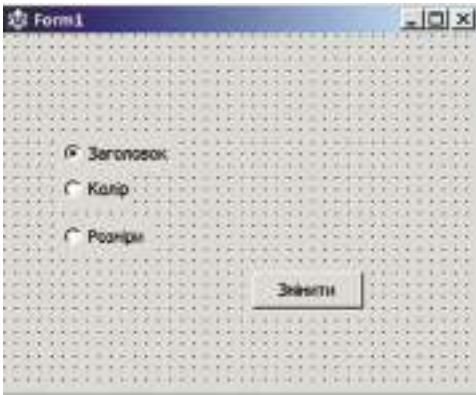
Для тих, хто вивчає Pascal

Використаємо в проєкті три перемикачі та кнопку (мал. 6.74).

Установимо значення властивостей перемикачів так, щоб перший з них було вибрано. Для цього для першого перемикача встановимо значення **true** властивості **Checked**, для всіх інших перемикачів залишимо значення цієї властивості **false**.

Створимо процедуру – обробник події **Click** для кнопки і до її тексту помістимо таку команду:

```
if RadioButton1.Checked
then Form1.Caption := 'Заголовок змінений'
else if RadioButton2.Checked
then Form1.Color := clRed
else begin
    Form1.Width := 600;
    Form1.Height := 300;
end;
```



Мал. 6.74. Використання перемикачів у Pascal

значення властивості **RadioButton2.Checked**. Оскільки буде вибрано другий перемикач, то це значення дорівнює **true**. Тому виконається команда змінення кольору вікна, а інші частини команди розгалуження виконуватися не будуть.

Якщо ж під час настання події **Click** для кнопки буде вибрано третій перемикач, то **RadioButton1.Checked = false**. Тому будуть виконуватися команди після першого ключового слова **Else**, тобто внутрішня команда розгалуження. Під час її виконання визначається значення властивості **RadioButton2.Checked**. Оскільки другий перемикач не вибрано, то його значення **false**. Тоді будуть виконуватися команди після другого ключового слова **Else**, тобто команди змінення розмірів вікна.

Якщо перед запуском цього проєкту на виконання в жодного перемикача не змінювати значення властивості **Checked**, установлені за замовчуванням, то під час запуску проєкту жоден з них не буде вибраним. Якщо і під час виконання проєкту не вибрати жоден з перемикачів, а вибрати кнопку, то зміняться розміри вікна (*подумайте самостійно, чому відбудуться саме такі змінення*). Щоб так не сталося, потрібно після другого ключового слова **Else** використати ще одну команду розгалуження:

Розпочинатися виконання цієї команди буде з визначення значення властивості **RadioButton1.Checked**.

Якщо під час настання події **Click** для кнопки буде вибрано перший перемикач, то **RadioButton1.Checked = true**. Тому виконається команда змінення заголовка вікна, а інші частини цієї команди розгалуження виконуватися не будуть.

Якщо під час настання події **Click** для кнопки буде вибрано другий перемикач, то **RadioButton1.Checked = false**. Тому будуть виконуватися команди після першого ключового слова **Else**. Цією командою є також команда розгалуження, що є частиною першої команди розгалуження. Під час її виконання визначається



```

if RadioButton1.Checked
then Form1.Caption := 'Заголовок змінений'
else if RadioButton2.Checked
then Form1.Color := clRed
else if RadioButton3.Checked
then begin
    Form1.Width := 600;
    Form1.Height := 300;
end;

```



Для тих, хто вивчає Python

Використаємо в проєкті три перемикачі та кнопку (мал. 6.75).

Створимо процедуру – обробник події **Click** для кнопки та до її тексту додамо таку команду:

```

if choice.get() == 1:
    root.title('Заголовок змінений')
elif choice.get() == 2:
    root['bg'] = 'red'
else:
    root.geometry('600x300')

```

Розпочинається виконання цієї команди буде з визначення значення змінної **choice**.

Якщо під час настання події **Click** для кнопки буде вибрано перший перемикач, то буде істинним значення логічного виразу **choice.get() == 1**. Тому виконається команда змінення заголовка вікна, а інші частини цієї команди розгалуження виконуватися не будуть.

Якщо під час настання події **Click** для кнопки буде вибрано другий перемикач, то значення логічного виразу **choice.get() == 1** буде **False**. Тому буде виконуватися команда перевірки умови після ключового слова **elif**. При її виконанні визначається значення змінної **choice**. Оскільки буде вибрано другий перемикач, то це значення дорівнює 2. Тому виконається команда змінення кольору вікна, а інші частини команди розгалуження виконуватися не будуть.

Якщо ж під час настання події **Click** для кнопки буде вибрано третій перемикач, то значення логічного виразу **choice.get() == 1** буде **False**. Тому буде виконуватися команди перевірки умови після ключового слова **elif**. Оскільки другий перемикач не вибрано, то значення логічного виразу **choice.get() == 2** буде **False**. Тоді будуть виконуватися команди після ключового слова **else**, тобто команди змінення розмірів вікна.

Якщо під час виконання проєкту не вибрати жодного з перемикачів, а вибрати кнопку, то зміняться розміри вікна (*подумайте самостійно, чому відбудуться саме такі змінення*). Щоб так не сталося, потрібно замість ключового слова **else** використати ще одну команду **elif**:

```

if choice.get() == 1:
    root.title('Заголовок змінений')
elif choice.get() == 2:
    root['bg'] = 'red'
elif choice.get() == 3:
    root.geometry('600x300')

```



Мал. 6.75. Використання перемикачів у Python



Прашуємо з комп'ютером

Виконайте тренувальні завдання за посиланням <https://cutt.ly/ShdrjKU> або QR-кодом.

Найважливіше в цьому пункті

Прапорці використовують, щоб вибрати або один, або кілька варіантів з набору можливих, або не вибрати жодного. З кількох прапорців може бути вибрано або один, або кілька, або всі, або жодний.

Тому прапорці використовують у тих випадках, коли під час виконання проекту потрібно вибрати або один з можливих варіантів, або кілька, або всі, або не вибрати жодного. Для опрацювання стану прапорців (вибраний або невібраний) у проект включають для кожного з них послідовно записані команди розгалуження, які незалежні одна від іншої.



Для тих, хто вивчає Pascal

Для розміщення на формі прапорця потрібно двічі клацнути на кнопці **TCheckBox** на вкладці **Standard** з компонентами.

Вибрано прапорець чи не вибрано визначається значенням його властивості **Checked** (англ. *checked* – перевірено): вибраний – **true**, невібраний – **false**. За замовчуванням ця властивість має значення **false**.



Для тих, хто вивчає Python

Команда для створення прапорця в мові **Python** має вигляд:

<ім'я прапорця> = Checkbutton(text = '<текст підпису>', variable = <ім'я змінної>), де змінна логічного типу використовується для визначення стану прапорця та створюється командою **<ім'я змінної> = BooleanVar()**.

Щоб під час запуску проекту прапорець відразу був у стані *вибраний*, потрібно використати команду виду **<ім'я змінної>.set(True)**, наприклад **choice.set(True)**. Щоб дізнатися значення цієї змінної під час виконання проекту, слід виконати команду виду **<ім'я змінної>.get()**, наприклад **choice.get()**. Значення цієї змінної можна використовувати в проектах у команді розгалуження як логічний вираз.

Перемикачі використовують для вибору тільки одного з кількох можливих варіантів. З кількох перемикачів однієї групи може бути вибрано тільки один.

Тому перемикачі використовують у тих випадках, коли під час виконання проекту потрібно вибрати тільки один з можливих варіантів. Для опрацювання стану прапорців (вибраний або невібраний) у проект включають для кожного з них команду розгалуження, кожна наступна з яких вкладена в попередню.



Для тих, хто вивчає Pascal

Для розміщення на формі перемикача потрібно двічі клацнути на кнопці **TRadioButton** на вкладці **Standard** з компонентами.

Вибрано перемикач чи не вибрано визначається значенням його властивості **Checked** аналогічно до прапорця.



На формі можна також розмістити кілька незалежних одна від одної груп перемикачів. Для цього потрібно двічі клацнути на кнопці **TRadioGroup** на вкладці з компонентами **Standard**. Тоді в кожній групі можна вибрати по одному перемикачу.



Для тих, хто вивчає Python

Команда для створення перемикача в мові **Python** має вигляд:

`<ім'я перемикача> = Radiobutton(text = '<тест підпису>', variable = <ім'я змінної>, value = <значення змінної>)`, де змінна використовується для визначення вибраного перемикача та створюється командою `<ім'я змінної> = IntVar()`.

Якщо потрібно, щоб один з перемикачів групи відразу був вибраним, то слід установити для змінної відповідне значення командою виду `<ім'я змінної>.set(<значення змінної>)`, наприклад `choice.set(1)`.

Щоб дізнатися під час виконання проекту, який з перемикачів однієї групи вибрано, потрібно виконати команду виду `<ім'я змінної>.get()`, наприклад `choice.get()`.

