ВИДАВНИЦТВО
РАНОК

НАВЧАННЯ
БЕЗ МЕЖ

Інформатика

8

УДК 004:37.016(075.3)
І-74

Авторський колектив:
Олена Бондаренко, Василь Ластовецький,
Олександр Пилипчук, Євген Шестопалов

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 21.02.2025 № 347)

Видано за рахунок державних коштів.
Продаж заборонено

Підручник створено відповідно до модельної навчальної програми «Інформатика. 7–9 класи» для закладів загальної середньої освіти (автори Бондаренко О. О., Ластовецький В. В., Пилипчук О. П., Шестопалов Є. А.)

Інформатика : підруч. для 8 кл. закл. загал. І-74 серед. освіти / [О. О. Бондаренко, В. В. Ластовецький, О. П. Пилипчук, Є. А. Шестопалов]. — Харків : Вид-во «Ранок», 2025. — 272 с. : іл.

ISBN 978-617-09-9597-1

УДК 004:37.016(075.3)

Інтерактивний електронний додаток до підручника доступний за QR-кодом або посиланням **rnk.com.ua/107276**

ЗРОБЛЕНО В УКРАЇНІ

ISBN 978-617-09-9597-1

© Бондаренко О. О., Ластовецький В. В., Пилипчук О. П., Шестопалов Є. А., 2025
© ТОВ Видавництво «Ранок», 2025

ДОРОГІ ВОСЬМИКЛАСНИКИ ТА ВОСЬМИКЛАСНИЦІ!

У 7 класі ви з'ясували, що таке хмарні технології, штучний інтелект і як їх краще застосовувати; навчилися розробляти мальовані анімації за допомогою редактора двовимірної анімації; оволоділи новими засобами математичного моделювання. Ви також продовжили вивчення мови програмування Python і дізналися, як використовувати метод покрокової деталізації для розв'язування складних задач.

У цьому навчальному році на вас очікує чимало цікавого та корисного, а пропонований підручник буде вашим надійним помічником. Пригадаймо, як із ним працювати.

Навчальний матеріал підручника налічує шість розділів. Кожен розділ складається з параграфів і практичних робіт. Кожен параграф містить теоретичні відомості за темою уроку, приклади практичного застосування отриманих знань, матеріали за рубриками «Повторюємо», «Питання для самоперевірки», «Вправи», «Комп'ютерне тестування».

«*Повторюємо*» допоможе вам пригадати відомості, які ви вивчали в попередніх класах і які будуть корисні для засвоєння нового матеріалу. «*Питання для самоперевірки*» дадуть змогу з'ясувати, чи зрозуміли ви вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт за комп'ютером. «*Вправи*» складаються із завдань теоретичного і практичного спрямування; вони призначені для самостійного та групового виконання.

У підручнику ви знайдете покроковий опис *практичних робіт*. Щоб виконати кожну з цих робіт, потрібно повторити матеріал, вивчений на попередніх уроках, — тоді ви зможете успішно застосувати свої знання та виконати завдання.

До підручника пропонується інтерактивний електронний додаток, який містить додаткові матеріали до вправ і практичних робіт, тестові завдання з автоматичною перевіркою відповідей, 3D-моделі, навчальні відео, інтерактивні вправи, довідкові матеріали, список використаних джерел та ін.

Бажаємо натхнення та успіхів!

Авторський колектив

У тексті використано такі позначення:

Запам'ятайте

Розгляньте приклад

Ознайомтеся з цікавою інформацією

Зверніть увагу

Дайте відповідь на питання

Виконайте завдання за комп'ютером

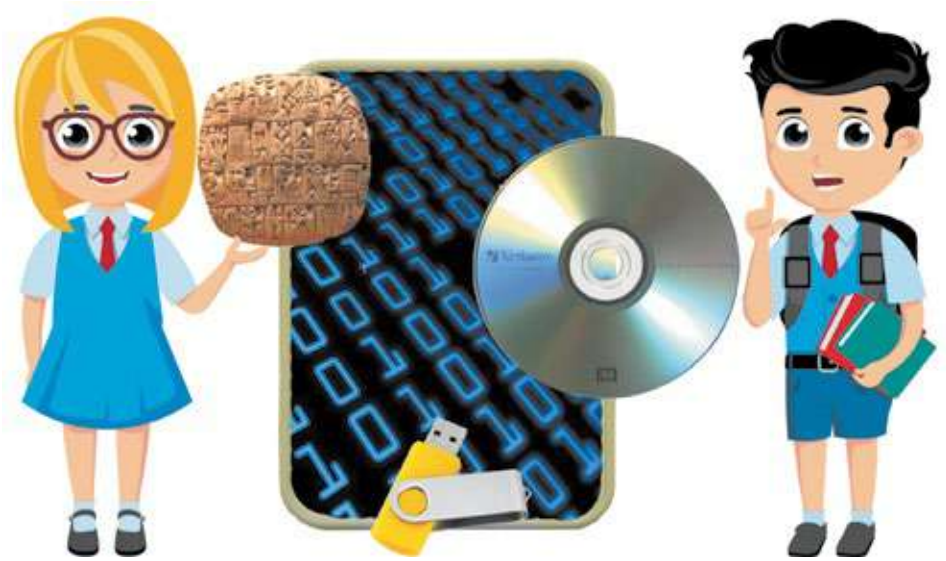
Виконайте завдання підвищеної складності

Виконайте завдання в парах

Знайдіть відповідь в інтернеті

Виконайте тестові завдання за комп'ютером

РОЗДІЛ 1
КОДУВАННЯ ДАНИХ ТА АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ



§ 1. Кодування та декодування повідомлень

§ 2. Двійкове кодування

§ 3. Кодування графічних даних

Практична робота 1. Визначення довжини двійкового коду текстових даних

§ 4. Архітектура комп'ютера. Технічні характеристики складових комп'ютера

§ 5. Історія опрацювання інформаційних об'єктів

Практична робота 2. Визначення конфігурації комп'ютера під потребу

ПОВТОРЮЄМО

Ви знаєте, що відомості про навколишній світ і процеси, які в ньому відбуваються, називають *інформацією*. Людина отримує інформацію у вигляді *повідомлень*, які сприймає органами чуття.

За *способом сприйняття* повідомлення поділяють на зорові (візуальні), смакові, нюхові, слухові (аудіальні) та дотикові (тактильні); за *способом подання* — на текстові, графічні, звукові, мультимедійні. Пристрої теж можуть приймати повідомлення за допомогою датчиків або сенсорів.

Вам відомо, що *стаціонарний комп'ютер* складається з системного блока, монітора, миші та клавіатури. Усі пристрої поділяються на пристрої введення, виведення, опрацювання, зберігання.

Дії, які можна виконувати з інформацією: пошук, опрацювання, передавання, зберігання, захист тощо, називають *інформаційними процесами*.

1. Що таке інформація?
2. Які бувають повідомлення за способом сприйняття; за способом подання?
3. Назвіть основні складові стаціонарного комп'ютера.
4. Які є пристрої зберігання інформації?
5. Які є пристрої введення інформації?
6. Назвіть пристрої виведення інформації.

У цьому розділі ви дізнаєтеся про кодування та декодування даних, детальніше ознайомитеся з апаратним забезпеченням комп'ютера, зі способами взаємодії комп'ютера і користувача.

§ 1. КОДУВАННЯ ТА ДЕКОДУВАННЯ ПОВІДОМЛЕНЬ

Із давніх-давен люди, сприймаючи навколишній світ, намагалися зберігати знання та враження у вигляді зарубок, вузликів на мотузці, наскельних малюнків тощо.

Опрацювання даних як інформаційний процес

Зрозуміло, що передавати збережену в такий спосіб інформацію від однієї людини до іншої було непросто. Проте до нас все таки дійшли відомості про життя наших предків (рис. 1.1).

а

б

в

Рис. 1.1

Обсяг інформації, потрібної людям, поступово зростав, а водночас удосконалювалися і способи її зберігання (рис. 1.2).

Рис. 1.2

Зібрану інформацію зазвичай *опрацьовують*, тобто виконують із нею певні дії (рис. 1.3, с. 8), результатом яких є нова інформація або змінення форми її подання. Інформацію, подану

у формі, зручній для зберігання, передавання та опрацювання, зазвичай за допомогою технічних засобів, називають **даними**. У сучасному світі інформацію опрацьовують здебільшого за допомогою комп'ютерів.

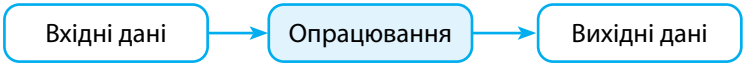


Рис. 1.3

Кодування та декодування

Повідомлення передаються від джерела до приймача каналами зв'язку. Щоб передати повідомлення за допомогою технічних засобів, його необхідно закодувати відповідно до типу каналу зв'язку. Після передавання повідомлення набуває вигляду, зрозумілого для приймача (рис. 1.4).

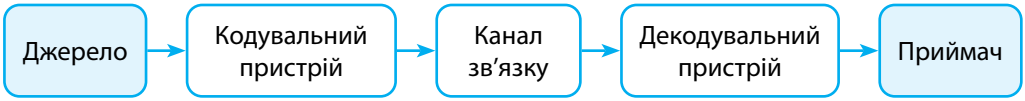


Рис. 1.4

Під час телефонної розмови звукові сигнали (голос) перетворюються на електричні (кодуються), які по мережі передаються до співрозмовника і знову перетворюються на звукові (декодуються).

Кодування — це перетворення повідомлення в зручну для передавання, опрацювання та зберігання форму.
Код — це набір символів або сигналів і правил їх використання для кодування повідомлень.
Декодування — це перетворення закодованого повідомлення у форму, прийнятну для приймача.

Після винайдення писемності як способу кодування природної мови найпоширенішим став спосіб подання повідомлень за допомогою символів, наприклад, ієрогліфів, літер, цифр тощо.

Окремими ієрогліфами позначаються певні об'єкти або поняття (рис. 1.5), літерами — окремі звуки усного мовлення, з яких складаються слова, речення тощо. Літери, що використовуються в мові, утворюють її *абетку*.

Рис. 1.5

Ті самі повідомлення, залежно від мети і використовуваних засобів, можна кодувати в різний спосіб: за допомогою цифр, знаків, символів, жестів, світлових, звукових сигналів та ін.

2 Звук [а] природної мови можна закодувати і літерою А, і положенням прапорців, і звуковими сигналами — коротким та довгим, які на папері позначають крапкою і тире (рис. 1.6).

Рис. 1.6

Для захисту текстових повідомлень можна вигадати власний код, наприклад, замінюючи літери певними символами, цифрами тощо. Для декодування повідомлення потрібно знати, як воно закодоване, це дасть змогу відновити його зміст.

3 Для секретного листування зі своїми воєначальниками давньоримський імператор Гай Юлій Цезар (100–44 рр. до н. е.) придумав власний шифр. Для шифрування кожна літера в тексті замінюється на літеру, віддалену від неї по колу за годинниковою стрілкою на задане число позицій.

Те саме позначення може мати різні значення залежно від галузі використання, мови тощо. Велика латинська літера V може позначати об'єм тіла, римську цифру 5, а P — звук, периметр.

Приймати, декодувати й опрацьовувати повідомлення може не лише людина, а й створений нею прилад, тварина (рис. 1.7) тощо. Способи кодування безперервно вдосконалюються.

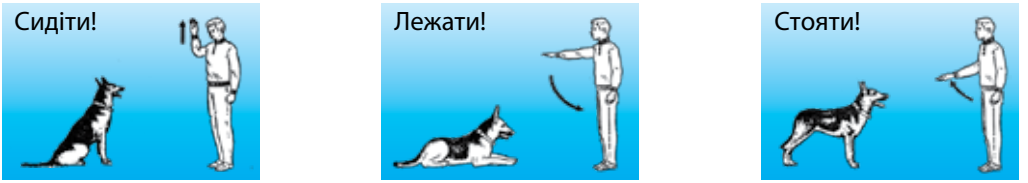


Рис. 1.7

Питання для самоперевірки

1. Як у давнину люди зберігали інформацію?
2. Що таке кодування; код; декодування?
3. Як повідомлення передається від джерела до приймача?
4. Як кодуються звуки природної мови?
5. Чи можна закодувати поняття? Наведіть приклад.
6. Опишіть кілька способів кодування звука [а].

Вправа 1

Виконати кодування й декодування повідомлень.

1. Запустіть текстовий редактор і створіть новий документ.
2. Розгляньте рис. 1.1. Яку інформацію ви отримали?
3. Знайдіть в інтернеті абетку Морзе і закодуйте своє прізвище.
4. Знайдіть в інтернеті інформацію про абетку Брайля та підготуйте коротке повідомлення про неї.
5. Вигадайте систему кодування букв, закодуйте повідомлення.
6. Обміняйтеся з однокласником або однокласницею кодами і декодуйте повідомлення. Збережіть файл із назвою Вправа 1.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107277).

§ 2. ДВІЙКОВЕ КОДУВАННЯ

Комп'ютер — універсальний пристрій для опрацювання даних різних типів. Розгляньмо, як у ньому кодуються дані.

Принцип двійкового кодування

Звичні вам числа називають десятковими, тому що для їх запису використовують 10 цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Дані ж у комп'ютері подаються електричними, світловими або магнітними сигналами, що набувають тільки двох станів: *вимкнено* (немає струму, розмагнічено) або *увімкнено* (є струм, намагнічено). Тому будь-які дані для опрацювання комп'ютером кодують за допомогою двох знаків — 0 і 1. Такий спосіб кодування називають **двійковим**. Двійкове кодування легко реалізувати в електронних пристроях.

Кодування символів

За допомогою однознакового двійкового коду можна закодувати лише два різні символи (див. 2-й стовпець таблиці). Тому для кодування символів у комп'ютері застосовують багатознакові двійкові коди, в яких код кожного символу складається з кількох записаних у рядок нулів або одиниць.

У таких кодах цифру 0 або 1 називають **бітом** (скор. від англ. *binary digit* — двійкова цифра). За кількістю бітів n, відведених для кодування одного символу, легко визначити число символів N, які можна закодувати: $N = 2^n$.

Буква	1 біт	2 біти	3 біти
А	0	00	000
Б	1	01	001
В	?	10	010
Г		11	011
Ґ		?	100
Д			101
Е			110
Є			111
Ж			?
...			

8-бітовий код придатний для кодування до $N = 2^8 = 256$ різних символів, а 16-бітовий код — до $N = 2^{16} = 65\,536$ символів.

Для кодування (декодування) тексту всі символи зводять у кодову таблицю, де поряд із двійковим кодом, а іноді й замість нього, записують десятковий код.

7-бітовий код ASCII

ASCII (*American Standard Code for Information Interchange* — американський стандартний код для обміну інформацією) є однією з найдавніших і найпоширеніших таблиць двійкових кодів символів. Її першу редакцію опубліковано 1963 року.

У цій таблиці десяткові числа від 0 до 31... і 127... є керувальними кодами, а решта — від 32 до 126 — кодують символи: великі й малі літери латинського алфавіту, розділові знаки, знаки математичних операцій тощо. Код 32 відповідає символу пропуск, код 33 — знаку оклику і т. д. Англійські літери в таблиці ASCII розташовані в алфавітному порядку.

Щоб побачити на екрані символи сучасної версії таблиці ASCII, можна скористатися такою програмою мовою Python:

```
for i in range(33, 127):
    print(chr(i), end = '')
```

Буде виведено:

```
!"#$%&'()*+,./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`
abcdefghijklmnopqrstuvwxyz{|}~
```

8-бітовий код KOI8-U

Для кодування літер інших алфавітів (зокрема кириличних) таблицю ASCII доповнюють до 256 символів. Для цього 7-бітові коди ASCII доповнюють зліва 0, а коди для нових символів утворюють додаванням зліва 1. Зокрема, для літер українського алфавіту використовують таблицю кодів KOI8-U (*Код Обміну Інформацією 8-бітовий Український*).

Розгляньмо фрагмент таблиці кодів KOI8-U:

192	11000000	ю	200	11001000	х	208	11010000	п
193	11000001	а	201	11001001	и	209	11010001	я
194	11000010	б	202	11001010	й	210	11010010	р
195	11000011	ц	203	11001011	к	211	11010011	с
196	11000100	д	204	11001100	л	212	11010100	т
197	11000101	е	205	11001100	м	213	11010101	у
198	11000110	ф	206	11001110	н	214	11010110	ж
199	11000111	г	207	11001111	о	215	11010111	в

Наприклад, мала літера а українського алфавіту в таблиці кодів KOI8-U має десятковий код 193, що відповідає двійковому коду 11000001.

▶ Стандарт Юнікод

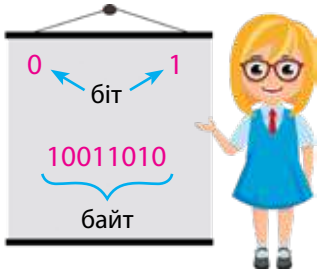
Із розвитком і поширенням комп'ютерів виникла потреба в кодуванні значно більшої кількості символів, ніж дозволяють 8-бітові коди.

Так, у сучасних текстових документах здебільшого використовується сімейство кодів, описаних у стандарті Юнікод (від англ. *Unicode* — уніфіковане кодування), що з'явився 1991 року. Поширений нині варіант Юнікоду — код UTF-8 — дає змогу закодувати до 1 112 064 різних символів.

Одиниці довжини двійкового коду

Двійкове кодування застосовують не лише для текстових, а й для графічних, звукових та інших даних. У всіх випадках обсяг даних вимірюють *довжиною двійкового коду*.

Як ви знаєте, послідовність із 8 двійкових розрядів (бітів) називають *байтом*.



3 Текст Марійка грає в шахи. складається з 20 символів із урахуванням пропусків і розділових знаків. Тому за 8-бітового кодування символів цей текст має обсяг 20 байтів.

Зверніть увагу, що на практиці зручніше користуватися більшими одиницями вимірювання: кібібайт (КіБ), мебібайт (МіБ), гібібайт (ГіБ), тебібайт (ТіБ).

1 КіБ = 2^{10} Б = 1024 Б ≈ 10^3 Б
1 МіБ = 2^{10} КіБ = 1024 КіБ ≈ 10^6 Б
1 ГіБ = 2^{10} МіБ = 1024 МіБ ≈ 10^9 Б
1 ТіБ = 2^{10} ГіБ = 1024 ГіБ ≈ 10^{12} Б

i Наведені в таблиці значення степенів двійки наближено дорівнюють множникам Міжнародної системи одиниць, які відповідають префіксам *кіло-*, *мега-*, *гіга-* тощо. Через те в повсякденному спілкуванні часто помилково використовують назви «кілобайт», «мегабайт», «гігабайт» тощо, навіть коли йдеться про їхні двійкові відповідники.

4 Нехай сторінка книжки містить 56 рядків по 64 символи в кожному рядку. Обчислимо обсяг даних (довжину двійкового коду) на сторінці за 8-бітового кодування.
Маємо: 56·64=3584 байти=3,5 КіБ. Щоб обчислити приблизний обсяг текстових даних у книжці, потрібно обсяг даних на сторінці помножити на кількість сторінок.
Розрахуємо обсяг даних у книжці з 256 подібних сторінок: 3584·256=917 504 Б=896 КіБ=0,875 МіБ.

! Не слід ототожнювати довжину двійкового коду текстового повідомлення з обсягом інформації, яку воно містить.

5 Двійковий код літери Р має довжину 1 байт. Сама ця літера, намальована на дорожньому знаку, несе інформацію для водія: *поряд зі знаком є облаштоване місце для стоянки*.

Тепер ви знаєте, в яких одиницях вимірюють обсяг даних, закодованих двійковим кодом, і як у комп'ютері кодуються текстові дані. Як кодуються дані інших типів, дізнаймося далі.

Питання для самоперевірки

1. Як кодуються дані в комп'ютері?
2. Чому в комп'ютері використовують двійкове кодування?
3. Як кодуються символи в комп'ютері?
4. Що таке біт; байт?
5. У яких одиницях вимірюють довжину двійкового коду?
6. Скільки байтів містить 1 КіБ; 1 МіБ?

Вправа 2

Дослідити двійкове кодування текстових даних.

1. Знайдіть в інтернеті таблицю кодів символів KOI8-U. У текстовому процесорі створіть таблицю з кодами малих українських букв, відсутніх у таблиці на с. 13.
2. Користуючись таблицею, розкодуйте повідомлення: 1001000 1100101 1101100 1101100 1101111 0100001.
3. Доповніть програму з прикладу 2 (с. 12) так, щоб поряд із символами виводилися їхні числові коди, і збережіть у файл із назвою Вправа 2-1.
4. Зробіть скриншот, на якому видно текст програми й результат її виконання, і збережіть у файл із назвою Вправа 2-2.
5. Розрахуйте кількість книжок (див. приклад 4), вміст яких можна зберегти на флешнакопичувачі обсягом 32 ГіБ.
6. Визначте приблизний обсяг текстових даних на одній зі сторінок підручника з інформатики.

Збережіть результати (п. 1, 2, 5 і 6).

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107278).

§ 3. КОДУВАННЯ ГРАФІЧНИХ ДАНИХ

Як ви вже знаєте, існує два основних способи кодування графічних даних: *растровий* і *векторний*.

Растрові зображення

Пригадаймо, що растровим називають зображення, подане як набір окремих крапок — **пікселів** (від англ. *picture element* — елемент малюнка). Піксель є найдрібнішою частинкою растрового зображення.

Як побудоване растрове зображення, легко зрозуміти, якщо відкрити в графічному редакторі фотографію і значно її збільшити. На рис. 3.1 показано пікселі, вишикувані в рядки і стовпці. Зображення на екрані монітора також має подібну будову, але через малий розмір пікселів ми бачимо його суцільним.

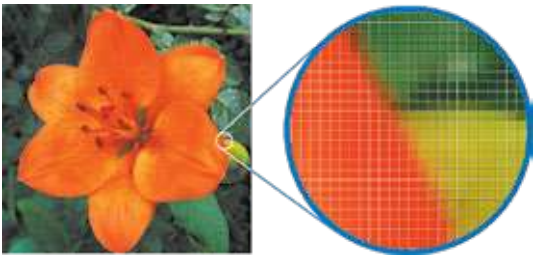


Рис. 3.1

! Кожен піксель має певний колір, і змінити його можна тільки повністю. Змінити колір частини пікселя неможливо.

Для **кодування зображення числами** потрібно:

1) поділити зображення на пікселі;
2) визначити розміри: кількість пікселів у довжину і в ширину;
3) визначити числовий код кольору кожного пікселя.

Зазвичай ці дії виконують за допомогою пристроїв, найвідомішими з яких є цифровий фотоапарат і сканер.

Кодування кольору

Щоб закодувати двоколірне (наприклад, чорно-біле) зображення, достатньо позначити колір кожного пікселя числом 0 (*перший колір*) або 1 (*другий колір*). Для зберігання коду пікселя вистачить одного біта, тому кажуть, що двоколірне зображення має глибину кольору 1 біт (рис. 3.2).

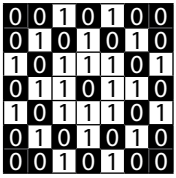


Рис. 3.2

Глибина кольору зображення — це кількість бітів, які використовуються для кодування кольору одного пікселя.

Зображення будиночка складається з пікселів восьми кольорів, які утворюють 32 рядки по 35 пікселів у кожному (рис. 3.3). Для кодування восьми кольорів достатньо десяткових чисел від 0 до 7, які можна подати двійковими послідовностями довжиною 3 біти, оскільки $2^3 = 8$. Отже, глибина кольору становить 3 біти.

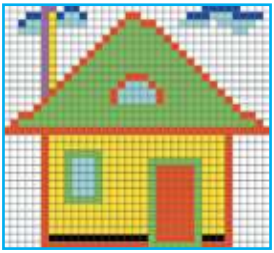


Рис. 3.3

Побудуємо таблицю десяткових і двійкових кодів цих кольорів:

Колір	Код	Колір	Код
Чорний	$0_{10} = 000_2$	Червоний	$4_{10} = 100_2$
Синій	$1_{10} = 001_2$	Фіолетовий	$5_{10} = 101_2$
Зелений	$2_{10} = 010_2$	Жовтий	$6_{10} = 110_2$
Коричневий	$3_{10} = 011_2$	Білий	$7_{10} = 111_2$

Закодуємо зображення послідовністю чисел, зазначивши ширину й висоту в пікселях і коди їх кольорів, перелічені зліва направо зверху вниз: 35 32 7 7 7 7 7 5 5 7 1 1 7 і т. д. За глибини кольору 3 біти довжина двійкового коду кольорів пікселів зображення будиночка становитиме:

$35 \cdot 32 \cdot 3 = 3360$ (бітів), або $3360 : 8 = 420$ (байтів).

За глибини кольору 8 бітів (або 1 байт) можна закодувати $2^8 = 256$ різних кольорів, а за глибини кольору 16 бітів (або 2 байти) — 2^{16}, що дорівнює 65 536 кольорам. Проте для досягнення фотографічної якості потрібна більша глибина кольору — 24 біти ($2^{24} = 16\,777\,216$ кольорів).

Як бачимо, зі збільшенням кількості кольорів глибина кольору зростає. За однакової кількості пікселів зображення це призводить до збільшення обсягу файлу.

У файлі растрового зображення зберігаються значення розмірів зображення та коди кольорів усіх пікселів. Файли більшості растрових форматів містять також іншу інформацію про зображення: дані про прозорість пікселів, тип фотокамери та ін.

2 Складемо програму мовою Python для виведення на екран растрового зображення, якщо відомий його код (приклад 1). Пікселі зобразимо квадратиками — відбитками Черепашки відповідних кольорів:

```
from turtle import*

pic = list(map(int,input().split()))  # Вводимо растровий код
W, H = pic.pop(0), pic.pop(0)     # Відокремлюємо ширину W і висоту H

pencolor('black')
shape('square'); shapesize(outline = 1)
size = 20              # Типовий розмір квадратика-Черепашки (пікселя)
colors = ['black', 'blue', 'green', 'brown', 'red', 'violet' ,'yellow', 'white']

up(); hideturtle()

for r in range(H):
    for c in range(W):              # Малюємо рядок пікселів
        fillcolor(colors[pic[r*W+c]])
        stamp()                     # Ставимо відбиток-піксель
        forward(size)
    back(W*size); right(90)         # Перехід на початок
    forward(size); left(90)         # наступного рядка
```

Векторні зображення

Пригадаймо, що **векторним** називають зображення, яке складається з *графічних примітивів* — геометричних об'єктів (відрізків, кіл, кривих тощо). Отже, базовим елементом векторного зображення є лінія, описана математичною формулою (рис. 3.4).

У найпростішому випадку криволінійний контур графічного примітива можна розглядати як ланцюжок прямолінійних відрізків, тобто ламану. У такому випадку векторний код контуру буде послідовністю чисел, яку можна побудувати так: перше число — кількість відрізків N, а наступні $2(N+1)$ чисел — координати послідовних вершин ламаної.

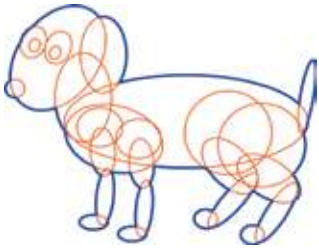


Рис. 3.4

3 Контур літери K (рис. 3.5) тричі зобразили ланцюжками відрізків, щоразу збільшуючи їх кількість на криволінійних ділянках. Як бачимо, зі збільшенням кількості відрізків зростає точність відтворення криволінійного контуру (рис. 3.6–3.8).

Початковий контур

Рис. 3.5

15 відрізків

Рис. 3.6

24 відрізки

Рис. 3.7

42 відрізки

Рис. 3.8

4 Векторний код контуру літери K із 15 відрізків (рис. 3.9) складатиметься з $1+2\cdot(15+1)=33$ чисел і може бути таким:

15 0 0 5 38 14 39 10 25 22 38 28 31 22 27 20 31 12 20 22 19 20 0 12 0 16 17 9 17 5 0 0 0

Координати першої та останньої точок однакові (0, 0), оскільки контур замкнутий.

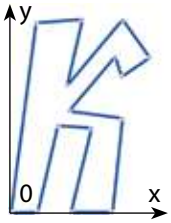


Рис. 3.9

Якщо малюнок складається з кількох контурів, можна послідовно записувати коди кожного з них. Це не ускладнить розкодування, оскільки кількість чисел у коді одного контуру цілком визначається кількістю відрізків, зазначеною на початку. Тому в програмі для виведення малюнка достатньо повторити виведення контуру, доки в коді залишаються неопрацьовані дані.

Складемо програму для виведення на полотно зображення контуру, заданого векторним кодом:

```
from turtle import *

size = 4                          # Масштабний множник
x0 = y0 = 0                       # Координати початкової точки

vector_code = input()             # Вводимо рядок з векторним кодом
# Перетворюємо рядок на список чисел
x = list(map(int, vector_code.split()))

N = x.pop(0)          # Відокремлюємо кількість відрізків у змінну N
y = x[1::2]                       # Формуємо список координат y
x = x[0::2]                       # Формуємо список координат x

up()
goto(x0, y0)                      # Йдемо в початкову точку
down()

for v in range(1, N+1):           # Будуємо відрізки
    goto(x0+size*x[v], y0+size*y[v])
```

Після запуску програми, увівши рядок із кодом, ми побачимо, як Черепашка послідовно малює відрізки, з яких утворюється векторне зображення (рис. 3.10).

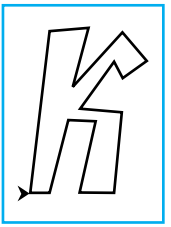


Рис. 3.10

Ми розглянули прості приклади растрового і векторного кодувань малюнків, проте в більшості випадків на практиці застосовуються значно складніші коди.

Питання для самоперевірки

1. Які є способи кодування графічних даних?
2. Опишіть особливості растрового зображення; векторного.
3. Якою буде довжина двійкового коду зображення розміром 2048×1536 пікселів, якщо глибина кольору 24 біти?
4. Поясніть, як виконується цикл for v in range(N) із прикладу 5.

Вправа 3

Виконати кодування графічних даних.

1. У середовищі програмування мовою Python запустіть програму з прикладу 2 (с. 18). Збережіть у файл із назвою Вправа 3-1 та випробуйте програму, ввівши такий код зображення: 5 4 6 4 4 4 6 4 1 6 1 4 4 1 6 1 4 6 4 4 4 6.
2. Поясніть результат, наведений на рис. 3.11.
3. Запишіть код кожного растрового зображення (рис. 3.12–3.14) і перевірте його за допомогою програми (див. п. 1).

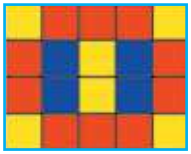


Рис. 3.11

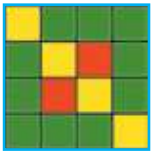


Рис. 3.12

Рис. 3.13

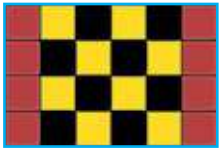


Рис. 3.14

4. Запустіть програму з прикладу 5, збережіть її у файл із назвою Вправа 3-2 і випробуйте, ввівши код із прикладу 4.
5. Змініть програму так, щоб масштабний множник size також уводився з клавіатури, і перевірте її роботу.
6. Змініть програму з прикладу 5, щоб зображення виводилося у випадковому місці. Збережіть файл із назвою Вправа 3-3.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107279).

ПРАКТИЧНА РОБОТА 1
ВИЗНАЧЕННЯ ДОВЖИНИ ДВІЙКОВОГО КОДУ ТЕКСТОВИХ ДАНИХ

Завдання: обчислити довжину двійкового коду текстових даних.
Обладнання: комп'ютер, простий текстовий редактор, підручник.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запишіть своє ім'я та прізвище. Визначте довжину двійкового коду записаного тексту за однобайтового кодування символів (див. приклад 3 на с. 14).
2. Запустіть простий текстовий редактор, наприклад, Блокнот. Наберіть своє ім'я та прізвище через пропуск і збережіть файл із назвою Практична робота 1.
   За допомогою контекстного меню файлу (команда Властивості) з'ясуйте розмір файлу в байтах і порівняйте зі значенням, отриманим у п. 1. Поясніть результат порівняння.
3. За допомогою фрагмента таблиці кодів KOI8-U (див. с. 13) закодуйте слово «байт» і запишіть результат за допомогою двійкових і десяткових кодів.
4. Користуючись програмою Калькулятор, визначте приблизний обсяг пам'яті, необхідний для зберігання тексту однієї сторінки підручника з інформатики і всього підручника (без малюнків) в електронному вигляді (див. приклад 4, с. 14).
5. Обчисліть, скільки подібних підручників може поміститися на флешнакопичувачі обсягом 64 ГіБ.
6. Усі розрахунки, отримані в п. 1–5, запишіть у текстовий документ і збережіть файл із назвою Практична робота 1. Завершіть роботу.

Зробіть висновок: як визначити довжину двійкового коду текстових даних і від чого залежить ця величина.

§ 4. АРХІТЕКТУРА КОМП'ЮТЕРА. ТЕХНІЧНІ ХАРАКТЕРИСТИКИ СКЛАДОВИХ КОМП'ЮТЕРА

Більшість сучасних комп'ютерів побудовано за принципами, які описав американський математик Дж. фон Нейман. Відтоді, попри розвиток технологій, зміни вигляду комп'ютерів та появу нових пристроїв, загальна схема роботи комп'ютера не змінилася (рис. 4.1).

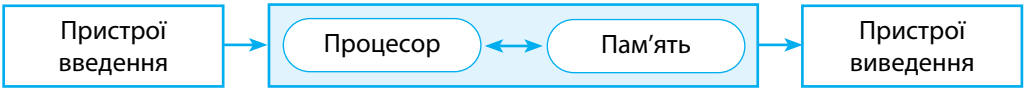


Рис. 4.1

Архітектура комп'ютера — це загальний опис будови комп'ютера та принципів взаємодії його складових.

У 1940-х роках найбільшого поширення отримали два типи архітектури комп'ютера: гарвардська (за місцем розробки — Гарвард, США) і фон Неймана. В обох основними вузлами є центральний процесор і пам'ять комп'ютера, але в першій програми та дані зберігаються в пам'яті комп'ютера окремо, а в другій — спільно. У сучасних комп'ютерах переважає простіша архітектура фон Неймана.

Основні пристрої персонального комп'ютера містяться в системному блоці. Це системна плата, процесор, оперативна пам'ять, відео- й аудіоадаптери, блок живлення та ін.

Системна (материнська) плата — пристрій комп'ютера, на якому розміщені процесор, внутрішня пам'ять, системна шина, порти для введення і виведення даних (з'єднувачі, слоти).

Системна шина (магістраль) — провідники і пристрої, призначені для обміну даними між іншими пристроями комп'ютера.

На рис. 4.2 наведено схему обміну даними.

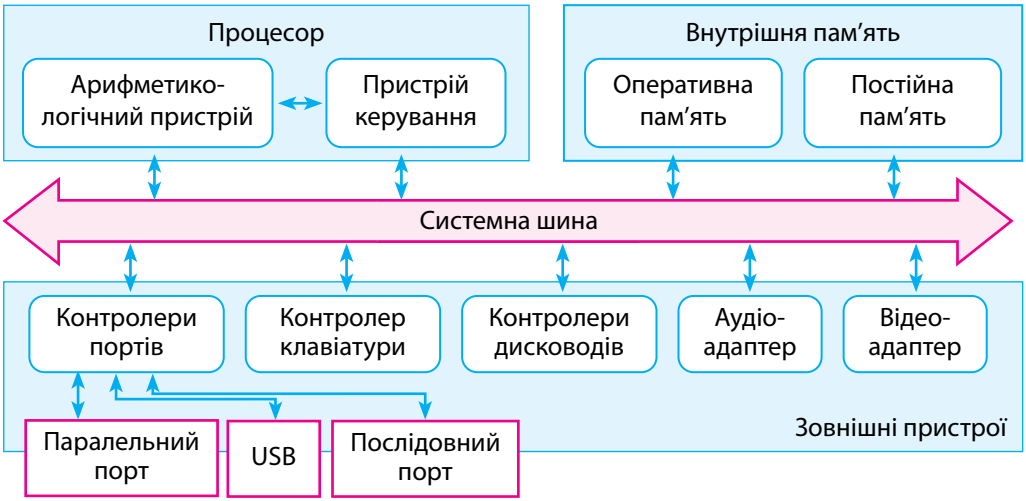


Рис. 4.2

Комп'ютер працює завдяки взаємодії двох складових: апаратної і програмної. Як ви знаєте, апаратну складову можна поділити на пристрої введення, виведення, опрацювання та зберігання. З'ясуймо, як саме пристрої введення та виведення приєднуються до комп'ютера і взаємодіють із ним.

Пристрої введення-виведення. Адаптери

Ви вже знайомі з такими пристроями введення-виведення даних, як клавіатура, миша, монітор (рис. 4.3). Їх називають *основними*.

Мишу та клавіатуру приєднують до відповідних портів комп'ютера. Проте спільна робота інших пристроїв введення-виведення і комп'ютера в цілому потребує узгодження. Тому для приєднання пристрою може знадобитися адаптер.

Рис. 4.3

Адаптер (контролер) — пристрій, призначений для приєднання до комп'ютера іншого пристрою та керування його роботою.

Адаптери бувають як порівняно прості, так і дуже складні, зі власним процесором, пам'яттю тощо. Наприклад:

- **відеоадаптер (відеокарта)** — пристрій для формування на основі коду зображення відеосигналу для монітора або проєктора, а також для захоплення відео з відеокамер та інших пристроїв (рис. 4.4, *а*);
- **аудіоадаптер (звукова карта)** — пристрій для відтворення та запису звуку, до якого приєднують мікрофон, звукові колонки, музичні інструменти тощо (рис. 4.4, *б*).

а

б

Рис. 4.4

Ці пристрої можуть бути як убудовані (інтегровані) в системну плату, так і встановлені в з'єднувач (слот) на ній. Відеоадаптер, монітор і спеціальні програми для забезпечення їх роботи утворюють *відеосистему комп'ютера*.

Основними характеристиками відеоадаптера є *частота оновлення зображення* та *обсяг відеопам'яті*. Від цих характеристик залежить якість зображення на екрані монітора під час перегляду фільмів, комп'ютерних ігор тощо.

Із винайденням USB (читається ю-ес-бі; від англ. *Universal Serial Bus* — універсальна послідовна шина), цей порт стали використовувати для приєднання до комп'ютера найрізноманітніших пристроїв.

Це, наприклад, миша та клавіатура (раніше — порт PS/2), принтер і сканер (раніше — паралельний порт Centronics), модем (раніше — послідовний порт RS-232), флешнакопичувач тощо. Застарілі пристрої до порту USB приєднують також за допомогою адаптерів (рис. 4.5).

Рис. 4.5

Пристрій опрацювання. Процесор

На системній платі встановлено процесор.

Процесор (центральний процесор) — це основний пристрій комп'ютера, призначений для виконання арифметичних і логічних операцій та керування роботою комп'ютера.

Конструктивно сучасний процесор (мікропроцесор) являє собою велику інтегральну схему, вміщену в захисний корпус (рис. 4.6). Від швидкодії процесора значною мірою залежить продуктивність комп'ютера.

Рис. 4.6

Розгляньмо основні технічні характеристики процесора:

Характеристика	Опис
Тактова частота	Кількість керувальних імпульсів (тактів) на секунду; вимірюється в герцах (Гц); визначає швидкодію процесора
Обсяг кеш-пам'яті	Обсяг надшвидкої пам'яті, яку процесор використовує для опрацювання часто вживаних даних
Розрядність	Кількість бітів даних, які процесор опрацьовує одночасно (розрядність сучасних процесорів — 64 біти)
Кількість ядер	Кількість обчислювальних ядер (окремих процесорів) в одній мікросхемі
Сокет	Тип з'єднувача для процесора на системній платі

Розглянемо стислий запис характеристик процесора:
Intel Core i9-14900KF 24C 3.2GHz 36MB LGA1700
Маємо: Intel — виробник; Core i9-14900KF — тип; 24C — кількість ядер; 3,2 ГГц — тактова частота; 36 МБ — обсяг кеш-пам'яті; LGA1700 — сокет.
Слід пам'ятати, що сокет для процесора на системній платі має відповідати сокету самого процесора, інакше встановити процесор на цю плату буде неможливо.

Пристрої збереження. Пам'ять

Звідки процесор отримує інформацію для опрацювання?

Пам'ять комп'ютера — це складова комп'ютера, призначена для тимчасового або постійного зберігання даних і програм.

Пам'ять поділяють на внутрішню і зовнішню; вона може бути енергозалежною й енергонезалежною (рис. 4.7).

Рис. 4.7

Після вимкнення комп'ютера дані з енергозалежної пам'яті зникають, а в енергонезалежній пам'яті зберігаються постійно.

Внутрішня пам'ять — це швидкодійна пам'ять, реалізована з використанням мікросхем і розміщена на системній платі.

Оперативна пам'ять (ОЗП — *Оперативний Запам'ятовувальний Пристрій*, або **RAM** — *Random Access Memory*) призначена для зберігання даних і програм під час роботи комп'ютера.

Основною технічною характеристикою ОЗП є його обсяг. За недостатнього обсягу комп'ютер працюватиме повільніше.

Постійна пам'ять (ПЗП — *Постійний Запам'ятовувальний Пристрій*, або **ROM** — *Read-Only Memory*) призначена для зберігання програм і даних під час початку роботи комп'ютера.

Зокрема, в постійну пам'ять записана базова система введення-виведення (**BIOS** — *Basic Input-Output System*) — програма для керування роботою пристроїв, завантаження операційної системи тощо.

Зовнішня пам'ять — це пам'ять, призначена для довготривалого енергонезалежного зберігання даних.

Розгляньмо деякі зовнішні запам'ятовувальні пристрої:

Пристрій	Назва пристрою	Принцип дії
	Накопичувачі на основі флешпам'яті (SSD, «флешка»)	Дані кодуються електричними зарядами в кристалі мікросхеми
	Накопичувач на жорстких магнітних дисках (HDD)	Дані записуються на диск завдяки намагнічуванню його поверхні
	Оптичні CD- і DVD-накопичувачі	Дані записуються на диск або зчитуються лазерним променем

Конфігурація комп'ютера

Складові комп'ютера бувають дуже різні, тому його продуктивність залежить від конфігурації. Конфігурацію комп'ютера визначає перелік його пристроїв та їхні технічні характеристики. Оскільки комп'ютерна техніка весь час розвивається і вдосконалюється, то й конфігурація порівняно швидко застаріває.

Стислий запис конфігурації комп'ютера може виглядати так: 27" (2560x1440) / AMD Ryzen 9 7950X (4.5–5.7 ГГц) / ОЗП 64 ГБ / SSD 2 ТБ / NVIDIA GeForce RTX 4090 24 ГБ / DVD+RW / LAN / Ubuntu.

Як бачимо, в описі конфігурації комп'ютера використовують скорочені позначення для характеристик монітора, процесора,

оперативної пам'яті, накопичувача (твердотільного або на жорстких магнітних дисках), відеоадаптера, оптичного дисковода, мережевої карти, операційної системи та ін.

Питання для самоперевірки

1. Що таке процесор?
2. Що визначає швидкодію комп'ютера?
3. Для чого потрібна оперативна пам'ять?
4. Які функції постійної пам'яті комп'ютера?
5. Яке призначення пристроїв зовнішньої пам'яті?
6. Що розуміють під конфігурацією комп'ютера?

Вправа 4

Визначити характеристики процесора й оперативної пам'яті.

1. Запустіть текстовий процесор і створіть новий документ.
2. З'ясуйте в інтернеті, як в операційній системі вашого комп'ютера дізнатися характеристики процесора та ОЗП. Знайдіть їх і запишіть у документ.
3. За каталогами інтернет-магазинів знайдіть опис одного з процесорів. Скопіюйте в документ посилання на сторінку та стислий запис характеристики процесора (див. приклад 1).
4. Визначте за цим записом і запишіть: розрядність; кількість ядер; тактову частоту; обсяг кеш-пам'яті; сокет.
5. З'ясуйте за допомогою чат-бота та скопіюйте в документ короткий опис таких характеристик ОЗП: пропускна здатність; тип; обсяг; частота. Обговоріть результат у групі.
6. З'ясуйте в інтернеті й запишіть у документ, що означають окремі частини скороченого запису конфігурації комп'ютера (див. приклад 2). Збережіть файл із назвою Вправа 4.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107280).

§ 5. ІСТОРІЯ ОПРАЦЮВАННЯ ІНФОРМАЦІЙНИХ ОБ'ЄКТІВ

Люди здавна розробляли засоби для полегшення опрацювання інформації, зокрема різних інформаційних об'єктів.

Етапи розвитку обчислювальних засобів

Залежно від технологій розрізняють кілька етапів.

На **домеханічному етапі** люди використовували для лічби різні підручні засоби: камінці, мотузки, зерна тощо. Згодом вони винайшли й почали використовувати найпростіші обчислювальні пристрої, наприклад, *абак* і його різновиди.

На **механічному етапі**, з розвитком промисловості й торгівлі, створено низку механічних пристроїв для обчислень. Попри простоту конструкції, вони допомагали значно прискорити розрахунки. Ідеї, закладені в XVII ст. у *калькуляторі Паскаля* та *арифмометрі Лейбніца*, використовували в різних пристроях аж до середини XX ст. Біля витоків механічної обчислювальної техніки стояли відомі науковці, інженери і дослідники (рис. 5.1).

Леонардо да Вінчі

Блез Паскаль

Ґотфрід Вільгельм Лейбніц

Чарлз Беббідж

Ада Лавлейс

Рис. 5.1

На **електронному етапі** створено *комп'ютери*, розвиток яких триває й нині. Значний внесок у вдосконалення обчислювальної техніки зробили такі провідні IT-корпорації світу, як Intel, IBM, AMD, Apple та ін. Комп'ютер із приладу, цікавого спочатку лише науковцям, наразі перетворився на цілком буденну річ.

Основні етапи розвитку комп'ютерів

Нинішні комп'ютери дуже різноманітні. Вони відрізняються особливостями архітектури, швидкістю обчислень, призначенням і багатьма іншими ознаками. Цьому розмаїттю передувала захоплива історія розвитку комп'ютерів як універсального засобу для роботи з різними інформаційними об'єктами.

На рис. 5.2 показано, як поділяють комп'ютери на кілька поколінь за елементною базою.

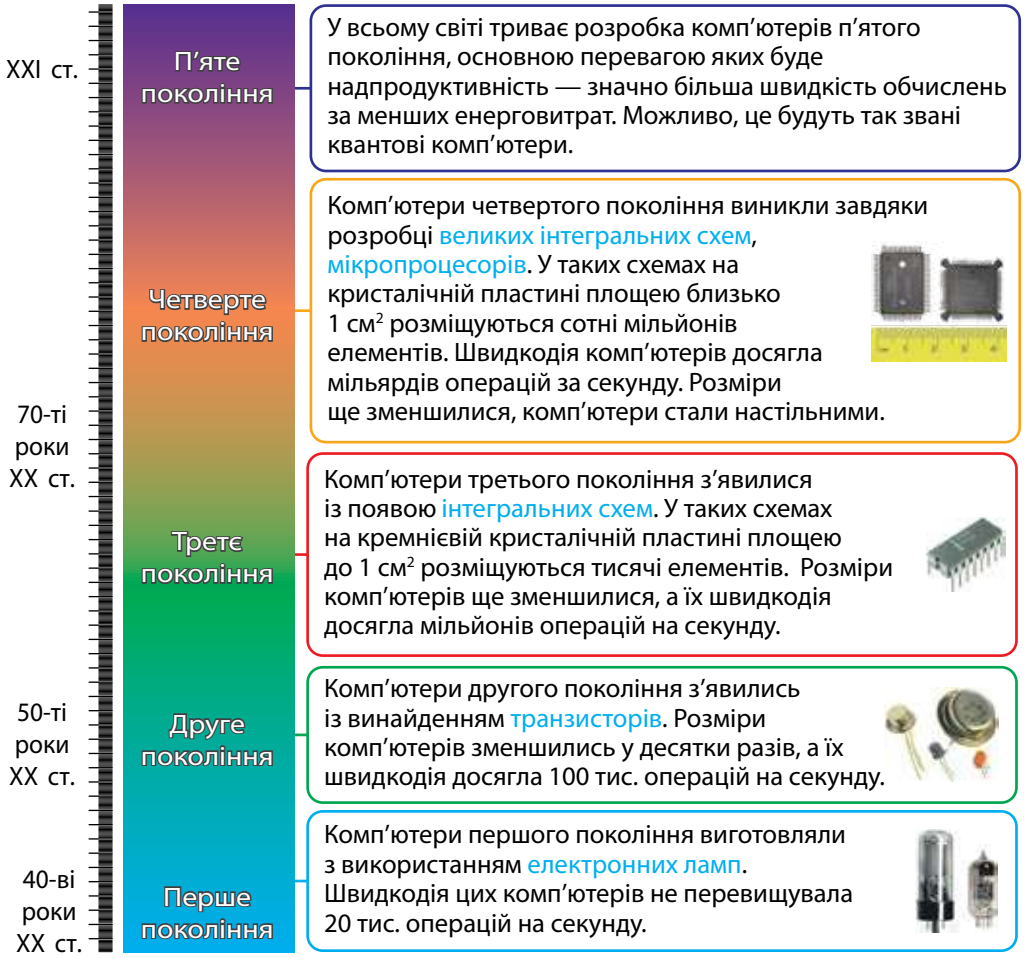


Рис. 5.2

Перші комп'ютери створено в різних країнах у 40–50-х роках XX століття: Z3 (Німеччина, 1941), Colossus (Велика Британія, 1943), Harvard Mark I (США, 1944), МЕСМ (СРСР, 1950).

Академік Сергій Лебедєв (рис. 5.3) — творець першого в континентальній Європі комп'ютера. Під його керівництвом 1951 року в Києві створено електронну обчислювальну машину «МЕСМ» (Мала електронна обчислювальна машина).

Рис. 5.3

У 60-х роках XX ст. в Інституті кібернетики Академії наук України створено серію машин для інженерних розрахунків (скорочено МІР). Колектив очолював Віктор Глушков (рис. 5.4), автор фундаментальних праць у галузі кібернетики, математики та обчислювальної техніки.

Рис. 5.4

Один із перших персональних комп'ютерів (ПК) створили в 1976 році Стів Джобс і Стів Возняк (рис. 5.5), засновники Apple Computer, США. Перший комп'ютер компанії — Apple I (див. рис. 5.5) — мав процесор, який працював на частоті 1 МГц; оперативну пам'ять обсягом 4 КіБ; постійну пам'ять обсягом 256 Б.

Рис. 5.5

Можливо, комп'ютерами п'ятого покоління стануть штучно створені біологічні системи, біопроцесорами в яких та пристроями пам'яті слугуватимуть молекули ДНК, білкові молекули тощо.

Ідею комп'ютера на базі ДНК запропонував американський вчений Леонард Адлеман у 1994 році. Нині в кількох країнах тривають дослідження в галузі біокомп'ютерних технологій.

Від 1980 року розвивається галузь квантових обчислень. Наприкінці 2020 року китайські вчені повідомили, що прототип квантового комп'ютера Цзю Чжан за 200 с виконав обчислення, на які звичайному суперкомп'ютеру знадобилося б 1 млрд років.

Види сучасних комп'ютерів

Сучасні комп'ютери відрізняються обчислювальними ресурсами. Дізнаймося про це докладніше.

Суперкомп'ютери — багатопроцесорні комп'ютерні системи, призначені для здійснення складних обчислень.

Найпотужніші в Україні комп'ютери мають центри суперкомп'ютерних обчислень у Національному технічному університеті «Київський політехнічний інститут імені Ігоря Сікорського», Інституті кібернетики імені В. М. Глушкова НАН України та Київському національному університеті імені Тараса Шевченка.

Рейтинг найпотужніших суперкомп'ютерів світу постійно змінюється. Так, японський суперкомп'ютер Fugaku утримував першість два роки, а в 2022 році поступився суперкомп'ютеру Frontier, установленому в Оук-Риджській національній лабораторії США. Frontier перетнув рубіж в 1 ексафлопс (1 квінтильйон операцій на секунду) і станом на 2024 рік очолював рейтинг [1]. Проте в листопаді 2024 року найшвидшим у світі став американський суперкомп'ютер El Capitan [2].

Мейнфрейми — високопродуктивні надійні сервери з великим обсягом пам'яті (рис. 5.6), призначені для використання в критично важливих системах (наприклад, керування електростанціями, продаж квитків, хмарні технології тощо).

Мікрокомп'ютери призначені для реалізації простих задач керування, наприклад, побутовою технікою, верстатами (рис. 5.7).

Рис. 5.6

Рис. 5.7

Персональні комп'ютери (ПК) призначені для використання однією людиною, тобто для персонального користування.

Конструкції та розміри ПК досить різноманітні (рис. 5.8).

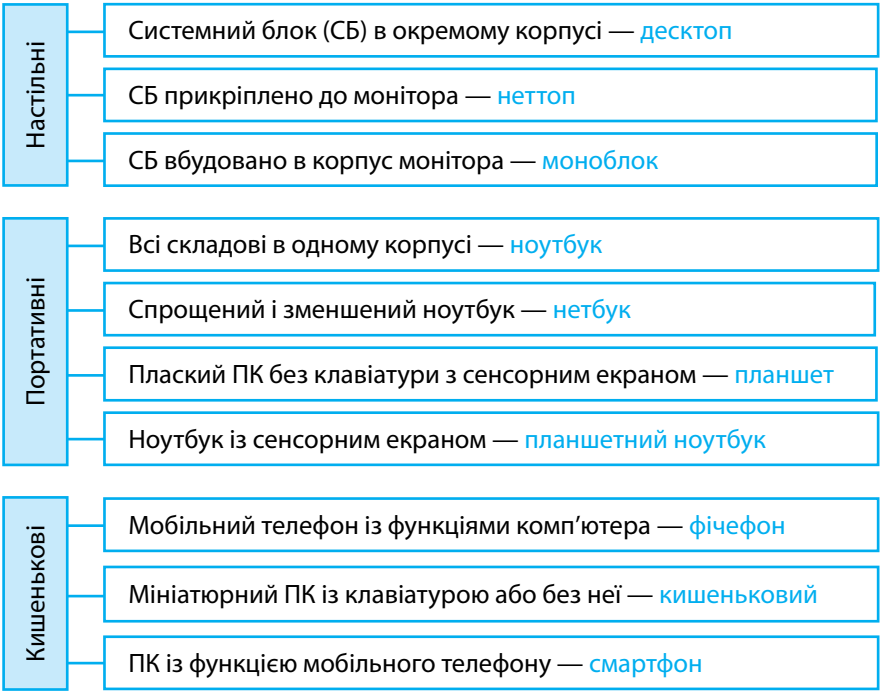


Рис. 5.8

ПК вибирають, зважаючи на потреби людини, яка його використовуватиме (рис. 5.9). Виробництво комп'ютерів постійно вдосконалюється, розширюється сфера їх застосування (рис. 5.10).

Рис. 5.9

Рис. 5.10

Без комп'ютерів складно уявити наш побут та дозвілля. Набувають поширення нові види комп'ютерів: роботизовані іграшки (рис. 5.11), ро́боти-домашні помічники тощо. Автоматика «розумного будинку» здатна діяти замість людини, спростити побутові складнощі людям похилого віку, особам з інвалідністю та ін.

Рис. 5.11

Питання для самоперевірки

1. Назвіть основні етапи розвитку обчислювальних засобів.
2. Коли створено перші комп'ютери?
3. Як із часом змінювалася елементна база комп'ютерів?
4. Яка елементна база персональних комп'ютерів?
5. Хто є основоположником розробки комп'ютерів в Україні?
6. Стисло опишіть види сучасних комп'ютерів.

Вправа 5

Підготувати повідомлення про історію розвитку ЕОМ.

1. Запустіть текстовий процесор і створіть новий документ.
2. Знайдіть у матеріалах параграфа або в інтернеті дані про швидкодію комп'ютерів першого покоління й сучасного комп'ютера та запишіть у документ.
3. Знайдіть в інтернеті та скопіюйте в документ інформацію про внесок Джона Вінсента Атанасова і Михайла Пилиповича Кравчука у створення перших комп'ютерів.
4. Складіть повідомлення на тему «Комп'ютери майбутнього».
5. Уведіть текст у документ. Збережіть файл із назвою Вправа 5.
6. Обговоріть з однокласниками та однокласницями перспективи розвитку комп'ютерів.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107281).

ПРАКТИЧНА РОБОТА 2
ВИЗНАЧЕННЯ КОНФІГУРАЦІЇ КОМП'ЮТЕРА ПІД ПОТРЕБУ

Завдання: визначити конфігурацію комп'ютера під потребу комп'ютерної гри з такими системними вимогами:
- 150 ГіБ вільного місця на диску;
- 16 ГіБ оперативної пам'яті;
- процесор Intel Core i9-10900K або AMD Ryzen 9 5900X;
- відеоадаптер AMD Radeon RX 5700 XT 8 GB або NVIDIA GeForce RTX 2070 SUPER 8GB.

Обладнання: комп'ютер зі встановленими браузером і текстовим процесором, з'єднаний з інтернетом.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. За каталогами інтернет-магазинів доберіть складові для комплектування комп'ютера з конфігурацією для гри.
2. Запустіть текстовий процесор і створіть новий документ. Уведіть назви та вартість (у гривнях) таких пристроїв:
   - корпус системного блока з блоком живлення;
   - системна плата;
   - процесор; монітор;
   - оперативна пам'ять;
   - накопичувач SSD;
   - відеоадаптер (якщо він не вбудований у системну плату);
   - клавіатура; миша.
3. Визначте, від яких пристроїв залежить відповідність конфігурації потребам гри. Скопіюйте назви пристроїв, технічні характеристики та посилання на сайт інтернет-магазину.
4. Запишіть, за якими характеристиками вибрано процесор; системну плату. Збережіть файл із назвою Практична робота 2.

Зробіть висновок: як визначити конфігурацію комп'ютера під потребу, які складові є обов'язковими та які характеристики вони повинні мати.

РОЗДІЛ 2
СТВОРЕННЯ ТА ПУБЛІКАЦІЯ ВЕБРЕСУРСІВ

§ 6. Адресація в інтернеті

§ 7. Обмін даними в інтернеті

§ 8. Створення сайтів

§ 9. Розмічання тексту засобами HTML

§ 10. Графічні зображення та гіперпосилання на вебсторінці

§ 11. Розмічання таблиць засобами HTML

Практична робота 3. Створення головної сторінки сайту засобами HTML

ПОВТОРЮЄМО

Ви знаєте, що комп'ютери в мережі мають різні функції. Комп'ютер, який надає ресурси в мережу, називають *сервером*, а комп'ютер, який ці ресурси використовує, — *клієнтом*.

Найбільша глобальна комп'ютерна мережа «Інтернет» надає користувачам і користувачкам усього світу можливості онлайн-спілкування, електронного листування, спільного опрацювання документів, обміну файлами тощо.

За допомогою *спеціальних сервісів* (хмарні диски, Facebook, YouTube та ін.) в інтернеті можна розміщувати файли та надавати до них спільний доступ багатьом особам.

Ви знаєте, що сайт складається з вебсторінок, взаємопов'язаних гіперпосиланнями й поєднаних спільною темою. Існують сайти *різного призначення* — навчальні, соціальні, розважальні тощо.

1. Який комп'ютер називають клієнтом; сервером?
2. Що таке інтернет? Назвіть кілька відомих вам сервісів інтернету.
3. Що таке електронна пошта?
4. Що таке хмарний диск; для чого використовується?
5. Що таке сайт? На яких сайтах ви зареєстровані?
6. Якими сайтами ви користуєтеся під час навчання?

У цьому розділі ви навчитеся створювати й оформлювати власні сайти, наповнювати їх умістом та розміщувати в інтернеті. Ви дізнаєтеся, що таке мова розмітки гіпертексту і як її застосовувати для форматування сторінок сайтів.

§ 6. АДРЕСАЦІЯ В ІНТЕРНЕТІ

Призначенням будь-якої комп'ютерної мережі є надання доступу до віддалених інформаційних, апаратних і програмних ресурсів і спільного їх використання.

Призначення і структура інтернету

Інтернет об'єднує окремі комп'ютери та локальні мережі в усьому світі. Розгляньмо структуру інтернету (рис. 6.1).

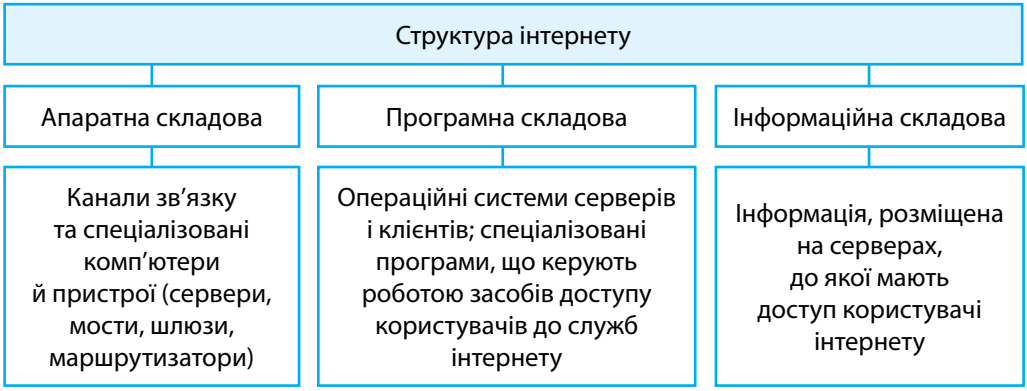


Рис. 6.1

Основу глобальної мережі «Інтернет» складають комп'ютери-сервери, з'єднані між собою лініями (каналами) зв'язку. Комп'ютери-клієнти можуть перебувати на значній відстані від сервера і з'єднуватися з ним різними каналами зв'язку.

IP-адреса

Для злагодженної роботи всіх зазначених складових структури інтернету потрібна система адресації.

IP-адреса — це ідентифікаційний номер комп'ютера в мережі.

Як і в локальній мережі, IP-адреса комп'ютера в інтернеті створюється за протоколом IPv4 і складається з чотирьох десяткових чисел від 0 до 255, розділених крапками. Наприклад, 78.111.176.233.

Із часом з'ясувалося, що кількості комбінацій чотирьох чисел (4,2 млрд) для потреб адресації недостатньо. Починаючи від 2008 року запроваджено протокол IPv6, за яким IP-адреса записується вісьмома чотирицифровими шістнадцятковими числами, розділеними двокрапками. Наприклад: 2011:0db2:11d3:087f:07a0:345e:8a2e:32c2. Це еквівалентно 16 десятковим числам від 0 до 255 і збільшує кількість можливих адрес до $3{,}4 \cdot 10^{38}$.

Комп'ютер у мережі може мати *статичну* (*постійну*) або *динамічну* (*тимчасову*) IP-адресу. Статичні адреси мають усі сервери, щоб інші комп'ютери «знали», де шукати інформацію, а динамічну комп'ютер отримує при тимчасовому з'єднанні.

Доменне ім'я

Для адресації ресурсів використовують доменні імена.

Доменне ім'я — це текстова адреса комп'ютера в інтернеті.

Домен (від англ. *domain* — територія, область) — певна зона в системі доменних імен, виділена власникові домену (країні, міжнародній організації, регіону, юридичній або фізичній особі) для забезпечення доступу через інтернет до певної інформації.

Доменне ім'я складається з назв (імен) кількох доменів, відокремлених крапками. Їх рівень зростає справа наліво (рис. 6.2).

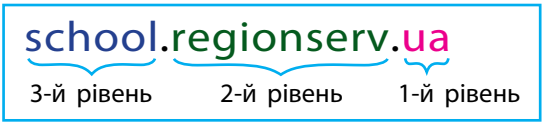


Рис. 6.2

До кожного домену *1-го* рівня можуть належати багато доменів *2-го* рівня; до кожного домену *2-го* — багато доменів *3-го*.

Розгляньмо, як класифікують домени *1-го* рівня (рис. 6.3).

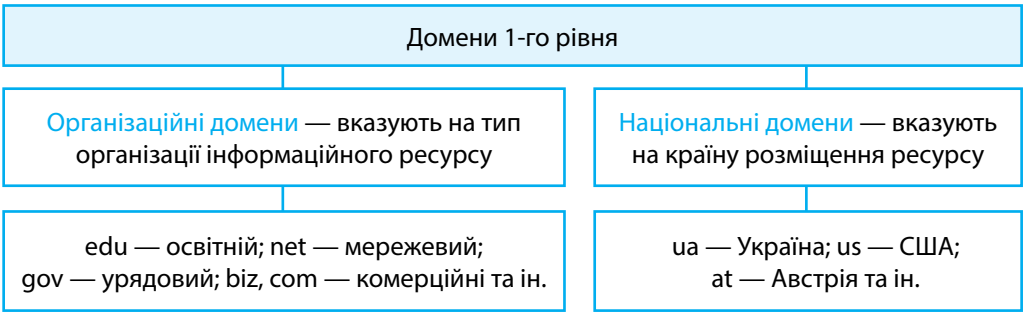


Рис. 6.3

Домен *1-го (верхнього)* рівня зазвичай вказує тип організації або державу; домен *2-го* рівня часто є назвою сервера даних; першою зліва в доменному імені є назва ресурсу.

Кожному доменному імені відповідає певна IP-адреса. IP-адресі комп'ютера може відповідати кілька доменних імен. Перетворення доменного імені на IP-адресу здійснюють **DNS-сервери** (*Domain Name Service* — служба доменних імен).

URL

Щоб отримати інформаційні матеріали з інтернету, адреси сервера недостатньо. Потрібна адреса ресурсу із зазначенням протоколу й унікального шляху до нього. Таку адресу називають уніфікованим вказівником на ресурс — **URL** (*Uniform Resource Locator*). Це стандартизована адреса ресурсу в інтернеті (документа, відео, вебсторінки, зображення тощо).

URL зазвичай складається з позначення протоколу обміну даними (http, https, ftp та ін.); доменного імені або IP-адреси сервера, де зберігається файл; шляху до файлу на сервері.

Адреса вебсторінки може мати такий вигляд:

https://www.example.com/32menu.html,

де https — захищений протокол передавання гіпертексту; www.example.com — доменне ім'я сервера; 32menu.html — ім'я файлу вебсторінки.

В адресному рядку браузера назву протоколу в адресі вебсторінки, наприклад https або http, часто зазначати не обов'язково.

ОС Windows має вбудовані засоби для перегляду вебсторінок. Якщо в адресному рядку вікна будь-якої папки замість шляху до неї набрати URL вебсторінки та натиснути клавішу Enter, вона відкриється для перегляду у вікні браузера.

Питання для самоперевірки

1. Що входить до апаратної і програмної складових інтернету?
2. Що таке IP-адреса? Що таке протокол IPv4; протокол IPv6?
3. Чим відрізняються постійна та динамічна IP-адреси?
4. Що таке доменне ім'я? У чому зручність його використання?
5. Назвіть кілька прикладів доменів першого рівня.
6. Що таке URL? Яке призначення URL?

Вправа 6

■ Визначити IP-адресу комп'ютера та складові URL ресурсу.

1. Запустіть текстовий процесор і створіть новий документ.
2. Визначте IP-адресу вашого комп'ютера на спеціалізованому сайті за вказівкою вчителя / вчительки. Запишіть IP-адресу. Яким сайтом (адреса, назва) скористалися для її визначення?
3. Знайдіть в інтернеті відомості про те, які символи є обов'язковими елементами великого Державного Герба України. Скопіюйте один з ескізів Державного Герба України.
4. Відкрийте сторінку з ескізом. За допомогою контекстного меню скопіюйте адресу зображення. Вставте в документ URL ресурсу та поясніть його складові: мережевий протокол обміну даними, ім'я сервера, шлях та ім'я графічного файлу.
5. Збережіть файл із назвою Вправа 6. Завершіть роботу.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107298).

§ 7. ОБМІН ДАНИМИ В ІНТЕРНЕТІ

Доступ до інформаційних ресурсів мережі «Інтернет» неможливий без організації швидкого та якісного передавання даних.

Передавання даних в інтернеті

В інтернеті застосовують пакетний принцип обміну повідомленнями. Спеціальні програми поділяють повідомлення на частини — **пакети даних**, долучають адреси відправника й одержувача, порядковий номер пакета й послідовно передають мережею.

Кожен пакет даних передається одним із мільйонів маршрутів. Якщо деякі мережеві комп'ютери й канали зв'язку вийдуть із ладу, то пакети дійдуть до одержувача іншими маршрутами.

Далі пакети даних збирає комп'ютер-отримувач, і початкові дані відновлюються. Завдяки тому що пакети даних пронумеровано, це легко зробити, навіть якщо вони дістались адресата в різний час і за різними маршрутами.

Якщо в якомусь із пакетів виявлено помилку, комп'ютер-отримувач надсилає комп'ютеру-відправнику запит на повторне передавання саме цього пакета даних. Розгляньмо рис. 7.1.

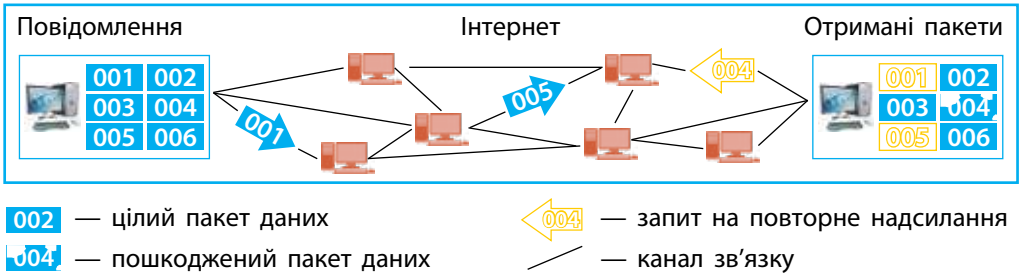


Рис. 7.1

Протоколи передавання даних

Для узгодженого передавання та приймання даних різних типів у мережах існують мережеві протоколи обміну даними.

Мережевий протокол — це сукупність правил (стандартів) для обміну даними між комп'ютерами.

Розгляньмо деякі найвідоміші протоколи та їхнє призначення:

Назва протоколу	Призначення протоколу
TCP	Керування передаванням і цілісністю пакетів даних
IP	Адресація та маршрутизація пакетів даних
Ethernet	Стандарт дротових локальних мереж
Wi-Fi	Стандарт бездротових мереж
DNS	Перетворення доменних імен на IP-адреси
FTP	Обмін файлами між комп'ютерами
HTTP	Передавання гіпертексту
POP/SMTP	Отримання/надсилання поштових повідомлень

Розробляють і впроваджують інтернет-протоколи і стандарти для вебтехнологій міжнародні організації Internet Engineering Task Force (IETF) та World Wide Web Consortium (W3C).

Провайдери

Послугами інтернету ми можемо користуватися цілодобово завдяки роботі провайдерів (від англ. *provider* — постачальник).

Провайдер (інтернет-провайдер) — це організація, яка надає послуги, пов'язані з доступом до глобальної мережі «Інтернет».

Сервер, який належить провайдеру, — це «точка доступу» до інтернету. Із ним з'єднуються комп'ютери або локальні мережі. Основні функції провайдерів такі: забезпечення доступу до інтернету, виділення дискового простору для сайтів, підтримка роботи поштових скриньок, передавання голосових повідомлень.

Щоб **почати роботу в інтернеті**, необхідно одержати доступ до сервера одного з провайдерів і встановити та налаштувати програми-клієнти служб, послуги яких потрібні.

Існують різні способи з'єднання з інтернетом, які відрізняються принципом роботи та швидкістю передавання даних: комутоване з'єднання, виділене з'єднання, технологія ADSL, технологія Wi-Fi, супутниковий зв'язок, мобільний інтернет. З особливостями способів з'єднання ви можете ознайомитися за QR-кодом або посиланням.

rnk.com.ua/108101

Популярним є з'єднання за допомогою мобільного пристрою: телефона чи модема. Такий доступ обмежений територією, на якій провайдер створив покриття. Існує кілька поколінь мобільного зв'язку (рис. 7.2), найпоширенішими з яких нині є 3G і 4G.

Рис. 7.2

Від 2018 року впроваджується мобільний зв'язок за стандартом 5G. Серед країн, що лідирують у сфері розвитку технологій 5G, — Китай, США, Південна Корея, Японія, Швеція та ін.

Швидкість обміну даними в мережі 5G в десятки разів перевищує швидкість мереж 4G. Стандарти, що розробляються для 6G, у тисячі разів перевищуватимуть показники 5G. Так, завантаження мережею 6G 100-годинного відеоматеріалу найвищої якості триватиме менше 1 с, на відміну від кількох годин у 4G. Очікується, що 6G буде впроваджено в період 2026–2030 років.

Питання для самоперевірки

1. Опишіть пакетний принцип передавання даних.
2. Що таке мережевий протокол?
3. Назвіть кілька мережевих протоколів та опишіть, яке їхнє призначення.
4. Назвіть основні функції провайдера.
5. Яка основна перевага кожного нового покоління технологій мобільного зв'язку порівняно з попереднім?
6. Технологією якого покоління мобільного зв'язку ви користуєтеся? Які її переваги?

Вправа 7

Визначити параметри з'єднання з інтернетом: спосіб з'єднання, провайдера, тип наданої провайдером IP-адреси (постійна чи тимчасова) та швидкість обміну даними.

1. Запустіть текстовий процесор і створіть новий документ.
2. Визначте спосіб приєднання вашої локальної мережі до інтернету та запишіть у створений документ.
3. Відкрийте один із сайтів для визначення параметрів з'єднання з інтернетом за вказівкою вчителя / вчительки. З'ясуйте назву вашого провайдера й запишіть у документ.
4. Визначте IP-адресу вашого комп'ютера, запишіть у документ. Порівняйте щойно визначену IP-адресу з адресою, яку було визначено на цьому комп'ютері у вправі 6.
   Зробіть висновок, чи змінилася надана провайдером IP-адреса, чи є вона постійною, і запишіть результат.
5. Визначте швидкість приймання та передавання даних в інтернеті вашим комп'ютером та запишіть у документ.
6. Збережіть файл із назвою Вправа 7. Завершіть роботу.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107297).

§ 8. СТВОРЕННЯ САЙТІВ

Розвиток інтернету й електронних технологій впливає на всі аспекти нашого життя. Нині кожна людина, що користується інтернетом, може створити власний сайт.

Вебсторінки і сайти

Щоб зрозуміти структуру головного сервісу інтернету — Всесвітньої павутини (англ. *World Wide Web*, або *WWW*), з'ясуймо значення понять сайт і гіпертекст.

Сайт, або вебсайт (від англ. *web* — мережа, *site* — місце), — це сукупність тематично пов'язаних вебсторінок і документів під одним доменним іменем.

Гіпертекст — це текст, що містить гіперпосилання, тобто зв'язки з іншими частинами документа або іншими документами чи файлами.

Термін «гіпертекст» увів у інформатику в 1963 році американський учений Теодор Нельсон, якого називають піонером інформаційних технологій. Вказівник, наведений у гіпертексті на гіперпосилання, набуває вигляду руки (рис. 8.1).

Рис. 8.1

Гіпертекст є основою вебсторінки, а ті у свою чергу об'єднуються в сайти; сукупність сайтів утворює Всесвітню павутину.

Розміщення сайтів у мережі

Інформацію із сайту можна отримувати в будь-який час із будь-якого комп'ютера, з'єднаного з інтернетом.

Де ж розташовуються сайти? Виявляється, у світі цілодобово працюють мільйони комп'ютерів — серверів, постійно підключених до інтернету, які надають хостинг. Саме на цих серверах і зберігаються папки зі вмістом більшості сайтів.

Хостинг (англ. *hosting*), — послуга з надання дискового простору сервера для розміщення, функціонування та обслуговування сайтів, файлових сховищ, електронної пошти тощо.

Хостингові компанії можуть забезпечувати реєстрацію доменного імені, обслуговування системи керування вмістом (див. далі), надання додаткового програмного забезпечення тощо. Хостинг може бути *платним* або *безплатним*.

Етапи створення сайтів

Створення сайту — це трудомісткий і тривалий процес, який умовно можна розділити на декілька основних етапів (рис. 8.2).

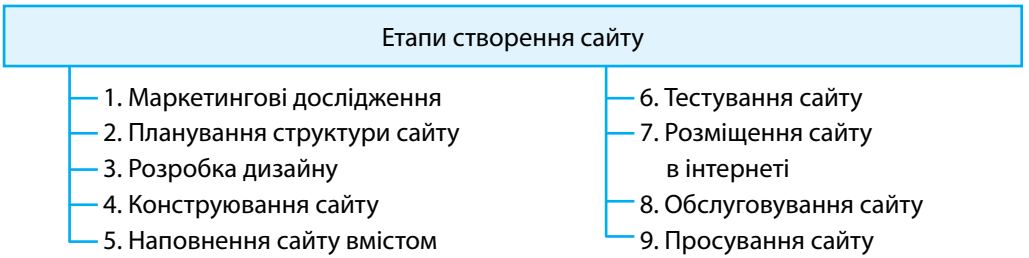


Рис. 8.2

Розгляньмо етапи розробки сайту докладніше.

▶ 1. Маркетингові дослідження

На цьому етапі з'ясовують призначення сайту та мету його створення, вивчають запити майбутніх відвідувачів, ознайомлюються з подібними сайтами і визначають оригінальні риси майбутнього сайту. Це впливає на спосіб його подальшого конструювання та наповнення.

▶ 2. Планування структури сайту

Продумана структура сайту спрощує його розробку та подальше користування й обслуговування. Слід обміркувати, кількість і розташування сторінок, меню, посилання, інші елементи, як відвідувачі / відвідувачки потраплятимуть на сайт, знаходитимуть сторінку з потрібною інформацією тощо. Вдала структура сайту спонукає відвідувачів повернутися на нього знову.

Для сайту з невеликою кількістю сторінок доцільно використовувати *ієрархічну структуру* (рис. 8.3), в якій перехід на потрібну сторінку можна здійснити за два-три клацання.

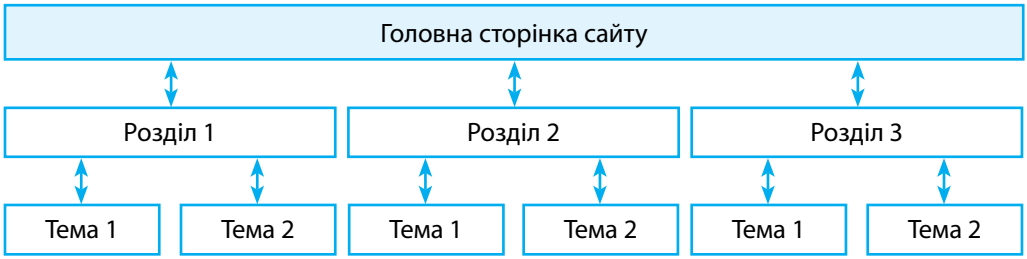


Рис. 8.3

Лінійну структуру варто використовувати для сайтів, на яких перегляд сторінок відбуватиметься послідовно (рис. 8.4).

Для більшості сайтів застосовують *мішану структуру*, яка поєднує в собі особливості ієрархічної та лінійної структур.