У вікні можна також розмістити кілька незалежних одна від одної груп перемикачів. Для цього потрібно кожен групу перемикачів пов'язати з окремою змінною. Тоді в кожній групі можна вибрати по одному перемикачу.





Дайте відповіді на запитання

- 1•. Для чого використовують прапорці, а для чого – перемикачі?
- 2◦. Які значення може мати властивість **Checked** прапорця та перемикача?
- 3•. Як визначити стан прапорця в проєкті, який розроблено мовою **Python**?
- 4•. Як потрібно записувати команди розгалуження, у яких перевіряється, вибрано прапорець чи ні?
- 5•. Як доцільно записувати команди розгалуження, у яких перевіряється, вибрано перемикач чи ні?
- 6•. Як визначити, який перемикач вибрано у проєкті, розробленому мовою **Python**?
- 7•. Для чого використовують групи перемикачів?





Виконайте завдання

- 1•. Намалюйте блок-схему та поясніть хід виконання послідовності команд:


 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>if CheckBox1.Checked then begin <команди 1> end; if CheckBox2.Checked then begin <команди 2> end;</pre>	<pre>if choice 1.get(): <команди 1> if choice 2.get(): <команди 2></pre> <p>де choice1 і choice2 – змінні логічного типу, які пов'язані з прапорцями</p>

2*. Намалюйте блок-схему та поясніть хід виконання команди:


 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre> if RadioButton1.Checked then begin <команди 1> end else if RadioButton2.Checked then begin <команди 2> end else begin <команди 3> end; </pre>	<pre> if choice.get() == 1: <команди 1> elif choice.get() == 2: <команди 2> else: <команди 3> </pre> <p>де choice – змінна цілого типу, яка пов'язана з групою перемикачів</p>

3*. Поясніть відмінності у виконанні команд розгалуження із завдання 1 та із завдання 2.


4*. Сформулюйте життєву задачу, у проєкті для розв'язування якої можна використати фрагмент із завдання 1.

 5*. Сформулюйте життєву задачу, у проєкті для розв'язування якої можна використати фрагмент із завдання 2.

6*. Побудуйте математичну модель і створіть проєкт, у якому можна обчислити суму, або різницю, або добуток, або частку від ділення двох довільних чисел залежно від вибраного перемикача. Збережіть проєкт у папці з іменем **Завдання 6.9.6**, створеній у вашій папці.

 7*. Побудуйте математичну модель і створіть проєкт, у якому можна обчислити суму, різницю, добуток, частку від ділення двох довільних дійсних чисел залежно від установлених позначок прапорців. Збережіть проєкт у папці з іменем **Завдання 6.9.7**, створеній у вашій папці.

8*. Робітник виготовляє деталі. Відомо кількість деталей, яку виготовив робітник за місяць, і вартість однієї деталі. Якщо робітник виконав план, то він одержує премію в розмірі 10 % від заробленої в результаті виготовлення деталей суми. Якщо робітник перевиконав план, то його премія становить 30 % від заробленої суми. Побудуйте математичну модель і створіть проєкт, який визначає суму грошей, яку заробить робітник за місяць. Збережіть проєкт у папці з іменем **Завдання 6.9.8**, створеній у вашій папці.

 9*. Робітниця протягом місяця виготовила x деталей, за кожену з яких вона отримує y грн. Якщо в робітниці є діти, вона отримує додаткову плату в розмірі 20 % від суми, заробленої за місяць. Якщо робітниця виготовить браковані деталі, то її зароблена сума за місяць буде зменшена на 15 %. Побудуйте математичну модель і створіть проєкт, який визначає суму грошей, яку заробить робітниця за місяць. Збережіть проєкт у папці з іменем **Завдання 6.9.9**, створеній у вашій папці.

10*. Створіть проєкт з використанням двох груп перемикачів і кнопки. Перемикачі першої групи визначають колір, у який зафарбовується вікно, а перемикачі другої групи визначають, на скільки будуть змінюватися розміри вікна. Збережіть проєкт у папці з іменем **Завдання 6.9.10**, створеній у вашій папці.



ПРАКТИЧНА РОБОТА № 11

«Проекти з розгалуженнями»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєвості та санітарно-гігієнічних норм.

1. Побудуйте математичну модель і створіть проєкт для обчислення значення виразу $x = (a + 3 * b) / (2 * a - b)$. Збережіть проєкт у папці з іменем **Практична 11_1**, створеній у вашій папці.
2. Побудуйте математичну модель і створіть проєкт для знаходження найменшого значення із чотирьох нерівних між собою чисел. Збережіть проєкт у папці з іменем **Практична 11_2**, створеній у вашій папці.
3. Вкладник хоче покласти в банк на депозит певну суму грошей на 2 роки. Банк пропонує три види депозитів: під 9 % річних, під 10 % річних і під 11 % річних. У перших двох депозитах прибуток додається до вкладу кожного року, у третьому – ні. Побудуйте математичну модель і створіть проєкт з використанням перемикачів для обчислення суми грошей, яку зможе отримати вкладник через 2 роки, якщо він вибере один із цих депозитів. Збережіть проєкт у папці з іменем **Практична 11_3**, створеній у вашій папці.

6.10. Цикл з лічильником



1. Які процеси називають циклічними? Наведіть приклади.
2. Що таке цикл в алгоритмі? Що таке тіло циклу?
3. Який вигляд має команда циклу з лічильником у мові програмування? Як вона виконується?

Команда циклу з лічильником

Ви вже знаєте, що в багатьох алгоритмах одна або кілька команд повинні повторюватися багато разів поспіль. Недоцільно включати в алгоритм ці команди стільки разів, скільки вони повинні повторюватися, тому що кількість разів повторення може бути дуже велика або навіть заздалегідь невідома.

У таких випадках потрібно в алгоритмах використовувати *цикли*. Нагадаємо, що **цикл в алгоритмі** – це фрагмент алгоритму, який може повторюватися поспіль більше ніж один раз.

У мовах програмування є кілька команд, які можуть реалізувати цикл. Одна з них – **команда циклу з лічильником**. Її доцільно використовувати в тих випадках, коли кількість повторень команд тіла циклу відома ще до початку виконання циклу.



Для тих, хто вивчає Pascal

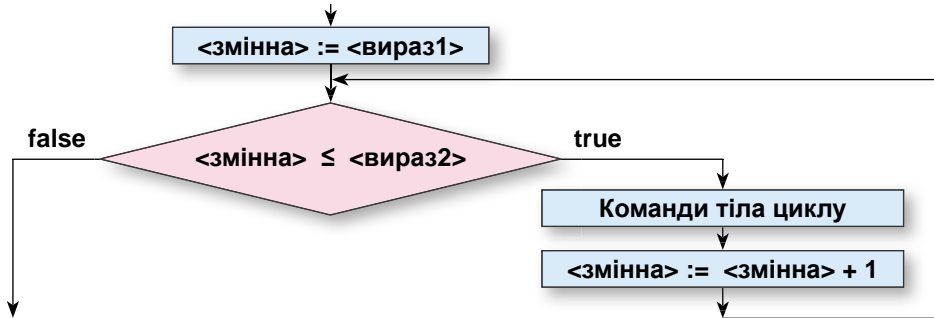
Загальний вигляд команди циклу з лічильником у Pascal такий:

```
for <ім'я змінної> := <вираз1> to <вираз2> do
begin
  <команди тіла циклу>
end;
```

Рядок **for <ім'я змінної> := <вираз1> to <вираз2> do** (англ. *for* – для, *to* – до, *do* – робити, виконувати) називається **рядком заголовка команди циклу з лічильником**. Змінна в рядку заголовка команди циклу з лічильником, що стоїть перед знаком присвоювання, називається **лічильником циклу**.

Лічильник циклу, вираз1 і вираз2 мають набувати тільки цілих значень. Якщо тіло циклу складається лише з однієї команди, операторні дужки **begin** і **end** можна не використовувати.

Блок-схему команди циклу з лічильником наведено на малюнку 6.76.



Мал. 6.76. Блок-схема команди циклу з лічильником

Виконується команда циклу з лічильником так:

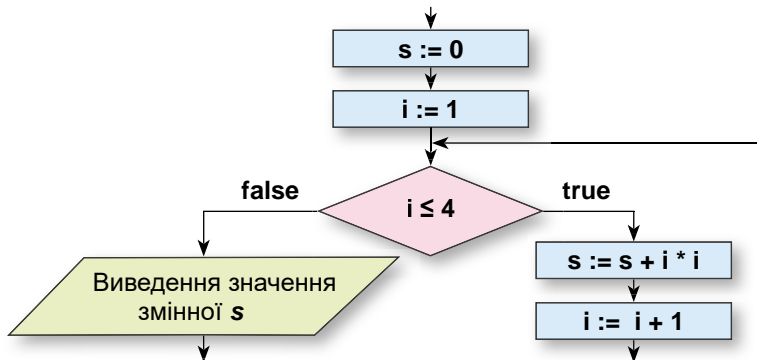
1. **Змінна**-лічильник циклу набуває значення **вираз1**.
2. Обчислюється значення логічного виразу **змінна ≤ вираз2**.
3. Якщо значення логічного виразу **true**, то виконуються команди тіла циклу і після цього виконується дія 4; якщо **false**, то виконання команди циклу закінчується і далі виконується команда, наступна за командою циклу.
4. Збільшується значення лічильника циклу на 1.
5. Повторюється виконання дії 2.