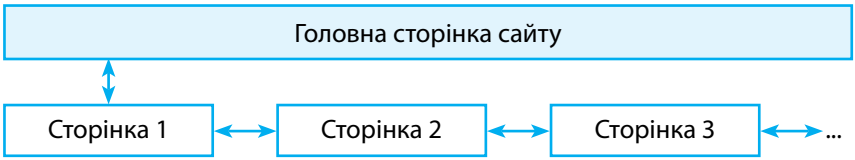


Рис. 8.4

▶ 3. Розробка дизайну

Вебсторінки мають відповідати основним правилам розробки сайтів і критеріям ергономічного розміщення матеріалів. Із критеріями ви можете ознайомитися за QR-кодом або посиланням.

rnk.com.ua/108159

▶ 4. Конструювання сайту

Сайт можна створити, використовуючи мову гіпертекстової розмітки (HTML, див. далі), мови опису стилів (CSS) і взаємодії з користувачем (JavaScript). Або скористатися системою керування вмістом — СКВ (англ. *Content Management System*, CMS). Слід подбати, щоб вебсторінки однаково виглядали в усіх браузерах.

▶ 5. Наповнення сайту вмістом

Для наповнення сайту варто заздалегідь зібрати матеріали (тексти, зображення тощо), які становитимуть вміст (контент). Пам'ятайте, на кого орієнтовано сайт (цільову аудиторію).

▶ 6. Тестування сайту

Створений і повністю або частково наповнений вмістом сайт доцільно випробувати на локальному сервері, тобто на комп'ютері користувача. Для створення локального сервера можна використати безплатні програми, такі як Apache, PHP, MySQL, phpMyAdmin, або програму XAMPP, яка їх об'єднує.

▶ 7. Розміщення сайту в інтернеті

Для доступу користувачів і користувачок інтернету сайт повинен мати доменне ім'я. Зазвичай послугу реєстрації доменних імен пропонують надавачі хостингу. Вони також забезпечують цілодобовий доступ до сайту.

! За наявності постійної IP-адреси можна розмістити сайт і на власному комп'ютері-сервері, який має бути постійно ввімкненим для надання доступу до сторінок сайту.

▶ 8. Обслуговування сайту

Обслуговування сайту полягає в періодичному оновленні матеріалів, супроводі (адмініструванні), удосконаленні навігації.

▶ 9. Просування сайту

Просування сайту полягає в його рекламуванні, оптимізації структури та змісту, організації опитувань і підтримки користувачів / користувачок тощо з метою збільшення відвідуваності.

Системи керування вмістом

Опанування мови гіпертекстової розмітки та вебпрограмування потребує часу. Для вебмайстрів-початківців зручнішим є автоматизоване створення сайтів за допомогою онлайнових СКВ.

Прості вебсторінки зручно створювати в СКВ Google Sites. Для більшого проєкту варто скористатися однією з досконаліших СКВ, яка включає різні модулі, з яких конструюють сайт.

Існує багато СКВ, і кожна має певні особливості. Логотипи деяких популярних безкоштовних СКВ наведено на рис. 8.5.

Рис. 8.5

Якщо СКВ установлено на власному комп'ютері, то готовий сайт потрібно буде вивантажити на сервер надавача хостингу.

На багатьох хостингах пропонується СКВ, яку вже встановлено на сервері. У такому разі можна вибрати дизайн вебсторінок із колекції шаблонів, створених фахівцями, і налаштувати відповідно до власних потреб. Критерії ергономічного розміщення відомостей на вебсторінках можна знайти за QR-кодом і посиланням rnk.com.ua/108159, розміщеними на с. 49.

Питання для самоперевірки

1. Дайте означення гіпертексту.
2. Що таке сайт? Назвіть основні етапи створення сайту.
3. Що таке хостинг?
4. Опишіть особливості ієрархічної структури сайту; лінійної структури; мішаної структури.
5. Що таке система керування вмістом? Наведіть приклади.
6. З'ясуйте за допомогою чат-бота призначення Apache, PHP, MySQL, phpMyAdmin.

Вправа 8

Спланувати структуру власного сайту.

1. Запустіть текстовий процесор і створіть новий документ. Запишіть відомості про майбутній сайт: тематику, назву, призначення, мету створення, на кого буде зорієнтовано сайт.
2. Знайдіть в інтернеті сайти схожого призначення та визначте, чим ваш сайт відрізнятиметься від решти подібних.

3. Запустіть редактор карт знань і створіть карту знань Структура, на якій сплануйте структуру майбутнього сайту.
4. У графічному редакторі схематично зобразіть головну сторінку, використовуючи як зразок головні сторінки наявних сайтів схожого призначення. Або зробіть оригінальний ескіз.
5. Вставте ескіз головної сторінки сайту в документ.
6. Продумайте й опишіть вміст однієї з вебсторінок: меню, посилання, тексти тощо. Збережіть документ із назвою Вправа 8.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107299).

§ 9. РОЗМІЧАННЯ ТЕКСТУ ЗАСОБАМИ HTML

Текстові процесори мають інструменти для форматування тексту. А як відбувається форматування гіпертексту для відображення у вікні браузера?

Поняття про мову HTML

Щоб браузер належно відображав дані на вебсторінці, до її неформатованого тексту додають спеціальні позначки — **теги**. Вони та правила їх використання утворюють мову розмітки. Одну з мов розмітки, а саме HTML, ми й розглянемо далі.

HTML (англ. *HyperText Markup Language* — мова розмітки гіпертексту) — набір тегів та правила їх застосування для створення гіпертекстових документів. Як видно з назви, HTML не є мовою програмування.

Отже, вебсторінка сайту являє собою текстовий документ, розмічений тегами HTML для відображення у вікні браузера.

HTML-документ

Існують спеціалізовані HTML-редактори: Microsoft SharePoint Designer, Notepad++, Phoenix Code Editor, Sublime Text, та онлайн-редактори: Online HTML Editor, HTML Instant, Real-Time HTML Editor та інші. Проте оскільки HTML-документ (HTML-файл) є текстовим документом, то створити його можна й у будь-якому текстовому редакторі, з яким ви вмієте працювати.

Тег — це позначка (мітка), записана в кутових дужках <>, яку опрацьовує браузер.

На рис. 9.1 наведено приклад HTML-коду та його відображення у вікні браузера.

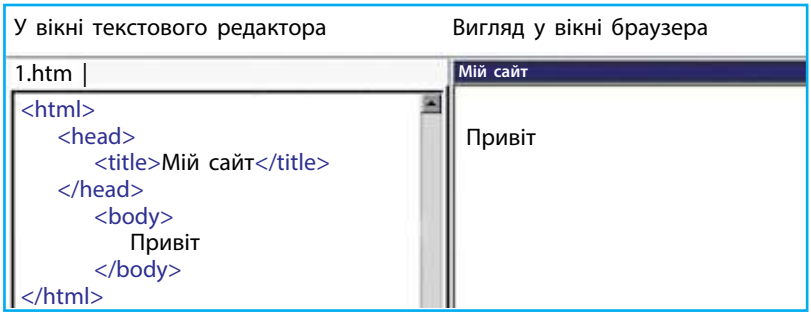


Рис. 9.1

Тут використано тільки теги, які визначають структуру HTML-документа:

<html> і </html>	Початок і кінець документа
<head> і </head>	Початок і кінець заголовка — службової частини коду
<title> і </title>	Початок і кінець назви, що виводитиметься на вкладці браузера (див. праву частину рис. 9.1)
<body> і </body>	Початок і кінець тіла документа, яке виводить браузер у робочому полі вікна

Форматування тексту

Текст із тегами в HTML-документі можна розташовувати довільно: розбивати на абзаци, робити відступи. Це не вплине на вигляд вебсторінки у вікні браузера, адже її вигляд залежить тільки від розмітки. Теги з помилками або такі, що не підтримуються браузером, буде знехтувано.

! Теги мови HTML визначають вигляд і функціонування об'єктів на вебсторінці сайту.

Теги бувають парними й непарними. **Парний тег** складається із відкривального тегу та закривального тегу, який починається символом «/».

... — парний тег жирного накреслення (рис. 9.2).

 або
 — непарний тег розриву рядка.

Фрагмент HTML-коду у вікні текстового редактора	Вигляд у вікні браузера
1.htm | <body> Інформатика – це наука </body>	Мій сайт **Інформатика –** це наука

Рис. 9.2

Розглянемо деякі теги форматування тексту:

<p>	Парний	Новий абзац із відступом
<h1>	Парний	Заголовок: <h1> — найвищий рівень, <h6> — найнижчий
<i>	Парний	Курсив

! Тег може містити атрибути, значення яких впливають на його дію. Атрибути записують у відкривальному тегу після назви, відокремлюючи один від одного пропусками.
Значення атрибутів, що містять пропуски, беруть в одинарні або подвійні лапки. Для решти значень це не обов'язково.

2 Застосуємо атрибути text і bgcolor (колір тексту і тла) (рис. 9.3):

Фрагмент HTML-коду у вікні текстового редактора

1.htm

```
<body text = "white" bgcolor = "green">
    Привіт!
</body>
</html>
```

Вигляд у вікні браузера

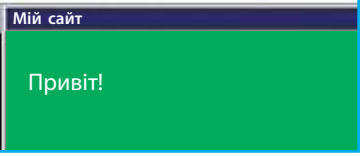


Рис. 9.3

Марковані та нумеровані списки

Для створення *маркованого списку* використовують тег … з атрибутом type, який задає вигляд маркера:

"disk" — ●; "circle" — ○; "square" — ■.

Для створення *нумерованого списку* використовують тег … з атрибутом type (тип нумерації: "1" — арабські цифри, "A" — латинські літери, "I" — римська нумерація тощо) та атрибутом start (задання початкового номера в списку з нумерацією арабськими цифрами).

Елементи списку розмічають тегами … (рис. 9.4).

Фрагмент HTML-коду у вікні текстового редактора

1.htm

```
<ul type = "square">
    <li>Іванова</li>
    <li>Бойко</li>
</ul>
<ol type = "I">
    <li>Інформатика</li>
    <li>Фізика</li>
</ol>
```

Вигляд у вікні браузера

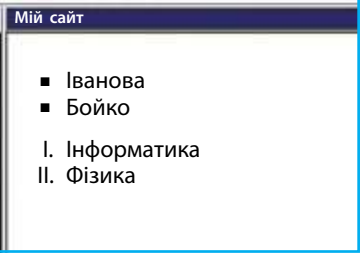


Рис. 9.4

Ідею гіпертексту вперше описав 1941 року аргентинський письменник Хорхе Луїс Борхес [3], а термін «гіпертекст» увів 1965 року американський учений Тед Нельсон [4].

Питання для самоперевірки

1. Що таке HTML?
2. Що таке тег? Наведіть приклади парних і непарних тегів.
3. Назвіть складові структури HTML-документа.
4. Що таке атрибут тегу?
5. Якими засобами HTML можна змінити колір тла вебсторінки сайту?
6. Які теги використовують для створення списків?

Вправа 9

Створити HTML-документ із відформатованим текстом.

1. Запустіть текстовий редактор Блокнот і введіть у документ набір тегів структури HTML-документа (див. рис. 9.1).
2. Збережіть створений документ як HTML-файл: Файл → → Зберегти як → Тип файлу (всі файли) → Ім'я: vprava9.html. У списку Кодування виберіть UTF-8.
3. Відкрийте папку зі збереженим файлом і за допомогою браузера відкрийте HTML-файл vprava9.
4. Змініть HTML-код так, щоб нумерований список відображався на синьому тлі; колір шрифту — жовтий; накреслення — жирний (рис. 9.5).
5. Збережіть файл vprava9.html і оновіть сторінку браузера.
   За потреби відкоригуйте HTML-код відповідно до п. 4.

2. Вівторок
3. Середа
4. Четвер

Рис. 9.5

6. Доповніть код сторінки так, щоб унизу з'явився список, який містить усі дні тижня. Кольори доберіть на свій розсуд. Обговоріть роботу з однокласником / однокласницею.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107300).

§ 10. ГРАФІЧНІ ЗОБРАЖЕННЯ ТА ГІПЕРПОСИЛАННЯ НА ВЕБСТОРІНЦІ

Сайт стає цікавішим і наочнішим, якщо його вебсторінки доповнити зображеннями. А пов'язати вебсторінки сайту між собою та з іншими сайтами допоможуть гіперпосилання.

Вставляння зображень

На вебсторінку можна додавати різноманітні зображення. Це можуть бути ілюстрації до тексту, графічні елементи оформлення.

Щоб **вставити на вебсторінку зображення з файлу**, використовують тег <img src = "my.jpg">, де img — назва тегу, src = "my.jpg" — атрибут, у якому вказують ім'я файлу із зображенням.

В атрибуті можна використовувати імена графічних файлів із розширеннями .jpg, .jpeg, .gif, .png або .bmp. Якщо зображення і HTML-файл містяться в різних папках, то, крім імені файлу, потрібно вказати шлях до нього із зазначенням назв папок.

Якщо зображення міститься у вкладеній папці images:
<img src = "images/my.jpg">.
Якщо зображення міститься у зовнішній папці images:
<img src = "../images/my.jpg">.

Тег img підтримує атрибути, які визначають розташування зображення на вебсторінці та обтікання його текстом. Приклад їх використання наведено на рис. 10.1.

Рис. 10.1

Проаналізуймо декілька тегів для вставляння зображення:

- зображення з лівого краю, обтікання текстом праворуч:
  <img src = "pr1.png" align = "left">
- відстань між текстом і зображенням по вертикалі 5 пікселів, по горизонталі — 30 пікселів (див. рис. 10.1):
  <img src = "pr1.png" vspace = "5" hspace = "30">
- опис зображення:
  <img src = "pr1.png" alt = "Моє фото">

  Примітка. Якщо навести вказівник на зображення, з'явиться текст — Моє фото. Крім того, значення параметра alt буде виведене на екран замість зображення, якщо в браузері вимкнуто показ зображень.
- ширина зображення 100 пікселів, висота — 20 % від висоти робочої області вікна браузера:
  <img src = "pr1.png" width = "100" height = "20%">

  Примітка. Одночасне задання значень width і height може порушити пропорції зображення.
- товщина рамки навколо зображення — 5 пікселів:
  <img src = "pr1.png" border = "5">

Додавання гіперпосилань

Сайт часто складається з багатьох вебсторінок, пов'язаних гіперпосиланнями. Для створення гіперпосилань на вебсторінки сайту та зовнішні ресурси інтернету використовують тег <a>....

Розгляньмо приклад додавання гіперпосилань до зображень і фрагментів тексту.

Нехай в одній папці з файлом index.html створено файл prf.html, який демонструє вашу фотографію. У файлі index.html до фрази Подивитися фото можна додати гіперпосилання, скориставшись такою розміткою:
 Подивитися фото

Щоб пов'язати гіперпосилання із зображенням, замість тексту вказують тег вставляння зображення:
<img src = "foto1.jpg">

HTML-код головної вебсторінки може мати такий вигляд:

```
<html>
    <body>
        <a href = "first.html">Перша сторінка</a><br>
        <a href = "second.html">Друга сторінка</a><br>
        <a href = "third.html">Третя сторінка</a><br>
    </body>
</html>
```

Після клацання напису Перша сторінка відкривається вебсторінка з файлу first.html, після клацання напису Друга сторінка — з файлу second.html, а напису Третя сторінка — third.html.

Для повернення на головну сторінку на інших вебсторінках сайту мають бути відповідні гіперпосилання. Наприклад:

```
<a href = "index.html">Головна</a>
```

Питання для самоперевірки

1. Як вставити на вебсторінку зображення?
2. У чому відмінність між значеннями атрибутів для вставлення малюнка з поточної, внутрішньої і зовнішньої папок?
3. Як вирівняти зображення за правим краєм вікна браузера?
4. Як задати розмір малюнка? Що означає розмір у відсотках?
5. Як створити гіперпосилання?
6. Як додати до малюнка гіперпосилання на іншу вебсторінку?

Вправа 10

Створити вебсторінку сайту з малюнками і гіперпосиланнями.

1. Створіть папку з назвою Вправа 10. Знайдіть в інтернеті малюнок шкільної тематики та збережіть у створеній папці.
2. Створіть HTML-документ (головну сторінку сайту) із заголовком Розклад уроків. Додайте малюнок із папки та текст (меню сайту): понеділок, вівторок, середа, четвер, п'ятниця. Збережіть створений документ з іменем index.html у папці.
3. Створіть п'ять HTML-документів (сторінок вебсайту) із розкладом уроків на кожний із цих днів тижня. До кожної сторінки додайте вгорі її назву та внизу слово Головна.

Слово Головна буде пов'язано гіперпосиланням із головною сторінкою. Збережіть сторінки сайту в папці з іменами 1.html; 2.html; 3.html; 4.html; 5.html.

4. У файлі index.html зробіть гіперпосилання на п'ять створених сторінок, а у файлах 1.html–5.html — на головну сторінку.
5. Відформатуйте тексти та графічні зображення.
6. Перевірте роботу сайту. Обговоріть роботу з однокласником / однокласницею.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107301).

§ 11. РОЗМІЧАННЯ ТАБЛИЦЬ ЗАСОБАМИ HTML

Відомості на вебсторінці сайту можна подати як таблицю.

Теги розмітки таблиці

Ознайоммося з основними тегами розмітки таблиці:

Тег	Призначення
<table>…</table>	Обмежують таблицю
<tr>…</tr>	Обмежують рядок таблиці
<td>…</td>	Обмежують клітинку таблиці
<th>…</th>	Використовують замість тегу <td>…</td> для виокремлення заголовків стовпців таблиці

Розгляньмо, як розмічати таблицю тегами (рис. 11.1):

Фрагмент HTML-коду

1.htm

```
<table border = 2>
<tr>
    <td>1-й рядок <br> 1-й стовпець</td>
    <td>1-й рядок <br> 2-й стовпець</td>
    <td>1-й рядок <br> 3-й стовпець</td>
</tr>
<tr>
    <td>2-й рядок <br> 1-й стовпець</td>
    <td>2-й рядок <br> 2-й стовпець</td>
    <td>2-й рядок <br> 3-й стовпець</td>
</tr>
</table>
```

Вигляд у вікні браузера

Мій сайт

1-й рядок 1-й стовпець	1-й рядок 2-й стовпець	1-й рядок 3-й стовпець
2-й рядок 1-й стовпець	2-й рядок 2-й стовпець	2-й рядок 3-й стовпець

Рис. 11.1

За допомогою атрибутів у тегах розмітки таблиці змінюють вигляд таблиці, клітинок і написів у них:

Атрибут	Опис
align	Вирівнювання таблиці у вікні браузера
background	Малюнок на тлі
bgcolor	Колір тла
border	Товщина рамки в пікселях
bordercolor	Колір рамки
cellspacing	Відстань між клітинками таблиці
cellpadding	Відстань між текстом клітинки та її межею
width	Ширина таблиці (клітинки) у відсотках або пікселях
height	Висота таблиці (клітинки) у відсотках або пікселях

1989 року британський учений Тім Бернерс-Лі представив інформаційну систему на основі гіпертексту [5]. А 1993 року він та Деніел Коннолі опублікували першу версію HTML [6].

Оскільки ширина клітинок одного стовпця не може бути різною в різних рядках, то її достатньо вказати лише для одного рядка.

<table width = "100%"> — ширина таблиці дорівнює ширині робочого поля вікна браузера; <td width = "50%"> — ширина клітинки дорівнює половині ширини таблиці.

Якщо в різних рядках таблиці все ж задати для одного стовпця різну ширину, то використано буде більше зі значень. Значеннями, які суперечать одне одному, буде знехтувано.

Відступи

Для налаштування відстаней між клітинками, а також між текстом і межами використовують атрибути відступів (рис. 11.2):

```
<table cellspacing = "10" cellpadding = "10"
        bgcolor = "blue">
    <tr bgcolor = "white">
        <td width = "100">1</td>
        <td width = "100">2</td>
    </tr>
</table>
```

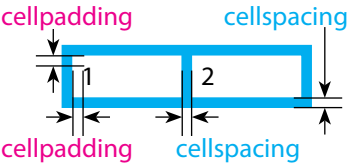


Рис. 11.2

Форматування вмісту клітинок

Теги рядків (<tr>) і клітинок (<td>) можуть включати атрибути для змінення вирівнювання тексту відносно меж клітинки.

Атрибут align задає горизонтальне вирівнювання:

align = "left" — за лівим краєм;

align = "center" — по центру клітинки;

align = "right" — за правим краєм.

Атрибут valign задає вертикальне вирівнювання:

valign = "middle" — посередині клітинки;

valign = "top" — вгорі клітинки;

valign = "bottom" — внизу клітинки.

На рис. 11.3 показано, як, комбінуючи значення атрибутів align та valign, отримують різні варіанти вирівнювання тексту в клітинках таблиці.

Фрагмент HTML-коду

1.htm

```
<table border = 2 widht = 100%>
<tr height = 100 align = center>
    <td valign = top>    1-й рядок <br> 1-й стовпець</td>
    <td valign = middle>1-й рядок <br> 2-й стовпець</td>
    <td valign = botton>1-й рядок <br> 3-й стовпець</td>
</tr>
<tr>
    <td align = left>   2-й рядок <br> 1-й стовпець</td>
    <td align = center>2-й рядок <br> 2-й стовпець</td>
    <td align = right>  2-й рядок <br> 3-й стовпець</td>
</tr>
</table>
```

Вигляд у вікні браузера

Мій сайт

1-й рядок 1-й стовпець	1-й рядок 2-й стовпець	1-й рядок 3-й стовпець
2-й рядок 1-й стовпець	2-й рядок 2-й стовпець	2-й рядок 3-й стовпець

Рис. 11.3

Іноді при розмічанні таблиці виникає потреба об'єднати кілька клітинок. Для цього використовують атрибути colspan (злиття клітинок у рядку) (рис. 11.4, *а*) і rowspan (злиття клітинок у стовпці) (рис. 11.4, *б*). Комбінуючи ці атрибути, можна створювати на вебсторінках таблиці як завгодно складної структури.

Фрагмент HTML-коду — У вікні браузера

1.htm

```
<table border = "1">
<tr>
    <td colspan = "2">Привіт!</td>
</tr>
<tr>
    <td>Петре,</td>
    <td>Марійко!</td>
</tr>
</table>
```

Мій сайт

Привіт!	
Петре,	Марійко!

а

Фрагмент HTML-коду — У вікні браузера

1.htm

```
<table border = "1">
<tr>
   <td rowspan = "2">Привіт!</td>
   <td>Петре,</td>
</tr>
<tr>
   <td>Марійко!</td>
</tr>
</table>
```

Мій сайт

Привіт!	Петре,
	Марійко!

б

Рис. 11.4

За допомогою HTML-таблиці можна створити макет вебсторінки, об'єднавши певні клітинки й розмістивши в них текст та об'єкти. Зараз частіше застосовують теги структурування документа і **CSS** (від англ. *Cascading Style Sheets*) — каскадні таблиці стилів. Більше про теги можна дізнатися з другого форзаца та за QR-кодом або посиланням.

rnk.com.ua/108160

Питання для самоперевірки

1. Як додати на вебсторінку таблицю?
2. Як вирівняти текст у клітинці таблиці по горизонталі?
3. Як вирівняти текст у клітинці таблиці по вертикалі?
4. Як установити товщину меж таблиці?
5. Як установити відступ від тексту в клітинці до її меж?
6. Як об'єднати клітинки одного рядка або стовпця таблиці?

Вправа 11

Створити таблицю за зразком (рис. 11.5).

1. Відкрийте текстовий редактор та створіть HTML-документ для відображення таблиці (рис. 11.6). Збережіть документ з іменем vprava11.html.
2. Змініть HTML-код, об'єднавши клітинки за зразком, і заповніть клітинки, як показано на рис. 11.7.
3. Задайте розміри таблиці в пікселях: ширина — 400, висота — 200.
4. Налаштуйте параметри клітинок у відсотках від розміру таблиці: висота верхнього рядка — 35%, ширина другого і третього стовпців — 60 і 20% відповідно.
5. Вирівняйте по вертикалі та горизонталі написи в клітинках (див. рис. 11.5).
6. Збережіть документ. Перегляньте таблицю в браузері. За необхідності відформатуйте клітинки таблиці й написи в них.

1111	2222	
	3333	4444

Рис. 11.5

1111	2222	3333
4444	5555	6666

Рис. 11.6

1111	2222	
	3333	4444

Рис. 11.7

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107302).

ПРАКТИЧНА РОБОТА 3
СТВОРЕННЯ ГОЛОВНОЇ СТОРІНКИ САЙТУ ЗАСОБАМИ HTML

Завдання: створити головну сторінку сайту засобами HTML за правилами ергономічного розміщення матеріалів.

Обладнання: комп'ютер, з'єднаний з інтернетом, комп'ютерний або онлайновий HTML-редактор.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

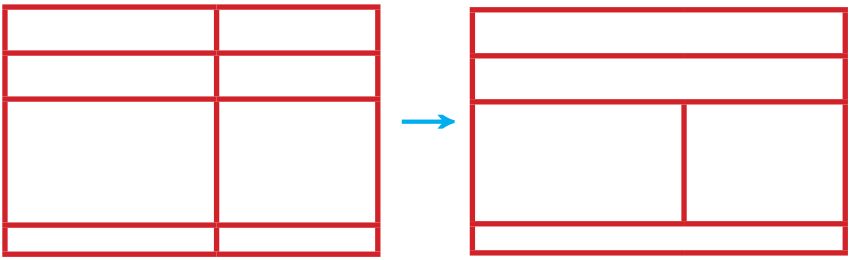
1. Придумайте тему свого сайту, продумайте вміст і розташування елементів на головній сторінці сайту.
2. Знайдіть в інтернеті (або створіть самостійно) матеріал (тексти, рисунки, посилання тощо) і збережіть у папці Сайт.
3. Розробіть ескіз головної сторінки сайту в текстовому чи графічному редакторі та створіть макет (рис. 1, *а*).
   Накресліть таблицю навколо елементів сторінки (рис. 1, *б*).

а б

Рис. 1

4. Запустіть на комп'ютері один із HTML-редакторів (можливо, онлайновий) за вказівкою вчителя / вчительки.

5. Створіть HTML-код вебсторінки із порожньою таблицею, доповніть кодом таблиці з потрібною кількістю рядків і стовпців (див. рис. 1, *б*), вказавши відносні розміри таблиці у відсотках.
   Збережіть HTML-файл з іменем index.html у папці Сайт.
6. Об'єднайте, де це потрібно, рядки і стовпці (рис. 2, *а*), щоб загальний вигляд таблиці відповідав запланованому розташуванню елементів на головній сторінці сайту (рис. 2, *б*).



а б

Рис. 2

7. Установіть розміри (відносні та / чи абсолютні) для клітинок таблиці згідно із макетом (див. рис. 2).
8. Додайте до таблиці написи й рисунки, меню, кнопки тощо відповідно до ескізу головної сторінки (див. рис. 1).
9. Відформатуйте елементи головної сторінки (колір, розмір, вирівнювання тощо) відповідно до критеріїв ергономіки.
10. Додайте до меню (і / або кнопок) необхідні гіперпосилання на інтернет-ресурси та перевірте їх роботу.
11. Вимкніть відображення меж таблиці. Збережіть HTML-файл. Перегляньте сторінку сайту в кількох браузерах, із різним масштабуванням. Перевірте роботу меню (і / або кнопок).
12. Надішліть HTML-файл учителеві / вчительці або збережіть файл на хмарному диску, надавши до нього доступ.

Зробіть висновок: у чому переваги та недоліки (труднощі) використання таблиць для макетування вебсторінки.

РОЗДІЛ 3
ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ

§ 12. Основні етапи створення текстового документа

§ 13. Робота з фрагментами тексту

§ 14. Форматування об'єктів текстового документа

Практична робота 4. Створення текстового документа, що містить об'єкти різних типів

§ 15. Структура документа. Колонтитули

§ 16. Автоматизоване створення змісту документа

Практична робота 5. Автоматизоване створення змісту документа

§ 17. Гіперпосилання в текстових документах. Спільна робота з документом

Практична робота 6. Створення документа на хмарному диску для спільного опрацювання

Практична робота 7. Створення мінігазети на хмарному диску

ПОВТОРЮЄМО

Ви вже маєте навички роботи з документом у середовищі текстового процесора MS Word. Знаєте, що документ може містити різні текстові об'єкти, основними з яких є *символ* і *абзац*. Властивостями символу є *шрифт*, *накреслення*, *розмір* і *колір*, а властивостями абзацу — *вирівнювання*, *міжрядковий інтервал*, *відступи* тощо.

До текстового документа можна додавати графічні зображення й таблиці. Властивостями зображення є його *розмір* і *спосіб розташування*, *стиль межі* тощо. Таблиця характеризується кількістю рядків і стовпців, параметрами меж і заповнення клітинок тощо. Значення властивостей об'єктів можна змінювати.

Ви також умієте виконувати операції над основними текстовими об'єктами документа.

1. Із яким текстовим процесором ви вже працювали?
2. Назвіть відомі вам об'єкти текстового документа.
3. Які властивості має символ?
4. Які властивості має абзац?
5. Назвіть властивості зображення.
6. Які властивості має таблиця?

У цьому розділі ви ознайомитеся з такими об'єктами текстового документа, як список, формула, недруковані символи, посилання, колонтитули, зміст тощо. А також дізнаєтеся про використання стильового оформлення документів і спільну роботу з ними.

§ 12. ОСНОВНІ ЕТАПИ СТВОРЕННЯ ТЕКСТОВОГО ДОКУМЕНТА

Текстовий редактор — це програмний засіб для створення та опрацювання текстових документів. Вибір текстового редактора залежить від особливостей опрацьовуваних документів.

Текстові редактори

Серед текстових редакторів користувач може вибрати той, що має набір функцій, потрібний для досягнення мети. Наприклад, до стандартного пакунка програм операційної системи Windows входять текстовий редактор Блокнот, призначений для створення та редагування простих текстових документів. Для програмістів корисним є безплатний текстовий редактор Notepad++.

Існує низка онлайнових текстових редакторів, які надають швидкий доступ до документів, їх редагування і перегляду. Google Документи дає змогу створювати нові документи, редагувати наявні, зберігати їх у хмарному сховищі Google Диск. Його інтерфейс схожий на Word. Основною перевагою Google Документів є можливість працювати з документом без доступу до інтернету.

Онлайн-версія Microsoft Word (у складі пакета Microsoft 365) має більше функцій для форматування, створення таблиць, графіків тощо. Його інтерфейс знайомий тим, хто користувався пакетом Microsoft Office. Докладніше про це — далі.

Текстовий процесор Microsoft Word

У 5 і 6 класах ви ознайомилися з текстовим редактором Microsoft Word (далі — Word) (рис. 12.1). Він надає можливість здійснювати надзвичайно багато процесів опрацювання об'єктів текстового документа. Тому його називають **текстовим процесором**.

Рис. 12.1

Щоб **запустити** Word, потрібно вибрати команду: Пуск → Всі програми → Microsoft Office → Microsoft Word.

Після запуску текстового редактора на екрані монітора з'являється перелік останніх документів і вікно онлайн-пошуку шаблонів. Можна знайти потрібний документ і відкрити шаблон.

Першу версію Word для Windows випущено в 1989 році, а всі наступні тільки додавали функцій. Остання версія — Word 2021. Ми будемо орієнтуватися на версію Word 2016.

Створення і збереження документа

Пригадаймо, що для створення нового документа потрібно вибрати зображення із написом Новий документ. Після цього відкриється вікно. Чи знайомі вам його об'єкти (рис. 12.2)?

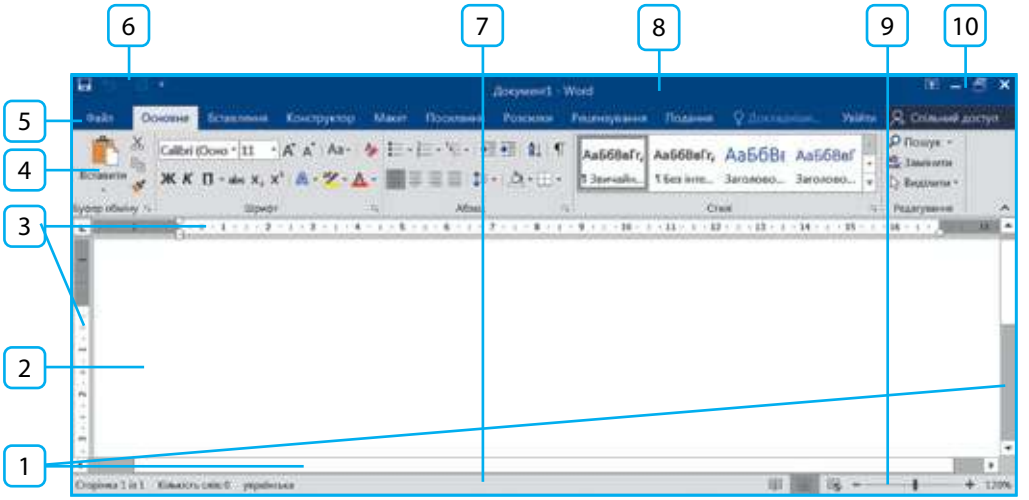


Рис. 12.2

1 — смуги прокручування
2 — сторінка документа
3 — лінійки
4 — стрічка інструментів
5 — вкладки головного меню
6 — панель швидкого доступу
7 — рядок стану
8 — рядок заголовка
9 — інструмент масштабування
10 — кнопки керування вікном

Як і інші офісні документи (презентації, таблиці тощо), текстовий документ для подальшого використання треба зберегти.

Щоб **зберегти текстовий документ**, слід виконати такі дії:

1) на вкладці ФАЙЛ вибрати команду Зберегти (або Зберегти як);
2) зазначити місце збереження в одну з наявних у списку папок або вибрати іншу папку, скориставшись кнопкою

;
3) увести назву файлу в поле Ім'я файлу і натиснути кнопку Зберегти (для збереження наявного документа в іншу папку і/або з іншим іменем слід вибрати команду Зберегти як…).

Щоб **повернутися в режим редагування** із будь-якого кроку алгоритму збереження документа, слід натиснути кнопку .

Набір тексту та перевірка правопису

Уміння швидко та правильно набирати текст — важлива навичка, яка допомагає в багатьох аспектах життя, від навчання до професійної діяльності. Повторити правила набору тексту ви можете за QR-кодом або посиланням rnk.com.ua/108777.

Текст, уведений за всіма правилами, сприяє швидкій автоматичній перевірці правопису, полегшує читання та сприймання змісту. Низку команд у Word можна викликати за допомогою «гарячих» клавіш. З ними можна ознайомитися на першому форзаці, а також за QR-кодом або посиланням rnk.com.ua/108783.

Word оснащений засобом, який дозволяє перевіряти й виправляти написання слів у тексті. Підкреслення червоною хвилястою лінією зазвичай свідчить про помилку.

Щоб **увімкнути автоматичну перевірку правопису**, потрібно вибрати Файл → Параметри → Правопис → Автоматична перевірка правопису.

Виклик контекстного меню для слова з помилкою (клацання по ньому правою кнопкою миші) дає змогу вибрати один із варіантів автоматичного виправлення помилки, або знехтувати нею (Пропустити все), або зберегти як правильний варіант для подальших перевірок (Додати до словника).

Мову перевірки правопису можна змінити, якщо в рядку стану вибрати команду Мова, а в діалоговому вікні вибрати мову. У разі вимкненої автоматичної перевірки правопису його можна **перевірити вручну**. Для цього потрібно:

1) виділити фрагмент тексту;
2) натиснути кнопку Мова, що в рядку стану;
3) вибрати в діалоговому вікні Мова потрібну мову і натиснути OK;
4) на вкладці Рецензування в групі інструментів Правопис натиснути кнопку Правопис і граматика (рис. 12.3).

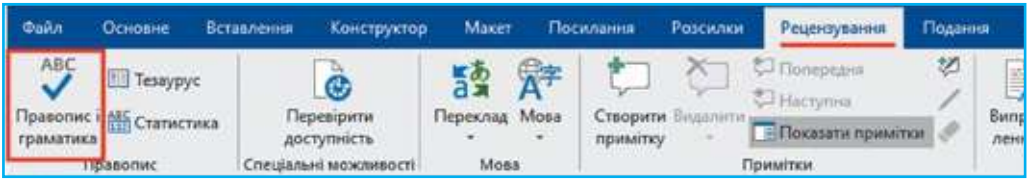


Рис. 12.3

- Одним із найстаріших текстових редакторів є ed (скор. від англ. *editor*), створений у 1970 році для операційної системи Unix. Він і досі використовується [7].
- Один із найпопулярніших текстових редакторів Microsoft Word був запущений у 1983 році і спочатку працював лише на комп'ютерах MS-DOS [8].

Питання для самоперевірки

1. Які види документів ви можете створювати в текстовому процесорі? Наведіть приклади.
2. Чому Word називають текстовим процесором?
3. Які клавіші швидкого доступу у Word ви знаєте, і як вони полегшують роботу з документами?
4. Доведіть необхідність дотримання правил набору тексту.
5. Як увімкнути перевірку правопису?
6. Які формати файлів можна використовувати для збереження текстового документа?

Вправа 12

Ввести в документ текст за правилами набору та виконати форматування об'єктів.

1. Запустіть текстовий процесор і створіть новий документ.
2. Установіть параметри сторінки та шрифту:
   - розмір сторінки — А4; поля — по 3 см;
   - шрифт — Arial; розмір шрифту — 11;
   - накреслення — курсив; колір шрифту — синій.
3. Установіть параметри абзацу:
   - міжрядковий інтервал — одинарний;
   - інтервали абзацу: Перед — 0 пт, Після — 10 пт.
4. Наберіть кілька рядків вірша Степана Чарнецького «Ой у лузі червона калина». Виправте орфографічні помилки. Які варіанти слова «підіймемо» пропонує текстовий процесор?
5. Установіть для приспіву: колір шрифту — червоний; накреслення — курсив, жирний.
6. Знайдіть в інтернеті зображення калини й скопіюйте в документ. За потреби змініть розмір зображення, розташуйте за зразком (рис. 12.4). Збережіть файл із назвою Вправа 12.

Ой у лузі червона калина похилилася,¶
Чогось наша славна Україна зажурилася.¶
А ми тую червону калину підіймемо,¶
А ми нашу славну Україну, гей-гей,
розveселимо!¶
Не хилися, червона калино,¶
Маєш білий цвіт.¶
Не журися, славна Україно,¶
Маєш добрий рід!¶

Рис. 12.4

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107750).

§ 13. РОБОТА З ФРАГМЕНТАМИ ТЕКСТУ

Будь-який текст зазвичай потребує подальшого опрацьовування. Редагування тексту — це процес змінення окремих його частин шляхом переміщення, форматування тощо.

Способи виділення фрагментів

Для роботи з фрагментом тексту його спочатку необхідно виділити. Це можна здійснити за допомогою клавіатури або миші.

Для **виділення за допомогою клавіатури всього тексту** можна скористатися сполученням клавіш Ctrl + A.

Щоб **виділити фрагмент тексту**, потрібно виконати такі дії:

1) установити текстовий курсор на початок або кінець фрагмента;
2) утримуючи натиснутою клавішу Shift, переміщувати курсор клавішами керування курсором (рис. 13.1).

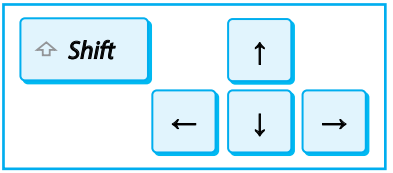


Рис. 13.1

Виділений фрагмент тексту підсвічується кольором (рис. 13.2). Щоб **зняти виділення**, потрібно натиснути будь-яку клавішу керування курсором (див. рис. 13.1).

Виділений і не виділений фрагменти тексту

Рис. 13.2

Для **виділення фрагмента за допомогою миші** слід установити вказівник на початок фрагмента і, утримуючи натиснутою ліву кнопку миші, протягнути вказівник до кінцевого положення. Щоб **зняти виділення фрагмента тексту**, слід натиснути будь-яке місце робочого поля (на сторінці документа).

Для **виділення кількох фрагментів тексту** описані дії мишею повторюють з утриманням натиснутою клавіші Ctrl. Зі швидкими способами виділення мишею ви можете ознайомитися за QR-кодом або посиланням rnk.com.ua/108778.

Пошук і заміна фрагментів

Функція Пошук та заміна здійснює пошук одного або декількох зразків тексту (зазвичай це символьний рядок) і замінює його відповідно до вимог користувача / користувачки.

До виділених фрагментів тексту можна застосовувати низку операцій, які виконуються за допомогою команд контекстного меню або сполучень клавіш. Наприклад, це може бути переміщення або вставлення, що передбачає копіювання в *буфер обміну* — ділянку пам'яті, призначену для тимчасового зберігання даних різних типів.

Знайти в тексті потрібні фрагменти (наприклад, повторювані) вручну буває важко, особливо у великих за обсягом документах. Текстовий процесор Word оснащений засобом для автоматичного пошуку об'єктів документа (це можуть бути фрагменти, рисунки, таблиці, формули та ін.).

Для **пошуку фрагмента тексту** потрібно виконати такі дії:

1) на вкладці Основне в групі інструментів Редагування вибрати команду Пошук (рис. 13.3) або натиснути сполучення клавіш Ctrl+F — відкриється бокова вкладка Навігація;

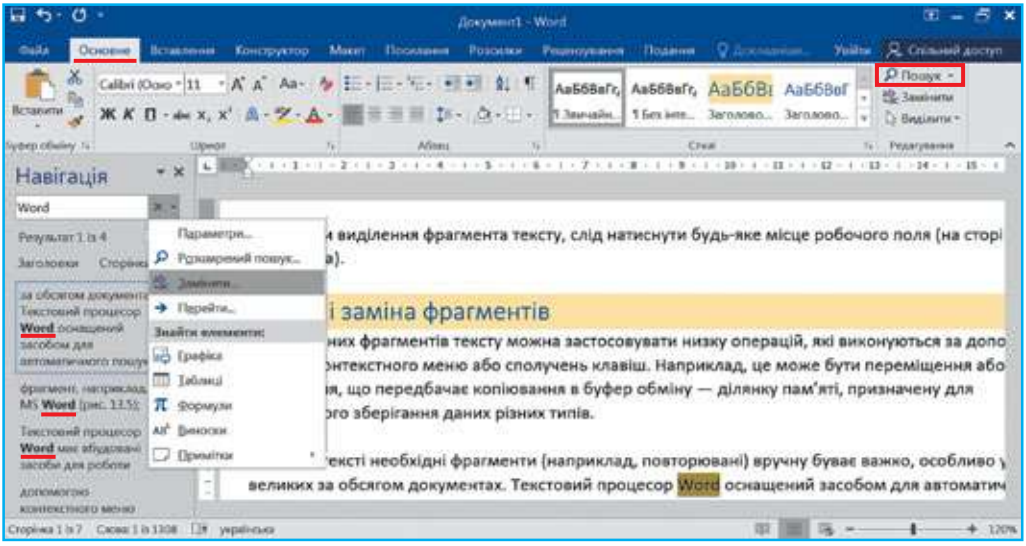


Рис. 13.3

2) у текстове поле пошуку ввести текст, який потрібно відшукати, — на вкладці Навігація з'являться уривки текстового документа зі знайденим фрагментом тексту;
3) для переходу до відповідного місця в документі клацнути вибраний уривок, який виділено жовтим тлом (див. рис. 13.3).

Якщо текстовий фрагмент перед пошуком виділити та скопіювати в буфер обміну (сполучення клавіш Ctrl+C), то пошук цього фрагмента можна здійснити за допомогою «гарячих» клавіш — сполучення Ctrl+F.

Досить часто буває потрібно не просто знайти фрагмент тексту в документі, а й замінити його на інший.

Щоб **замінити фрагмент**, потрібно виконати такі дії:
1) на боковій вкладці Навігація у списку, що розгортається, вибрати команду Замінити (рис. 13.4);
2) у вікно Пошук і замінювання, що відкриється, у полі Замінити на ввести текст, яким потрібно замінити знайдений текстовий фрагмент, наприклад, MS Word (рис. 13.5);
3) натиснути кнопку Замінити (або Замінити все).

За необхідності можна налаштувати додаткові параметри пошуку, які з'являються після розгортання вікна кнопкою Більше.

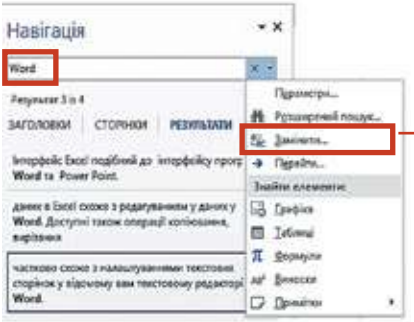


Рис. 13.4

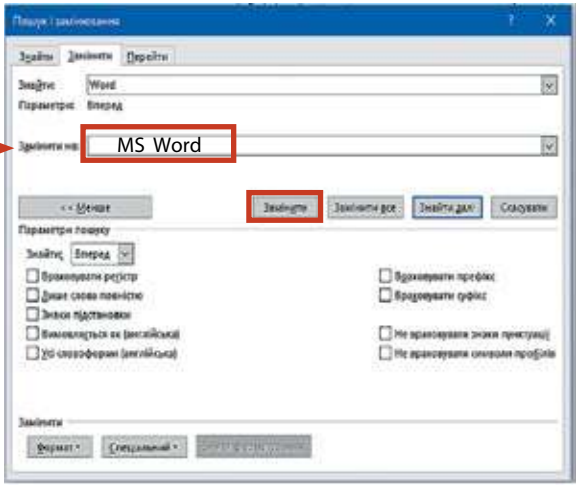


Рис. 13.5

Діалогове вікно Пошук і замінювання (див. рис. 13.5) можна відкрити також сполученням клавіш Ctrl+H.

! У документі можна здійснювати пошук тексту не лише за змістом, а й такого, що має певне форматування.

Для пошуку інших об'єктів: графічних зображень, таблиць, формул тощо слід вибрати відповідну команду зі списку, що розгортається біля текстового поля пошуку на вкладці Навігація.

i Незважаючи на розробку низки сучасних інтегрованих середовищ, багато фахівців надають перевагу текстовим редакторам завдяки їх швидкості та гнучкості.

Питання для самоперевірки

1. Як виділити в документі різні текстові фрагменти?
2. Які операції можна виконувати з виділеним фрагментом?
3. Чим відрізняються операції Вирізати і Видалити?
4. Як здійснити автоматичний пошук фрагмента тексту?
5. Опишіть, як у документі виконати автоматичний пошук графічних зображень, таблиць, списків тощо.
6. Як виконати автоматичну заміну в документі одного текстового фрагмента на інший?

Вправа 13

Виконати дії з фрагментами тексту.

1. Запустіть текстовий процесор і створіть новий документ.
2. Знайдіть в інтернеті стислий опис однієї з визначних пам'яток історії або природи місцевості (області), де ви мешкаєте.
3. Скопіюйте в буфер обміну декілька важливих, на вашу думку, абзаців знайденого опису та вставте в документ.
4. За потреби відредагуйте та відформатуйте текст.
   Видаліть гіперпосилання командою контекстного меню Видалити гіперпосилання.

Виправте помилки, якщо вони є. Чи для всіх слів це вдалося зробити? Слідкуйте за тим, щоб не було червоного та/або синього хвилястого підкреслення тексту.

5. Виберіть у тексті одне з повторюваних слів або словосполучень і замініть його в усьому документі найбільш доречним, на вашу думку, синонімом.
   Наприклад, словосполучення «визначна пам'ятка» можна замінити на слово «перлина» тощо.
   За необхідності налаштуйте параметри розширеного пошуку.
6. Збережіть документ із назвою Вправа 13.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107751).

§ 14. ФОРМАТУВАННЯ ОБ'ЄКТІВ ТЕКСТОВОГО ДОКУМЕНТА

Уведений та відредагований текст потребує форматування.

Форматування тексту — це процес надання йому потрібного зовнішнього вигляду. Для того щоб відформатувати символи введеного тексту, їх необхідно попередньо виділити.

Форматування документа в текстовому процесорі полягає в зміненні властивостей окремих його об'єктів. Ви вже ознайомилися з форматуванням символів, абзаців, рисунків, таблиць, списків.

Розгляньмо опрацювання інших об'єктів текстового документа: недрукованих і/або відсутніх на клавіатурі символів, формул.

Робота із символами і формулами

Бувають випадки, коли до тексту потрібно додати символи, яких немає на клавіатурі: наприклад, знак авторського права, знаки арифметичних дій, формули тощо.

Недруковані символи

У ході форматування текстів часто використовуються **недруковані символи** — приховані знаки, що належать до текстових об'єктів. На моніторі вони зазвичай не відображаються і на друк не виводяться (звідси й назва). Найуживаніший із них — текстовий пропуск (пробіл).

Недруковані символи в тексті позначають місця натискань клавіш Пропуск, Tab, Enter тощо. Щоб **увімкнути** або **вимкнути режим відображення** їх на екрані монітора, слід на вкладці Основне в полі Абзац натиснути кнопку ¶ — Відобразити всі знаки.

Якщо увімкнути режим відображення недрукованих символів, наведений текст набуде такого вигляду (рис. 14.1):

Між·словами·з'являються·крапки,·а·також·знаки·абзацу·та·інші·символи,·які·раніше·не·відображалися.¶

Рис. 14.1

Отже, режим відображення дозволяє швидко усунути недоліки форматування за допомогою відповідних клавіш:

Вигляд недрукованого символу	Назва недрукованого символу	Клавіші (сполучення клавіш)
·	Пропуск	**Пропуск**
¶	Кінець абзацу	**Enter**
↵	Розрив рядка	**Shift + Enter**
¬	М'який перенос	**Ctrl + «–»**
°	Нерозривний пробіл	**Ctrl + Shift + Пропуск**

Символи, відсутні на клавіатурі

Зазвичай люди користуються тими символами, що є на клавіатурі (літери, цифри, знаки пунктуації та ін.). Лише іноді відчувають потребу в додаванні символів, яких на клавіатурі немає (наприклад, символ параграфа, літери іншого алфавіту).

Для вставлення в текст символів, які відсутні на клавіатурі, потрібно:

1) установити курсор у потрібне місце в тексті;
2) вибрати вкладку Вставлення;
3) у групі Символи розгорнути список Символ (рис. 14.2);
4) у списку, що відкриється, вибрати потрібний символ.

Якщо потрібного символу в колекції немає, слід під списком натиснути інструмент Інші символи, а потім знайти й вибрати потрібний символ, натиснути кнопку Вставити.

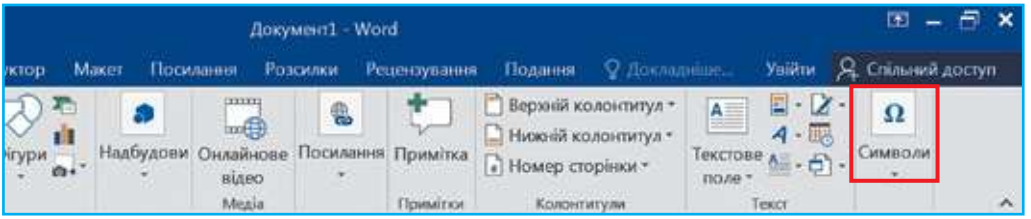


Рис. 14.2

Формули в текстовому документі

Які формули ви вивчили на уроках із математики та фізики? Як ви знаєте, одні формули мають лінійний вигляд, тобто всі елементи записані в одному рядку, інші — нелінійний. Вони багаторівневі й містять звичайні дроби, функції, дужки тощо.

Текстовий процесор Word має вбудовані засоби для роботи з формулами. Для **вставлення в документ готової формули** потрібно на вкладці Вставлення в групі Символи розгорнути колекцію формул Формула значком ▼, після чого вибрати потрібну.

Для **створення формули** необхідно під розгорнутим списком формул вибрати команду Вставити нову формулу або натиснути кнопку π Формула.

За необхідності можна створити формулу самостійно, скориставшись інструментом вкладки Конструктор (Знаряддя для формул). Назва цієї вкладки з'являється в рядку назв інших вкладок, якщо натиснути віконце створюваної формули, як наведено на рис. 14.3.

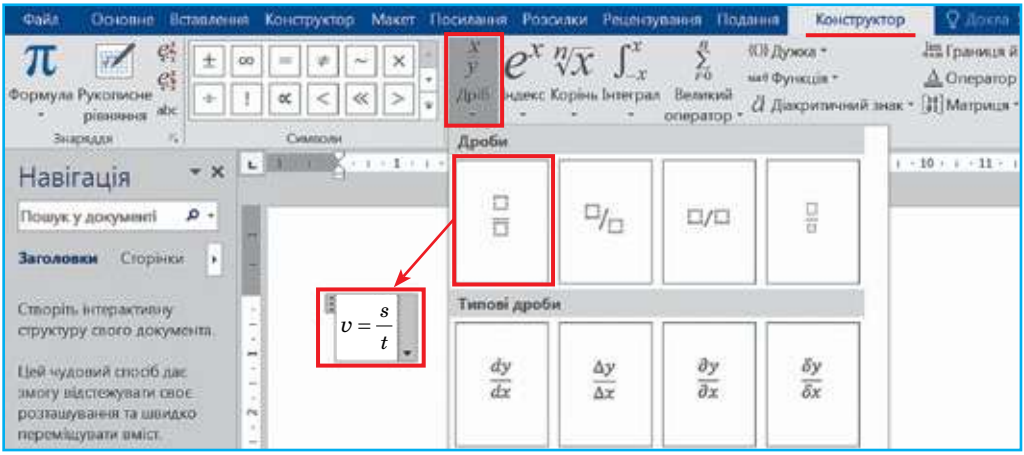


Рис. 14.3

Для змінення вигляду формули використовують відповідні інструменти вкладки Конструктор (Знаряддя для формул) та / або інструменти контекстного меню формули.

Нескладні формули та вирази мають лінійний вигляд.

Для зведення формули до лінійного (однорядкового) вигляду скористаємося інструментом Лінійний (рис. 14.4).

Зверніть увагу на те, що після цього формула, зображена на рисунку, матиме такий вигляд:

$$v=s/t.$$

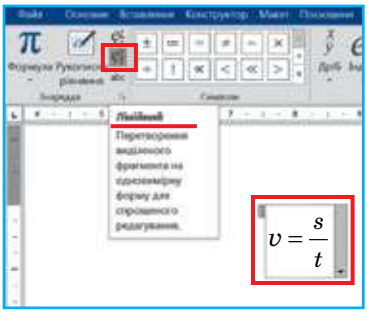


Рис. 14.4

Питання для самоперевірки

1. Назвіть кілька об'єктів текстового документа; поясніть, у чому полягає їх форматування.
2. Що таке недруковані символи і як їх відобразити на екрані?
3. Як вставити в документ символ, відсутній на клавіатурі?
4. Як створити власну формулу в текстовому документі?
5. Як перетворити кілька абзаців на маркований список?
6. Який алгоритм створення таблиці в текстовому документі?

Вправа 14

Створити й опрацювати нумеровані списки та формули.

1. Запустіть текстовий процесор і створіть новий документ.
2. Установіть такі параметри сторінки та шрифту:
   - розмір сторінки — A5; поля — по 2 см;
   - шрифт — Arial; розмір шрифту — 12.
3. Знайдіть в інтернеті прізвища й імена всіх президентів України і створіть нумерований список, розмістивши їхні прізвища й імена в порядку каденції.
4. Пригадайте або знайдіть в інтернеті формулу для розрахунку питомої теплоємності речовини, створіть її у текстовому документі (нижче, під списком). Перегляньте колекцію формул. Чи наявна в ній формула питомої теплоємності? За допомогою контекстного меню збережіть її в колекції Word.
5. Додайте до документа під формулою новий абзац із написом: § 8. Форматування.
6. Установіть для доданого абзацу: розмір шрифту — 16; накреслення — жирне. Знак § можна знайти у вікні Вставлення → Символ на вкладці Спеціальні символи.

Збережіть файл із назвою Вправа 14.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107752).

ПРАКТИЧНА РОБОТА 4
СТВОРЕННЯ ТЕКСТОВОГО ДОКУМЕНТА, ЩО МІСТИТЬ ОБ'ЄКТИ РІЗНИХ ТИПІВ

Завдання: створити текстовий документ із використанням об'єктів різних типів — картку для індивідуальної роботи.

Обладнання: комп'ютер зі встановленим текстовим процесором, з'єднаний з інтернетом.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть текстовий процесор і створіть новий документ.
2. Установіть такі параметри:
   - формат сторінки — А4; поля — по 2 см;
   - орієнтація — книжкова;
   - інтервали абзацу: Перед — 0 пт, Після — 0 пт;
   - міжрядковий інтервал — одинарний;
   - шрифт — Times New Roman; розмір шрифту — 14;
   - накреслення — жирний.
3. Уведіть:
   - у *першому* абзаці знак і номер параграфа — § 1;
   - у *другому* абзаці — заголовок Теплові властивості речовин;
   - задайте вирівнювання — По центру.
4. Знайдіть в інтернеті зображення для заданої теми, скопіюйте його та вставте в документ під заголовком.
5. Задайте формат обтікання малюнка текстом Навколо рамки. Пропорційно змініть розміри малюнка так, щоб за шириною він зайняв пів сторінки.
6. Установіть для заголовка розмір шрифту — 24. Відформатуйте документ так, щоб заголовок розмістився в одному рядку, змінивши за потреби міжсимвольний інтервал.
7. Створіть нумерований список теплових властивостей речовини за зразком, наведеним на рисунку.

8. Додайте в документ таблицю, що складається із 5 рядків і 4 стовпців, і розташуйте під списком.
9. Об'єднайте клітинки верхнього рядка таблиці та введіть назву таблиці — Основні теплові властивості.
10. Заповніть таблицю за зразком, уведіть формули.
11. Оформте (відформатуйте) таблицю за зразком, розташуйте написи по центру, перемістіть межі таблиці тощо.
12. Збережіть файл із карткою з назвою Практична робота 4.

§ 1
Теплові властивості речовин

1. Питома теплоємність
2. Теплопровідність
3. Температура плавлення
4. Температура кипіння
5. Теплове розширення
6. Теплове випромінювання
7. Температурна стійкість

Основні теплові властивості			
Фізична величина	Символ для позначення	Одиниця	Формула для визначення
Питома теплоємність	c	$\frac{\text{Дж}}{(\text{кг}\cdot{}^\circ\text{C})}$	$c=\frac{Q}{m\Delta t}$
Питома теплота плавлення	λ	$\frac{\text{Дж}}{\text{кг}}$	$\lambda=\frac{Q}{m}$
Питома теплота пароутворення	r	$\frac{\text{Дж}}{\text{кг}}$	$r=\frac{Q}{m}$

Зробіть висновок: як створити документ, що містить об'єкти різних типів; які об'єкти містить створена картка.

§ 15. СТРУКТУРА ДОКУМЕНТА. КОЛОНТИТУЛИ. ГІПЕРПОСИЛАННЯ

Для оформлення великих за обсягом документів важлива строга впорядкованість розміщення розділів, заголовків, означень.

Структурні елементи документа

Під оформленням документа розуміють структуру: *зовнішню* (параметри сторінок, поля тощо) і *внутрішню* (заголовки, підписи тощо).

Структура документа — ієрархічна схема розміщення складових текстового документа.

Складні документи розбивають на частини, розділи, параграфи та інші структурні елементи, відформатовані з дотриманням певних правил (рис. 15.1). Розгляньмо таблицю, наведену на с. 86.

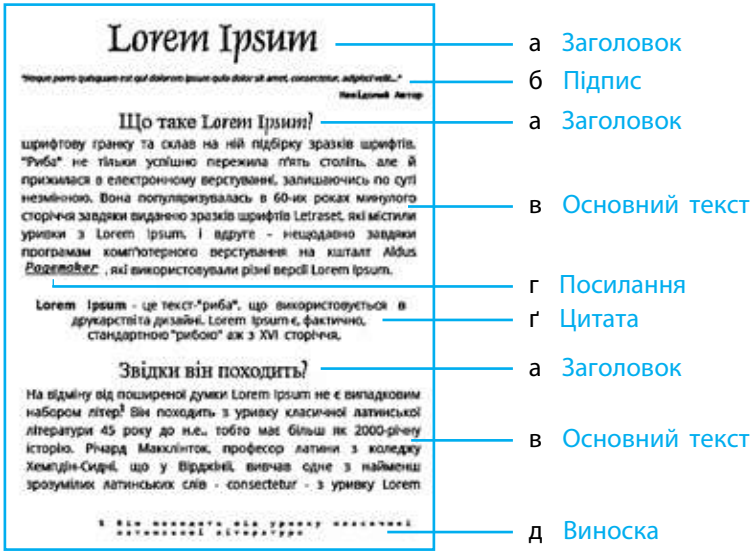


Рис. 15.1

Елемент	Опис	Форматування	
		шрифт	абзац
Заголовок, рис. 15.1, *а*	Назва документа або його частини. Заголовки розподіляють за рівнями. Чим більш загальну інформацію описує заголовок, тим вищий його рівень	Накреслення жирне, міжрядковий інтервал щільніший, ніж основна частина. Заголовки вищого рівня оформляються вагоміше за нижчий (більші символи тощо)	Зазвичай вирівнювання За серединою
Підпис (або Автор), рис. 15.1, *б*	Супровідна інформація до назви розділу або цитати	Шрифт менш помітний, ніж заголовок, або цитата. Зазвичай накреслення — курсив	Зазвичай вирівнювання За правим краєм
Основний текст, рис. 15.1, *в*	Найбільша частина за обсягом інформації	Шрифт нещільний і нефігурний, легкий для сприйняття	Зазвичай вирівнювання За шириною
Цитати, адреси, посилання на інші документи, рис. 15.1, *г*	Виділена частина основного тексту	Зазвичай накреслення курсив	Фрагмент того самого абзацу
Важлива інформація, рис. 15.1, *ґ*	Означення, висновки, попередження, інша інформація, що потребує уваги	Шрифт більший і/або щільніший порівняно з основним текстом	Зазвичай вирівнювання За серединою
Додаткова інформація рис. 15.1, *д*	Виноски, примітки, пояснення тощо	Шрифт дрібніший порівняно з основним текстом	Іноді застосовують більші відступи, ніж в основному тексті

Стильове оформлення документа

Форматування складних документів займає багато часу. Щоб прискорити цей процес, застосовують стильове оформлення.

Стиль — це іменована сукупність параметрів форматування об'єктів документа, які визначають його зовнішній вигляд.

Щоб **задати стильове оформлення об'єкта**, слід вибрати стиль на вкладці Основне в групі Стилі (рис. 15.2) або створити власний.

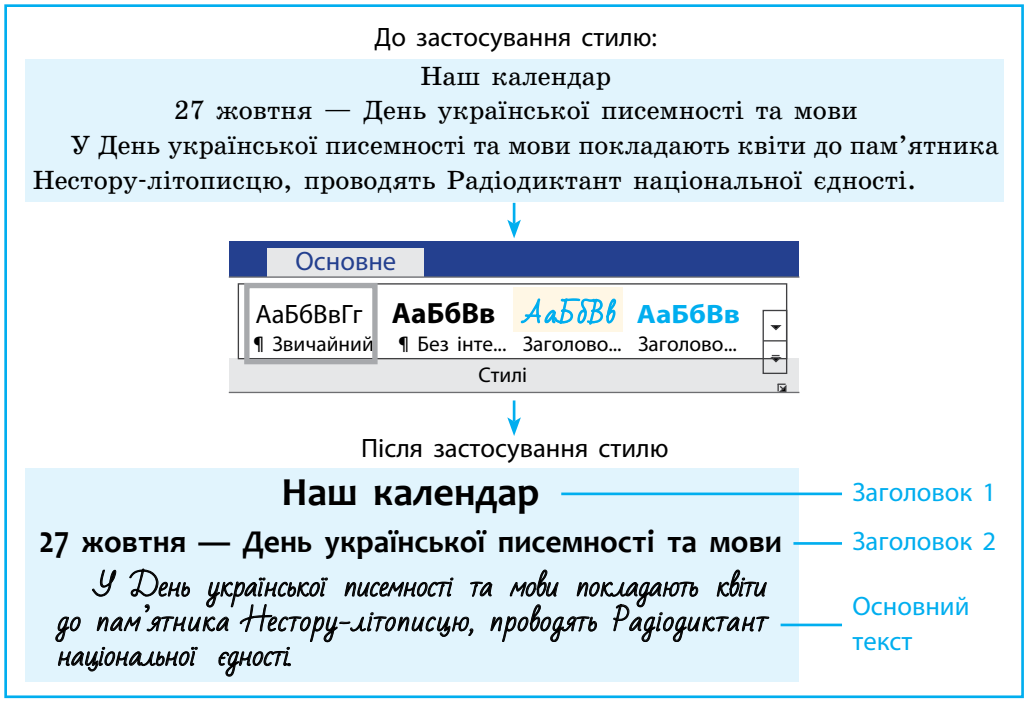


Рис. 15.2

Щоб **змінити стиль**, потрібно:

1) відкрити контекстне меню зразка стилю;
2) вибрати команду Змінити; у вікні Зміна стилю натиснути Формат;
3) налаштувати параметри стилю — після змінення параметрів зміниться й вигляд об'єктів, які мали такий стиль.

Шаблони документів

У випадку, коли потрібно створювати документи зі схожою структурою, доречно скористатися заздалегідь підготовленими та збереженими шаблонами документів.

Шаблон — це документ, що зберігається у файлі особливого формату та використовується як основа для створення нових документів.

У текстовому процесорі файли шаблонів мають розширення .dotx (.dot). Якщо користувач вирішить зберегти як шаблон документ, оформлений на власний розсуд, то слід відкрити вкладку Файл, зазначити місце збереження, назву і тип файлу — Шаблон Word (*.dotx). Після цього з'являється можливість створити новий документ на основі шаблону.

Збережений шаблон (на відміну від файлу документа) буде захищений від випадкових змін доти, доки користувач не збереже інший документ як шаблон із таким самим іменем.

! Новий документ у текстовому процесорі створюється на основі шаблону Normal.dotx.

За наявності доступу до інтернету Word дозволяє шукати й використовувати різні шаблони документів. Щоб **скористатись онлайновим шаблоном** для створення документа, потрібно на вкладці Файл вибрати команду Створити, тематику онлайн-шаблонів і певний шаблон.

Розділи

Коли для сторінок, які містяться в різних частинах документа, слід установити різні формати, документ потрібно поділити на розділи (рис. 15.3).

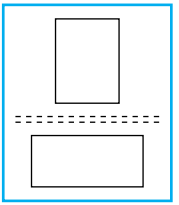


Рис. 15.3

Розділ — структурна складова текстового документа, яка завершується недрукованим символом розриву розділу.

За відсутності розривів розділів Word розглядає документ як один розділ із однаковим налаштуванням формату сторінок всього документа.

Розрив розділу зазначає місце, після якого можна задати нові параметри розмітки сторінки, притаманні лише цьому розділу.

Для **вставлення розриву розділу** потрібно вибрати на вкладці Розмітка сторінки в групі Параметри сторінки елемент Розриви та вибрати потрібний тип розриву розділу. Тепер у документі для окремих розділів можна встановити різні налаштування, наприклад, розміри та орієнтацію сторінки, ширину полів тощо.

Для **видалення розриву розділу** потрібно на вкладці Основне увімкнути кнопку ¶ — Відобразити всі знаки. Розриви розділів будуть показані подвійними пунктирними лініями, які можна видалити, скориставшись клавішами Delete або Backspace.

У разі вилучення розриву розділу видалиться форматування розділу, який містився перед розривом. Його текст стане частиною наступного розділу та набуде його формату.

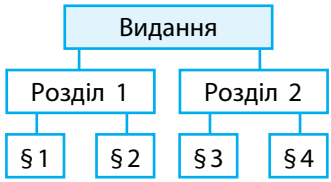


Рис. 15.4

Упорядковувати складові (частини, розділи, параграфи) текстового документа великого обсягу (рис. 15.4), наприклад підручника з інформатики, більш зручно в **режимі структури документа**.

Режим структури дозволяє приховати основний текст і побачити лише заголовки розділів, параграфів тощо. Можна змінювати рівень заголовків, переміщувати параграфи та розділи тощо. Щоб увімкнути цей режим, потрібно на вкладці Вигляд у групі Подання вибрати команду Структура (рис. 15.5).

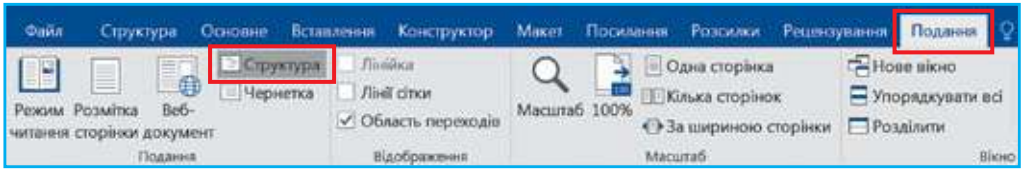


Рис. 15.5

Колонтитули

Додаткова інформація може міститися в колонтитулах.

Колонтитули — це текст або зображення, які розміщуються у верхній (верхній колонтитул) або нижній (нижній колонтитул) частині сторінок документа.

Інформація, подана в колонтитулі, відображається на всіх сторінках документа або сторінках одного чи кількох розділів. Він може містити назву документа, номер сторінки, дату або іншу інформацію, яка повторюється на кожній сторінці.

Для **встановлення колонтитула** необхідно:

1) на вкладці Вставлення в групі Колонтитули вибрати відповідну команду: Верхній колонтитул, Нижній колонтитул, Номер сторінки;
2) у полі введення колонтитула ввести його текст або на вкладці Конструктор (Знаряддя для колонтитулів) у групі Вставлення вибрати потрібний інструмент;
3) налаштувати відображення колонтитула інструментами вкладки Конструктор (Знаряддя для колонтитулів) (рис. 15.6).

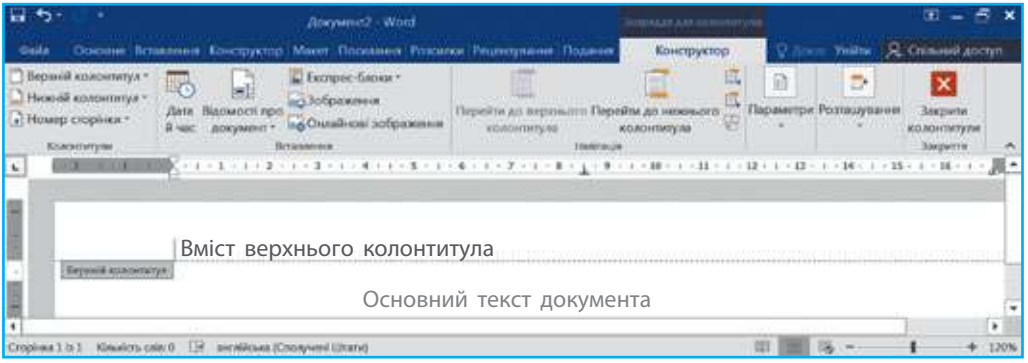


Рис. 15.6

Для **повернення до основного тексту документа** на ньому слід натиснути кнопку Закрити колонтитули на вкладці Конструктор (Знаряддя для колонтитулів) (див. рис. 15.6).

Для **видалення колонтитула** треба на ньому двічі клацнути, виділити його вміст і натиснути клавішу Delete або скористатися відповідними інструментами групи Колонтитули вкладки Конструктор (Знаряддя для колонтитулів).

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 Конструктор (Знаряддя для колонтитулів) (див. рис. 15.6).

Щоб **створити різні колонтитули для певних частин документа,** потрібно:

1) розбити текст на розділи; двічі клацнути на верхньому (або нижньому) полі сторінки потрібного розділу;
2) на вкладці Конструктор (Знаряддя для колонтитулів) у групі Навігація вимкнути кнопку Як у попередньому;
3) увести потрібний вміст колонтитула;
4) подвійним клацанням перейти до основного тексту документа.

Після розбиття на розділи можна змінювати вміст окремих колонтитулів, і це не впливатиме на колонтитули інших розділів документа.

Попередниками сучасних колонтитулів були примітки на полях стародавніх манускриптів, які переписувачі книг залишали для того, щоб читацька аудиторія могла досить легко знайти потрібну інформацію або розуміти контекст.

Гіперпосилання

У текстовому процесорі Word разом із підготовкою документів, які призначено для друку, можна розробляти також електронні документи, які, зокрема, містять гіперпосилання (посилання) для зручного користування документом. Наприклад, це може бути миттєвий доступ до використаних джерел тощо.

Посилання (гіперпосилання) — об'єкт документа, клацанням по якому здійснюється перехід у певне місце документа або до іншого файлу на комп'ютері або в інтернеті.

Для **створення посилання (гіперпосилання)** потрібно:

1) виділити фрагмент тексту або графічний об'єкт;
2) на вкладці Вставлення (група Посилання) вибрати інструмент Гіперпосилання (рис. 15.7);
3) зазначити, на який саме об'єкт створюється посилання.

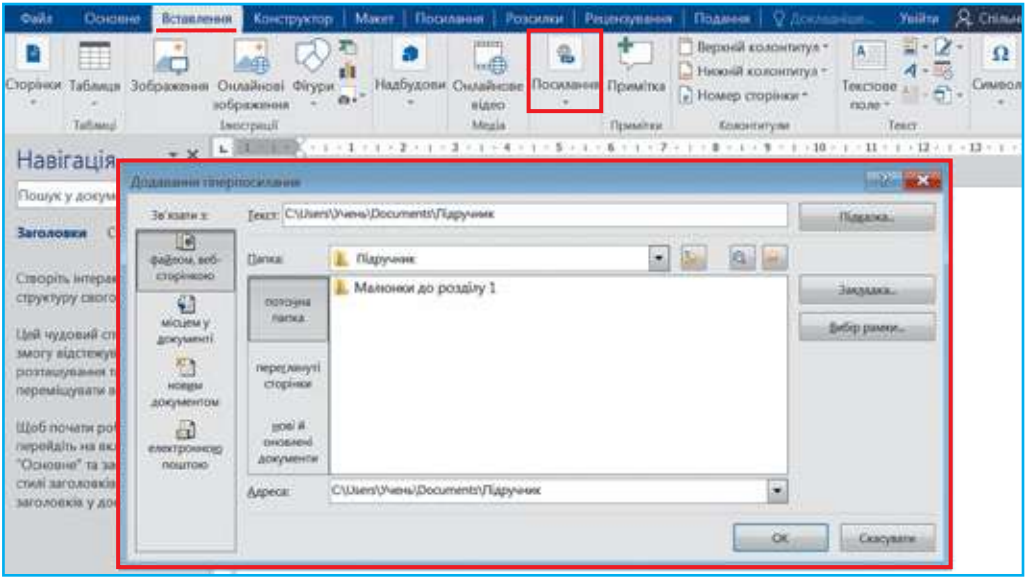


Рис. 15.7

Посилання можна створити або змінити командою контекстного меню виділеного об'єкта Гіперпосилання. Щоб **створити посилання на певне місце**, у тексті документа слід створити закладку; щоб **перейти за посиланням**, — натиснути його лівою кнопкою миші, утримуючи натиснутою клавішу Ctrl.

Для **створення закладки** потрібно виділити об'єкт або встановити курсор у потрібне місце, вибрати інструмент Закладка, а потім увести назву нової закладки й натиснути кнопку Додати у вікні зі списком закладок.

Питання для самоперевірки

1. Назвіть і охарактеризуйте кілька структурних елементів текстового документа.
2. У чому полягає стильове оформлення документа Word?
3. Чим відрізняється використання шаблонів і звичайних документів?
4. Що таке розділи в текстовому документі; у чому полягає особливість форматування розділу?
5. Наведіть приклад колонтитулів багатосторінкового документа. Поясніть, який вміст допоможе впорядкувати документ.
6. Як створити посилання (гіперпосилання) у тексті документа?

Вправа 15

Створити й оформити текстовий документ із використанням різних стилів.

1. Запустіть текстовий процесор і створіть новий документ.
2. Знайдіть в інтернеті й скопіюйте одну з дій п'єси І. Карпенка-Карого «Сто тисяч» (або за вказівкою вчителя / вчительки).
3. Для основного тексту дії встановіть стиль Звичайний, для приміток — курсив.
   Якщо назви твору немає, введіть її на початку документа і встановіть стиль Назва. У кінці документа введіть ім'я і прізвище автора твору, задайте стиль Строгий.
4. Установіть колонтитули: верхній — із поточною датою, нижній — із номерами сторінок.
5. Зробіть назву твору посиланням на вебсторінку із ним.
6. Збережіть документ у форматі шаблону з назвою Вправа 15. Покажіть свою роботу однокласнику / однокласниці та вчителю / вчительці.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107753).

§ 16. АВТОМАТИЗОВАНЕ СТВОРЕННЯ ЗМІСТУ ДОКУМЕНТА

Що таке зміст і яке його призначення?

ЗМІСТ

Рис. 16.1

Зміст документа — це перелік назв структурних частин документа, впорядкований відповідно до його ієрархічної схеми, із зазначенням номерів сторінок.

Завдяки нумерації сторінок (рис. 16.1) можна легко орієнтуватися в структурі великого за обсягом документа. Зміст розташовується на початку або в кінці документа, і часто перегляд починають саме з нього. Де розташований зміст у вашому підручнику?

Посилання на відповідні розділи в змісті дають можливість швидко знаходити потрібне та переміщуватися від одного розділу до іншого. Інструментами вікна Навігація також зручно переміщуватися. Їх можна ввімкнути прапорцем Область переходів на вкладці Подання (рис. 16.2).

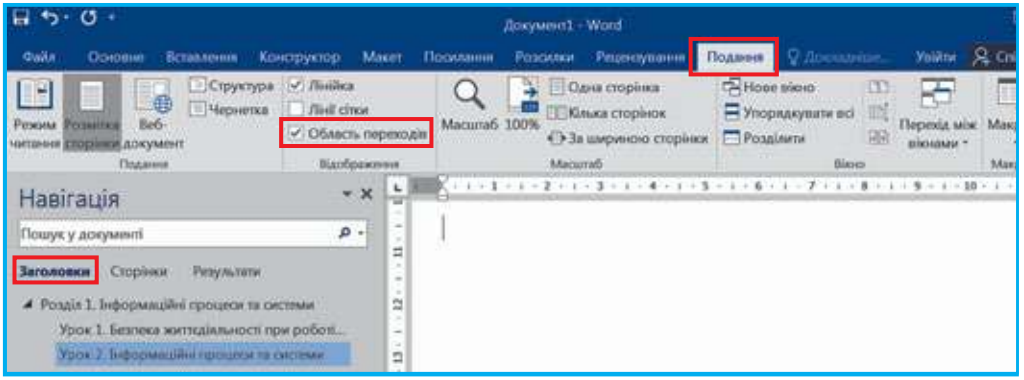


Рис. 16.2

Зміст можна створити в ручному або автоматичному режимі.

Створення змісту в ручному режимі доцільне для невеликих документів, і єдино можливе для документів, які мають понад 9 рівнів заголовків (за автоматичного створення зміст може мати не більше 9 рівнів заголовків).

Автоматичне створення змісту

Автоматизована побудова змісту документа стає можливою після застосування до заголовків і підзаголовків різного степеня вкладеності відповідних стилів: Заголовок 1, Заголовок 2 і т. д.

Для **автоматизованого створення змісту** потрібно:

1) відформатувати назви розділів, параграфів тощо відповідними стилями (рис. 16.3);
2) вибрати місце вставлення змісту (це може бути на початку або в кінці документа);
3) на вкладці Посилання в групі Зміст розгорнути список Зміст;
4) вибрати стиль змісту та натиснути клавішу Enter.