Для ілюстрації виконання команди циклу з лічильником розглянемо фрагмент проекту, у якому обчислюється сума $1^2 + 2^2 + 3^2 + 4^2$:

```

var s, i: integer;
begin
  s := 0;
  for i := 1 to 4 do
  begin
    s := s + i * i;
  end;
  Label1.Caption := IntToStr(s);
end;
  
```

Блок-схему цього фрагмента проекту наведено на малюнку 6.77.



Мал. 6.77. Блок-схема фрагмента проекту



Виконаємо цей фрагмент проекту.

Команда	Результат виконання
<code>s := 0</code>	<code>s = 0</code>
<code>i := 1</code>	<code>i = 1</code>
<code>i <= 4</code>	<code>(1 <= 4) = true</code>
<code>s := s + i * i</code>	<code>s = 0 + 1 * 1 = 1</code>
<code>i := i + 1</code>	<code>i = 2</code>
<code>i <= 4</code>	<code>(2 <= 4) = true</code>
<code>s := s + i * i</code>	<code>s = 1 + 2 * 2 = 5</code>
<code>i := i + 1</code>	<code>i = 3</code>
<code>i <= 4</code>	<code>(3 <= 4) = true</code>
<code>s := s + i * i</code>	<code>s = 5 + 3 * 3 = 14</code>
<code>i := i + 1</code>	<code>i = 4</code>
<code>i <= 4</code>	<code>(4 <= 4) = true</code>
<code>s := s + i * i</code>	<code>s = 14 + 4 * 4 = 30</code>
<code>i := i + 1</code>	<code>i = 5</code>
<code>i <= 4</code>	<code>(5 <= 4) = false</code>
<code>Label1.Caption := IntToStr(s)</code>	<code>Label1.Caption = 30</code>

Звертаємо вашу увагу, що після закінчення виконання команди циклу з лічильником значення лічильника циклу (змінної `i`) на 1 більше, ніж значення **вираз2**, і це значення, за потреби, можна використовувати в наступних командах.



Для тих, хто хоче знати більше

Існує різновид команди циклу з лічильником, у якому лічильник не збільшується на 1, а зменшується на 1:

```
for <ім'я змінної> := <вираз1> downto <вираз2> do
begin
  <команди тіла циклу>
end;
```

(англ. *downto* – униз до).

Ця команда циклу продовжує виконуватися, поки значення лічильника циклу більше або дорівнює (не менше) значення **вираз2**.



Для тих, хто вивчає Python

Загальний вигляд команди циклу з лічильником у мові **Python** такий:

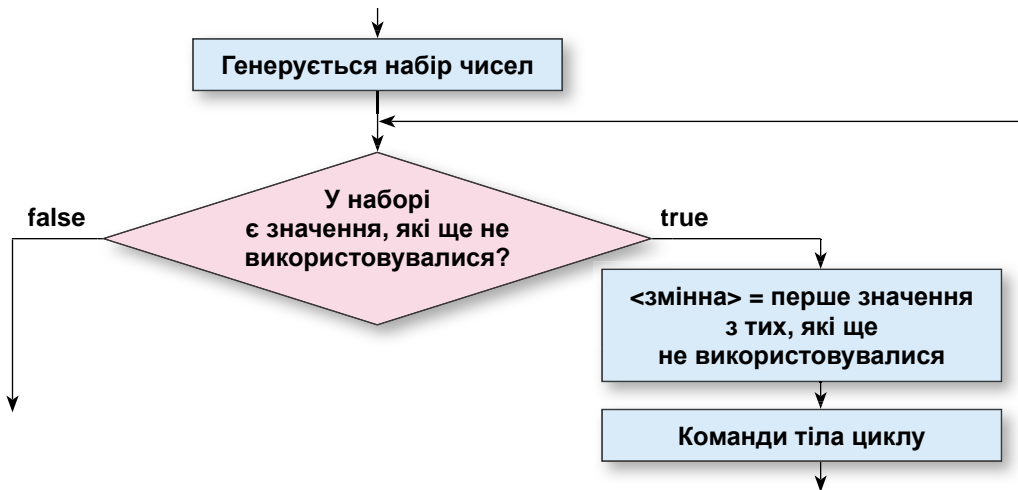
```
for <ім'я змінної> in range(<кількість повторень>):
  <команди тіла циклу>
```

Рядок `for <ім'я змінної> in range(<кількість повторень>)` (англ. *for* – для, *in range* – в діапазоні) називається **рядком заголовка команди циклу з лічильником**. Змінна в рядку заголовка команди циклу з лічильником називається **лічильником циклу**.

Функція `range(<кількість повторень>)` утворює (генерує) набір послідовних цілих чисел, перше з яких 0, а останнє – на 1 менше від заданої кількості повто-

рень (усього чисел стільки, скільки задано кількість повторень). Наприклад, функція `range(3)` генерує набір чисел 0, 1, 2, усього 3 числа. Змінна-лічильник циклу послідовно набуває значень із цього набору чисел.

Блок-схему команди циклу з лічильником у **Python** наведено на малюнку 6.78.



Мал. 6.78. Блок-схема команди циклу з лічильником

Виконується команда циклу з лічильником так:

1. Функція `range(<кількість повторень>)` генерує набір чисел від 0 до (**кількість повторень – 1**).
2. Виконується перевірка, чи є в наборі числа, які ще не використовувалися і могли б стати новим значенням **змінної**-лічильника циклу.
3. Якщо результат перевірки **True**, то **змінна**-лічильник циклу набуває значення першого з набору чисел, які ще не використовувалися, і після цього виконується дія 4; якщо **False**, то виконання команди циклу закінчується і далі виконується команда, наступна за командою циклу.
4. Виконуються команди тіла циклу.
5. Повторюється виконання дії 2.

У записі команди циклу з лічильником усі команди тіла циклу повинні бути записані з однаковим відступом від першої літери слова **for**.

Для ілюстрації виконання команди циклу з лічильником розглянемо проект, у якому обчислюється сума $1^2 + 2^2 + 3^2 + 4^2$.

s = 0

for i in range(4):

s = s + (i + 1) * (i + 1)

print('Сума дорівнює', s)

Виконаємо цей проект.

Команда	Результат виконання
<code>s = 0</code>	<code>s = 0</code>
<code>range(4)</code>	<code>0, 1, 2, 3</code>
У наборі ще є невикористані числа?	<code>True</code>
Змінна <code>i</code> набуває першого значення з набору невикористаних чисел	<code>i = 0</code>



Продовження таблиці

Команда	Результат виконання
$s = s + (i + 1) * (i + 1)$	$s = 0 + (0 + 1) * (0 + 1) = 1$
У наборі ще є невикористані числа?	True
Змінна i набуває першого значення з набору невикористаних чисел	$i = 1$
$s = s + (i + 1) * (i + 1)$	$s = 1 + (1 + 1) * (1 + 1) = 5$
У наборі ще є невикористані числа?	True
Змінна i набуває першого значення з набору невикористаних чисел	$i = 2$
$s = s + (i + 1) * (i + 1)$	$s = 5 + (2 + 1) * (2 + 1) = 14$
У наборі ще є невикористані числа?	True
Змінна i набуває першого значення з набору невикористаних чисел	$i = 3$
$s = s + (i + 1) * (i + 1)$	$s = 14 + (3 + 1) * (3 + 1) = 30$
У наборі ще є невикористані числа?	False
<code>print('Сума дорівнює', s)</code>	Сума дорівнює 30



Для тих, хто хоче знати більше

Функція `range(<початок>, <кінець>, <крок>)` у загальному випадку має три аргументи цілого типу:

- **початок** – перше ціле число, яке потрапить до згенерованого набору;
- **кінець** – ціле число, яке не увійде до згенерованого набору і яке більше за всі числа з набору, якщо **крок** > 0, і менше від всіх чисел з набору, якщо **крок** < 0.
- **крок** – різниця між кожними двома сусідніми цілими числами в згенерованому наборі чисел.

Наприклад, функція `range(3, 9, 2)` згенерує такий набір чисел: 3, 3 + 2 = 5, 5 + 2 = 7. Наступне число 7 + 2 = 9 не увійде в набір чисел, оскільки воно не менше, ніж другий аргумент функції.

Тестування проєкту із циклом з лічильником

Під час тестування проєкту, що містить цикл з лічильником, доцільно підбирати такі набори вхідних даних, щоб тіло циклу виконувалося різну кількість разів: 1 раз; 2 рази; 3 рази; 5 разів, іншу кількість разів.