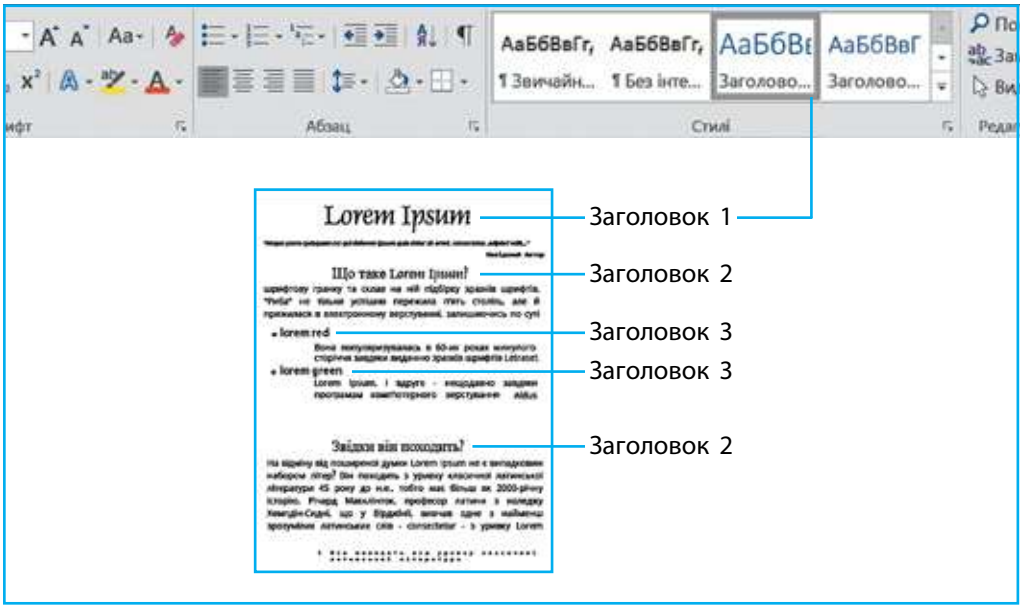


Рис. 16.3

Для **оновлення змісту** потрібно:

1) на вкладці Посилання в групі Зміст вибрати Оновлення таблиці;
2) вибрати команду Оновити лише номери сторінок або Оновити цілком.

Для **видалення змісту документа** потрібно на вкладці Посилання в групі Зміст вибрати Оновлення таблиці, а потім вибрати команду Видалення змісту.

Автоматичне створення покажчика

Для пришвидшення пошуку термінів у кінці документа зазвичай розміщують алфавітний покажчик (рис. 16.4).

Рис. 16.4

Покажчик (алфавітний покажчик) — список слів із зазначенням номерів сторінок, на яких вони згадуються.

Для **створення покажчика** потрібно:

1) виділити в тексті термін (слово або фразу);
2) на вкладці Посилання, що в групі Покажчик, вибрати команду Позначити елемент;
3) у вікні Визначення елемента покажчика налаштувати такі параметри:
   а) змінити текст у полі Основний — за потреби;
   б) щоб елемент указував на кілька термінів (наприклад, архівний файл; стиснений файл; архів), у полі додатковий уписати через двокрапку інші терміни;
   в) натиснути кнопку Позначити або Позначити все (щоб знайти й позначити в документі всі терміни);
4) вибрати місце в документі, де буде розміщено покажчик, і на вкладці Посилання в групі Покажчик вибрати Покажчик.

Щоб **видалити елемент покажчика**, потрібно відобразити недруковані символи тексту (Основне → ¶) і видалити біля елемента покажчика позначку у фігурних дужках.

Опрацювання складних документів

Щоб робота зі складним текстовим документом була ефективною, користувачам слід дотримуватися певної послідовності дій.

Ознайоммося з **основними етапами опрацювання** складного текстового документа, розгляньмо таблицю.

Номер	Етап	Опис
1	Набір і збереження	Уведення тексту з клавіатури і збереження у вигляді файлу на носії
2	Редагування документа	Додавання або вилучення фрагментів тексту, а також інших об'єктів, змінення їх розташування, виправлення помилок
3	Форматування документа	Надання документу бажаного вигляду. — Використання вбудованих стилів заголовків або користувацьких параметрів форматування, що визначають рівень абзацу — Поділ документа на розділи й налаштування параметрів сторінки для всього документа (рідше для кожного розділу) — Перегляд структури документа в режимі структури або за допомогою засобів навігації (за потреби коригування)
4	Підготовка документа до друку	Оформлення заголовків, розбивка на сторінки, їх нумерування, створення змісту
5	Друк документа	Виведення на папір усіх або окремих сторінок створеного документа

Під час роботи зі складним текстовим документом робоче поле можна розділити на кілька частин: вкладка Вигляд → Розділити (група Вікно). Зверніть увагу на те, що кожну частину документа можна прокручувати окремо, що зручно робити під час копіювання (або переміщення) фрагментів тексту.

Питання для самоперевірки

1. У чому полягає відмінність між змістом звичайного (паперового) і електронного документів?
2. Опишіть алгоритм автоматизованого створення змісту.
3. У чому полягають переваги автоматизовано створеного змісту перед ручним?
4. Коли автоматизований зміст документа створити неможливо?
5. Що таке покажчик? Яке його призначення?
6. Опишіть послідовність дій з опрацювання складного текстового документа.

Вправа 16

Створити зміст багатосторінкового структурованого текстового документа.

1. Запустіть текстовий процесор і відкрийте документ Вправа 15. Або виконайте такі дії:
   1) створіть новий документ;
   2) знайдіть в інтернеті першу дію п'єси Івана Карпенка-Карого «Сто тисяч» (або вибраного вами твору);
   3) скопіюйте в документ.
2. Установіть стилі заголовків твору:
   - для назви дії — Заголовок 1,
   - для яв — Заголовок 2.
3. Проставте номери сторінок (автоматично).
4. Додайте на початку *порожню* сторінку вставленням розриву сторінки (Розмітка сторінки → Розриви → Сторінка).
5. Створіть на доданій сторінці автоматизований зміст.
6. Збережіть файл із назвою Вправа 16. Покажіть свою роботу однокласнику / однокласниці та вчителю / вчительці.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107754).

ПРАКТИЧНА РОБОТА 5
АВТОМАТИЗОВАНЕ СТВОРЕННЯ ЗМІСТУ ДОКУМЕНТА

Завдання: створити текстовий документ, який складається з декількох розділів, здійснити форматування його об'єктів, створити зміст і покажчик в автоматизованому режимі.

Обладнання: комп'ютер із текстовим процесором, під'єднаний до інтернету.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть текстовий процесор і створіть новий документ. Знайдіть в інтернеті три-чотири вірші українських поетів або поетес і скопіюйте в документ.
   Якщо текст міститься в таблиці, то витягніть його. Для цього виділіть усю таблицю й виберіть команду Конструктор → → Дані → Перетворити на текст (розділювач — знак абзацу).
2. Відформатуйте текст так, щоб рядки (окрім першого) не починалися з нового абзацу, видаліть порожні рядки та розмістіть назву вірша в окремому абзаці.
3. Уведіть у першому абзаці назву створеної вами збірки, наприклад: Вірші Лесі Українки (Василя Симоненка, Миколи Вінграновського, Ліни Костенко та інших — на ваш вибір).
4. Установіть стиль:
   - для документа — Звичайний;
   - для назв віршів — Заголовок 1;
   - для назви збірки — Назва.
5. Знайдіть в інтернеті портрет автора чи авторки віршів і скопіюйте в документ після назви збірки. Наприклад, як наведено далі.

За потреби відформатуйте зображення: обріжте, змініть розміри, обтікання, розташування тощо.

6. Установіть окремо для першої сторінки верхній колонтитул зі своїми прізвищем, ім'ям і класом.
   У нижньому колонтитулі кожної сторінки вставте номери сторінок.
7. Створіть під портретом автора чи авторки автоматизований зміст. Змініть напис Зміст на Зміст збірки.
8. Поділіть документ на розділи. Для цього після портрета й останнього рядка кожного вірша вставте розриви Поточна сторінка.
   Після змісту вставте розрив Наступна сторінка.
9. Установіть орієнтацію перших сторінок документа включно зі змістом: альбомна. Для решти сторінок орієнтація — книжкова.
10. Установіть для всього документа міжрядковий інтервал — 1,5. Оновіть зміст.
11. Додайте до назви збірки посилання на сторінку, з якої копіювалися тексти віршів. Перевірте, чи працює посилання.
12. Збережіть файл із назвою Практична робота 5.

Вірші Лесі Українки

Зробіть висновок: якими є особливості роботи зі складним текстовим документом, як автоматизувати процес створення змісту.

§ 17. КОРИСТУВАННЯ ХМАРНИМ ДИСКОМ. СПІЛЬНА РОБОТА З ДОКУМЕНТОМ

Нині стрімко зростає впровадження і використання хмарних технологій, які дозволяють працювати з документами в онлайн-режимі редагувати один і той самий документ одночасно кільком особам. Хмарні сервіси надають постійний доступ до віддалених інтернет-ресурсів (серверів, сховищ). Одним із хмарних сервісів є хмарний диск.

Створення документів на хмарному диску

Створити текстові або інші документи й опрацьовувати їх за допомогою відповідних редакторів можна в хмарному сховищі OneDrive корпорації Microsoft (рис. 17.1) і/або на Google Диску корпорації Google (рис. 17.2) тощо.

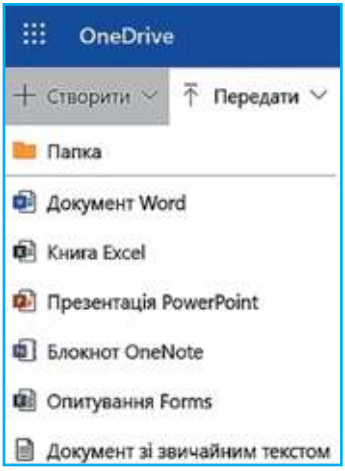


Рис. 17.1

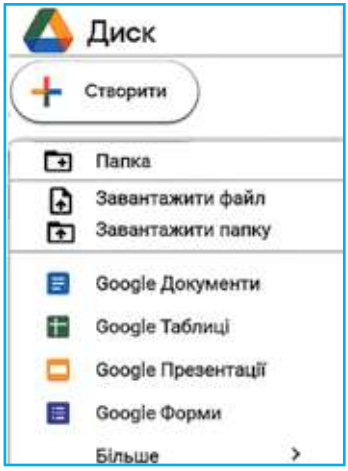


Рис. 17.2

Хмарний диск — сховище даних, розташоване на віддаленому сервері, цілодобово доступне користувачам завдяки інтернету.

Щоб отримати можливість користуватися хмарними сервісами корпорацій Microsoft і Google, досить створити відповідні облікові записи.

На хмарному диску можна створити документ, таблицю, презентацію, малюнок, форму опитування. Крім браузера, жодного додаткового програмного забезпечення встановлювати не треба. Користування онлайновими текстовими редакторами схоже на користування текстовим процесором, про що ми вже згадували.

Для **створення документа** на OneDrive або Google Диск слід:

1) увійти до свого облікового запису Microsoft (microsoft.com) або Google (google.com);
2) кнопкою відкрити Запускач програм для Microsoft (рис. 17.3, *а*) або Додатки Google для Google (рис. 17.3, *б*);
3) відкрити хмарний диск, вибравши для цього OneDrive або Google Диск;
4) натиснути кнопку Створити; вибрати тип створюваного документа: або (див. рис. 17.1 і 17.2 відповідно).

а

б

Рис. 17.3

Збереження файлів на хмарному диску

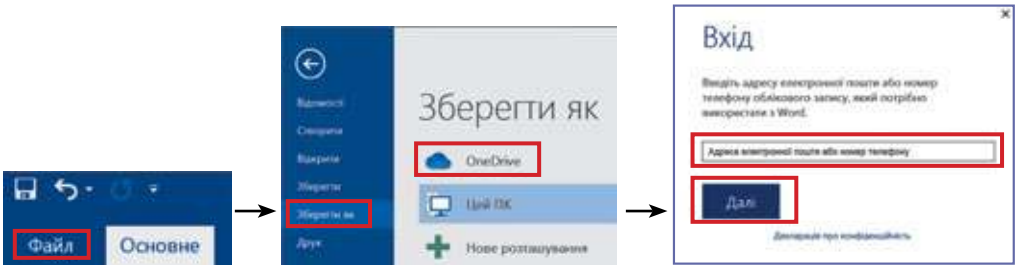
Створений на комп'ютері текстовий документ за наявності інтернету можна зберегти на OneDrive або Google Диску.

▶ Збереження документа на OneDrive під час його опрацювання

Якщо на хмарному диску OneDrive треба зберегти текстовий документ, відкритий у Word, то потрібно:

1) відкрити вкладку Файл (рис. 17.4, *а*) і вибрати команду Зберегти як (рис. 17.4, *б*);
2) вибрати для збереження хмарний диск OneDrive (рис. 17.4, *б*);
3) увійти до свого облікового запису Microsoft (рис. 17.4, *в*);
4) вибрати потрібну папку на хмарному диску, в якій планується зберегти файл опрацьовуваного документа;

5) зазначити (або змінити) назву та/або тип файлу, що зберігається; натиснути кнопку Зберегти.



а б в

Рис. 17.4

Для надання кільком користувачам / користувачкам спільного доступу до документа, збереженого на OneDrive, необхідно вибрати на вкладці Файл команду Надати спільний доступ (про це — далі).

! Збереження документа на хмарному диску OneDrive під час опрацювання в текстовому редакторі можливе лише за умови, що Word активовано.

Для організації спільної роботи над документом можна переслати файл іншим користувачам поштою. Автор документа (керівник проєкту) може відстежувати зміни в надісланих йому файлах і на свій розсуд приймати внесені зміни або ні.

Проте наведений спосіб спільного опрацювання документів є досить громіздким і малоефективним. Значно зручнішим способом є робота з файлами, збереженими на хмарному диску за допомогою онлайн-редакторів.

i На OneDrive зареєстрованим користувачам / користувачкам безкоштовно надається 5 Гб дискового простору для зберігання користувацьких файлів, 15 Гб для зберігання поштової кореспонденції, пакет сервісів Microsoft 365 різного призначення, зокрема й текстовий редактор Word Online [9].

Для **збереження файлу** на OneDrive потрібно:

1) зайти до свого облікового запису Microsoft;
2) відкрити хмарне сховище OneDrive; за необхідності відкрити папку на OneDrive для зберігання файлу;
3) натиснути кнопку Передати; зазначити об'єкт для збереження: Файл або Папка;
4) знайти у вікні провідника файл або папку й натиснути кнопку Відкрити — після цього вибране буде збережено на OneDrive.

На Google Диск можна завантажити файли й папки з комп'ютера або створити їх безпосередньо на хмарному диску. Алгоритм збереження файлу чи папки на Google Диску подібний до алгоритму для OneDrive.

На Google Диску власникам облікового запису Google для збереження кореспонденції та інших файлів безкоштовно надається простір обсягом 15 Гб, а також низка онлайн-сервісів (додатків) Google з онлайн-редактором Google Документа [9].

Синхронізація даних

Хмарні сховища можна синхронізувати з користувацьким комп'ютером, що дає змогу отримувати доступ до них із файлової системи комп'ютера, під'єднаного до інтернету. Після синхронізації можна бачити файли у хмарному сховищі у вікні провідника та виконувати з ними різні дії як зі звичайними комп'ютерними файлами: видаляти, копіювати або переміщати з папки в папку, з комп'ютера на хмарний диск і навпаки.

Спільна робота з документом

Під час спільної роботи з документом його можуть використовувати одночасно кілька осіб. Проте працювати з документом, збереженим на хмарному диску, можуть тільки ті користувачі й користувачки, які мають до нього доступ. Потрібне лише під'єднання до інтернету та наявність браузера. Доступ до файлів надає власник сховища, в якому вони розміщені. Одним файлом можна поділитися з сотнями електронних адрес (до 600).

Доступ на OneDrive

Для надання доступу до файлу на OneDrive слід позначити потрібний файл «галочкою» і натиснути кнопку Спільний доступ (рис. 17.5) або відкрити файл і натиснути кнопку Спільний доступ.

Далі у вікні Параметри посилань потрібно налаштувати параметри доступу (будь-кому, хто має посилання, чи лише користувачам із певною адресою; дозволити редагування документа чи лише перегляд; обмежувати термін чи надати безстроковий доступ; установлювати пароль доступу чи ні) і натиснути кнопку Застосувати (рис. 17.6).

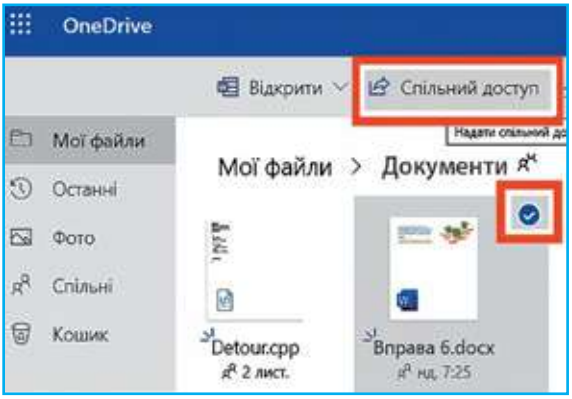


Рис. 17.5

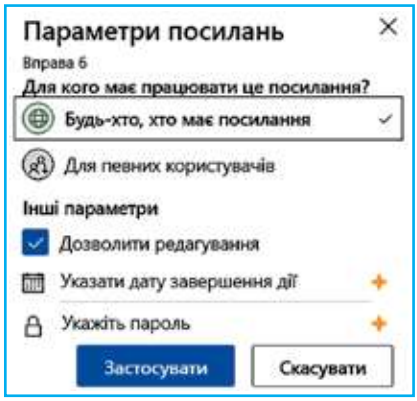


Рис. 17.6

Доступ на Google Диск

Про типи й рівні доступу на Google Диск ви дізналися в 7 класі.

Ви знаєте, що на Google Диск існують два типи доступу до файлу: всім, хто має посилання на файл, і лише тим, кому надано доступ за їх іменами або електронними адресами, та три рівні доступу до документа: перегляд; коментування; редагування. Кожен наступний рівень надає користувачеві всі можливості попередніх рівнів доступу. Так, найвищий рівень — редагування — включає можливість перегляду та коментування у файлах.

Для **надання доступу до файлів** на Google Диску необхідно в контекстному меню файлу вибрати команду Надати доступ.

Для надання доступу до файлу в режимі онлайн-перегляду особі, що є власником файлу, потрібно скористатися кнопкою Спільний доступ (рис. 17.7).

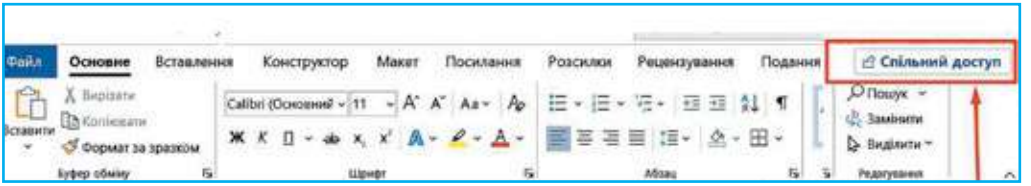


Рис. 17.7

У формі, що відкриється, потрібно встановити тип і рівень **доступу** користувача до файлу й натиснути Готово (рис. 17.8).

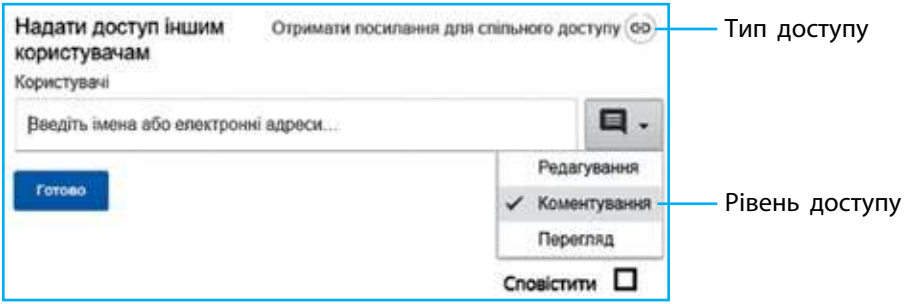


Рис. 17.8

Відстеження змін у документі

Іноді користувачам і користувачкам важливо відстежувати, хто і які зміни вносив у документ. Розгляньмо, як це здійснити.

Якщо документ опрацьовується (редагується, форматується, коментується) кількома особами одночасно, то всі зміни відстежуються в режимі реального часу, тобто кожен бачить, хто і які зміни вносить.

Щоб **відстежувати зміни**, які були внесені за відсутності користувача / користувачки, потрібно в онлайн-редакторі документа Word на вкладці Рецензування увімкнути відстеження виправлень (рис. 17.9).

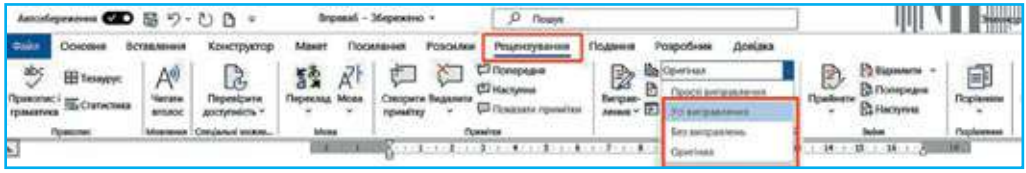


Рис. 17.9

У вбудованому текстовому редакторі Google Документа виправлення й дати їх внесення можна побачити на вкладці Файл, скориставшись командою Історія версій.

Спілкування під час колективного опрацювання документів може відбуватися засобами текстового процесора: за допомогою приміток — кнопка Примітки у Word та коментарів у google-документі — кнопка у правому верхньому куті вікна.

Для **додавання приміток** до виділених фрагментів тексту чи до графічного зображення в документі використовують команду Рецензування → Створити примітку (рис. 17.10).

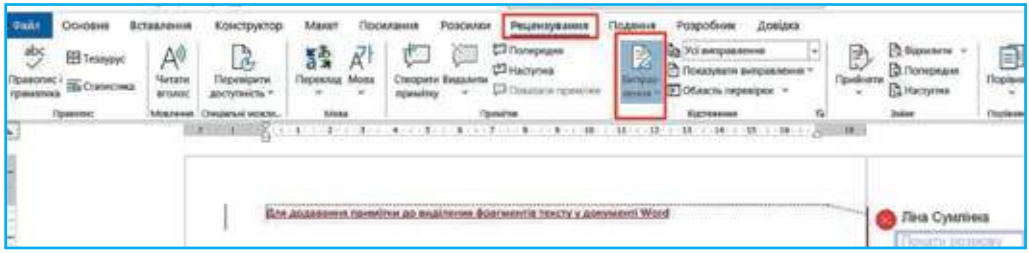


Рис. 17.10

Для **додавання коментарів до фрагмента** в google-документі є значок , що з'являється після виділення тексту (рис. 17.11).

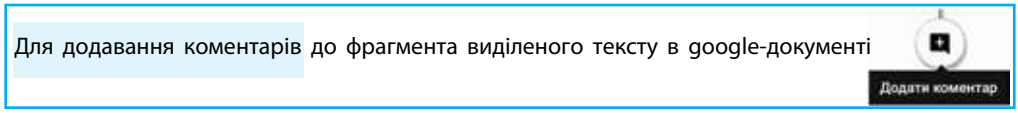


Рис. 17.11

Питання для самоперевірки

1. Що таке хмарний диск?
2. Які переваги надає хмарний диск у користуванні збереженими на ньому документами?
3. Як створити документ на OneDrive або Google Диску?
4. Як зберегти документ на OneDrive під час роботи у Word?
5. Як зберегти файл на OneDrive або Google Диску?
6. Як надати доступ для перегляду файлів на OneDrive або Google Диску з автоматичним сповіщенням про це?
7. Як додати примітку або коментар до текстового фрагмента на хмарному диску?
8. Як відстежити зміни в документі?

Вправа 17

Надати доступ до текстового документа, збереженого на хмарному диску.

1. Зайдіть до свого облікового запису Google (або Microsoft).
2. Відкрийте хмарний диск Google Диск або OneDrive.
3. Збережіть на хмарному диску текстовий документ з іменем Вправа 17.docx.
4. Надайте доступ для перегляду збереженого документа вчителю / вчительці та кільком однокласникам / однокласницям, автоматично сповістивши їх поштою.
5. Попросіть однокласника чи однокласницю надати вам доступ до свого документа, надайте адресу своєї електронної пошти.
6. Відкрийте свою поштову скриньку. Перевірте, чи надано доступ; якщо так, то перегляньте документ.
   Спробуйте відредагувати текст або залишити коментар (примітку). Поясніть, чи вдалося відредагувати документ.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107755).

ПРАКТИЧНА РОБОТА 6
СТВОРЕННЯ ДОКУМЕНТА НА ХМАРНОМУ ДИСКУ ДЛЯ СПІЛЬНОГО ОПРАЦЮВАННЯ

Завдання: колективно створити й опрацювати документ онлайн.
Обладнання: комп'ютер, під'єднаний до інтернету.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Об'єднайтеся в групи для роботи над створенням реферату.
2. Виберіть тему реферату.
3. У своїй групі виберіть керівника, який створює документ на Google Диску або OneDrive і надає доступ до редагування.
4. Зайдіть до свого облікового запису Google або Microsoft.
5. Перейменуйте документ відповідно до теми.
6. На титульній (першій) сторінці документа напишіть тему.
7. Знайдіть в інтернеті вимоги до учнівських рефератів. Колективно опрацюйте приблизний обсяг і зміст (за розділами) свого реферату та розподіліть завдання.
8. Під темою реферату з робочого місця впишіть свої прізвище й ім'я, біля яких у дужках зазначте назву свого розділу.
9. Доберіть в інтернеті матеріали для свого розділу реферату та скопіюйте до спільно опрацьовуваного документа.
10. Колективно відредагуйте та відформатуйте документ. До заголовка свого розділу додайте гіперпосилання на джерело інформації. Для спілкування використовуйте коментарі.
11. Надайте вчителю / вчительці доступ до документа із можливістю перегляду та коментування.

Зробіть висновок: у чому полягають переваги й недоліки колективного опрацювання документа в онлайн-редакторі на хмарному диску порівняно з іншими способами спільної роботи з документом.

ПРАКТИЧНА РОБОТА 7
СТВОРЕННЯ МІНІГАЗЕТИ НА ХМАРНОМУ ДИСКУ

Завдання: створити текстовий документ, який містить об'єкти різних типів, і автоматизований зміст.

Обладнання: комп'ютер, з'єднаний з інтернетом.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Ви — редактор газети. Виберіть тему. На хмарному диску створіть текстовий документ. Для цього:

а) розмістіть на титульній сторінці назву, логотип, гасло або підзаголовок; укажіть дату випуску та номер мінігазети;

б) додайте внизу сторінки своє прізвище та клас.

2. Запросіть у команду кілька однокласників / однокласниць, яких цікавить вибрана тема. Розподіліть завдання.
   - Кореспондент: пише статтю (2–4 абзаци), наприклад, про новини школи, огляд подій чи опис шкільного проєкту.
   - Дизайнер: виконує форматування сторінок газети.
   - Спеціаліст із реклами: формує розділ оголошень або реклами.
3. На наступній сторінці розмістіть текст, зображення, підпис.
4. Додайте таблицю, наприклад, розкладу шкільних заходів.
5. Налаштуйте колір меж у таблиці, розмір клітинок тощо.
6. Додайте нумерований (маркований) список цікавих фактів.
7. Оформте колонтитули: у верхньому (нижньому) колонтитулі додайте назву газети або номер сторінки.
8. Вставте текстове поле для оголошення або реклами. Використайте яскраві кольори для привернення уваги.
9. Для назв статті й таблиці встановіть стиль Заголовок 1. Створіть автоматизований зміст мінігазети.
10. Збережіть файл. Надайте доступ вчителю / вчительці.

Зробіть висновок: які особливості форматування об'єктів використано під час оформлення текстового документа.

РОЗДІЛ 4
ОПРАЦЮВАННЯ МУЛЬТИМЕДІЙНИХ ОБ'ЄКТІВ

§ 18. Поняття мультимедіа. Кодування аудіоданих. Формати аудіофайлів

§ 19. Кодування відеоданих. Формати відеофайлів

Практична робота 8. Використання аудіоданих у програмах мовою Python

§ 20. Програмне забезпечення для опрацювання звуку

§ 21. Програмне забезпечення для опрацювання відео

§ 22. Побудова аудіо- й відеоряду

§ 23. Опрацювання, зберігання та розміщення відеофільму в інтернеті

Практична робота 9. Створення відеофільму

Практична робота 10. Розміщення відеоматеріалів в інтернеті

ПОВТОРЮЄМО

Згадаймо, що під поняттям *мультимедіа* розуміють поєднання повідомлень, поданих у різний спосіб. *Об'єктами мультимедіа* є текст, зображення, звук, відео, анімація.

До пристроїв, що входять до складу *мультимедійного обладнання*, належать інтерактивні дошки, проєктори, відеокамери, акустичні системи тощо. Щороку кількість пристроїв збільшується.

Мультимедіа застосовують у різних галузях: освіті, кіно, телебаченні, рекламі, медицині тощо.

Файли мультимедіа створюють та опрацьовують за допомогою різних комп'ютерних програм і зберігають у *спеціальних форматах*. Ви вже навчилися користуватися деякими з цих програм, наприклад, програмою для створення комп'ютерних презентацій, мультимедійним програвачем, графічними редакторами тощо.

1. Що таке мультимедіа?
2. Що є об'єктами мультимедіа?
3. Наведіть приклади мультимедійних пристроїв.
4. Назвіть галузі застосування мультимедіа.
5. Що називають графічним редактором?
6. Яка програма призначена для перегляду відео?

У цьому розділі ви навчитеся створювати та опрацьовувати такі об'єкти мультимедіа, як звук і відео, опануєте процес створення відеофільму та способи його розміщення в інтернеті.

§ 18. ПОНЯТТЯ МУЛЬТИМЕДІА. КОДУВАННЯ АУДІОДАНИХ. ФОРМАТИ АУДІОФАЙЛІВ

Повідомлення, які через тисячоліття надійшли від наших предків, належать до графічної (наскельні малюнки, настінні розписи) і текстової (стародавні написи й документи) інформації. А от історія запису звуку та відео налічує близько 150 років. Але нині уявити без них інформаційний простір неможливо.

Мультимедіа

В історії розвитку комп'ютерів також простежуються подібні етапи. На перші комп'ютерні монітори виводилася текстова інформація. Пізніше з'явилися відеосистеми з підтримкою виведення графіки. А подальше зростання обчислювальних ресурсів дозволило опрацьовувати звукові та відеодані — так комп'ютер став мультимедійним.

Мультимедіа (англ. *multimedia*) — комбінування в одному інформаційному продукті інформації, поданої різними способами: текстової, графічної, звукової, відео, а також можливості її інтерактивного використання.

Кодування звуку

Людина отримала змогу зберігати і відтворювати звукові повідомлення, коли навчилася записувати звукові коливання на грамплатівки (рис. 18.1), магнітофонні стрічки тощо. Такий звукозапис називають **аналоговим**.

Рис. 18.1

Фонограф став першим пристроєм, який міг відтворювати та записувати звуки. Американський винахідник Томас Едісон оголосив про винайдення фонографа у 1877 році [10].

Ви вже знаєте, як кодують текстові та графічні дані. Тепер з'ясуймо, як у комп'ютері кодується звук.

Цифровий звукозапис — це процес подання звуку у вигляді двійкового коду, а також результат цього процесу.

Звук — це хвиля, тобто явище поширення коливань у речовині (докладніше про це ви дізнаєтеся на уроках фізики). Зобразімо ці коливання у вигляді графіка залежності рівня сигналу від часу (рис. 18.2).

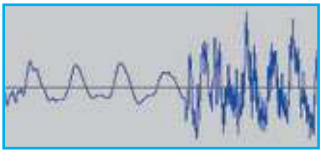
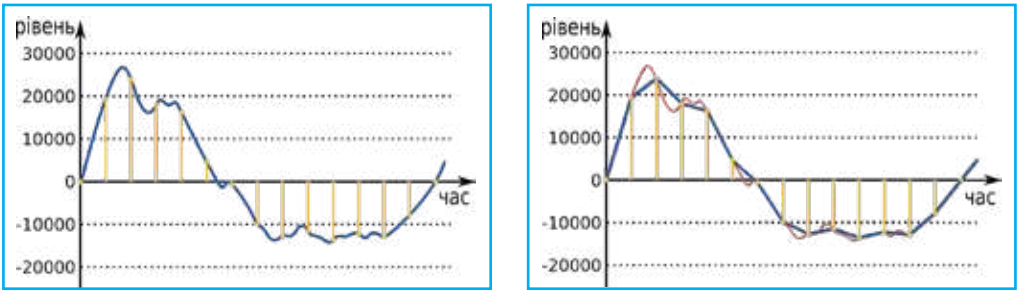


Рис. 18.2

Щоб закодувати звук, потрібно описати числами форму побудованого графіка. Для цього вимірюємо через однакові проміжки часу рівень сигналу, тобто відстані від горизонтальної осі до точок графіка (на рис. 18.3 — жовті відрізки).



а б

Рис. 18.3

Процес кодування неперервного сигналу послідовністю окремих значень його рівня називається **дискретизацією сигналу**.

Результати вимірювання кодуються двійковим способом. Якщо для зберігання одного значення рівня сигналу виділити 2 байти (тобто 16 бітів), то за їх допомогою можна закодувати $N = 2^{16} = 65\,536$ різних значень рівня — від $-32\,768$ до $32\,767$. Таким чином, чим більше бітів виділено для кожного значення, тим точніше можна описати форму графіка.

2 Фрагмент звукової хвилі (див. рис. 18.3, *а*) можна закодувати такою послідовністю чисел: 0, 19 630, 24 274, 18 180, 16 457, 4672, –735, –9997, –12 928, –11 681, –13 815, –12 444, –13 128, –7928, 0. За цими значеннями можна *приблизно* відновити початкову форму хвилі (синя лінія на рис. 18.3, *б*).

За такого способу кодування неминучими є певні втрати форми початкової хвилі, а отже, і якості звуку. Адже за час між двома вимірюваннями рівень може різко змінитись і цю зміну не буде зафіксовано (див. червоні ділянки графіка, наведені на рис. 18.3, *б*). Це означає, що збережений звук трохи відрізнятиметься від того, який записували.

Отже, щоб під час кодування точніше зберегти форму звукової хвилі, потрібно вимірювати рівень сигналу якомога частіше.

Частота дискретизації — це число вимірювань рівня сигналу, які виконуються за 1 с; вимірюється в герцах (Гц).

Збільшення частоти дискретизації призводить до збільшення обсягу даних, які потрібно зберегти або передати.

Під час розкодовування за послідовністю чисел відтворюється форма звукової хвилі, після чого сигнал подають на посилювач звуку й далі, наприклад, на звукові колонки.

Під час телефонної розмови (рис. 18.4) іноді погіршується якість звуку. Це означає, що через перевантаження ліній, погіршення передавального каналу чи з інших причин знижується частота дискретизації. Завдяки цьому вдається зменшити обсяг даних, які передаються, і не переривати розмову.

Рис. 18.4

3 Для передавання людського мовлення достатньо виконувати 8000 вимірювань на секунду (частота дискретизації 8000 Гц). На звукових дисках стандарту CD-DA застосовують частоту дискретизації 44 100 Гц, а DVD-Audio — до 192 000 Гц.

Формати аудіофайлів

Для зберігання оцифрованого звуку використовують різні алгоритми стиснення, а отже, й формати аудіофайлів:

Піктограма	Формат	Особливості
	AIFF, WAV	Числовий код звуку не стискається, тому файли мають найбільший обсяг (подібно до графічного формату BMP)
	FLAC, APE	Дані стискаються без втрати якості, тобто потім можна повністю відновити початковий код
	Ogg, MP3	Стиснення зі втратами якості через спотворення найвищих частот. Початковий код відновити неможливо

Відтворення звуку в програмах

Закодований звук із файлу можна відтворити в програмі мовою Python. Найпростіше це зробити за допомогою функції playsound (із модуля playsound, playsound2 або playsound3).

4 Після запуску програми буде відтворено вміст файлу Гарна музика.mp3, якщо він є в одній папці з програмою:

```
from playsound import playsound          # Імпортуємо функцію
playsound('Гарна музика.mp3')            # Починаємо відтворення
```

Якщо ж файлу немає, то у вікно консолі буде виведено повідомлення про помилку.

Після виклику функції playsound звук із файлу відтворюється до кінця, а потім продовжується виконання програми. Для повноцінного керування відтворенням звуку (пауза, примусова зупинка) слід використовувати інші модулі.

5 Складемо програму для озвучення введеного числа. Для цього нам знадобляться звукові файли з назвами цифр (0.mp3, 1.mp3, 2.mp3 і т. д.):

```
from playsound import playsound
N = input()
for C in N:              # Перебираємо символи рядка
    S = C + '.mp3' # Формуємо ім'я файлу з символу і розширення
    playsound(S)     # Відтворюємо звук
```

Якщо запустити програму й увести ціле число, то для кожної його цифри буде відтворено звук із відповідного файлу.

Як бачимо, кодування, збереження та подальше використання звукової інформації значно розширили можливості людини.

Питання для самоперевірки

1. Що таке цифровий звукозапис?
2. Поясніть принцип кодування звуку.
3. У чому полягає дискретизація звукового сигналу?
4. Якими форматами аудіофайлів ви користуєтеся?
5. Як запрограмувати відтворення звукового файлу?
6. Поясніть, як виконується програма з прикладу 4.

Вправа 18

Скласти програму з використанням аудіофайлів.

1. Відтворіть на комп'ютері музичний аудіофайл.
2. Засобами медіаплеєра і/або операційної системи відшукайте й запишіть докладні дані про цей файл.
3. Складіть програму для озвучення абеткою Морзе введеного рядка великих англійських букв. Завантажте звукові файли з Вікісховища за вказівкою вчителя / вчительки.
4. Скориставшись наданими вам файлами зі звуками нот (папка notes*), складіть програму для відтворення мелодії, закодованої рядком із назвами нот (наприклад, "до мі соль мі до").

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107812).

§ 19. КОДУВАННЯ ВІДЕОДАНИХ. ФОРМАТИ ВІДЕОФАЙЛІВ

Надзвичайно багато інформації з навколишнього середовища людина сприймає за допомогою зору, тому саме відеоматеріали є найбільш інформативними. Розгляньмо, як кодують відеодані.

Принципи кодування відео

Як ви вже знаєте, рухоме зображення — відео — отримують, демонструючи почергово окремі кадри. Отже, щоб закодувати відео, потрібно закодувати зображення кожного його кадру приблизно так, як це описано для зображень у § 3.

Оскільки для того, щоб рухи на екрані виглядали плавними, потрібно демонструвати 24 кадри на секунду або навіть більше, то обсяг відеоданих може бути досить значним.

Здебільшого всі закодовані кадри фільму зберігають в одному файлі, разом із закодованими даними звукового супроводу та іншими даними. Проте іноді кожен кадр відео зберігають в окремому графічному файлі (наприклад, формату PNG).

Мультфільм тривалістю 1 год 40 хв (6000 с) за частоти кадрів 24 кадри/с складається з $6000 \cdot 24 = 144\,000$ кадрів.

За розмірів кадру 720×400 пікселів і глибини кольору 3 байти обсяг даних одного кадру становить:

$720 \cdot 400 \cdot 3 = 864\,000$ байтів.

Отже, загальний обсяг відеоданих мультфільму становить:

$144\,000 \cdot 864\,000 = 124\,416\,000\,000$ байтів $\approx$

≈ 120 ГіБ.

Вам цікаво, як ці дані разом зі звуковою доріжкою вміщаються в значно меншому файлі обсягом 1,6 ГіБ? Поміркуємо разом.

Як і у випадку графічних або звукових даних, оцифровані відеодані стискають. Існують різні методи зменшення обсягу файлу до прийнятної для використання величини.

Розглянемо два методи стиснення даних докладніше.