Працюємо з комп'ютером



Виконайте тренувальні завдання за посиланням <https://cutt.ly/lhdkPJ> або QR-кодом.



Найважливіше в цьому пункті

Цикл в алгоритмі – це фрагмент алгоритму, який може повторюватися поспіль більше ніж один раз. Цикл з лічильником використовується і в тих випадках, коли перед його виконанням відомо, скільки разів повинні виконатися команди тіла циклу.



Для тих, хто вивчає Pascal

Загальний вигляд команди циклу з лічильником у **Pascal** такий:

```
for <ім'я змінної> := <вираз1> to <вираз2> do
begin
  <команди тіла циклу>
end;
```

Рядок **for <ім'я змінної> := <вираз1> to <вираз2> do** називається **рядком заголовка команди циклу з лічильником**. Змінна в рядку заголовка команди циклу з лічильником, що стоїть перед знаком присвоювання, називається **лічильником циклу**. Лічильник циклу, **вираз1** і **вираз2** мають набувати тільки цілих значень.

Виконується команда циклу з лічильником так:

1. **Змінна**-лічильник циклу набуває значення **вираз1**.
2. Обчислюється значення логічного виразу **змінна ≤ вираз2**.
3. Якщо значення логічного виразу **true**, то виконуються команди тіла циклу й після цього виконується дія 4; якщо **false**, то виконання команди циклу закінчується і далі виконується команда, наступна за командою циклу.
4. Збільшується значення лічильника циклу на 1.
5. Повторюється виконання дії 2.



Для тих, хто вивчає Python

Загальний вигляд команди циклу з лічильником у мові **Python** такий:

```
for <ім'я змінної> in range(<кількість повторень>):
  <команди тіла циклу>
```

Рядок **for <ім'я змінної> in range(<кількість повторень>)** називається **рядком заголовка команди циклу з лічильником**. Змінна в рядку заголовка команди циклу з лічильником називається **лічильником циклу**.

Функція **range(<кількість повторень>)** утворює (генерує) набір послідовних цілих чисел, перше з яких 0, а останнє – на 1 менше від заданої кількості повторень. **Змінна**-лічильник циклу послідовно набуває значень із цього набору чисел.

Виконується команда циклу з лічильником так:

1. Функція **range(<кількість повторень>)** генерує набір чисел від 0 до (**кількість повторень – 1**).
2. Виконується перевірка, чи є в наборі числа, які ще не використовувалися і могли б стати новим значенням **змінної**-лічильника циклу.
3. Якщо результат перевірки **True**, то **змінна**-лічильник циклу набуває значення першого з набору чисел, які ще не використовувалися, і після цього виконується дія 4; якщо **False**, то виконання команди циклу закінчується і далі виконується команда, наступна за командою циклу.
4. Виконуються команди тіла циклу.
5. Повторюється виконання дії 2.

У записі команди циклу з лічильником усі команди тіла циклу повинні бути записані з однаковим відступом від першої літери в слові **for**.



Дайте відповіді на запитання

- 1°. Що таке цикл в алгоритмі?
- 2°. Який вигляд має команда циклу з лічильником у мові програмування?





- 3°. Як виконується команда циклу з лічильником у мові програмування?
- 4°. Який вигляд має блок-схема циклу з лічильником? Поясніть виконання такого фрагмента алгоритму.
- 5°. Чи можуть команди тіла циклу з лічильником виконуватися тільки один раз? Поясніть свою відповідь. Наведіть приклад.
- 6°. Чи можуть команди тіла циклу з лічильником не виконуватися жодного разу? Поясніть свою відповідь. Наведіть приклад.
- 7°. Чи може виконання циклу з лічильником ніколи не закінчитися? Поясніть свою відповідь. Наведіть приклад.
- 8°. На яких тестових наборах вхідних даних доцільно тестувати проект із циклами?
- 9°. У чому полягають відмінності у виконанні основних алгоритмічних структур: слідування, розгалуження, цикл?





Виконайте завдання

- 1°. Складіть блок-схему фрагмента алгоритму та виконайте його.

	 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
а)	<pre>n := 1; for i := 1 to 4 do n := n * i;</pre>	<pre>n = 1 for i in range(4): n = n * (i + 1)</pre>
б)	<pre>p := 0; a := 2; for i := 3 to 6 do begin a := 3 * a + 1; p := p + a; end;</pre>	<pre>p = 0 a = 2 for i in range(3, 7, 1): a = 3 * a + 1 p = p + a</pre>



- 2°. Складіть блок-схему фрагмента алгоритму і виконайте його.

	 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
а)	<pre>n := 0; for i := 1 to 4 do n := n + i * i * i;</pre>	<pre>n = 1 for i in range(4): n = n + (i + 1)**3 Операція ** означає піднесення до степеня, у даному випадку значення i + 1 буде піднесено до степеня 3</pre>
б)	<pre>p := 0; a := 4; for i := 2 to 5 do begin a := 2 * a - 1; p := p + a; end;</pre>	<pre>p = 0 a = 4 for i in range(2, 6, 1): a = 2 * a - 1 p = p + a</pre>

- 3°. Побудуйте математичну модель і складіть блок-схему алгоритму знаходження суми п'яти чисел, перше з яких дорівнює 7, а кожне із чотирьох наступних на 3 більше за попереднє. Виконайте складений алгоритм.



4°. Побудуйте математичну модель і складіть блок-схему алгоритму знаходження суми шести чисел, перше з яких дорівнює 100, а кожне з п'яти дорівнює попередньому, поділеному на 5. Виконайте складений алгоритм.

5°. Побудуйте математичну модель і складіть проєкт для обчислення суми n чисел, перше з яких дорівнює додатному числу x , а кожне наступне у 2 рази більше за попереднє. Запишіть у зошит набори вхідних даних, для яких ви тестували проєкт. Збережіть проєкт у папці **Завдання 6.10.5**, створеній у вашій папці.



6°. Побудуйте математичну модель і складіть проєкт для обчислення суми n чисел, перше з яких дорівнює x , а кожне наступне дорівнює квадрату попереднього. Запишіть у зошит набори вхідних даних, для яких ви тестували проєкт. Збережіть проєкт у папці **Завдання 6.10.6**, створеній у вашій папці.

7°. Людина поклала в банк певну суму грошей. Щорічно банк додає до вкладу 10 % від суми, що зберігається на рахунку після попереднього року. Побудуйте математичну модель і складіть проєкт для визначення суми, яка буде на рахунку через n років. Запишіть у зошит набори вхідних даних, для яких ви тестували проєкт. Збережіть проєкт у папці **Завдання 6.10.7**, створеній у вашій папці.



8°. Людина поклала в банк певну суму грошей. Щорічно банк додає до вкладу заздалегідь визначений відсоток від суми, що зберігається на рахунку на кінець поточного року. Побудуйте математичну модель і складіть проєкт для визначення, на скільки зросте сума на рахунку через за n років. Запишіть у зошит набори вхідних даних, для яких ви тестували проєкт. Збережіть проєкт у папці **Завдання 6.10.8**, створеній у вашій папці.

9*. Спортсмен-паралімпієць, готуючись до участі в марафоні, у перший день тренувань подолав дистанцію в s км. Кожного наступного дня його денна дистанція збільшувалася на p % від дистанції попереднього дня. Побудуйте математичну модель і створіть проєкт для визначення:

- дистанції, яку спортсмен-паралімпієць здолає на n -ний день тренувань;
- сумарної дистанції за n днів тренувань;
- на скільки більшою буде дистанція у n -ний день тренувань за дистанцію першого дня.

Збережіть проєкт у папці **Завдання 6.10.9**, створеній у вашій папці.

6.11. Цикл з передумовою



- Який вигляд має і як виконується команда циклу з лічильником?
- Який вигляд має і як виконується блок-схема циклу з передумовою?
- У яких випадках в алгоритмах використовується цикл з лічильником, а в яких – цикл з передумовою?

Цикл з передумовою

Ви вже знаєте, якщо перед виконанням циклу невідомо, скільки разів мають виконатися команди тіла циклу, то використовувати команду циклу з лічильником не можна.

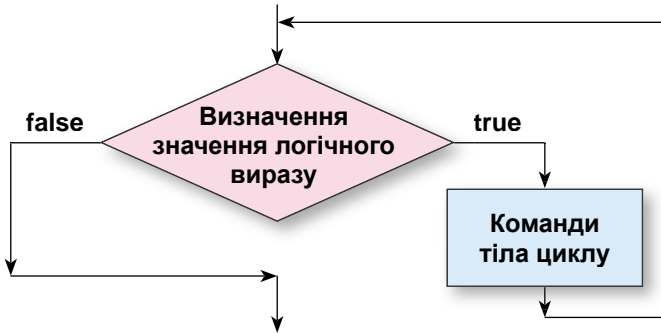
У таких випадках використовують інші команди циклу, зокрема **команду циклу з передумовою**.

На початку виконання команди циклу з передумовою визначається значення логічного виразу. Якщо воно **true**, то виконуються команди тіла циклу і після цього



знову визначається значення логічного виразу. І якщо воно знову дорівнює **true**, команди тіла циклу виконуються знову. І так буде відбуватися доти, доки значення логічного виразу не стане **false**, і тоді виконання команди циклу з передумовою закінчиться і буде виконуватися наступна команда.



Блок-схему циклу з передумовою наведено на малюнку 6.79.



Мал. 6.79. Блок-схема циклу з передумовою

Команда циклу з передумовою в мовах програмування

Команда циклу з передумовою в мовах програмування має такий вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>while <логічний вираз> do begin <команди тіла циклу> end;</pre>	<pre>while<логічний вираз>: <команди тіла циклу></pre>

(англ. *while* – поки).

Виконання команди циклу з передумовою відбувається так: обчислюється значення логічного виразу; якщо це значення дорівнює **true**, то виконуються **команди тіла циклу**, після чого знову обчислюється значення логічного виразу, і якщо це значення знову дорівнює **true**, то знову виконуються **команди тіла циклу**, після чого знову обчислюється значення логічного виразу; якщо значення логічного виразу дорівнює **false**, то **команди тіла циклу** не виконуються, а виконується команда, наступна за циклом.

Звертаємо вашу увагу:

- якщо в тілі циклу лише одна команда, то операторні дужки **begin** і **end** можна не використовувати;
- серед команд тіла циклу можуть бути й лінійні фрагменти, і розгалуження, й інші цикли.

Розглянемо задачу, алгоритм розв'язування якої містить цикл з передумовою.

Задача 1. Скільки потрібно взяти доданків, перший з яких дорівнює заданому дійсному числу, а кожний наступний – на 3 більший за попередній, щоб їх сума перевищила 100?

Побудуємо математичну модель для цієї задачі:

- **вхідні дані:** перший доданок (**a**), число, на яке наступний доданок більше за попереднє, (**3**), число, яке має перевищити сума чисел, (**100**);

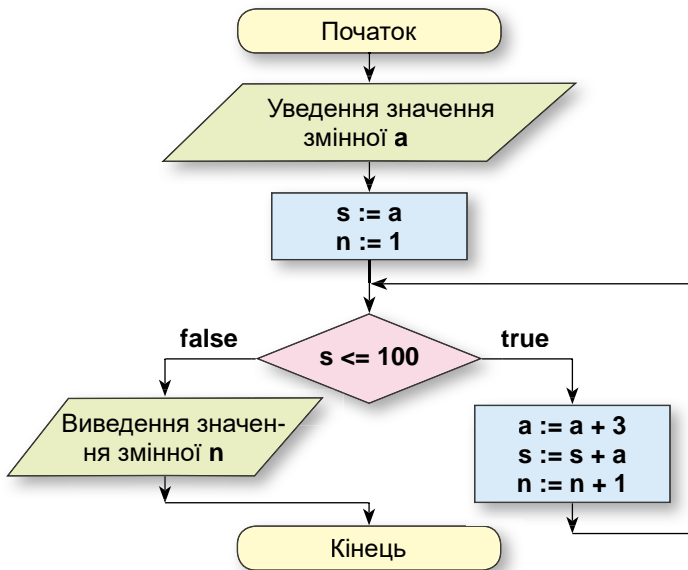
- **кінцеві результати:** кількість доданків (n);
- **формули:** $a = a + 3$, $s = s + a$, $s \leq 100$.

З трьох вхідних даних одне потрібно вводити (a), а два інші – конкретні числа. У задачі має бути один кінцевий результат, який потрібно виводити.

Перед початком циклу присвоїмо змінній s , яку використаємо для збереження суми, значення першого доданка. А змінній n , яку використаємо для збереження кількості взятих доданків, присвоїмо значення 1, бо в сумі вже враховано один (перший) доданок.



До суми потрібно додавати наступні доданки, доки ця сума буде менша або дорівнюватиме 100. Тому логічний вираз у заголовку циклу матиме вигляд $s \leq 100$. У тілі циклу шукатимемо наступний доданок ($a := a + 3$), додаватимемо його до поточного значення суми ($s := s + a$) та збільшуватимемо кількість доданих доданків на 1 ($n := n + 1$).

Блок-схему алгоритму розв'язування цієї задачі наведено на малюнку 6.80.



Мал. 6.80. Блок-схема алгоритму розв'язування задачі 1

Відповідний фрагмент проекту має такий вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre> var a, s: real; n: integer; begin a := StrToFloat(Edit1.Text); s := a; n := 1; while s <= 100 do begin a := a + 3; s := s + a; n := n + 1; end; Label1.Caption := IntToStr(n); end; </pre>	<pre> a = int(input('Уведіть перший доданок: ')) s = a n = 1 while s <= 100: a += 3 s = s + a n += 1 print(n) </pre>



Тестувати проєкт, що містить цикл з передумовою, доцільно для таких наборів даних, щоб тіло циклу не виконувалося жодного разу; виконувалося 1 раз; 2 рази; 3 рази; 5 разів і за потреби при інших наборах даних.

Так, наприклад, для тестування фрагмента проєкту

<pre>a := StrToFloat(Edit1.Text); s := 0; n := 0; while s < a do begin s := s + 5; n := n + 1; end;</pre>	<pre>a = float(input('Уведіть число a: ')) s = 0 n = 0 while s < a: s += 5 n += 1</pre>
--	--

можна взяти такі значення **a**:

- **a = -2** – тіло циклу не виконуватиметься жодного разу;
- **a = 3** – тіло циклу буде виконано 1 раз;
- **a = 7** – тіло циклу буде виконано 2 рази і т. д.



 **Працюємо з комп'ютером**




Виконайте тренувальні завдання за посиланням <https://cutt.ly/6hdrID2> або QR-кодом.

 **Найважливіше в цьому пункті**

Блок-схему циклу з передумовою наведено на малюнку 6.79.
Команда циклу з передумовою в мовах програмування має такий вигляд:

 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
<pre>while <логічний вираз> do begin <команди тіла циклу> end;</pre>	<pre>while <логічний вираз>: <команди тіла циклу></pre>

Виконання команди циклу з передумовою відбувається так: обчислюється значення логічного виразу; якщо це значення дорівнює **true**, то виконуються **команди тіла циклу**, після чого знову обчислюється значення логічного виразу; якщо значення логічного виразу дорівнює **false**, то **команди тіла циклу** не виконуються, а виконується команда, наступна за циклом.

 **Дайте відповіді на запитання**

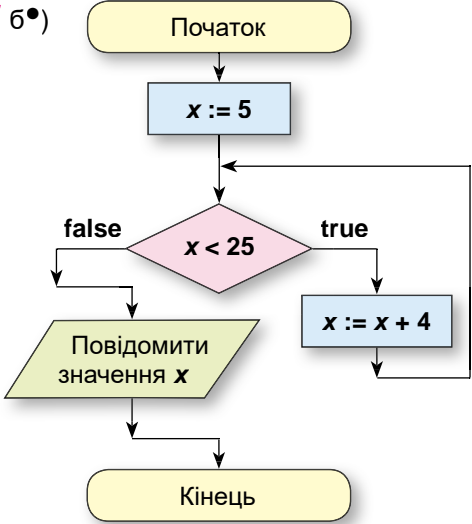
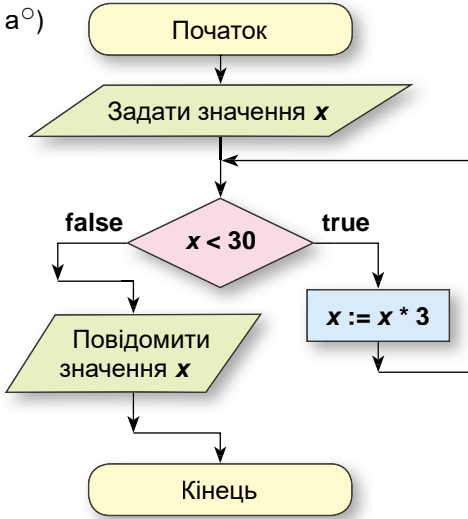
- 1°. Який вигляд має блок-схема циклу з передумовою? Поясніть виконання цього циклу.
- 2°. Як виконується цикл з передумовою в алгоритмі?
- 3°. Який вигляд має команда циклу з передумовою в мові програмування?
- 4°. Як виконується команда циклу з передумовою?
- 5°. Чи можуть команди тіла циклу з передумовою не виконуватися жодного разу? Поясніть свою відповідь. Наведіть приклади.
- 6°. Чи можуть команди тіла циклу з передумовою виконуватися тільки один раз? Поясніть свою відповідь. Наведіть приклади.