1. Можна стиснути кожен кадр так само, як це робиться з фотографіями у форматах JPG (зі втратою якості) або PNG (без втрати якості). У такому разі кожний кадр під час перегляду фільму потрібно буде розпакувати, що вимагає додаткових обчислень, тобто збільшує навантаження на процесор.
2. Можна повністю зберігати лише ключові кадри, тобто ті, які дуже відрізняються від попередніх. Тоді з кадрів, що йдуть після ключового, достатньо зберегти лише фрагменти, які відрізняються від попереднього кадру. Це одночасно зменшить як обсяг файлу, так і обсяг обчислень, необхідних для його відтворення.

Під час *скринкастингу* передається через інтернет або записується до файлу те, що відбувається на екрані комп'ютера. Якщо ведучий демонструє слайди презентації та водночас щось пояснює, зображення може не змінюватися протягом декількох секунд. А під час переміщення вказівника миші змінюється тільки маленька ділянка екрана поблизу нього. У такому разі другий метод є дуже ефективним.

На практиці ж у різних форматах відеофайлів використовують не тільки ці, а й інші методи та їх різноманітні комбінації.

Формати відеофайлів

Як нам уже відомо, відеодані являють собою набір кодів статичних (нерухомих) зображень, призначених для почергового виведення з певною частотою кадрів. Для стиснення відеоданих і відтворення відео використовують спеціальні програми — **кодеки** (від слів *кодування* — *декодування*).

Формат відеофайлу визначається методом стиснення. Більшість відеофайлів можуть містити статичні зображення, аудіо, відео різних форматів. Такі файли називають **медіаконтейнерами**.

Розгляньмо наведені в таблиці формати відеофайлів:

Формат	Опис
AVI	Медіаконтейнер, розроблений спочатку для ОС Windows. Відтворюється на смартфонах, планшетах тощо
WMV	Формат компанії Microsoft, може містити аудіо- й відеодані. Відтворюється плеєрами MPlayer, Windows Media Player та ін.
MOV	Формат компанії Apple для QuickTime-плеєра, може містити відео, анімацію, графіку. Підтримує будь-які аудіо- та відеокодеки
MPG	Часто вживаний формат завдяки невеликим втратам під час стиснення та сумісності з багатьма програмами й пристроями
MP4	Формат для зберігання відео, аудіо та деяких видів анімації. Підтримує багатоканальний звук. Часто використовується в портативних пристроях і для передавання відеофайлів в інтернеті
MKV	За можливостями перевищує AVI, може зберігати інформацію про титри, навігаційні меню тощо. У контейнер MKV можна впакувати відео й аудіо найрізноманітніших форматів
WebM	Відкритий формат (не вимагає ліцензії) компанії Google для вебмедіа. Спрощений аналог MKV із функціями, потрібними для мережі. Підтримується більшістю сучасних браузерів і медіаплеєрів

Існує значно більше форматів відео- та аудіофайлів, ніж ми розглянули. Їхня кількість зростає, оскільки вони створюються для різних пристроїв і мають різне призначення.

Окрім того, технології опрацювання мультимедійних даних розвиваються, і це також сприяє відмові від застарілих форматів файлів та появі нових.

Виведення відео в програмах

Для перегляду відео за допомогою програм-програвачів його зберігають в одному файлі. Якщо ж, наприклад, потрібно зробити анімаційного персонажа у власній ігровій програмі, то зручніше мати відеофрагмент у вигляді окремих графічних файлів.

Пізніше, вивчаючи основи програмування, ви дізнаєтеся, як у власних програмах вивести зображення, а також, як послідовно вивести низку зображень, щоб отримати анімаційний ефект.

Як бачимо, розуміння принципів отримання рухомого зображення із сукупності нерухомих і цифрового кодування відеоданих дає змогу не лише зберігати відеоматеріали для перегляду, але й створювати цікаві власні програми з анімаційним оформленням.

Питання для самоперевірки

1. Як отримують рухоме зображення?
2. Навіщо «стискають» цифрове відео?
3. Назвіть відомі вам формати відео.
4. Чому існує велика кількість відеоформатів?

Вправа 19

Дослідити файл із відеоданими.

1. Знайдіть в інтернеті відеофільм О. Довженка «Земля» (або інший за вказівкою вчителя / вчительки) для завантаження. Запишіть формат і розмір відеофайлу; тривалість фільму.
2. Розрахуйте обсяг відеоданих без упакування за прикладом 1.
3. Поясніть, чому саме цей формат відеофайлу використовується для передавання відео в інтернеті.
4. Скориставшись чат-ботом, підготуйте повідомлення: чому відеоігри іноді мають великий обсяг встановлювального файлу.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107813).

ПРАКТИЧНА РОБОТА 8
ВИКОРИСТАННЯ АУДІОДАНИХ У ПРОГРАМАХ МОВОЮ PYTHON

Завдання: створити програму Колекція звуків для доступу до звукових файлів та їх прослуховування; перегляд колекції має здійснюватися через систему текстових меню.

```
-=#[ КОЛЕКЦІЯ ЗВУКІВ ]#=-

1. Звуки англійської абетки Морзе
2. Основні звуки першої октави

0. Закінчити роботу

Введіть номер колекції: 2

Назва колекції: Ноти
1 do.mp3
2 re.mp3
3 mi.mp3
Введіть номер файлу: 3
```

rnk.com.ua/110469

Обладнання: комп'ютер із середовищем програмування IDLE; папка sounds зі збірками звукових файлів формату MP3*, кожна збірка — в окремій папці; доступ до папки — за QR-кодом або посиланням rnk.com.ua/110469.

▶ Теоретичні відомості

Засоби мови Python, необхідні для створення програми, вже розглянуті в 7 і 8 класах. Для роботи знадобляться:

- модуль playsound (див. §18);
- списки, в яких зберігатимемо інформацію про окремі папки зі збірками звукових файлів, у такому форматі:
  <Назва змінної> = ['<Назва збірки>', '<Шлях до файлів>', "<ім'я файлу1>", "<ім'я файлу2>", ...]
- власні функції (§39, 40, 7 клас) для доступу й програвання файлів; розгалуження, цикли, відомі вам елементи мови.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Імпортуйте функцію для відтворення звуків із файлів:
```
from playsound import playsound
```
2. Створіть списки для кожної зі збірок файлів (у нас їх дві):
```
album1 = ['Абетка Морзе', './sounds/morse/', 'A.mp3', 'B.mp3', 'C.mp3']
album2 = ['Ноти', './sounds/notes/', 'do.mp3', 're.mp3', 'mi.mp3']
```
3. Створіть функцію для відтворення файлу зі збірки.
Параметри функції: album — список з описом збірки; number — номер файлу в збірці.
```
def play_from_album(album, number):
    playsound(album[1]+album[number+1])
```
4. Створіть функцію для виведення на екран інформації про збірку у вигляді нумерованого списку.
Параметр функції: album — список з описом збірки.
```
def show_album(album):
    print('\nНазва колекції: ', album[0])
    for i in range(2, len(album)):
        print(i–1, album[i])
```
5. Створіть функцію для введення з клавіатури номера файлу та перевірки його правильності.
Параметр функції: maxN — найбільший допустимий номер.
```
def select_file(maxN):
    n = int(input('Введіть номер файлу: '))
    while n<1 or n>maxN:
        n = int(input('Хибний номер. Спробуйте ще раз: '))
    return n
```
6. Створіть функцію для опрацювання вибраної збірки за допомогою створених раніше функцій: виведення списку, вибору номера й відтворення вибраного файлу.
```
def process_album(album):
    show_album(album)
    number = select_file(len(album)–2)
    play_from_album(album, number)
```

7. Створіть основну програму, в якій виводиться список збірок, а після вибору однієї з них — викликається функція для її опрацювання. Введення замість номера збірки значення 0 має закінчувати роботу програми.
Хибні номери мають спричиняти повторне введення номера.

```
section = –1
while section != 0:
    while section<0 or section>2:
        print('\n-=#[ КОЛЕКЦІЯ ЗВУКІВ ]#=–\n')
        print('1. Звуки англійської абетки Морзе')
        print('2. Основні звуки першої октави (піаніно)')
        print('\n0. Закінчити роботу')
        section = int(input('\nВведіть номер колекції: '))
    if section > 0:
        if section == 1:
            process_album(album1)
        elif section == 2:
            process_album(album2)
        section = –1

print('Бажаю успіхів! :)')
```

8. Запустіть програму, виправте за потреби помилки. Збережіть програму у файлі Практична робота 8.
Перевірте роботу в різних ситуаціях: введення неправильних номерів збірки або файлу тощо.
9. Доповніть списки іменами файлів, які є в папках morse і notes, але відсутні в програмі. Перевірте роботу програми.
10. Додайте до програми ще одну збірку — з папки notes_b, яка містить записи звучання на баяні всіх звуків (основних і альтерованих) першої октави.
Перевірте роботу програми.

Зробіть висновок: які можливості відкриває винайдення способу кодування й зберігання звукових файлів.

§ 20. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПРАЦЮВАННЯ ЗВУКУ

Для роботи зі звуком використовують різноманітні програми. Одні призначені для захоплення або відтворення звуку, інші — редагування аудіоданих, змінення форматів аудіофайлів тощо.

Захоплення звуку

Щоб опрацьовувати на комп'ютері звукові дані, отримані із зовнішніх пристроїв, наприклад, мікрофона, електрогітари, синтезатора тощо, їх потрібно перевести в цифрову форму.

Захоплення звуку — це процес перетворення звукового сигналу із зовнішніх джерел у цифрову форму та запису в аудіофайл.

Для **захоплення звуку** слід приєднати до комп'ютера аудіопристрій, приміром мікрофон / електрогітару; запустити програму для записування звуку; виконати мелодію, прочитати перед мікрофоном текст тощо; зберегти записані аудіодані у файл.

Звукові дані у файл записуються для різних потреб. Безпосередньо для запису звукових даних (наприклад, для того щоб працювати з ними пізніше) можна скористатися найпростішими програмами. Їх існує багато для різних ОС — із назвами Диктофон, Звукозапис, Записувач звуку або подібними.

Аудіоредактори

Якщо звук потрібно не лише записати у файл, а й опрацювати, варто скористатися досконалішою програмою.

Аудіоредактор (звуковий редактор) — це програма для захоплення звуку та опрацювання аудіоданих.

За допомогою аудіоредактора можна записувати звук із різних джерел, накладати різноманітні ефекти, обрізати аудіофайли, монтувати запис із декількох файлів, створювати звуки тощо.

Аудіоредактор Audacity

Рис. 20.1

Аудіоредактор Audacity (рис. 20.1) належить до вільного програмного забезпечення (ліцензія GNU GPL) і може бути завантажений із офіційного сайту програми.

Аудіоредактор Audacity надає низку функцій.

Це запис звуку з мікрофона або іншого джерела; імпортування звукових даних із файлів різних форматів; редагування звукових даних; одночасна робота з багатьма звуковими доріжками; застосування до вибраних фрагментів різноманітних ефектів; експортування результату в потрібний формат (WAV, MP3 та ін.).

Після запуску Audacity відкривається головне вікно (рис. 20.2). Детальніші налаштування та команди доступні в меню програми.

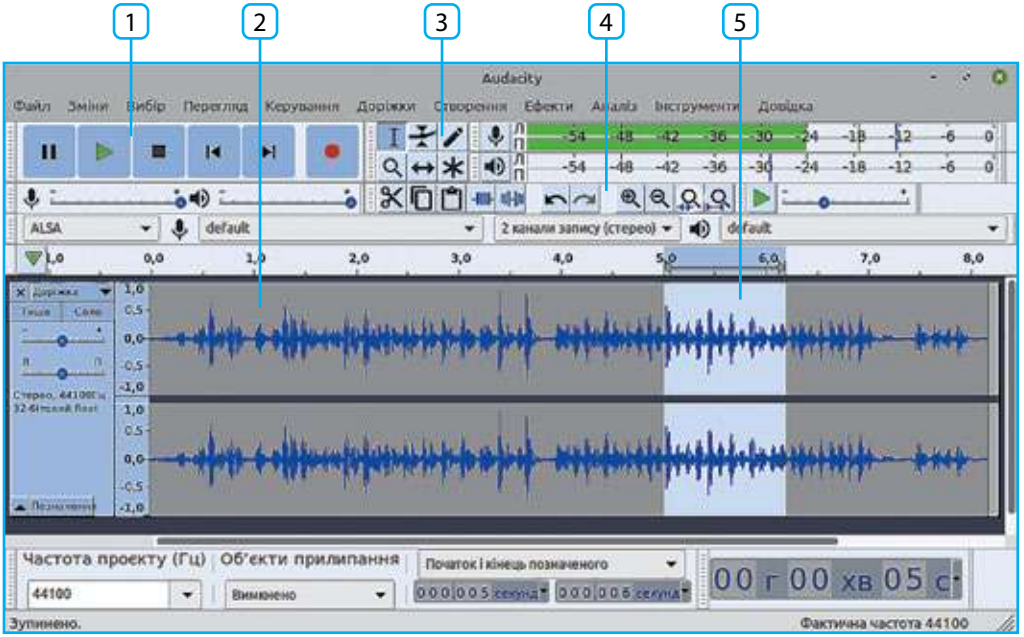


Рис. 20.2

Тут: 1 — панель Керування; 2 — звукова доріжка; 3 — панель Інструменти; 4 — панель Зміни; 5 — аудіофрагмент (блакитне тло).

Загальне керування програмою

Кнопки панелі Керування нагадують кнопки в медіаплеєрах:

Вигляд	Опис
	Призупиняє запис або відтворення звуку. Для продовження призупиненого процесу слід натиснути кнопку ще раз
	Запускає відтворення виділеного фрагмента або починаючи від поточної позиції, яку задають клацанням на звуковій доріжці, до кінця
	Зупиняє процес відтворення або запису
	Переносить поточну позицію на початок або в кінець звукової доріжки
	Виконує запис звуку з джерела, приєднаного до комп'ютера

На рис. 20.3 подано алгоритм запису звуку за допомогою мікрофона, приєднаного до комп'ютера або вбудованого в нього.

Рис. 20.3

Робота з фрагментами

Фрагмент, над яким потрібно виконати певні операції, виділяють перетягуванням безпосередньо на звуковій доріжці. Якщо на його початку або в кінці рівень не дорівнює нулю, то в цьому місці після виконання операції може з'явитися різкий перепад рівня, який під час прослуховування сприйматиметься як шум.

Щоб цього уникнути, після виділення фрагмента слід натиснути клавішу Z — межі виділення зміститься так, щоб фрагмент починався та закінчувався в точках із нульовим рівнем сигналу, а саме в тих, де хвиля перетинає осьову лінію в напрямку знизу вгору.

На рис. 20.4 показано розташування початку виділеного фрагмента і форму хвилі до та після зменшення рівня. У випадку *а* клавішу Z не натискали, тому з'явився різкий перепад рівня.

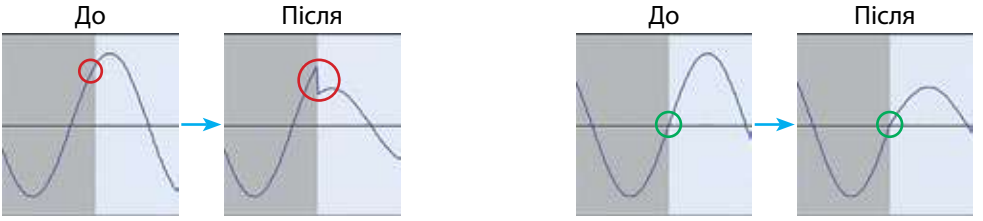


Рис. 20.4

Щоб краще роздивитися хвилю на виділеному фрагменті, потрібно на панелі Зміни натиснути кнопку — масштаб відображення зміниться так, що фрагмент займе всю ширину вікна. Якщо ж натиснути кнопку , то можна буде побачити у вікні всю доріжку.

Для **видалення зайвого фрагмента звукової доріжки** його слід виділити, натиснути клавішу Z, а потім клавішу Delete.

Так само як і фрагмент тексту, звуковий фрагмент можна скопіювати або вирізати в буфер обміну та вставити в інше місце. Зробити це можна за допомогою команд меню Зміни або кнопки панелі з такою самою назвою.

Копія виділеного фрагмента поміщується в буфер обміну командою Копіювати (кнопкою) або Вирізати (кнопкою). В останньому випадку фрагмент із доріжки вилучається, а її частина, що була праворуч від фрагмента, підсовується вліво.

Щоб **вставити фрагмент із буфера обміну**, слід:

1) клацнути на потрібному місці доріжки;
2) натиснути клавішу Z, щоб поточна позиція змістилася до найближчого перетину осі хвилею в напрямку знизу вгору;
3) вибрати команду меню Вставити або натиснути кнопку .

Звукову доріжку в поточній позиції буде розрізано і розсунуто, а в місце розриву вставиться фрагмент із буфера обміну.

Збереження звукових даних

Кінцевий результат роботи слід експортувати в один із форматів, підтримуваних іншими програмами (WAV, OGG, MP3 тощо).

Щоб отримати файл формату MP3, потрібно:

1) вибрати команду Файл → Експорт → Експортувати як MP3;
2) унизу вікна Експорт звукових даних вибрати параметри кодування формату MP3 (рис. 20.5).
   Якість (бітова частота) 192 Кб/с вважається високою. За використання змінної бітової частоти різні ділянки звукової доріжки кодуються з різною частотою дискретизації. Звучання файлу в цілому сприймається якіснішим, ніж файлу такого ж розміру, але зі сталою бітовою частотою;
3) вибрати папку та назву файлу й натиснути Зберегти.

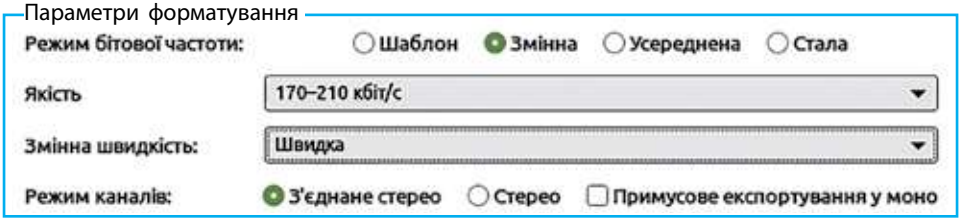


Рис. 20.5

Залежно від формату, в який експортується файл, можуть додатково з'являтися діалогові вікна для уточнення параметрів.

Отже, аудіоредактор дає змогу не лише захопити звук, а й опрацювати аудіодані та експортувати результат у файл бажаного звукового формату, придатний для подальшого використання в інших програмах.

Питання для самоперевірки

1. У чому полягає процес захоплення звуку?
2. Яке призначення аудіоредактора?
3. Опишіть процес запису звуку з мікрофона.
4. Як прибрати зі звукової доріжки зайві фрагменти?
5. Поясніть, як перемістити виділений фрагмент в інше місце звукової доріжки.
6. Опишіть послідовність створення файлу формату MP3 із записаної звукової доріжки.

Вправа 20

Створити аудіофайл за допомогою аудіоредактора Audacity.

1. Створіть у власній папці папку Медіа.
2. Якщо потрібно, підключіть мікрофон та навушники до комп'ютера (рис. 20.6). Запустіть аудіоредактор.
3. Натисніть кнопку Запис. Промовте в мікрофон свої прізвище, ім'я, клас. Зупиніть запис.
4. Прослухайте запис. Якщо якість звуку вас не влаштовує, повторіть запис, змінивши розташування мікрофона (ближче або далі), його нахил тощо.
5. Видаліть зайві звукові фрагменти (паузи на початку та в кінці).

Рис. 20.6

6. Експортуйте запис у файл з іменем Вправа 20.mp3 у папці Медіа. Завершіть роботу програми, вимкніть мікрофон. Прослухайте запис. Визначте його частоту дискретизації.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107814).

§ 21. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПРАЦЮВАННЯ ВІДЕО

Ми переглядаємо відео на смартфоні або комп'ютері за допомогою спеціальної програми — медіаплеєра. Проте з відеоданими на комп'ютері можна виконувати значно більше різних дій, використовуючи інше програмне забезпечення та пристрої.

Захоплення відео

Процес перетворення відеосигналу із зовнішнього пристрою в цифрову форму з подальшим записом у відеофайл називають захопленням відео.

За джерелом і типом відеосигналу розрізняють захоплення відео з аналогового пристрою, цифрових відеокамер, екрана монітора. Ознайомимося зі схемою (рис. 21.1).

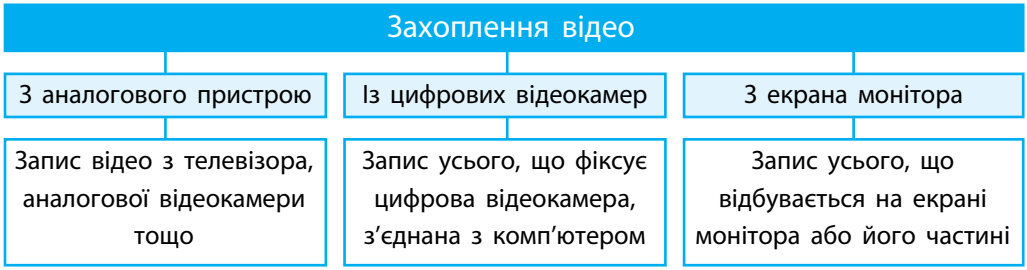


Рис. 21.1

Для оцифрування аналогового відеосигналу є спеціальні пристрої — **плати відеозахоплення**. Серед програм для захоплення відео — OBS Studio, Icecream Screen Recorder, CamStudio тощо. Вони дають змогу захоплювати не лише відео з екрана та інших джерел, а й звук із мікрофона, колонок, зберігати відеофайли у форматах, підтримуваних медіаплеєрами. Редактор презентацій PowerPoint також має засоби захоплення відео і звуку з екрана.

Програма для захоплення відео OBS Studio

OBS Studio — вільна багатоплатформна програма з відкритим кодом для запису відео або трансляції з багатьма функціями. Розгляньмо елементи інтерфейсу вікна програми (рис. 21.2).

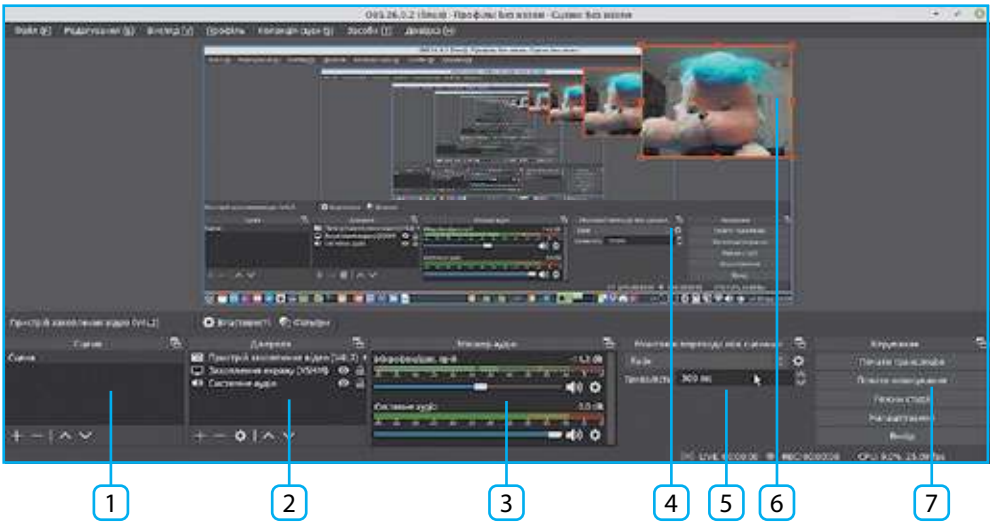


Рис. 21.2

Тут: 1 — перелік сцен; 2 — перелік джерел поточної сцени; 3 — регулятори гучності аудіовиходу й мікрофона; 4 — зразок захоплення з екрана; 5 — налаштування переходів між сценами; 6 — зразок захоплення з камери; 7 — загальне керування програмою.

Програма OBS Studio дає можливість захоплювати одночасно відео з екрана та вебкамери; захоплювати звук із мікрофона та програм; довільно компонувати сцену, тобто розміщувати на екрані джерела відео; створювати декілька сцен і перемикатися між ними під час запису або трансляції; здійснювати пряму трансляцію відео на популярні відеосервіси (YouTube, Instagram тощо), для чого в програмі є готові набори налаштувань; експортувати записане відео в багато різних відеоформатів.

Усіма функціями OBS Studio можна користуватися безоплатно й без будь-яких обмежень.

Щоб отримати відео з таким макетом екрана, як на рис. 21.2, потрібно додати джерела Пристрій захоплення відео і Захоплення екрану. Для запису звуку програм і мікрофона слід додати джерело Захоплення аудіовиводу.
Якщо ж потрібен лише звук мікрофона, то замість останнього джерела слід вибрати джерело Захоплення аудіовходу.

Дізнаймося, які кнопки містить вікно програми OBS Studio. Кнопка відкриває діалогове вікно для вибору джерела зі списку; кнопка — вилучає зі списку вибране джерело; кнопки і — переміщують вибране джерело вгору або вниз.

Засоби перетворення відеоформатів

Різні пристрої та програми накладають різні вимоги до формату даних. Наприклад, для перегляду відео на великому екрані бажано мати файл із великими розмірами кадру, а для смартфона вистачить файлу з меншими розмірами кадру, а отже, й обсягом.

Конвертер — програма, призначена для перекодування медіафайлів (відео- та / або аудіофайлів) з одного формату в інший.

Конвертер відеофайлів HandBrake

HandBrake — вільна багатоплатформна програма, інструмент для перекодування відеофайлів з одного формату в інший.

Розгляньмо роботу з відеофайлом (рис. 21.3, с. 134; використано кадр із анімаційного фільму Sintel, випущеного Blender Institute).

Для **конвертування відеофайлу** за допомогою HandBrake слід:
1) натиснути кнопку Open Source (Відкрити джерело) і в діалоговому вікні вибрати відеофайл для перетворення;
2) задати параметри відеофайлу, який потрібно отримати;
3) в поле Save As (Зберегти як) увести ім'я файлу, у списку To: (У:) вибрати папку, куди буде збережено перекодоване відео;
4) натиснути кнопку Start (Почати).

Далі розпочнеться процес перекодування, який триватиме досить довго (його перебіг показує індикатор). Після цього в папці з'явиться файл, який можна переглянути й оцінити результат.

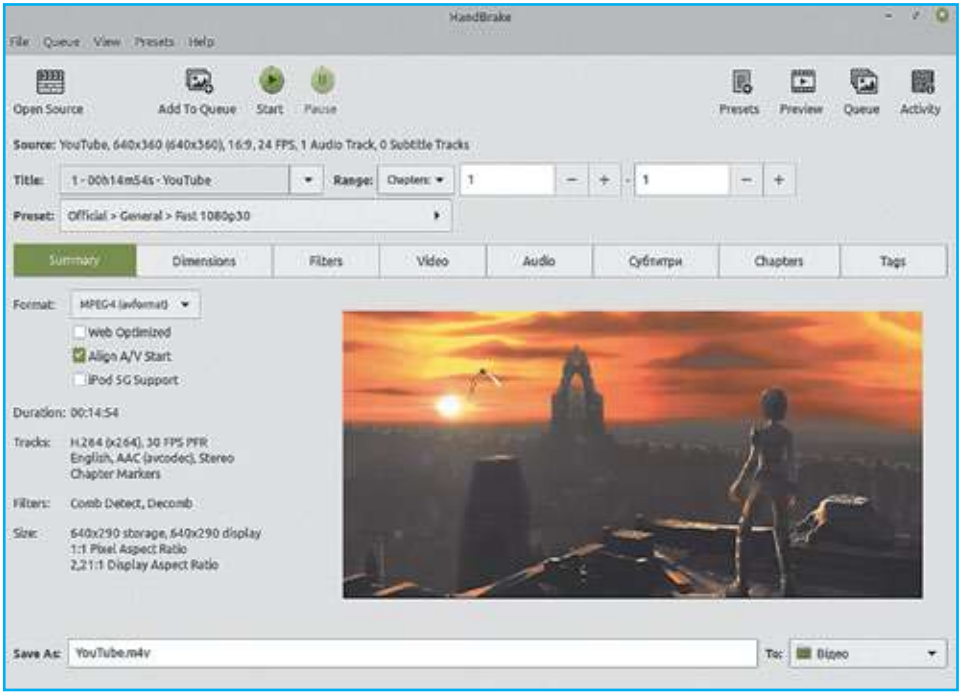


Рис. 21.3

Із розвитком хмарних інтернет-технологій з'явилася можливість конвертування аудіо- та відеофайлів за допомогою онлайнових сервісів, відшукати які допомагають пошукові системи.

Результатом розвитку ідей штучного інтелекту (ШІ) стала поява в інтернеті в 2020-х роках низки сайтів, де ШІ надає різноманітні послуги. Є серед них і зорієнтовані на роботу зі звуком і відео. Тут можна перетворити звуковий файл із голосом людини на текст, озвучити власний текст голосом популярного актора, відокремити голос співака чи співачки від музичного супроводу, створити відеофрагмент за словесним описом і багато іншого.

! Якість матеріалів, створених за допомогою ШІ, не завжди висока, проте вона поступово зростає завдяки так званому *машинному навчанню* (це галузь ШІ, що вивчає методи побудови алгоритмів, здатних навчатися).

Питання для самоперевірки

1. Що таке захоплення відео?
2. Із яких пристроїв можна захоплювати відео?
3. Які можливості надає програма OBS Studio?
4. Якими програмами для перегляду відеофайлів ви користуєтеся? Опишіть, які можливості вони надають.
5. Коли виникає потреба змінити формат відеофайлу?
6. Для чого призначені програми-конвертери?

Вправа 21

Створити відеофайл, захопивши відео з екрана комп'ютера, за допомогою програми захоплення відео OBS Studio.

1. Виберіть тему малюнка (наприклад, «Небо України», «Моя вулиця» або на свій розсуд).
   Запустіть графічний редактор і підготуйте його до малювання (налаштуйте розмір полотна, форму пензля, кольори тощо).
2. Запустіть програму OBS Studio. Додайте до сцени джерела Захоплення екрану і, за наявності вебкамери, Відеопристрій.
3. Розташуйте джерела відео на макеті екрана на свій розсуд.
   Підказка. Відео з вебкамери краще розташувати в одному з кутів, де немає важливих для виконання малюнка елементів керування.
4. Увімкніть запис, перейдіть до вікна графічного редактора й виконайте малюнок.
5. Зупиніть запис і збережіть файл із назвою Вправа 21 у папці Медіа. Завершіть роботу програми.
6. За допомогою медіаплеєра перегляньте відеофільм і обговоріть його з однокласниками й однокласницями.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107815).

§ 22. ПОБУДОВА АУДІО- Й ВІДЕОРЯДУ

Відеофрагменти, отримані під час захоплення відео, зняті на цифрову камеру, записані з екрана, можна використати для створення відеоролика або відеофільму за допомогою відеоредактора.

Відеоредактор — програма для створення, редагування та монтування відеофайлів з окремих відеофрагментів.

Існує низка популярних відеоредакторів, логотипи деяких зображено на рис. 22.1. Серед них є вільні програми, а також поширювані за власницькими ліцензіями.

Lightworks

Pinnacle Studio

OpenShot

Sony Vegas Pro

Рис. 22.1

Відеоредактор OpenShot

Відеоредактор OpenShot належить до вільного програмного забезпечення (ліцензія GNU GPL). Останню версію програми можна отримати на офіційному сайті.

Розгляньмо основні елементи інтерфейсу програми OpenShot версії 2.5.1 (рис. 22.2). Вікно програми поділено на декілька ділянок, розміри яких можна змінювати, перетягуючи межі. У меню Вигляд → Перегляди можна вибрати між простим і розширеним поданням вікна.

Основні дії з об'єктами виконуються на монтажному столі, розташованому в нижній частині вікна програми. На доріжках монтажного столу формується послідовність кліпів (фрагментів відеоряду та звукового супроводу, титрів) відповідно до задуму, змінюються тривалості кліпів та інші властивості, налаштовуються відеоефекти та відеопереходи тощо.

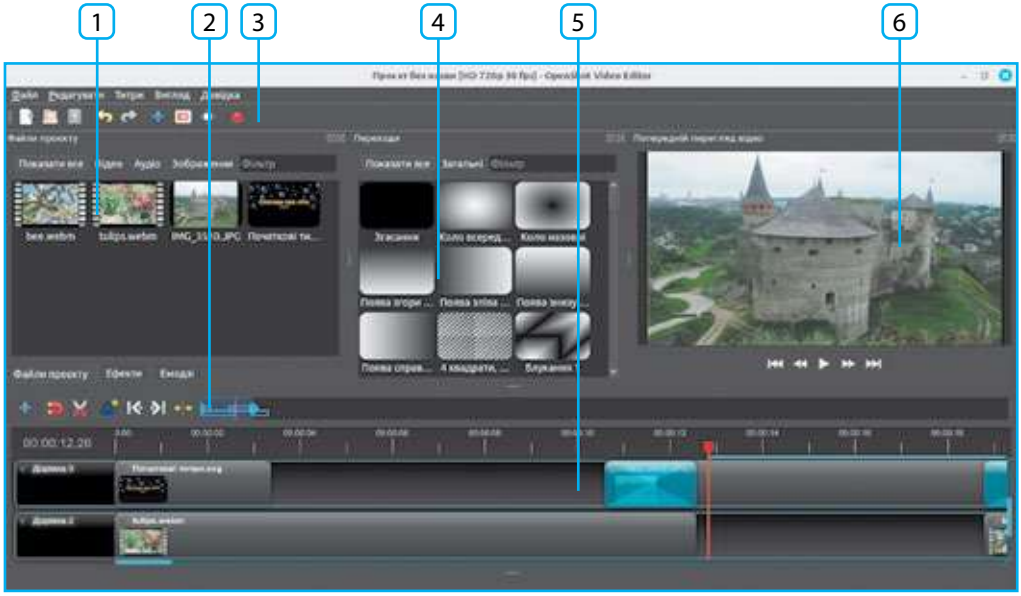


Рис. 22.2

Тут: 1 — файли проєкту; 2 — засіб керування масштабом; 3 — панель інструментів; 4 — переходи; 5 — монтажний стіл; 6 — екран перегляду.

Процес створення відеофільму

Процес розробки, створення й редагування відеофільму за допомогою відеоредактора OpenShot складається з декількох основних кроків. А саме:

1) імпортування аудіо- та відеофрагментів у проєкт;
2) редагування аудіо- та відеофрагментів;
3) побудова відеоряду;
4) налаштування переходів та ефектів;
5) підготовка та додавання титрів;
6) налаштування звукового супроводу;
7) експорт відеофільму у відеофайл.

Більшість кроків не обов'язково виконувати саме в такій послідовності, але в процесі роботи слід приділити увагу кожному.

1. Імпортування аудіо- та відеофрагментів у проєкт

На цьому кроці користувач готує добірку зображень, відео- та аудіофрагментів, із яких формуватиметься відеофільм. Процес додавання до проєкту файлів із зовнішніх джерел називають **імпортом**. Підготовлені файли потрібно імпортувати за допомогою відповідної команди меню Файл або просто перетягнувши з вікна папки в ділянку Файли проєкту вікна програми.

2. Редагування аудіо- та відеофрагментів

Наступним кроком роботи є редагування (розділення, обрізання тощо) відео- й аудіофрагментів. Для цього їх потрібно перетягнути з ділянки Файли проєкту на доріжки монтажного столу.

Відео-, аудіофрагмент або зображення, розташоване на доріжці монтажного столу, називають кліпом.

На рис. 22.3 показано кліп на доріжці монтажного столу та основні елементи керування, пов'язані з доріжкою та кліпом.

Рис. 22.3

Тут: 1 — кнопка виклику меню доріжки; 2 — індикатор поточної позиції; 3 — назва доріжки; 4 — кнопка виклику меню кліпу; 5 — кліп; 6 — часова шкала; 7 — «відтворювальна головка» (курсор); 8 — маркер.

Відеоредактор OpenShot дозволяє розділити довгий кліп на два коротші, а також швидко обрізати початок або кінець кліпу. Для **розділення кліпу на дві частини** потрібно натиснути кнопку інструмента Лезо і натиснути на тому місці кліпу, де має бути розріз. Щоб вимкнути Лезо, слід натиснути кнопку ще раз.

Кліпи деяких відеоформатів можуть, із різних причин, розрізатися неправильно. Якщо виникне така проблема, то слід вилучити файл із проєкту, конвертувати відео в підтримуваний формат, наприклад MP4 (H.264), і знову імпортувати до проєкту.

Щоб **обрізати початок або кінець кліпу**, потрібно перетягнути на доріжці лівий або правий край його рамки (червоні лінії на рис. 22.3) в потрібне місце, визначаючи його візуально на екрані перегляду або за часовими параметрами на шкалі часу вгорі монтажного столу. Обрізану частину кліпу можна відновити перетягуванням країв рамки в протилежний бік.

! Неможливо збільшити тривалість кліпу понад його повну тривалість. Це не стосується зображень і титрів: для них можна задати яку завгодно тривалість показу.

Під час роботи над кліпом можна кадр відео, який видно на екрані перегляду, зберегти у файл формату PNG (меню Файл → → Збереження поточного кадру). Цей файл можна додати до проєкту й використати під час монтажу для оформлення відеофільму.

▶ 3. Побудова відеоряду

Наступним кроком є побудова аудіо- та відеоряду.

Ознайомимося з призначенням кнопок монтажного столу:

Кнопка	Призначення
	Додати доріжку
	Увімкнути (вимкнути) прилипання кліпів до маркерів, кінців інших кліпів, курсора під час перетягування
	Інструмент Лезо (див. вище)
	Додати маркер (позначає певну позицію на часовій шкалі)
	Перейти до попереднього або наступного маркера
	Показати курсор

На одній доріжці можна розташувати кілька кліпів. Для загального керування доріжкою достатньо викликати контекстне меню на її назві й вибрати потрібну команду (рис. 22.4). Доріжку можна заблокувати від випадкових переміщень і змін тривалості кліпів. Можливість розрізати кліпи та змінювати їхні властивості залишається.

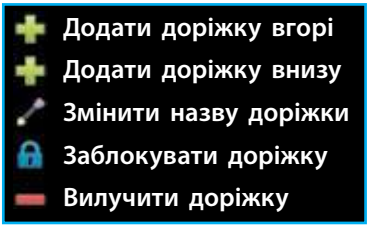


Рис. 22.4

4. Підготовка та додавання титрів

Титри — написи, що з'являються на початку або наприкінці відеофільму, на певному фрагменті чи зображенні тощо.

Відеоредактор OpenShot дає змогу оформити відеоряд за допомогою титрів різних типів: звичайних та анімованих.

Звичайні титри **додають до проєкту** командою меню Титри → → Титри. У діалоговому вікні Титри, яке з'явиться на екрані (рис. 22.5), потрібно виконати такі дії:

1) вибрати шаблон та вписати текст титрів у відповідні поля;
2) змінити, за потреби, шрифт, кольори тексту і тла;
3) увести назву, під якою файл титрів буде збережено в проєкті, і натиснути кнопку Зберегти.

Після цього у вікні Файли проєкту з'явиться піктограма файлу формату SVG із зазначеною назвою, в якому збережено створені титри. Файл слід додати на доріжку так само, як це робиться з кліпами, в тому місці відеоряду, де потрібно показати титри.

Якщо у вікні Титри натиснути кнопку Скористатись додатковим редактором, то для детальнішого редагування титрів відкриється вікно графічного редактора (наприклад Inkscape).

Ефектні анімовані титри можна створити, якщо на комп'ютері встановлено редактор тривимірної графіки Blender, а в OpenShot зроблено відповідні налаштування. Після уведення в діалоговому вікні тексту та інших параметрів титрів і подальшого опрацювання цих даних комп'ютером (дещо тривалого) до файлів проєкту додається відеофрагмент з анімованими титрами.

Зверніть увагу, що час відтворення титрів коригується так само, як і час відтворення кліпів або переходів — перетягуванням меж на часовій шкалі.

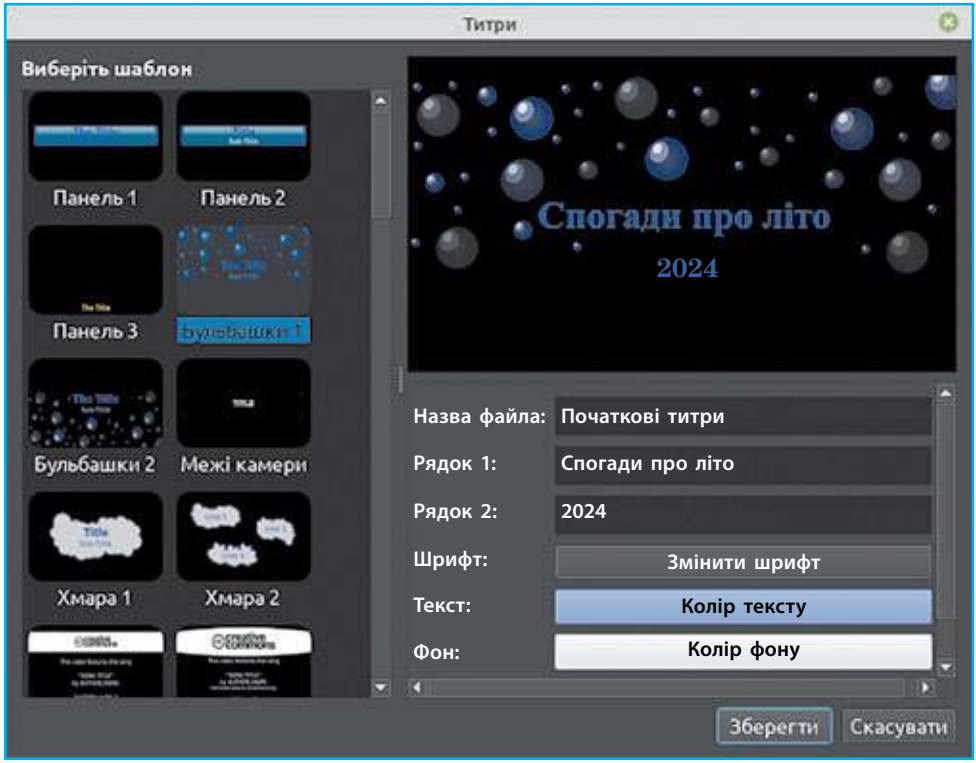


Рис. 22.5

Зберігання проєкту відеофільму

Зазвичай робота над відеофільмом потребує багато часу, тому незавершений проєкт потрібно зберегти для подальшої роботи.

Під час зберігання проєкту створюється файл із розширенням .osp, який містить інформацію про всі файли та проєкт у цілому, але не містить відео- та аудіоданих. Тому, коли користувач відкриє файл проєкту на іншому комп'ютері, обов'язково виникнуть проблеми через відсутність потрібних аудіо- і відеофайлів,

зображень тощо. Щоб уникнути такої ситуації, необхідно зберегти всі потрібні файли та файл проєкту в окремій папці й уже з неї імпортувати їх в OpenShot.

Ви ознайомились із деякими основними кроками створення відеофільму. На цьому процес не завершується, з подальшими кроками ви ознайомитеся далі. Ви дізнаєтеся про те, як покращити сприйняття відеофільму й опублікувати його в інтернеті.

Питання для самоперевірки

1. Опишіть основні елементи вікна відеоредактора OpenShot.
2. Як можна імпортувати матеріали в проєкт?
3. Опишіть порядок роботи з кліпами на монтажному столі.
4. Як збільшити час відтворення зображення?
5. Опишіть послідовність створення й додавання титрів.
6. Що містить файл проєкту з розширенням *.osp?

Вправа 22

Створити відеофільм зі звуковим супроводом.

1. Знайдіть в інтернеті й завантажте файл для звукового супроводу відеофільму в папку, зазначену вчителем / вчителькою. Збережіть у тій самій папці відеофайли, власні або підготовлені вчителем / вчителькою.
2. Запустіть відеоредактор OpenShot, імпортуйте до нього підготовлені аудіо- та відеофайли.
3. Створіть файл титрів із назвою відеофільму.
4. Розташуйте титри та імпортовані відео- й аудіофайли на доріжках монтажного столу.
   Збережіть проєкт (Файл → Зберегти проєкт) із назвою Вправа 22 у відповідній папці.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107816).

§ 23. ОПРАЦЮВАННЯ, ЗБЕРІГАННЯ ТА РОЗМІЩЕННЯ ВІДЕОФІЛЬМУ В ІНТЕРНЕТІ

Після того як основну роботу з матеріалом відеофільму виконано (див. §22), до нього можна додати переходи між кліпами та уточнити часові параметри відеоряду. Далі відеофільм слід експортувати в бажаний відеоформат, опублікувати в інтернеті тощо.

▶ 5. Налаштування переходів між кліпами

Для уникнення різкої зміни одного кліпу іншим, у відеофільмі використовують переходи. Саме додавання переходів між відеофрагментами надає відеофільму більш професійного та естетичного вигляду. Якщо в проєкті кілька доріжок, то на кінцевому відео в кожен момент часу буде видно кадри тієї, яка розташована у списку найвище.

! Плавний перехід від одного кліпу до іншого можна отримати, поступово змінюючи прозорість окремих ділянок кадру.

Якщо два кліпи помістити на одну доріжку так, щоб вони накладалися, то між ними буде автоматично додано перехід із поступовим збільшенням прозорості першого з наявних кліпів (рис. 23.1).

Рис. 23.1

У вікні Переходи можна вибрати інші варіанти переходу від одного кліпу до іншого. Щоб скористатися вибраним варіантом, його піктограму з вікна Переходи слід перетягнути на перший кліп. Опрацьовувані кліпи можуть лежати й на різних доріжках.

Результатом переходу, як і раніше, є збільшення прозорості першого з кліпів. Проте за використання варіантів із вікна Переходи прозорість змінюється не рівномірно на всій площі кадру: швидше зникають ділянки кадру, яким відповідають світліші ділянки спеціального зображення — **маски** (рис. 23.2, с. 144).

Із певних причин доданий перехід може відбуватися «навпаки»: спочатку з'являється кадр із другого кліпу, потім поступово відновлюється кадр із першого кліпу і продовжується другий кліп.

Щоб виправити цю помилку, необхідно змінити напрямок переходу, вибравши в його контекстному меню команду Обернути перехід.

Рис. 23.2

6. Налаштування звукового супроводу

Звукові файли для супроводу відеофільму потрібно імпортувати до проєкту, а потім додавати на доріжки так само, як відеофрагменти. Щоб **змінити тривалість звучання аудіокліпу**, слід перетягнути його правий або лівий край на часовій шкалі.

Усі команди керування звуком зібрано в контекстному меню кліпу. Розгляньмо деякі з них детальніше.

Команда Гучність → Увесь кліп → Поява (увесь кліп) забезпечує плавне наростання гучності від повної тиші на початку до максимального рівня в кінці. Команда Гучність → Кінець кліпу → Зникнення (повільне) забезпечує згасання гучності протягом останніх 3 секунд звучання тощо.

Досить часто відеокліпи мають власні звукові дані, які у відеофільмі можуть бути не потрібні. Щоб **вимкнути звук кліпу**, необхідно в його контекстному меню вибрати команду Гучність → Увесь кліп → Рівень 0%.

7. Експорт відеофільму у відеофайл

Після завершення монтажу відеоряду, звуку та титрів відеофільм експортують у потрібний відеоформат. За командою Файл → Експортувати проєкт → Експортування відео відкривається діалогове вікно, в якому можна задати параметри майбутнього відеофайлу: назву файлу, шлях, формат, розміри кадру, якість.

Після натискання кнопки Експортування відео починається опрацювання проєкту (час залежить від складності, тривалості відео). Після завершення в цільовій папці з'являється відеофайл — і він вартий зусиль, які ви доклали для його реалізації.

Сервіси розміщення аудіо- та відеофайлів в інтернеті

Щоб надати іншим користувачам інтернету можливість переглянути ваш відеофільм, його можна зберегти на хмарному диску (Google Диск, Microsoft OneDrive тощо) і через пошту або соціальні мережі надіслати посилання на файл.

Існує й інший спосіб поширення відео — опублікування на спеціалізованих інтернет-сервісах або в соціальних мережах. Розроблено низку популярних інтернет-сервісів для поширення відео, на яких користувачі / користувачки можуть не лише публікувати, а й коментувати, оцінювати, ділитися ними.

На рис. 23.3 наведено логотипи популярних сервісів, які відрізняються умовами використання та переліком послуг.

YouTube Instagram Primcast Vimeo

Рис. 23.3

Для **розміщення відеофільму на YouTube** потрібно:

1) увійти в обліковий запис YouTube (Google);
2) натиснути кнопку + Створити;
3) вибрати операцію ▶ Завантажити відео (рис. 23.4) і відеофільм для завантаження;
4) увести інформацію про відеофільм;
5) вибрати рівень доступу (приватне; не для всіх; для всіх);
6) вибрати Зберегти або Опублікувати (залежно від рівня доступу) — надійде електронний лист із посиланням, поширюючи яке, можна зробити відеофільм більш доступним для аудиторії.

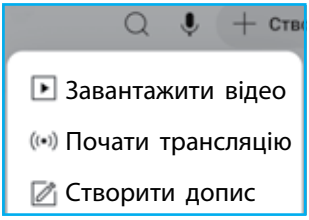


Рис. 23.4

Питання для самоперевірки

1. Яке призначення переходів між відеокліпами?
2. Опишіть різні способи додавання переходів.
3. Як обрізати аудіо- або відеофрагмент?
4. Опишіть процес налаштування звукового супроводу.
5. Як експортувати відеофільм у відеофайл?
6. Які є способи поширення відеофільму через інтернет?
7. Опишіть процес опублікування відеофільму в інтернеті.

Вправа 23

Налаштувати часові параметри відеофільму, додати відеопереходи між кліпами.

1. Запустіть OpenShot і відкрийте проєкт із назвою Вправа 22. Перегляньте відеоряд на екрані перегляду.
2. Додайте переходи між відеокліпами.
3. Перемістіть аудіокліп звукового супроводу так, щоб його відтворення починалося на початку фільму. Збережіть проєкт.
4. Експортуйте відеофільм у файл із назвою Вправа 23-1, вибравши профіль Інтернет і не змінюючи параметрів експорту.
5. Експортуйте відеофільм, вибравши профіль із невеликим розміром (наприклад, 720×576) у файл із назвою Вправа 23-2. Порівняйте розміри файлів.
6. Завантажте файл Вправа 23-2 на хмарний диск і надішліть посилання вчителю / вчительці та однокласнику / однокласниці.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107817).

ПРАКТИЧНА РОБОТА 9
СТВОРЕННЯ ВІДЕОФІЛЬМУ

Завдання: створити відеофільм зі звуковим супроводом і переходами на основі раніше підготовленого проєкту.

Обладнання: комп'ютер із відеоредактором, навушники (звукові колонки).

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть відеоредактор OpenShot. Відкрийте файл Вправа 23.
2. Перегляньте відеофрагменти, збережіть із кожного по кадру.
3. Додайте збережені зображення до проєкту.
4. На монтажному столі розмістіть додані зображення перед відповідними кліпами.
5. Налаштуйте переходи між усіма об'єктами відеоряду.
6. На одному із захоплених кадрів додайте титри з назвою або стислим описом відповідного відеофрагмента.
7. Додайте титри з назвою фільму на початку відеоряду.
8. Додайте наприкінці відеофільму титри, в яких зазначте автора та дату створення відеофільму.
9. Вимкніть власний звук деяких відеокліпів.
10. Синхронізуйте час відтворення аудіо та відеоряду. Для цього змініть тривалість звукового супроводу, час показу картинок, написів або обріжте деякі відеокліпи.
11. Перегляньте відеофільм. Перевірте, щоб він не закінчувався раптово або недоречно, виправте виявлені недоліки.
12. Збережіть проєкт із назвою Практична робота 9.

Зробіть висновок: як створити аудіо- та відеокліпи, додати переходи між відеокліпами й титри, налаштувати часові параметри аудіо- та відеоряду, а також як синхронізувати аудіо- та відеоряд відеофільму.

ПРАКТИЧНА РОБОТА 10
РОЗМІЩЕННЯ ВІДЕОМАТЕРІАЛІВ В ІНТЕРНЕТІ

Завдання: опублікувати відеофільм в інтернеті, надати доступ.
Обладнання: комп'ютер із браузером, відеоредактором, відеоконвертером, з'єднаний з інтернетом.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть відеоредактор OpenShot та відкрийте проєкт із назвою Практична робота 9.
2. Експортуйте відеофільм із назвою Практична робота 10. У вікні експортування виберіть профіль Інтернет.
3. Закрийте відеоредактор і запустіть відеоконвертер.
4. Сконвертуйте відеофільм у формат MP4.
5. Запустіть браузер і відкрийте електронну скриньку.
6. Зайдіть на хмарний диск.
7. Збережіть на хмарному диску відеофільм у форматі mp4.
8. Відкрийте доступ усім, хто має посилання на відеофільм.
9. Надішліть посилання на адресу, зазначену вчителем / вчителькою.

10. Зареєструйтеся на сервісі YouTube (Google) і / або увійдіть у свій обліковий запис.
11. Опублікуйте відеофільм на YouTube для загального доступу.
12. Надішліть посилання на відеофільм на адресу, зазначену вчителем / вчителькою.

Завершіть роботу за комп'ютером.

Зробіть висновок: як розмістити аудіо- та відеоматеріали в інтернеті, які переваги різних способів розміщення.

РОЗДІЛ 5
АЛГОРИТМИ ТА ПРОГРАМИ

§ 24. Сучасні мови програмування

§ 25. Налагодження програмного коду

§ 26. Графічний інтерфейс користувача

§ 27. Опрацювання подій

§ 28. Організація діалогу з користувачем

Практична робота 11. Введення і виведення даних

§ 29. Величини. Числові типи даних

§ 30. Математичні функції

§ 31. Рядковий тип даних

§ 32. Робота з текстовими файлами

§ 33. Логічний тип даних

§ 34. Елементи вибору

§ 35. Алгоритми з повтореннями. Цикли

Практична робота 12. Створення програми з графічним інтерфейсом

§ 36. Графічні методи модуля tkinter

§ 37. Створення анімації

Практична робота 13. Моделювання руху об'єктів

ПОВТОРЮЄМО

У 5–7 класі ви вже опанували основи алгоритмізації та програмування мовою Python, знаєте, як цією мовою реалізувати базові алгоритмічні структури, і вмієте створювати програми з використанням розгалужень і повторень.

У курсі 7 класу ви розширили уявлення про типи величин і алгоритми опрацювання даних різних типів.

Ви також оволоділи прийомами роботи в середовищі програмування IDLE, навчилися вести діалог в інтерактивному режимі у вікні консолі, створювати у вікні програми програмний код і зберігати його для подальшого використання.

1. Що таке тип величини?
2. Із якими типами даних ви працювали?
3. Які базові алгоритмічні структури використовуються при конструюванні алгоритмів?
4. Опишіть алгоритм створення програмного коду.
5. Які види вікон використовують у IDLE?
6. Які повідомлення з'являються у вікні консолі?

У цьому розділі ви продовжите вивчення основ програмування мовою Python і навчитеся розробляти програми з графічним інтерфейсом, створюючи об'єкти класів графічних компонентів. Ви дізнаєтеся, як задавати властивості об'єктів графічного інтерфейсу та програмувати обробку подій для них.

§ 24. СУЧАСНІ МОВИ ПРОГРАМУВАННЯ

Історія створення мов програмування розпочалася в XIX ст., бурхливо розвивалась у XX — із появою ЕОМ. За цей час створено близько 9000 мов програмування та їх модифікацій.

Програма та мова програмування

Розв'язування задач полягає в розробці алгоритму у вигляді програмного коду. Тож пригадаймо основні означення.

Програма — це алгоритм, який записано мовою програмування та призначено для розв'язування певного завдання на комп'ютері. Команди в програмі (програмному коді) подаються мовою програмування.

Мова програмування — це штучна мова, що становить систему позначень і правил для запису алгоритмів у формі, придатній для їх виконання комп'ютером.

Будь-яка мова програмування має такі основні складові:

Алфавіт мови	Синтаксис мови	Семантика мови
Набір символів, із яких утворюються команди та інші мовні конструкції	Правила побудови команд мови програмування	Правила виконання команд, записаних мовою програмування

Під час створення програм використовуються такі символи:

- літери латинського алфавіту A...Z, a...z; цифри 0...9;
- знаки арифметичних операцій, спеціальні символи: + – * / \^ = < > ( ) . , : ; ‘ # _;
- комбінації символів: <=, >=, <>, ==; !=, **;
- службові слова, що мають фіксований зміст: and, elif, if, for, while, as, else, import тощо.

Вважають, що теоретичні основи програмування заклала британська науковиця Ада Августа Лавлейс, яка розробила перші програми для аналітичної машини Чарлза Беббіджа.

Класифікація мов програмування

Розгляньмо розподіл мов за основними ознаками (рис. 24.1).

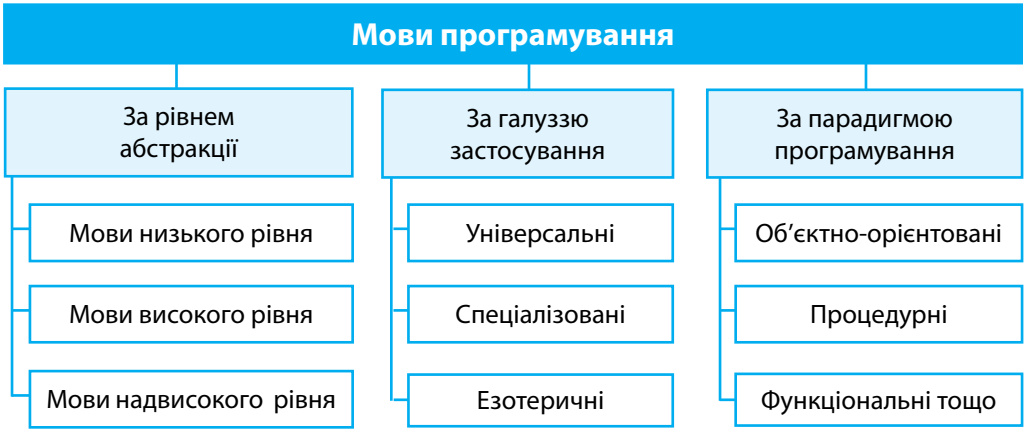


Рис. 24.1

Розподіл за рівнем абстракції

Першими (у 40-х роках XX ст.) створено **мови низького рівня** — машинно-орієнтовані, тобто зорієнтовані на конкретні тип процесора й архітектуру комп'ютера. Наприклад, машинний код і мова асемблера. Програмування на них вимагає від програміста керування пам'яттю й апаратними ресурсами. Це як зводити будинок цеглина за цеглиною. Мови низького рівня використовують для програмування мікроконтролерів або системного програмного забезпечення, де потрібна висока швидкодія.

Мови високого і надвисокого рівнів нагадують людську мову: для побудови команд вони використовують звичайні слова або скорочення (зазвичай із англійської мови). Програмування на таких мовах дає можливість людині не зважати на особливості апаратних засобів, а зосередитися на логіці програми. Це як зводити будинок із готових блоків.

Програми, розроблені мовами високого і надвисокого рівнів, можна виконувати на будь-яких комп'ютерах, для яких існують транслятори вибраної мови (про це дізнаємося далі).

Перша мова високого рівня, що набула широкого визнання серед фахівців, — Fortran (рис. 24.2). Її розроблено в 1954 році в електронній корпорації IBM (США) [11].

Рис. 24.2

Розподіл за галуззю застосування

Мовою програмування люди доносять інформацію до комп'ютера. **Універсальні** мови використовують для розв'язування різних завдань; **спеціалізовані** мови — для розв'язування завдань певного типу або галузі, наприклад, вебпрограмування; **езотеричні** — експериментальні, іноді створені для розваги, проте багато з них є повноцінними мовами програмування.

На рис. 24.3 і 24.4 наведено логотипи універсальних і спеціальних мов відповідно.

Python C/C++ Java Object Pascal

Рис. 24.3

PHP Perl VBScript JavaScript

Рис. 24.4

Поділ за парадигмою програмування

Систему ідей і понять, які визначають стиль написання комп'ютерних програм, а також як програміст уявляє роботу програми, називають **парадигмою програмування**. Парадигм існує ціла низка, а більшість мов програмування (зокрема й Python) підтримують декілька з них. Ви вже використовували елементи структурної та процедурної парадигм.

За **структурної** парадигми програма розглядається як послідовність дій. Базовими поняттями структурного програмування є: *команда* (інструкція), що задає дію з опрацювання даних; *змінна*, якій можна присвоїти значення; *базові алгоритмічні конструкції*, які керують послідовністю виконання команд; кожна з конструкцій має один вхід і один вихід.

За **процедурної** парадигми програма складається з окремих блоків команд — процедур або функцій. Це дозволяє використовувати певний фрагмент коду багато разів, записавши його один раз і надавши назву.

За **об'єктно-орієнтованої** парадигми програма розглядається як сукупність об'єктів, що взаємодіють між собою. Об'єкт має набір властивостей, здатний виконувати певний набір дій над даними й може реагувати на події, які з ним відбуваються.

За **декларативної** парадигми програма в певний спосіб описує, *що потрібно отримати* як результат, а не *як це треба зробити*. Тому декларативне програмування часто називають описовим. Так, декларативним є опис вебсторінок на HTML — теги описують, що містить сторінка й що має відображатися (заголовок, шрифт, текст, зображення), але не містять інструкцій, як її слід відображати. Прикладами декларативних мов є також мова логічного програмування Prolog, мова опису запитів до баз даних SQL, мова опису документів XML тощо.

Інтерпретація та компіляція програм

Для перекладу програм із мов програмування високого та надвисокого рівнів на машинну мову застосовують спеціальні програми — **транслятори**. Вони поділяються на дві категорії: інтерпретатори й компілятори.

Інтерпретатор перетворює невеликий фрагмент програмного коду на машинний код, який процесор одразу виконує. Далі інтерпретується наступний фрагмент програмного коду. Машинний код не зберігається для повторного виконання (рис. 24.5). Прикладом надвисокорівневої інтерпретованої мови є Python.

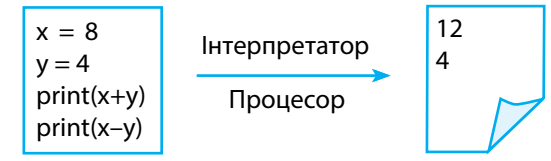


Рис. 24.5

Компілятор перетворює відразу всю програму на машинні коди і вміщує їх у пам'ять комп'ютера, не виконуючи (рис. 24.6). Скомпільовану програму можна зберегти для подальшого використання. Збережений результат компіляції називають **виконуваним файлом** (наприклад, із розширенням *.exe в ОС Windows).

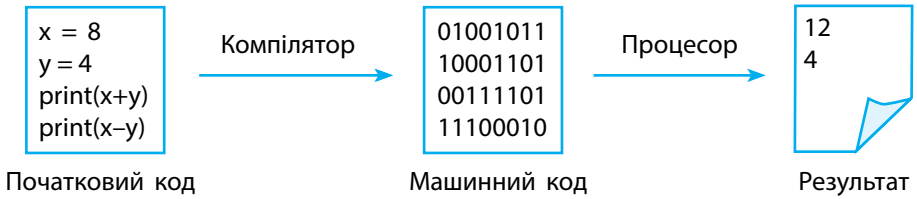


Рис. 24.6

Середовище програмування IDLE

Ви вже опанували основи роботи у середовищі програмування IDLE (рис. 24.7). І знаєте, що файл для встановлення системи програмування можна завантажити із сайту python.org [12]. Разом із версією Python 3 на комп'ютер буде встановлено IDLE, в якому є всі засоби для роботи: текстовий редактор, транслятор, налагоджувач тощо.

Рис. 24.7

Після запуску IDLE відкриється **вікно консолі** IDLE Shell, у якому можна вводити команди і бачити результати в інтерактивному режимі. Команди записують після запрошення «>>>» (рис. 24.8).

```
>>> a = 3.5
>>> b = 8
>>> a**b
22518.75390625
>>>
```

Рис. 24.8

Для створення великих програм, які необхідно зберігати й редагувати, використовують вікно програми. Тоді у вікні консолі виводяться результати виконання програми та повідомлення про помилки в програмному коді (див. далі).

Вікно програми призначене для введення та редагування тексту програми. Щоб відкрити його, необхідно вибрати в меню вікна консолі команду File → New File — відкриється вікно програми з заголовком Untitled.

На рис. 24.9 наведено алгоритм роботи з програмою в середовищі програмування IDLE.

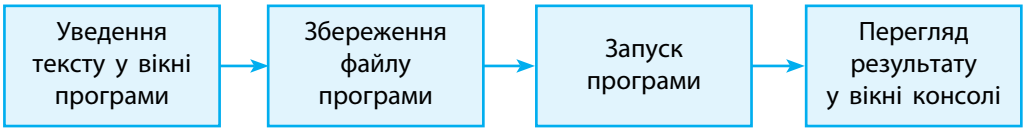


Рис. 24.9

Створимо програму обчислення площі прямокутного трикутника з катетами a і b.

1. Уведемо код у вікні програми (рис. 24.10).
2. Виберемо команду File → Save As і збережемо програму у файлі square.py. (Згадайте: в IDLE програму не можна запустити на виконання, поки її не збережено.)
3. Виберемо команду Run → Run Module (або натиснемо F5).
4. У вікні консолі введемо значення змінних a і b та розглянемо результат виконання програми (рис. 24.11).

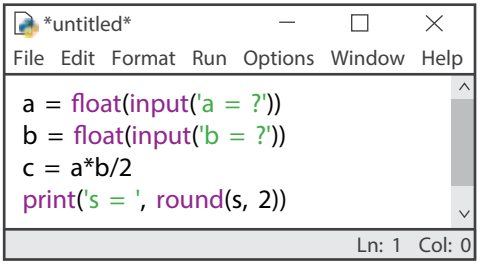


Рис. 24.10

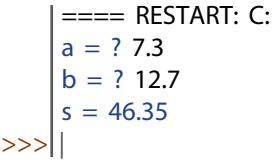


Рис. 24.11

Опанування основ алгоритмізації та програмування потребує від вас зусиль і наполегливості, але результат того вартий.

Карен Спарк Джон — британська дослідниця, яка розробила технології пошуку, що дозволили користувачам і користувачкам працювати з комп'ютерами, використовуючи слова замість рівнянь і кодів [13].

Питання для самоперевірки

1. Опишіть базові алгоритмічні структури, на яких ґрунтується парадигма структурного програмування.
2. Поясніть різницю між інтерпретатором і компілятором.
3. Які парадигми програмування підтримує мова Python?
4. Поясніть призначення команд у програмі (див. приклад).
5. Знайдіть в інтернеті інформацію про рейтинг мов програмування. Обговоріть у групах, які мови є найпопулярнішими в професійному програмуванні.
6. Запустіть веббраузер і введіть у рядок пошуку запит «Hello world різними мовами програмування». Порівняйте коди різними мовами, що виводять рядок Hello world.

Вправа 24

Виконати дії в інтерактивному режимі.

1. Виберіть команду ПУСК → Всі програми → Python → IDLE.
2. У вікні консолі введіть команду завантаження модуля random:
   >>> from random import*
3. Уведіть команди присвоєння змінній x цілого випадкового значення в діапазоні від 1 до 100 і отримання значення x:
   >>> x = randint(1, 100)
   >>> x
4. Уведіть у вікні консолі в інтерактивному режимі команди для обчислення виразів: x//5, x%5; x**3–x+10.
5. Уведіть команду введення значення змінної name:
   >>> name = input ("Як Вас звати?")
6. Уведіть команду print() для виведення у вікно консолі рядка такого змісту: <уведене ім'я користувача>, привіт від Python!.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107835).

§ 25. НАЛАГОДЖЕННЯ ПРОГРАМНОГО КОДУ

Розробка програми, як і будь-яка інша діяльність людини, не обходиться без помилок. Розуміння класифікації помилок і способів їх виправлення є важливою навичкою для програміста.

Помилки в програмах бувають різних типів:

- синтаксичні помилки (англ. *syntax error*);
- помилки часу виконання (англ. *run-time error*);
- логічні (алгоритмічні) помилки.

Процес усунення помилок називають **налагодженням програмного коду**. Інтерпретатор Python виявляє два типи помилок — синтаксичні помилки і винятки (виняткові ситуації).

Виправлення синтаксичних помилок

Синтаксичною називають помилку, що виникає, коли порушено правила запису команд, наприклад, назву змінної почали з цифри або коли забули поставити двокрапку перед початком блока коду. Інтерпретатор Python не може зчитувати або виконувати код, у якому є синтаксичні помилки.

У команді введення не вистачає лапок після текстової константи. Під час спроби виконати програму з'являється повідомлення про помилку, місце помилки виділяється червоним кольором (рис. 25.1).

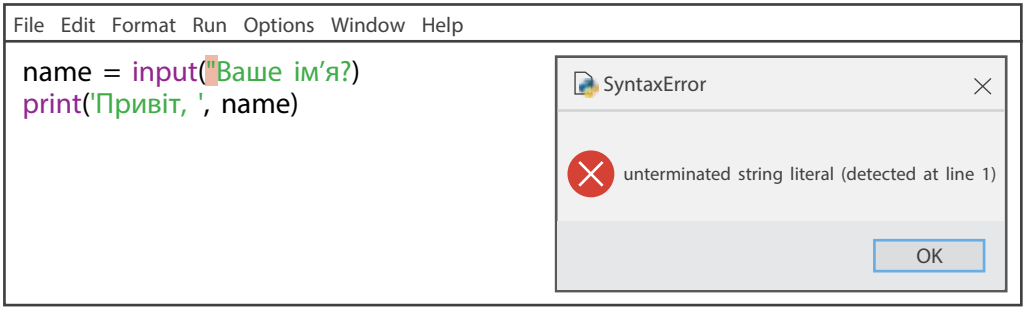


Рис. 25.1

У вікні консолі спробуємо ввести команду присвоювання, де порушено правила створення назви змінної (рис. 25.2).

```
>>> 2x = 3
    SyntaxError: invalid decimal literal
>>>
```

Рис. 25.2

Винятки

Програма може бути правильно написана з точки зору синтаксису, але вона припиняє роботу через якусь помилку.

Помилку, яка не є синтаксичною і спричиняє аварійне припинення виконання програми, називають винятком.

Після запуску програми на виконання у вікні консолі буде виведено номер рядка коду, в якому виявлено помилку, команду, яка спричинила аварійне припинення виконання програми, тип винятку та опис помилки.

Розгляньмо причини виникнення винятків на прикладі.

Виняток NameError виникає під час спроби використати змінну, якій не було присвоєно значення (рис. 25.3). Із точки зору інтерпретатора Python це означає, що такої змінної не існує.

```
>>> c=a*2
    Traceback (most recent call last):
      File "<pyshell#0>", line 1, in <module>
        c=a*2
    NameError: name 'a' is not defined
```

Рис. 25.3

Крім того, зауважимо таке. Останній рядок повідомлення про помилку описує, що сталося.

Виводиться тип винятку NameError — це назва вбудованого винятку, що стався. Решта рядка містить опис подробиць на основі типу винятку та причини його виникнення.

Розгляньмо ще декілька прикладів.

4 Якщо ми неправильно записали назву функції в команді виклику функції, інтерпретатор повідомляє про невизначений ідентифікатор і пропонує варіант виправлення (рис. 25.4).

```
>>> prit(a)
Traceback (most recent call last):
  File "<pyshell#1>", line 1, in <module>
    prit(a)
NameError: name 'prit' is not defined. Did you mean: 'print'?
```

Рис. 25.4

5 Виняток ZeroDivisionError виникає, якщо в програмі трапилася спроба ділення на нуль (рис. 25.5).

```
>>> x = 0
>>> y = 5/x
Traceback (most recent call last):
  File "<pyshell#3>", line 1, in <module>
    y = 5/x
ZeroDivisionError: division by zero
```

Рис. 25.5

6 Під час спроби виконати команду a = int("Hi") виникає виняток ValueError — помилка значення: рядок "Hi" не можна перетворити на ціле число (рис. 25.6).

```
>>> a = int('Hi')
Traceback (most recent call last):
  File "<pyshell#4>", line 1, in <module>
    a = int("Hi")
ValueError: invalid literal for int() with base 10: 'Hi'
```

Рис. 25.6

7 У разі спроби виконати команду a = 10+"3" виникає виняток TypeError — помилка типу, тобто операцію застосовано до об'єкта невідповідного типу.

Число 10 і рядок "3" належать до різних типів даних, тому над ними не можна виконати операцію додавання (рис. 25.7).

```
>>> a = 10+'3'
Traceback (most recent call last):
  File "<pyshell#0>", line 1, in <module>
    a = 10+'3'
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Рис. 25.7

Перехоплення винятків

Команда try-except використовується для обробки винятків, які можуть виникнути під час виконання програми. Вона дає змогу продовжити виконання програми навіть після виникнення помилки, замість того щоб аварійно його завершити.

Синтаксис неповної команди try-except:

try:
    <блок коду, у якому перехоплюється виняток>
except <назва винятку, який перехоплюється>
    <блок коду, який виконується в разі виникнення винятку>

На рис. 25.8 проілюстровано перехоплення винятку «Ділення на нуль».

```
try:                          # Пробуємо виконати ділення
        c = a/b
except ZeroDivisionError:     # Відслідковуємо помилку
        print("Дільник b дорівнює нулю")
        c = 0                 # Змінна c тепер дорівнюватиме 0
print("c = ", c)              # Виводимо значення c
```

Рис. 25.8

Результат виконання команди, якщо a = 5, b = 4, наведено на рис. 25.9, якщо a = 5, b = 0 — на рис. 25.10.

```
c = 1.25
```

Рис. 25.9

```
Дільник b дорівнює 0
c = 0
```

Рис. 25.10

У блоці except можна не вказувати конкретний тип винятку. У такому разі перехоплюватимуться винятки майже всіх типів (рис. 25.11). Результат виконання наведено на рис. 25.12.

```
a = input('a = ?')
try:                    # Пробуємо перетворити тип
        a = int(a)      # уведеного значення
except:                 # Відслідковуємо помилку
        a = 0           # Змінна a тепер дорівнюватиме 0
print('a = ', a)
```

Рис. 25.11

```
==== RESTART:
a = ? 34abc
a = 0
>>>
```

Рис. 25.12

Отже, у блоці try ви пишете код, який може спричинити помилку. Якщо під час виконання цього коду виникає виняток, програма виконує команди блока except.

Логічні помилки

Найскладніше виявити та виправити **логічні помилки**, тобто помилки, яких припустилися під час розроблення алгоритму. Навіть якщо програма успішно опрацьована інтерпретатором, результат її виконання може виявитися хибним. Якщо ви, бажаючи збільшити число a на 1, замість a = a+1 випадково напишете a = a+2, то інтерпретатор не виявить помилки, адже синтаксично ця команда цілком правильна, але результат роботи програми буде неправильним.

Для виявлення логічних помилок програму потрібно перевірити за допомогою **тестів** — наборів вхідних даних і відповідних їм результатів, і виправити виявлені помилки. Тестовий набір даних допомагає перевірити програму на різних вхідних даних, щоб упевнитися, що вона працює коректно в всіх можливих ситуаціях. Тестування полягає в порівнянні результатів зі зразком. Часто розбіжність спричиняють недоліки математичної моделі та помилки в алгоритмі.

Нам потрібно виконати тестування програми для пошуку кореня лінійного рівняння вигляду $ax+b=c$.

```
a, b, c = map(int, input('a, b, c = ?').split())
if a == 0:
    if b == c: print('Безліч коренів')
    else: print('Корені відсутні')
else:
    x = (c–b)/a
    print('x = ', x)
```

Створімо тестові набори значень:

Вхідні дані	Очікуваний результат
a = 5; b = 2; c = 17	x = 3
a = 0; b = 2; c = 17	'Корені відсутні'
a = 0; b = 4; c = 4	'Безліч коренів'

Якщо результати виконання програми збігаються з результатами тестових наборів, наша програма працює коректно, і її можна використовувати для розв'язування лінійних рівнянь.

Питання для самоперевірки

1. Поясніть причину виникнення помилки:
   а) >>> 2a = 10 в) >>> print (a + b ґ) >>> 8 + "3"
   б) >>> a, b = 0 г) >>> int("Hi") д) >>> 1/0
2. Що таке виняток? Наведіть приклади.
3. Опишіть структуру повідомлення в разі виникнення винятку.
4. У результаті роботи програми одержано відповідь: «1,5 спортсмена». Визначте, до якого типу належить помилка.
5. Напишіть програму, яка виконує ділення і множення введених із клавіатури чисел a і b. Які винятки можуть виникнути під час виконання програми?
6. Напишіть програму для визначення, скільки квадратних метрів зелених насаджень потрібно населеному пункту, де ви мешкаєте, якщо за нормами Всесвітньої організації охорони здоров'я, площа озеленення міст має становити 21 м^2 зелених насаджень на одну людину.

Вправа 25

Виправити помилки в програмі й виконати обчислення. *Задача*. Відомі назва планети, радіус орбіти, швидкість руху по орбіті. Обчислити тривалість року на планеті (період обертання). Перевірити роботу програми на тестових даних.

Назва планети	Радіус орбіти, км	Швидкість руху, км/год	Період обертання, діб
Земля	149 600 000	107 250	365,18
Венера	108 200 000	126 110	
Нептун	4 496 600 000	19 550	
Марс	227 900 000	86 870	

В IDLE виберіть команду File → New File.

1. Уведіть текст програми (*він містить помилки*).