- 7*. Чи може виконання циклу з передумовою ніколи не закінчитися? Поясніть свою відповідь. Наведіть приклади.
- 8*. У яких випадках в алгоритмі потрібно використовувати цикл з лічильником, а коли – цикл з передумовою?
- 9*. Чим відрізняються між собою цикл з лічильником і цикл з передумовою?
- 10*. У чому полягають відмінності у виконанні основних алгоритмічних структур: слідування, розгалуження, цикл з лічильником і цикл з передумовою?



Виконайте завдання

1. Виконайте алгоритм:





2*. Виконайте фрагмент програми та з'ясуйте, якими будуть значення змінних після його завершення:

	Для тих, хто вивчає Pascal	Для тих, хто вивчає Python
а)	<pre>s := 100; while s > 30 do s := s - 20;</pre>	<pre>s = 100 while s > 30: s = s - 20</pre>
б)	<pre>k := 1; a := 12; while a < 100 do begin a := 2 * a - 4; k := k + 1; end;</pre>	<pre>k = 1 a = 12 while a < 100: a = 2 * a - 4 k += 1</pre>
в)	<pre>s := 0; a := 1; k := 1; while a < 50 do begin s := s + a; k := k + 1; a := k * k; end;</pre>	<pre>s = 0 a = 1 k = 1 while a < 50: s = s + a k += 1 a = k * k</pre>



3*. Виконайте фрагмент програми та з'ясуйте, якими будуть значення змінних після його завершення:

	 Для тих, хто вивчає Pascal	 Для тих, хто вивчає Python
а)	<pre>k := 1; a := 100; while a > 10 do begin a := a / 2; k := k + 1; end;</pre>	<pre>k = 1 a = 100 while a > 10: a = a / 2 k += 1</pre>
б)	<pre>s := 0; a := 5; while a <= 100 do begin s := s + a; a := a * 2; end;</pre>	<pre>s = 0 a = 5 while a <= 100: s = s + a a = a * 2</pre>



4*. Складіть текстові задачі за алгоритмами їх розв'язування із завдання 1.



5*. Складіть текстові задачі за алгоритмами їх розв'язування із завдання 2.

6*. Побудуйте математичну модель і складіть проєкт для визначення, за скільки годин робітник виконає план в x деталей, якщо за першу годину роботи він виготовив y деталей, а за кожну наступну годину виготовляє на 1 деталь більше, ніж за попередню. Збережіть проєкт у папці з іменем **Завдання 6.11.6**, створеній у вашій папці.



7*. Побудуйте математичну модель і складіть проєкт для визначення, за скільки годин турист подолає відстань 20 км, якщо за першу годину він пройшов 6 км, а за кожну наступну годину проходить на 1 км менше, ніж за попередню. Збережіть проєкт у папці з іменем **Завдання 6.11.7**, створеній у вашій папці.

8*. Створіть проєкт для обчислення середнього арифметичного додатних чисел, перше з яких дорівнює 100, а кожне наступне менше від попереднього на задане додатне число. Збережіть проєкт у папці з іменем **Завдання 6.11.8**, створеній у вашій папці.

9*. Побудуйте математичну модель і створіть проєкт для визначення, на скільки років вкладниця має покласти x грн до банку, щоб отримати від банку більше за a грн прибутку, якщо банк щорічно збільшує вклад на p %. Збережіть проєкт у папці з іменем **Завдання 6.11.9**, створеній у вашій папці.

10*. Створіть проєкт для знаходження найбільшого спільного дільника двох натуральних чисел за алгоритмом Евкліда: поки числа різні, більше з них замінювати різницею більшого та меншого; коли числа стануть рівними, то кожне з них дорівнюватиме їх НСД. Збережіть проєкт у папці з іменем **Завдання 6.11.10**, створеній у вашій папці.

11*. Спортсменка, готуючись до участі в марафоні, у перший день тренувань пододала дистанцію в s км. Кожного наступного дня її денна дистанція збільшувалася на p % від дистанції попереднього дня. Побудуйте математичну модель і створіть проєкт для визначення:

а) у який день тренувань спортсменка пододала дистанцію в s_1 км;

б) у який день тренувань дистанція збільшиться на s_2 км порівняно з попереднім днем.

Збережіть проєкт у папці з іменем **Завдання 6.11.11**, створеній у вашій папці.



12*. Об'єднайтеся в групу з 2–3 осіб для розробки спільного дослідницького проєкту. Ознайомтеся з поняттям систем числення (наприклад, на сайті <https://nrs.rozh2sch.org.ua/>), відшукайте відомості про їх значення в комп'ютерній галузі. Розробіть проєкт мовою програмування для переведення чисел між системами числення з різними основами. Презентуйте свій проєкт у класі.

ПРАКТИЧНА РОБОТА № 12

«Проєкти із циклами»

Увага! Під час роботи з комп'ютером дотримуйтеся вимог безпеки життєдіяльності та санітарно-гігієнічних норм.

1. Побудуйте математичну модель і створіть проєкт для розв'язування задачі:
Перед початком повені рівень води в річці становив N см. Під час повені кожну годину рівень води зростає на P см. Яким буде рівень води через N год після початку повені? Через скільки годин після початку повені рівень води буде не менший від K см?
2. Використайте в проєкті поля для введення вхідних даних, поле для виведення відповіді на перше запитання задачі й дві кнопки. Біля кожного поля розмістіть написи для пояснювальних коментарів.
3. Складіть обробник події **Click** для першої кнопки, виконання якого приведе до виведення в поле відповіді на перше запитання задачі.
4. Складіть обробник події **Click** для другої кнопки, виконання якого приведе до знаходження відповіді на друге запитання задачі й виведення його у вікно повідомлень.
5. Збережіть проєкт у папці з іменем **Практична 12**, створеній у вашій папці.



СЛОВНИК

А

Алфавіт мови програмування – множина символів, з яких можна утворювати слова і речення цієї мови, с. 149.

Б

Багаторівневий список – список, у якому окремі елементи підпорядковані іншим, с. 70.

Байт – це послідовність з восьми бітів, с. 16.

Біт – це цифра 0 або 1 у двійковому коді повідомлення, с. 15.

В

Висловлювання – речення, яке містить твердження про певний об'єкт або про зв'язки між об'єктами і про яке можна однозначно сказати, **істинне** воно чи **хибне**, с. 208.

Відеоадаптер – пристрій для опрацювання графічних даних, що виводяться на екран монітора, с. 37.

Відеоредактори – програми для редагування відеоданих, с. 128.

Відеостудії – програми, що використовуються в професійних цілях для редагування відеоданих, с. 128.

Внутрішнє гіперпосилання – гіперпосилання для переходу до об'єктів у межах одного документа, с. 88.

Внутрішня пам'ять – складова пам'яті комп'ютера, що призначена для забезпечення роботи процесора. Складається внутрішня пам'ять з оперативної, постійної і кеш-пам'яті, с. 23.

Г

Грабер – див. *рекордер*, с. 122.

Графічний планшет – пристрій введення графічних даних. Використовується властивість робочої поверхні планшета реагувати на дотик стилуса. Користувач створює малюнок, дотикаючись стилусом до поверхні графічного планшета, с. 29.

Д

Двійкове кодування – це кодування повідомлень з використанням сигналів лише двох видів, с. 15.

Двійковий код повідомлення – це повідомлення, отримане в результаті двійкового кодування початкового повідомлення, с. 15.

Декодування повідомлення – це процес отримання початкового повідомлення із закодованого, с. 7.

Державні стандарти оформлення документів – єдині вимоги та правила оформлення офіційних документів різних типів, їх структури та змісту, які встановлюються нормативними актами, с. 80.

Дизайн вебсторінки – особливості розміщення контенту та елементів навігації, їх форматування, оформлення сторінки, с. 105.

Диз'юнкція двох логічних виразів x і y – логічна операція, результатом якої є **true**, якщо значення хоча б одного з логічних виразів x або y є **true**, і є **false**, якщо значення кожного з логічних виразів x і y є **false**, с. 211.

Довжина двійкового коду повідомлення (обсяг даних) – це кількість бітів або байтів у двійковому коді цього повідомлення, с. 16.

Е

Експрес-стилі – вбудовані стилі, у яких значення властивостей об'єктів підібрані гармонійно, з урахуванням основних вимог до дизайну, с. 78.

Елементи навігації – засоби для переходу з вебсторінки до інших сторінок або до інших об'єктів на тій самій сторінці, с. 105.

Ергономіка – це наука, яка вивчає особливості виробничої діяльності людини з метою забезпечення ефективності, безпеки та зручності цієї діяльності, с. 106.

Ергономічний сайт – це сайт, що забезпечує необхідні зручності відвідувачеві, не викликає фізичної та психологічної втоми, не впливає на здоров'я та працездатність, с. 106.

З

Закладка – місце в текстовому документі, яке має унікальне ім'я, с. 88.

Заперечення логічного виразу x – логічна операція, результат якої дорівнює **true**, якщо значення логічного виразу x дорівнює **false**, і дорівнює **false**, якщо значення логічного виразу x дорівнює **true**, с. 210.

Змінні логічного типу – змінні, які можуть набувати одного з двох значень **true** або **false**, с. 209.

Зміст документа – перелік назв структурних частин документа із зазначенням номерів відповідних сторінок, с. 87.

Зовнішнє гіперпосилання – гіперпосилання для переходу з документа на зовнішні ресурси, с. 88.

Зовнішня пам'ять – складова пам'яті комп'ютера, що призначена для тривалого зберігання даних. До її складу входять пристрої на магнітних та оптичних дисках, флешнакопичувачі тощо, с. 24.

I

Інтерпретатор – комп'ютерна програма, що здійснює пооператорне (покомандне) перетворення в машинний код і виконання команд програми, яку записано деякою мовою програмування, с. 156.

Інтерфейс користувача – засоби керування (кнопки, меню та ін.), засоби введення даних (поля, лічильники та ін.), засоби виведення даних (вікно повідомлень, написи, текстові поля та ін.) та інші, а також методи їх використання, с. 149.

K

Клавіатура – пристрій введення, що призначений для введення символічних даних і команд, с. 30.

Кліп – невеликий за розміром (тривалістю) кіно- або відеофільм, с. 130.

Кодеки – програми, що містять алгоритми кодування і декодування мультимедійних даних, с. 122.

Кодування повідомлення – це процес заміни однієї послідовності сигналів, якою подане повідомлення, іншою послідовністю сигналів, с. 6.

Колірні моделі – це спосіб кодування різних кольорів спектра у вигляді впорядкованого набору числових значень певних базових компонентів, с. 12.

Колонтитули – службові повідомлення, які розміщують на полях сторінки і повторюються на інших сторінках документа, с. 73.

Коментар – текст, який ігнорується компілятором (інтерпретатором) і включається до тексту проекту з метою полегшення його розуміння та пошуку логічних помилок, с. 205.

Компілятор – комп'ютерна програма, що перетворює (компілює) програму, написану певною мовою програмування, на еквівалентний код іншою мовою програмування, який може виконати комп'ютер, с. 153.

Комп'ютерна програма – алгоритм опрацювання даних, записаний спеціальною мовою та призначений для виконання комп'ютером, с. 148.

Контент – змістове наповнення вебсторінки, що доступне користувачу: тексти, зображення, відео, звукові дані та інше, с. 105.

Кон'юнкція двох логічних виразів x і y – логічна операція, результатом якої є **true**, якщо значення кожного з логічних виразів x і y є **true**, і є **false**, якщо значення хоча б одного з логічних виразів x або y є **false**, с. 210.

Кратні одиниці вимірювання довжини двійкового коду:

$$1 \text{ кБ (кілобайт)} = 2^{10} \text{ Б} = 1024 \text{ Б};$$

$$1 \text{ МБ (мегабайт)} = 2^{10} \text{ кБ} = 2^{20} \text{ Б} = 1\,048\,576 \text{ Б};$$

$$1 \text{ ГБ (гігабайт)} = 2^{10} \text{ МБ} = 2^{20} \text{ кБ} = 2^{30} \text{ Б};$$

$$1 \text{ ТБ (терабайт)} = 2^{10} \text{ ГБ} = 2^{20} \text{ МБ} = 2^{30} \text{ кБ} = 2^{40} \text{ Б}, \text{ с. 17.}$$

L

Логічний вираз – вираз, який може набувати одне з двох значень: **true** або **false**, с. 209.

Логічні помилки – помилки, що проявляються як неочікуване переривання виконання проекту або як не-



правильні результати його виконання, с. 204.

М

Мапа сайту – схема, що відображає ієрархію вебсторінок і зв'язки між ними, с. 113.

Маркери меж – умовні позначки, перетягуючи які можна змінювати розміри об'єкта, с. 72.

Мікрофон – пристрій, що здійснює перетворення звукових коливань в електричні, с. 29.

Мова програмування – мова, яка використовується для запису програм, с. 149.

Монітор – основний пристрій для виведення даних у персональних комп'ютерах. Монітори поділяють за системою створення зображення на монітори на рідинних кристалах, або LCD-монітори, плазмові, OLED, на електронному чорнилі – e-ink тощо, с. 36.

Музичні редактори – програми для редагування звукових (музичних) даних, с. 128.

Музичні студії – програми, що використовуються в професійних цілях для редагування звукових (музичних) даних, с. 128.

Н

Налагодження проєкту – процес виявлення і виправлення помилок у проєкті, с. 204.

О

Обробник події – фрагмент проєкту, який буде виконуватися одразу після настання цієї події, с. 161.

Операторні дужки – слова **begin** і **end**, с. 162.

Опрацювання даних – це інформаційний процес, який полягає в отриманні нових даних з наявних, с. 6.

П

Підпрограма – окремо виділена частина проєкту, яка має своє ім'я та яку можна за цим іменем запустити на виконання, с. 163.

Плотер – пристрій виведення даних на папір, плівку або інші матеріали, що мають розмір A2 і більше, с. 36.

Подкаст – мультимедійний файл, що розповсюджується з використанням Інтернету і зазвичай містить думки авторів з різноманітних питань: політичних, економічних, технічних, літературних, музичних тощо, с. 145.

Подкастер – той, хто створює подкасти, с. 145.

Посилання – спеціальний запис у документі, який указує на якийсь об'єкт цього або іншого документа, с. 88.

Пріоритет виконання логічних операцій – спочатку виконуються операції заперечення, потім – операції кон'юнкції, а за ними – операції диз'юнкції, с. 211.

Принтер – пристрій виведення даних на папір, плівку або інші матеріали, що мають розмір зазвичай не більше ніж A3, с. 36.

Приховані (недруковані) символи – символи, які вводяться користувачем з клавіатури під час набору тексту, але зазвичай не відображаються на екрані, і не виводяться на папері при друкуванні, с. 64.

Проекційна клавіатура – тип пристрою введення даних, у якому зображення клавіатури проєктується на плоску поверхню і введення даних здійснюється «натисненням» на зображення клавіш, с. 30.

Процедура – окремо записана частина проєкту, яка має своє ім'я і яку можна за цим іменем запустити на виконання, с. 161.

Процесор – пристрій, який забезпечує опрацювання даних у комп'ютері. Він містить пристрій керування, який забезпечує виконання команд комп'ютерної програми, та арифметично-логічний пристрій, який і здійснює операції над даними, с. 22.

Р

Рекордер – програма для запису (захоплення) аудіо- та відеоданих, с. 122.

С

Семантика в мові програмування – установлене однозначне тлумачення мовних конструкцій, правил їх виконання, с. 149.

Сенсорний екран – пристрій введення і виведення даних. Для введення даних використовується властивість поверхні екрана реагувати на дотик пальця або стилуса, с. 29.

Синтаксис мови програмування – правила складання і запису мовних конструкцій: не словникових слів і речень, с. 149.

Синтаксичні помилки – помилки, які виникають у результаті використання символів, що не входять до алфавіту мови програмування, неправильне написання словникових слів, порушення синтаксичних правил, с. 149.

Системи опрацювання текстів – прикладне програмне забезпечення, яке призначене для створення й опрацювання текстів, с. 61.

Системи управління вебконтентом, WCMS – програми, що призначені для створення, редагування та керування вмістом вебсайту, с. 111.

Сканер – пристрій для введення графічних даних. За принципом дії виділяють планшетні, з протяжним механізмом, ручні та скан-камери, с. 32.

Словник мови програмування – набір слів, які використовуються в цій мові, с. 149.

Списки – особливий вид форматування абзаців текстового документа, які використовують для переліку об'єктів, опису порядку дій тощо, с. 70.

Стиль – набір значень властивостей певного типу об'єктів, який має власне ім'я, с. 78.

Структура документа – ієрархічна схема розміщення складових частин документа, с. 82.

Сублімаційний друк – друк, що використовує явище сублімації – фізичне явище переходу речовини з твердого стану в газоподібний, минаючи рідкий стан. Види сублімаційного друку – пря-

мий «мокрый» друк, прямий «сухий» друк, непрямий друк, с. 39.

Т

Таблиця кодів символів – це таблиця, у якій кожному символу, що може бути використаний у текстовому повідомленні, поставлено у відповідність деяке число, с. 9.

Табуляція – засіб, який дає змогу розміщувати об'єкти в рядку текстового документа у строго визначених місцях (позиціях табуляції), с. 74.

Тачпад – пристрій введення, що використовується в ноутбуках і нетбуках, для керування роботою комп'ютера, с. 29.

ТВ-тюнер – пристрій, що забезпечує перегляд телевізійних передач на комп'ютері, с. 29.

Теги – команди, що визначають розмітку тексту мовою **HTML**: його структуру, формат і розташування фрагментів тексту, забезпечують усталення нетекстових об'єктів на сторінку та інше, с. 107.

Текстовий документ – основний об'єкт опрацювання в системах опрацювання текстів, у якому можуть міститися символи, слова, речення, абзаци, сторінки, малюнки, таблиці, діаграми та інше, с. 62.

Тема документа – стиль оформлення текстового документа, який має ім'я і визначає для його об'єктів значення деяких властивостей (копір, розмір і шрифт символів, спосіб вирівнювання абзаців, міжрядковий інтервал, ефекти графічних зображень тощо), с. 79.

Тестовий набір вхідних даних – це набір значень вхідних даних, для яких можна заздалегідь визначити правильний результат виконання проекту, після чого потрібно порівняти його з реальним результатом виконання проекту для цих значень вхідних даних, с. 204.

Тестуванням проєкту – процес пошуку логічних помилок у проєкті з використанням тестових наборів вхідних даних, с. 204.



Тривимірні, або **3D-принтери**, – принтери, що використовуються для створення об'ємних об'єктів складної форми. Розрізняють тривимірні принтери за типом речовини, з якої «будується» об'єкт, та способом її отвердіння, с. 40.

Ф

Флешпам'ять – один з видів зовнішньої пам'яті, який використовує як носій даних електронні мікросхеми. Розрізняють такі пристрої флешпам'яті: USB-флешнакопичувач («флешка»), флешкарта (карта пам'яті), твердотілий, або SSD-накопичувач, с. 25.

Форма – це об'єкт, на якому під час створення проекту можна розмістити різні компоненти (елементи керування), зокрема, кнопки, поля, написи, меню, смуги прокручування та ін. Під час виконання проекту форма відкривається як вікно прикладної програми, на якому будуть відображатися всі об'єкти, розміщені на формі, с. 154.

Функція – це підпрограма, результатом виконання якої є одне або кілька значень: числа, рядки тексту та ін., с. 163.

Ю

Юнікод – таблиця кодів символів, яка дає можливість закодувати 1 114 112 різних символів, тобто усі символи писемності всіх світових мов, с. 10.

ASCII – таблиця кодів символів, у якій літерам англійського алфавіту, цифрам, розділовим знакам, символам редагування та форматування тексту поставлені у відповідність числа від 0 до 127, с. 9.

CMYK – колірна модель, у якій базовими компонентами є чотири кольори спектру – *блакитний* (англ. *Cyan*), *пурпурний* (англ. *Magenta*), *жовтий* (англ. *Yellow*), *чорний* (англ. *black*), с. 12.

DOCX – формат файлу текстового документа, у якому зберігається текст, рисунки, вставлені об'єкти, значення їх властивостей, використовується тек-

стовим процесором **Microsoft Word**, с. 62.

FLAC – формат файлів для стиснення аудіоданих без втрат, який використовується серед авторів і користувачів вільного програмного забезпечення, с. 121.

HSB – колірна модель, у якій базовими компонентами є *відтінок* (англ. *Hue*), *насиченість* (англ. *Saturation*) та *яскравість* (англ. *Brightness*), с. 12.

HTML – мова розмітки гіпертексту, що використовується для розробки вебсторінок, с. 107.

IDLE – інтегроване середовище розробки проектів мовою **Python**, с. 155.

MIDI – формат файлів, які містять команди для відтворення звуку спеціальним пристроєм або програмою-синтезатором, с. 121.

MP3 – один з найрозповсюдженіших форматів аудіофайлів. Використовується в глобальних мережах, файлообмінних системах, с. 121.

MP4 – формат файлів для зберігання мультимедійних даних у телебаченні, відеофільмах тощо, с. 121.

ODT – формат файлу текстового документа, створеного текстовим процесором **Open Writer**, с. 63.

PDF – формат файлу текстового документа, у якому зберігаються форматування, усі вставлені об'єкти, використовується програмою **Adobe Reader** або браузером, с. 63.

RGB – колірна модель, у якій базовими компонентами є три кольори спектру – *червоний* (англ. *Red*), *зелений* (англ. *Green*) і *синій* (англ. *Blue*), с. 12.

RTF – формат файлу текстового документа, у якому зберігається текст, рисунки, вставлені об'єкти, значення їх властивостей, використовується різними системами опрацювання текстів для різних операційних систем, с. 63.

QuickTime – формат файлів, що базується на технології збереження і відтворення відео- та звукових даних, яку було розроблено корпорацією **Apple**, с. 121.



SmartArt – графічні об'єкти **Microsoft Office**, які призначені для подання в текстовому документі структурованих даних у вигляді різноманітних схем, с. 98.

ТХТ – формат файлу текстового документа, у якому зберігається тільки текст з розбиттям на абзаци, без форматування, с. 62.

WAV – формат звукових файлів, який використовується як основний для нестиснутих аудіоданих в операційних системах сімейства **Windows**, с. 121.

WMA – формат файлів для аудіоданих, який розроблено корпорацією **Microsoft** і стандартно використовується в ОС **Windows**, с. 121.

WMV – формат файлів для відеоданих, який розроблено корпорацією **Microsoft** і стандартно використовується в ОС **Windows**, с. 121.

WYSIWYG – технологія, яка дає змогу відображати документ на екрані в такому самому вигляді, у якому він буде надрукований, с. 62.

YouTube – сервіс для розміщення відеофайлів в Інтернеті, с. 140.

3D-ручки – пристрої виведення даних для «малювання» в просторі. Використовуючи ці пристрої, будують зображення шляхом видавлювання тонкого струменя (0,5–1,5 мм) розпавленої пластмаси, яка відразу твердіє, утворюючи об'ємний об'єкт, с. 42.

З М І С Т

<i>Шановні восьмикласниці та восьмикласники!</i>	3
--	---

Розділ 1

КОДУВАННЯ ДАНИХ

1.1. Опрацювання даних як інформаційний процес. Кодування та декодування повідомлень	5
1.2. Кодування текстових і графічних даних	9
1.3. Двійкове кодування	14
<i>Практична робота № 1. «Розв'язування задач на визначення довжини двійкового коду повідомлень»</i>	20

Розділ 2

АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРА

2.1. Персональний комп'ютер, його основні складові	21
2.2. Пристрої введення даних	29
2.3. Пристрої виведення даних	36
2.4. Історія обчислювальних і комп'ютерних пристроїв	44
2.5. Види сучасних комп'ютерів та їх застосування	52
<i>Практична робота № 2. «Конфігурація комп'ютера під потребу»</i>	60

Розділ 3

ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ

3.1. Системи опрацювання текстів. Пошук і замінування фрагментів тексту	61
3.2. Створення в текстовому документі багаторівневих списків, розділів, стовпців, колонтитулів	69
<i>Практична робота № 3. «Створення текстового документа, що містить об'єкти різних типів»</i>	77
3.3. Використання стилів у текстових документах. Структура текстового документа	77
3.4. Зміст документа. Гіперпосилання. Колективна робота над документом	86
<i>Практична робота № 4. «Стилі. Структура документа. Автоматизоване створення змісту. Гіперпосилання»</i>	94



3.5. Створення спеціальних графічних об'єктів у текстовому документі	95
--	----

Розділ 4

СТВОРЕННЯ ТА ПУБЛІКАЦІЯ ВЕБРЕСУРСІВ

4.1. Структура вебсторінок. Поняття про мову HTML	105
4.2. Засоби автоматизованого створення та публікації вебресурсів	110
<i>Практична робота № 5. «Створення сайту з використанням онлайн-системи конструювання сайтів»</i>	117

Розділ 5

ОПРАЦЮВАННЯ ОБ'ЄКТІВ МУЛЬТИМЕДІА

5.1. Опрацювання мультимедійних об'єктів	118
5.2. Редагування аудіо- та відеоданих	128
<i>Практична робота № 6. «Створення відеофільму»</i>	139
5.3. Сервіси для роботи з аудіо- і відеоданими та публікування їх в Інтернеті	139
<i>Практична робота № 7. «Розміщення аудіо- та відеоматеріалів в Інтернеті»</i>	147

Розділ 6

АЛГОРИТМИ ТА ПРОГРАМИ

6.1. Комп'ютерні програми і мови програмування	148
6.2. Події. Обробники подій	160
<i>Практична робота № 8. «Проекти, що змінюють значення властивостей вікна та використовують вікно повідомлень»</i>	173
6.3. Кнопка	173
6.4. Напис	180
<i>Практична робота № 9. «Проекти з кнопками та написами»</i>	188
6.5. Поле. Проекти з вхідними даними і кінцевими результатами	190
<i>Практична робота № 10. «Проекти з полем, з уведенням даних і виведенням результатів»</i>	203
6.6. Налаштування проекту	204
6.7. Логічні вирази. Змінні логічного типу. Логічні операції	208
6.8. Розгалуження	214
6.9. Прапорці. Перемикачі	225
<i>Практична робота № 11. «Проекти з розгалуженнями»</i>	235
6.10. Цикл з лічильником	235
6.11. Цикл з передумовою	242
<i>Практична робота № 12. «Проекти із циклами»</i>	248
Словник	249

Відомості про користування підручником

№ з/п	Прізвище та ім'я учня / учениці	Клас	Навчальний рік	Оцінка	
				на початку року	в кінці року
1					
2					
3					
4					
5					