```
Nazva = int(input("Назва планети?")
R = int(input("Радіус орбіти?"))
V = input("Швидкість руху? ')
Period = 2*3.14*R/V*24
Print ('Рік на планеті ', 'Nazva', ' триває ', round(Period,3), ' діб');
```

2. Виправте синтаксичні помилки в програмі.
3. Перевірте роботу програми для даних Землі. Чи збігається відповідь із табличною? Якщо ні — проаналізуйте вираз, за яким обчислюється значення Period: період обертання = довжина орбіти/швидкість руху*24, виправте логічну помилку.
4. Виконайте обчислення для Венери, Нептуна, Марса. Заповніть таблицю.
5. Внесіть зміни, щоб результат виводився в земних роках.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107836).

§ 26. ГРАФІЧНИЙ ІНТЕРФЕЙС КОРИСТУВАЧА

Ознайоммося з тим, як організувати взаємодію людини та комп'ютера з використанням графічних елементів керування.

Графічний інтерфейс користувача (ГІК) — це сукупність зображених на екрані віджетів (вікна, меню, кнопки тощо), за допомогою яких людина взаємодіє з комп'ютером.

Користувач має довільний доступ (за допомогою клавіатури або миші) до всіх видимих екранних об'єктів — елементів інтерфейсу. Кожний елемент керування — це об'єкт певного класу, що має властивості та методи.

Для створення програм із ГІК мовою Python застосовують модуль tkinter (від англ. *Tk interface*). Він містить багатоплатформну подійно-орієнтовану бібліотеку класів популярних елементів керування.

Перед використанням цих класів слід завантажити модуль tkinter за допомогою команди:

```
from tkinter import*
```

Створення вікна програми

Побудова програми з ГІК починається зі створення вікна. Далі у вікно додаються інші потрібні складові графічного інтерфейсу користувача, тобто вікно програми є контейнером для інших графічних об'єктів.

Створімо порожнє вікно:

```
from tkinter import*
root = Tk()
root.title('Графічна програма')
root.geometry('250x150')
root.mainloop()
```

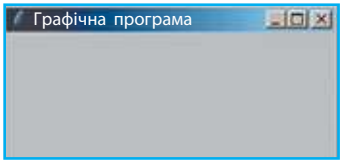


Рис. 26.1

Після збереження програми й запуску на виконання відображується порожнє вікно (рис. 26.1).

Проаналізуймо наведений код.

- Створюється об'єкт root як новий об'єкт класу Tk(), який визначено в модулі tkinter; через змінну root можна керувати виглядом і «поведінкою» вікна.
- Метод title() установлює заголовок вікна.
- Метод geometry() установлює розміри вікна (в пікселях). Як аргумент методу передається рядок у форматі <ширина>x<висота>. Якщо під час створення вікна програми не викликати метод geometry(), то вікно займе простір, необхідний для розміщення елементів керування.
- Метод mainloop запускає головний цикл обробки подій. Рядок root.mainloop() має бути останнім у програмному коді.

За замовчуванням вікно розміщується у верхньому лівому куті екрана. Але ми можемо змінити його положення, передавши потрібні значення в метод geometry():

```
root.geometry('250x150+300+250')
```

Тепер рядок у методі geometry має таку структуру:

<ширина>x<висота>+<координата X>+<координата Y>.

Тобто під час запуску вікно розміститься на 300 пікселів праворуч і 250 пікселів униз від верхнього лівого кута екрана.

Додавання віджета

Користувач / користувачка в основному взаємодіє з програмою за допомогою різних кнопок, меню, значків, уводячи інформацію в спеціальні поля, вибираючи певні значення в списках тощо. Ці графічні компоненти називатимемо віджетами (від англ. *widget*).

Віджети — це блоки для створення графічного інтерфейсу користувача, тобто «будівельні цеглинки» програми. Багато віджетів у різних бібліотеках для побудови ГІК — подібні.

Значення властивостей віджета визначають, який він має вигляд на екрані (наприклад, ширина й висота, колір тощо). Їх початкові значення встановлюють під час створення об'єкта.

Атрибути об'єкта — це назви змінних, у яких зберігаються значення його властивостей.

Віджети створюються викликом конструктора відповідного класу:

[<змінна> =]<клас>([<батьківський_віджет>,][<атрибут> = <значення>])

Тут і далі елементи, зазначені у квадратних дужках, задавати не обов'язково. Значення властивостей можуть задаватися (змінюватися) пізніше.

Батьківський віджет — це елемент керування, до якого буде поміщено новостворений віджет.

Якщо батьківський віджет не задано, то новий віджет буде розміщено в головному вікні. Далі можуть задаватися значення атрибутів. Більшість віджетів у Tkinter мають такі властивості, як висота (атрибут height), ширина (width), шрифт (font), колір віджета (bg), колір тексту (fg) тощо.

Розгляньмо створення елемента керування на прикладі напису — об'єкта класу Label. Віджет Label містить рядок тексту й застосовується для виведення заголовків, підписів інших елементів інтерфейсу, результатів обчислень тощо.

Синтаксис створення віджета:

[змінна =]Label([батьківський_віджет,][атрибут = значення])

Під час створення об'єкта Label можна задати такі атрибути: text — текстовий рядок; width, height — ширина й висота в знакомісцях (кількість символів); bg, fg — колір фону й символів; font — параметри шрифту (тип, кегль).

Наприклад, font = 'Arial 18' — шрифт Arial, кегль 18.

Створений об'єкт потрібно відобразити у вікні. Для **задання розташування віджета** використовують методи-пакувальники:

- place(x, y), де параметри x і y установлюють зміщення елемента по горизонталі й вертикалі відносно верхнього лівого кута вікна;
- pack(), виклик якого розташовує віджети один біля одного.

Зверніть увагу: якщо метод-пакувальник не використати, то напис у вікні так і не з'явиться, хоча він є в програмі.

Розгляньмо кілька прикладів.

1 Додати у вікно напис Hello World!.

```
lab = Label(root, text = 'Hello World!',
            font = 'Arial 18', bg = 'blue',
            fg = 'white')
lab.pack()
```

Напис у вікні «прилип» до верхнього краю вікна (рис. 26.2). Але в будь-якому застосунку віджети не розкидані по вікну, а розміщені на визначених місцях.

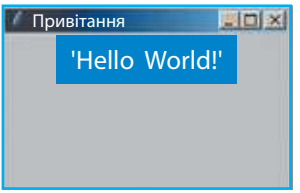


Рис. 26.2

2 Замінити команду виклику методу lab.pack() викликом методу place() (рис. 26.3): lab.place(x = 80, y = 30)

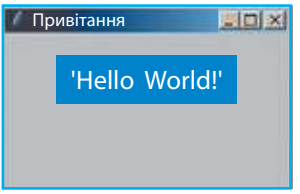


Рис. 26.3

Змінити властивість напису або будь-якого іншого віджета можна так: ім'я_віджета ["атрибут"] = значення

3 Змінити для створеного об'єкта lab текст напису.

```
lab["text"] = 'Я вивчаю Python'
```

Загальні методи віджетів

Класи графічних компонентів мають вбудовані методи об'єктів, за допомогою яких можна змінювати властивості об'єктів. **Метод** — фрагмент коду, який вбудований в об'єкт та передбачає виконання деяких дій із ним.

Запис команди для виклику методу складається з назви об'єкта та назви методу, що розділені крапкою:

<назва об'єкта>.<назва методу>()

Хоча в кожного класу графічних компонентів свій набір методів, деякі (pack(), place(), config(), destroy()) є спільними для всіх віджетів. Метод config() призначений для зміни властивостей раніше створеного віджета: ім'я_віджета.config(атрибут = значення). Метод destroy() викликається для знищення віджета.

4 Змінити для створеного об'єкта lab колір тла й символів:

```
lab.config(bg = 'grey', fg = 'white')
```

Питання для самоперевірки

1. Із чого починається побудова програми з ГІК?
2. Опишіть порядок дій під час створення вікна програми.
3. Як створити об'єкт класу Label?
4. Як змінити властивості вже створеного віджета Label?
5. Змініть у створеного віджета Label значення атрибута text на 'Погода на завтра'.

Вправа 26

Створити об'єкт класу Label.
В IDLE виберіть команду File → New File.

1. Створіть вікно програми розмірами 300×300 із заголовком Перша програма.
```
from tkinter import*
root = Tk()
root.title('Перша програма')
root.geometry('300x300')
```
2. Додайте віджет lab1 класу Label. Задайте значення властивостей: текст — Привіт, друже!; шрифт — Arial, 18 кегль; колір тла — 'yellow'; колір символів — 'grey'; ширина — 20.
```
lab1 = Label(root, text = 'Привіт, друже!', width = 20,
             font = 'Arial 18', bg = 'yellow', fg = 'grey')
lab1.pack()
```
3. Завершіть програму командою запуску циклу обробки подій.
```
root.mainloop()
```
4. Збережіть код із назвою Vprava26 і випробуйте програму.
5. Змініть для об'єкта lab1 текст напису, колір тла та літер — на свій розсуд; перемістіть напис у центр вікна.
6. Збережіть програмний код і випробуйте програму.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107837).

§ 27. ОПРАЦЮВАННЯ ПОДІЙ

Основною рисою програм із графічним інтерфейсом є інтерактивність. Для забезпечення інтерактивності елементи керування пов'язують із подіями, що трапляються під час користування програмою, й забезпечують обробку цих подій через відповідну функцію або метод. З'ясуємо, як примусити програму реагувати на дії користувача.

Клас Button

Командна кнопка — об'єкт класу Button, один із найрозповсюдженіших віджетів, який застосовується для запуску чи закінчення певного процесу.

Синтаксис створення об'єкта класу Button:

[змінна =]Button([батьківський_віджет,][атрибут = значення])

Під час створення об'єкта класу Button можна задати значення атрибутів для налаштування вигляду кнопки, які ми розглянули в ході знайомства з об'єктами класу Label. Це text (напис), width, height (ширина та висота в знакомісцях), bg, fg (колір фону й тексту), font (параметри шрифту).

Створимо командну кнопку (рис. 27.1).

```
btn = Button(root, text = 'ok', width = 10,
             font = 'Arial 16')
btn.pack()
```

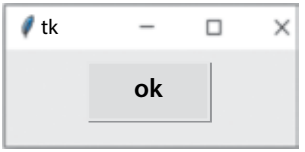


Рис. 27.1

Під час створення ігрових і навчальних програм буває доречно додати на кнопку не текст, а зображення. Для цього потрібно задати значення атрибута image. Спочатку за допомогою функції PhotoImage(file = 'шлях до файлу') потрібно зберегти в змінній шлях до графічного файлу, після чого значення цієї змінної присвоїти атрибуту image. Засобами tkinter можна завантажувати зображення форматів GIF, BMP, PNG.

Створимо командну кнопку й додамо на неї малюнок із файлу Picture.gif, що міститься в тій самій папці, що й файл програми (рис. 27.2).

```
my_image = PhotoImage(file = 'Picture.gif')
btn = Button(root, image = my_image)
btn.pack()
```

Рис. 27.2

Наша програма виконується — створено вікно з командною кнопкою. Але після клацання кнопки немає жодного результату, бо не створено функцію — обробник події Натискання на кнопку.

Опрацювання події Натискання на кнопку

Команди програми із графічним інтерфейсом не просто виконуються від початку до кінця, хід їх виконання залежить від втручання користувача, який взаємодіє з програмою (натискає кнопки, вводить дані тощо, тобто створює події).

Подія — зовнішній вплив на об'єкт, на який цей об'єкт може реагувати.

У графічній програмі від віджетів очікуються події, а у відповідь на них викликаються функції — обробники подій, прикріплені до цих віджетів.

Обробник події — блок програмного коду, який виконується в разі настання події, з якою він пов'язаний.

Щоб прикріпити до кнопки обробник події Натискання на кнопку, необхідно під час створення об'єкта в переліку вказати атрибут command і присвоїти йому назву функції, яку буде викликано в разі натискання (рис. 27.3).

```
def btn_click():
    ...
btn = Button(root, text = 'ok', command = btn_click)
```

Рис. 27.3

У 7 класі ви ознайомилися з правилами створення й використання функцій користувача. Згадаймо правила опису функцій:

```
def <назва функції>(<перелік параметрів>):
    <тіло функції>
```

Якщо функції не треба передавати значення, після її назви ставлять порожні дужки. Ці правила дійсні і для функцій — обробників подій. Отже, додамо обробник події:

```
from tkinter import*
def btn_click():
    root['bg'] = 'green'
root = Tk()
btn = Button(root, text = 'ok', command = btn_click)
btn.pack()
root.mainloop()
```

Тепер програма реагує на натискання кнопки — колір тла головного вікна змінюється на зелений.

Зауважте! Обробник події має бути описаний вище, ніж виклик обробника події. Найчастіше описи функцій розміщують на початку тексту програми.

Створимо кнопку, котра стрибає (рис. 27.4). Після натискання на цю кнопку координати її розташування змінюються випадковим чином.

```
from random import randint
from tkinter import*
def btn_click():
    x1 = randint(5, 190)
    y1 = randint(5, 95)
    btn.place(x = x1, y = y1)
root = Tk()
root.geometry('200x100')
btn = Button(root, text = 'OK', command = btn_click)
btn.place(x = 10, y = 10)
root.mainloop()
```

Рис. 27.4

Функція btn_click() спрацьовуватиме після кожного натискання кнопки, і її положення у вікні змінюватиметься.

Опрацювання інших подій

Можна зробити так, щоб об'єкт змінював властивості у відповідь на натискання певної клавіші клавіатури або миші.

Для цього віджет із подією й функцією — обробником події пов'язують викликом методу bind() (рис. 27.5):

<віджет>.bind(<подія>, <функція>),

де <подія> — текстова константа, системна назва події; <функція> — назва функції — обробника події.

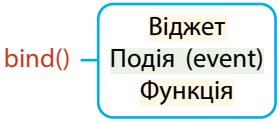


Рис. 27.5

Наприклад, віджет — lab (об'єкт класу Label); подія — '<Button-1>' (клацання лівою кнопкою миші); функція — lab_click (описана в коді вище).

4 Функцію action буде викликано щоразу, коли віджет lab у фокусі й натиснуто клавішу W:

lab.bind('W', action)

Ознайоммося з системними назвами інших подій. Назва події розміщується в лапках і кутових дужках <>.

Системна назва події	Дія
'<Return>'	Натискання клавіші Enter
'<KeyPress-Up>' '<KeyPress-Down>' '<KeyPress-Left>' '<KeyPress-Right>'	Натискання клавіш керування курсором
'<Key>'	Натискання будь-якої клавіші на клавіатурі
'<Button-1>'	Натиснуто ліву клавішу миші
'<Button-3>'	Натиснуто праву клавішу миші
'<Motion>'	Рух миші
'<Enter>'	Входження вказівника миші в межі віджета
'<Leave>'	Вихід вказівника миші за межі віджета

Функція, зазначена у виклику методу bind(), у разі настання події отримує один аргумент. Це об'єкт класу Event, що має опис подробиць події, корисних під час створення програм, наприклад, щоб змусити об'єкти рухатися вікном:

- x і y — координати вказівника миші на віджеті;
- char — набраний на клавіатурі символ;
- keysym_num — код набраного на клавіатурі символу тощо.

5 Запрограмуємо змінення кольору тла віджета lab класу Label під час наведення на нього вказівника миші.

```
from tkinter import *
def lab_in(event):              # Обробник події "<Enter>"
    lab['bg'] = 'green'
def lab_out(event):             # Обробник події "<Leave>"
    lab['bg'] = 'yellow'
root = Tk()
lab = Label(root, text = 'Hello World!', bg = 'yellow', font = 'Arial 16')
lab.place(x = 50, y = 50)
lab.bind("<Enter>", lab_in)  # Входження вказівника миші в межі віджета
lab.bind("<Leave>", lab_out) # Вихід вказівника миші за межі віджета
root.mainloop()
```

Надання об'єкту фокуса

Як бути, коли у вікні програми є кілька віджетів, які можуть реагувати на однакові події? Щоб на подію відреагував певний об'єкт, слід надати йому фокус (зробити активним) викликом методу focus_set(). Подію потрібно прив'язати до вікна, і tkinter буде опрацьовувати події всіх віджетів вікна.

6 У вікні програми розташовано два об'єкти класу Label (рис. 27.6). Запрограмуємо ЛКМ вибір віджета у вікні та його пересування за натисканням клавіш зі стрілками. Після кожного натискання віджет має зміщуватися на 5 пікселів у відповідний бік.

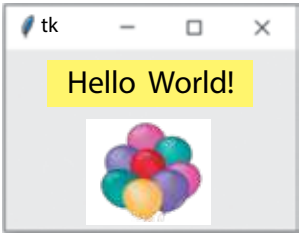


Рис. 27.6

Події пов'яжемо з об'єктом root — вікном програми. Під час клацання лівою кнопкою миші віджета викликається функція focus_widget, яка надає віджету фокус.

При натисканні будь-якої клавіші викликається функція move_lab. У ній з об'єкта event за допомогою атрибута widget здійснюється доступ до об'єкта, для якого обробляється подія. Методи winfo_x() і winfo_y() цього об'єкта повертають його координати x і y відносно вікна.

У змінних dx і dy формується величина зсуву віджета за кожною з осей після натискання на клавішу.

```
from tkinter import*
def focus_widget(event):  # Обробник клацання лівою кнопкою миші
    event.widget.focus_set()
def move_lab(event):      # Обробник натискання клавіші зі стрілкою
    dx = dy = 0
    if event.keysym == 'Up': dy = –5
    elif event.keysym == 'Down': dy = 5
    elif event.keysym == 'Left': dx = –5
    elif event.keysym == 'Right': dx = 5
    # Пересування віджета на dx по горизонталі, dy по вертикалі
    xn = event.widget.winfo_x()+dx
    yn = event.widget.winfo_y()+dy
    event.widget.place(x = xn, y = yn)
root = Tk()
lab1 = Label(root, text = 'Hello World!', bg = 'yellow', font = 'Arial 16')
lab1.place(x = 10, y = 10)
my_image2 = PhotoImage(file = 'кульки.png')
lab2 = Label(root, image = my_image2)
lab2.place(x = 10, y = 50)
root.bind('<1>', focus_widget)
root.bind('<Key>', move_lab)
root.mainloop()
```

Отже, програмуючи опрацювання дій користувача та описуючи функції — обробники подій, ви можете створити програми зі зручним графічним інтерфейсом.

Українська науковця Катерина Ющенко (рис. 27.7) стояла біля витоків вітчизняної школи теоретичного програмування. Вона розробила Адресну мову, одну з перших мов програмування високого рівня у світі, у якій застосовуються операції над адресами об'єктів у пам'яті комп'ютера [14].

Рис. 27.7

Ґрейс Мюррей Гоппер (рис. 27.8) — американська вчена в галузі комп'ютерних наук та контр-адмірал військово-морських сил США, одна з перших програмістів комп'ютера Марк I. Популяризувала ідею машинонезалежних мов програмування, що сприяло розробці мови програмування високого рівня COBOL [15].

Рис. 27.8

Питання для самоперевірки

1. Побудуйте алгоритм створення програми з графічним інтерфейсом на Python.
2. Як прикріпити до об'єкта класу Button обробник події Натискання на кнопку?
3. Обговоріть у групі відмінності в застосуванні обробника натискання кнопки та обробника натискання зазначеної клавіші клавіатури або миші.
4. Запишіть команду виклику для об'єкта lab методу bind() для обробки натискання клавіші ↓.
5. Опишіть функцію (обробник події) Натискання на літеру A, що пересуває об'єкт на 5 пікселів угору.

Вправа 27

Створити програму з різними кнопками й написом. В IDLE виберіть команду File → New File.

1. Створіть вікно програми розмірами 250 × 200 із заголовком Цікаві кнопки. Завершіть програму оператором root.mainloop().
2. Розмістіть угорі вікна кнопку із написом Червоний колір, яка займає всю ширину вікна. Для цього пакувальник pack() викличте з параметрами side = TOP, fill = X:
   b_red = Button(text = 'Червоний колір', command = button_red)
   b_red.pack(side = TOP, fill = X)
3. Аналогічно створіть кнопки із написами Синій колір, Збільшити шрифт, Зменшити шрифт, Змінити напис.
4. Додайте внизу вікна віджет label_text класу Label. Задайте значення властивостей fg, bg (колір фону, колір шрифту).
   label_text = Label(text = 'Hello, World!', font = 'Arial 12')
   label_text.pack(side = BOTTOM, fill = X)
5. Створіть обробник події Натискання на кнопку для об'єкта b_red (рис. 27.9):
   def button_red():
       root.config(bg = 'red')
6. За аналогією створіть обробник події для кнопки Синій колір.
7. Створіть обробник події для кнопки Збільшити шрифт:
   def button_plus()
       label_text.config(font = 'Arial 18')

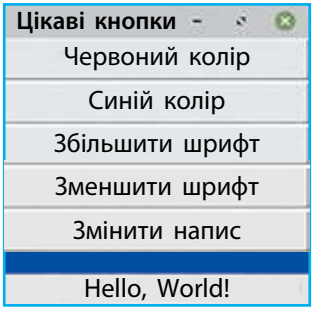


Рис. 27.9

8. За аналогією створіть обробник події для кнопки Зменшити шрифт.
9. Створіть обробник події для кнопки Змінити напис, який змінює текст напису на Привіт від Python!.
10. Запустіть програму, перевірте роботу кнопок.
11. Змініть розміри вікна, поясніть поведінку кнопок і напису.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107838).

§ 28. ОРГАНІЗАЦІЯ ДІАЛОГУ З КОРИСТУВАЧЕМ

У консольному режимі ви користувалися для введення й виведення даних функціями input() і print().

Функція input() призначена для введення даних із клавіатури:
<змінна> = input(<підказка>)

Функція print() потрібна, щоб вивести текст у вікно консолі:
print(<список виведення>)

Розгляньмо елементи керування, призначені для введення й виведення даних у програмі з графічним інтерфейсом. Віджет Label дозволяє виводити текстові повідомлення. Але програма має надавати користувачеві / користувачці можливість вводити дані. Це можна зробити за допомогою віджетів класу Entry.

Клас Entry

Компонент Entry — це поле для введення тексту, також його можна використовувати і для виведення.

Синтаксис створення об'єкта класу Entry такий:
[змінна =]Entry([батьківський_віджет,][властивість = значення])

Для налаштування вигляду віджета Entry слід задати значення властивостей, спільних із віджетами класів Button і Label: ширина й висота в знакомісцях (width, height), колір фону та символів (bg, fg), параметри шрифту (font).

Створімо об'єкт entry1 із такими значеннями властивостей:
- ширина текстового поля — 14 знакомісць;
- параметри шрифту — Arial, 18 кегль.

```
entry1 = Entry(root, width = 14, font = 'Arial 18')
```

Методи Entry

Які дії може виконувати елемент Entry у відповідь на певні події? Поміркуймо разом.

Розгляньмо методи Entry на прикладі об'єкта entry1.

Метод	Призначення	Приклад виклику
get()	Отримати значення, що міститься в текстовому полі	Змінній a цілого типу присвоїти значення, що міститься в полі entry1: a = int(entry1.get())
insert(index, str)	Вивести в текстове поле рядок str, починаючи зі знакомісця з номером index	Вивести в поле entry1 значення змінної x: entry1.insert(0, x)
delete(first, last)	Вилучити символи, починаючи зі знакомісця з номером first до знакомісця з номером last (нумерація символів з 0)	Очистити текстове поле. Щоб вилучити весь текст, як другий параметр потрібно вказати END: entry1.delete(0, END)

1 Уведемо число до поля об'єкта entry1 і виведемо квадрат цього числа до поля об'єкта entry2 (рис. 28.1). Зчитування значення з текстового поля, піднесення до квадрата й виведення результату в поле entry2 виконує функція kvadrat() — обробник події Натискання клавіші Enter.

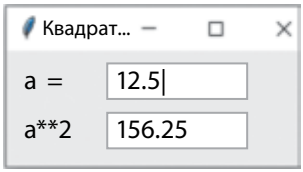


Рис. 28.1

```
from tkinter import*
def kvadrat(event):
    a = float(entry1.get())
    entry2.delete(0, END)
    entry2.insert(0, str(a**2))
root = Tk()
Label(root, text = 'a = ', font = 'Arial 18').place(x = 10, y = 10)
entry1 = Entry(root, width = 8, font = 'Arial 18')
entry1.place(x = 80, y = 10)
Label(root, text = 'b = ', font = 'Arial 18').place(x = 10, y = 50)
entry2 = Entry(root, width = 8, font = 'Arial 18')
entry2.place(x = 80, y = 50)
entry1.bind('<Return>', kvadrat)
root.mainloop()
```

Метод grid()

Вам уже відомі методи pack() і place(), призначені для розміщування віджетів у вікні. Якщо у вікні потрібно в певному порядку розмістити кілька віджетів, то швидко та правильно зробити це допоможе метод grid() (з англ. сітка). Цей пакувальник розміщує віджети в клітинках уявної таблиці.

Адреса кожної клітинки таблиці складається з номера рядка row і номера стовпця column.

Нумерація в рядках і стовпцях починається з нуля.

Клітинки можна об'єднувати як по вертикалі, так і по горизонталі.

Атрибути методу grid() мають такий зміст:

- row, column — номери рядка й стовпця відповідно;
- padx, pady — ширина (в пікселях) вільного місця навколо віджета по горизонталі й по вертикалі відповідно;
- columnspan, rowspan — об'єднання декількох клітинок у рядку або стовпці відповідно.

Пронумеруймо уявні рядки й стовпці (рис. 28.2).

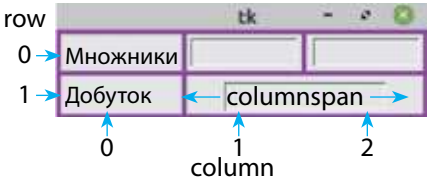


Рис. 28.2

Наведене розташування віджетів забезпечить фрагмент коду:

```
Label(text = 'Множники').grid(row = 0, column = 0, padx = 5, pady = 5)
entry1 = Entry(width = 12)
entry1.grid(row = 0, column = 1, padx = 5, pady = 5)
entry2 = Entry(width = 12)
entry2.grid(row = 0, column = 2, padx = 5, pady = 5)
Label(text = 'Добуток').grid(row = 1, column = 0, padx = 5, pady = 5)
entry3 = Entry()
entry3.grid(row = 1, column = 1, columnspan = 2, padx = 5, pady = 5)
```

Уведемо арифметичний вираз до поля об'єкта entry1 і виведемо значення виразу до поля об'єкта entry2 (рис. 28.3). Для обчислення значення виразу використовується функція eval(), яка обробляє рядок клавіатурних символів так само, як і оболонка консоль Python.

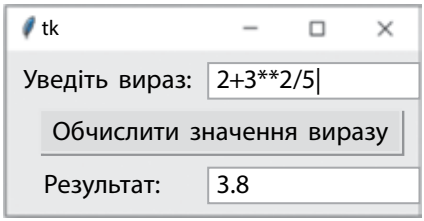


Рис. 28.3

```
from tkinter import*
def btn_click ():
        s = entry1.get()
        a = eval(s)
        entry2.delete(0, END)
        entry2.insert(0, str(a))
root = Tk()
Label(text = 'Уведіть вираз: ').grid(row = 0, column = 0)
entry1 = Entry(width = 20)
entry1.grid(row = 0, column = 1)
btn = Button(root, text = 'Обчислити значення виразу',
                command = btn_click)
btn. grid(row = 1, column = 0, columnspan=2)
Label(text = 'Результат: ').grid(row = 2, column = 0)
entry2 = Entry(width = 20)
entry2.grid(row = 2, column = 1)
root.mainloop()
```

Виклик методу grid (а також place, pack) можна записати в одному рядку з викликом конструктора тоді, коли в програмі не потрібно звертатися до цього об'єкта.

Отже, перевагами табличного способу розміщення є гнучкість та зручність під час розробки складного інтерфейсу.

Адель Голдберг (рис. 28.4) — американська вчена в галузі інформатики, відома своєю роботою в галузі об'єктно-орієнтованого програмування і графічних інтерфейсів та розробкою мови програмування Smalltalk [16].

Рис. 28.4

Вікно повідомлень

Іноді виникає потреба вивести певне повідомлення про роботу програми в діалоговому вікні, не створюючи спеціального віджета. Пакет tkinter містить модуль messagebox, який надає доступ до вікон повідомлень.

Модуль messagebox потрібно імпортувати додатково:

```
from tkinter import messagebox
```

Від вибраного методу залежить вигляд піктограми у вікні повідомлення (рис. 28.5). Для генерації вікна повідомлень із кнопкою OK цей модуль надає такі методи:

- messagebox.showinfo() — створює вікно для відображення деякої інформації;
- messagebox.showerror()() — створює вікно з повідомленням про помилку;
- messagebox.showwarrning() — створює вікно для відображення попереджень.

Синтаксис виклику методу:

```
messagebox.showinfo([заголовок, ][текст])
```

3 Створимо вікно повідомлення інформаційного характеру (рис. 28.6):

```
messagebox.showinfo('Інформація', 'Обчислення закінчено')
```

Рис. 28.5

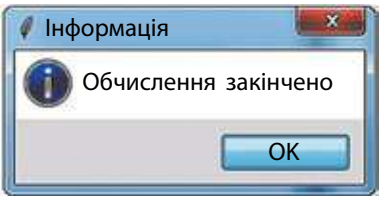


Рис. 28.6

4 Напишемо обробник події Натискання на кнопку, в якому з полів об'єктів entry1 і entry2 класу Entry зчитуються значення для змінних a і b; якщо b = 0 (рис. 28.7), з'являється вікно з повідомленням про помилку (рис. 28.8), інакше — з'являється вікно зі значенням a/b.

```
def b1_click():
    a = float(entry1.get())
    b = float(entry2.get())
    if b == 0:
        messagebox.showerror('Помилка', 'На 0 ділити не можна!')
    else:
        messagebox.showinfo('Відповідь', str(a/b))
```

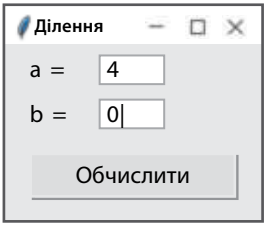


Рис. 28.7

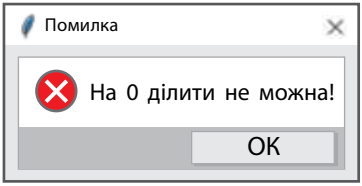


Рис. 28.8

Отже, віджети для обміну інформацією та вікна повідомлень дають змогу в програмі з графічним інтерфейсом організувати діалог людини з комп'ютером.

Питання для самоперевірки

1. Поясніть синтаксис створення об'єкта класу Entry. Яким атрибутам слід задати значення для налаштування вигляду віджета Entry?
2. Опишіть основні методи об'єкта класу Entry.
3. У поле віджета Entry введено рядок Інформатика. Який рядок міститься в полі після виконання оператора:
   а) entry1.delete(2, 6);
   б) entry1.insert(3, '111')?
4. Як зчитати рядок із поля віджета Entry за натисканням Enter?
5. Поясніть, у яких випадках доцільно використовувати кожен із трьох методів-пакувальників: pack, place, grid.
6. Який вигляд має вікно повідомлення, що створене такою командою:
   messagebox.showinfo('Підказка', 'Уведіть число')?

Вправа 28

Відомо, що в результаті переробки 60 кг макулатури зберігається 1 дерево і 0,0005 га лісу. Скласти програму Збережи дерево (рис. 28.9), яка за масою макулатури визначає, скільки дерев і яку площу лісу допоможе зберегти її переробка.

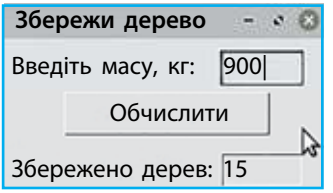


Рис. 28.9

В IDLE виберіть команду File → New File.

1. Створіть вікно програми із заголовком Збережи дерево. Завершіть програму оператором root.mainloop().
2. Створіть віджет класу Label із підписами і віджет entry1 класу Entry для введення даних.
```
Label(text = 'Введіть масу, кг:').grid(row = 0, column = 0,
      padx = 8, pady = 8)
entry1 = Entry(width = 10)
entry1.grid(row = 0, column = 1, padx = 8, pady = 8)
```
3. Розмістіть у вікні віджет b1 класу Button:
```
b1 = Button(text = 'Обчислити', command = b1_click, width = 14)
b1.grid(row = 1, column = 0, columnspan = 2, padx = 8, pady = 8)
```
4. Додайте решту написів та поля entry2, entry3.
5. Опишіть функцію b1_click() для обчислення кількості збережених дерев і площі врятованого лісу.
```
def b1_click():
    n = int(entry1.get())
    entry2.delete(0, END)
    entry2.insert(0, str(int(n/60)))
    entry3.delete(0, END)
    entry3.insert(0, str(n/60*0.0005))
```
6. Випробуйте програму для різних значень маси макулатури.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/107839).

ПРАКТИЧНА РОБОТА 11
ВВЕДЕННЯ І ВИВЕДЕННЯ ДАНИХ

Завдання: скласти програму Касир, під час виконання якої комп'ютер буде виконувати такі дії:
а) повідомляти вартість одиниці товару (рис. 1);
б) запитувати кількість товару, що купується;
в) обчислювати вартість покупки;
г) повідомляти вартість покупки;
ґ) запитувати, скільки грошей дає покупець;
д) повідомляти решту.

Обладнання: комп'ютер із середовищем програмування IDLE.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

В IDLE виберіть команду File → New File.

1. Створіть вікно програми із заголовком Діалог із касиром.
2. Завантажте модулі tkinter, messagebox.
3. Завершіть програму оператором root.mainloop().
4. Створіть об'єкт lab1 класу Label для виведення назви товару та вартості одиниці товару.
```
lab1 = Label(root, text = 'Яблука "Чемпіон": 1 кг – 15 грн', fg = 'red')
lab1.grid(row = 0, column = 0, columnspan = 2)
```
5. Додайте ще два написи відповідно до рис. 2.

Рис. 1

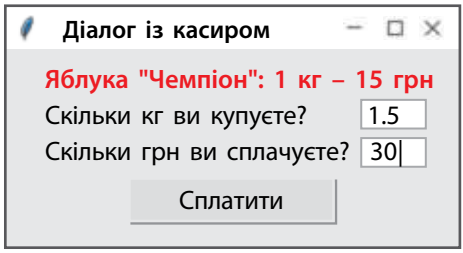


Рис. 2

6. Додайте віджети entry1, entry2 класу Entry згідно з рис. 2.
7. Розмістіть у вікні віджет btn класу Button:
   btn = Button(root, text = 'Сплатити', command = btn_click)
   btn.grid(row = 3, column = 0, columnspan = 2)
8. Організуйте діалог із касиром за таким планом:
   а) у поля вводяться дані для розрахунку;
   б) обчислюється вартість товару й різниця між сумою грошей і вартістю товару;
   в) у діалоговому вікні виводиться повідомлення про решту грошей. Для реалізації такого діалогу опишіть функцію btn_click():

```
def btn_click():
    a = float(entry1.get()); a = a*15
    b = float(entry2.get()); c = b–a
    messagebox.showinfo('Ваша решта', str(c) + ' грн')
```

9. Збережіть файл із назвою Практична робота 11.
   Запустіть програму, проаналізуйте її роботу в неприпустимих ситуаціях (наприклад, сума менша за вартість покупки). Додайте навколо віджетів вільний простір.
10. Додайте до опису функції btn_click() перевірку значення c:
    if c<0: messagebox.showerror('Помилка', 'Недостатньо грошей!')
    else: messagebox.showinfo('Ваша решта', str(c) + ' грн')
11. Змініть виведення повідомлення про решту так, щоб виводилася сума у гривнях і копійках (див. рис. 2).
12. Помістіть у папку з файлом програми графічний файл із зображенням яблук. Завантажте зображення до змінної my_image за допомогою функції PhotoImage:
    my_image = PhotoImage(file = 'яблука.gif')
    Додайте до вікна віджет lab4 класу Label, надавши атрибуту image значення змінної my_image.
    lab4 = Label(root, image = my_image)
    За допомогою методу grid() помістіть малюнок праворуч або нижче від решти віджетів.

Зробіть висновок: як організувати діалог користувача з комп'ютером із використанням модуля tkinter.

§ 29. ВЕЛИЧИНИ. ЧИСЛОВІ ТИПИ ДАНИХ

Будь-яка програма оперує певними величинами. Так, у програмі обчислення шляху, подоланого автомобілем за певний час, використані такі величини, як швидкість руху, час, шлях.

Величина — це окремий інформаційний об'єкт (число, символ, рядок тощо), який характеризується типом і значенням, розміром пам'яті, потрібної для зберігання, назвою (ідентифікатором).

Характеристики величини

Правила вибору назв величин такі самі, як і для інших об'єктів. Кожна величина належить до певного типу. З попередніх класів ви знаєте, що характеристиками величини є назва, тип і значення.

Тип величини визначає множину допустимих значень величини й операції, які можна виконувати над цими значеннями. Згадаємо, що ви вже знайомі з такими типами даних, як int (цілі числа), float (дробові числа), bool (логічні значення), str (рядки), list (списки).

Python — це мова з динамічною типізацією. Це означає, що тип даних змінної визначається за значенням, яке їй присвоєно. У разі присвоєння цілого числа автоматично створюється змінна типу int.

Щоб визначити змінну як об'єкт float, їй присвоюється дробове число, в якому роздільником цілої і дробової частини є крапка.

Тип величини можна визначити за допомогою функції type (рис. 29.1).

```
>>> a = 5
>>> type(a)
<class 'int'>
```

Рис. 29.1

Об'єкти та змінні

У Python усі величини є об'єктами певного класу. Так, число 7 — це об'єкт, який належить до класу (типу) int. Для оперування величинами створюються змінні.

! Змінні в Python зберігають посилання на значення в пам'яті комп'ютера.

Змінну часто порівнюють зі скринькою, в якій зберігається значення величини. Змінні в Python більше схожі на ярлики, що прикріплюються до об'єктів, розміщених у пам'яті комп'ютера (рис. 29.2).

Рис. 29.2

1 Створимо змінну a:
a = 7
Змінна a — це посилання на об'єкт, який містить значення 7.

Значення величини — це конкретні дані, які вона містить у момент звернення до неї. Щоб створити змінну в Python, необхідно дати їй назву й присвоїти значення (рис. 29.3).

Загальний вигляд команди присвоєння:
A = B, де A — назва змінної, B — константа, змінна або вираз.

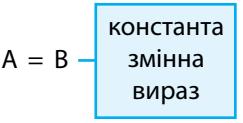


Рис. 29.3

2 Створимо декілька змінних, що посилаються на один і той самий об'єкт (див. рис. 29.2):
a = b = c = 7

Опрацювання даних числових типів

Метою створення перших комп'ютерів була автоматизація обчислень. Уже тоді з'ясувалося, що цілі й дробові числа зручно розміщувати в пам'яті комп'ютера по-різному. Так виникла потреба в декількох типах для числових даних. Вдалий вибір типів даних для величин дає змогу побудувати ефективніший алгоритм опрацювання даних, економно використовувати пам'ять.

Розуміння характеристик числових типів важливе не тільки для правильного розв'язування обчислювальних задач. У програмах із графічним інтерфейсом часто потрібні точні обчислення. Координати пікселів, розміри вікон та елементів інтерфейсу задаються цілочисельними величинами.

Числові типи даних

Цілі числа в мові Python належать до класу int.

Тип величини можна визначити за допомогою функції type().

```
>>> a = 5
>>> type(a)
<class 'int'>
```

Дробові числа, або числа з рухомою комою, належать до типу float. Число float може мати не більше 18 значущих цифр. Тому якщо число занадто велике або занадто мале, то його можна записати в стандартному чи в експоненційному вигляді.

Число після експоненти вказує ступінь числа 10, на яке треба помножити значення мантиси (число 3.9) (рис. 29.4):

```
x = 3.9e3
print(x)        # 3900.0
x = 3.9e-3
print(x)        # 0.0039
```

Рис. 29.4

Ви вже знаєте, що під час розробки програми може виникнути необхідність перетворення значення одного типу даних на значення іншого типу: перетворити введений рядок на числове значення, дробове число перетворити на ціле тощо.

Для перетворення типів призначені такі функції:

- float() — створює дробове число з рядка або цілого числа;
- int() — створює ціле число з рядка або дробового числа. Під час перетворення дробового числа воно не округлюється, а лише відкидається дробова частина;
- str() — створює рядок із числа.

Проаналізуйте приклади застосування функцій перетворення типів для різних аргументів, наведені в таблиці:

Вираз	int('49')	int(4.9)	int('49abc')	str(49)	float(49)
Результат виконання	49	4	помилка	'49'	49.0

Правила запису арифметичних виразів

У 7 класі ви ознайомилися з математичними операторами Python і правилами лінійного запису арифметичних виразів:

- вираз записується у вигляді лінійного ланцюжка символів;
- не можна опускати знак операції множення;
- аргументи функцій записуються в дужках;
- порядок виконання операцій одного пріоритету регулюється дужками.

6 Запишемо за правилами мови Python команду присвоєння змінній t значення виразу $t=\frac{x^2-y^2}{(x-1)\cdot 12{,}34}$.

```
t = (x**2–y**2)/((x–1)*12.34)
```

Виведення дробових чисел

Для підвищення наочності даних, що виводяться програмою, можна задавати кількість десяткових знаків, які з'являться після десяткової крапки, за допомогою методу format():

```
"{:<загальна кількість знаків>.<кількість десяткових знаків>f}".
format(<число>)
```

Проаналізуймо синтаксис форматованого виведення:

- у лапках між фігурними дужками стоїть двокрапка, після якої записуються два цілі числа, розділені крапкою, — загальна кількість знаків у числі, яке виводимо, й кількість десяткових знаків у дробовій частині числа;
- літера f вказує на тип значення, що виводиться (float);
- після лапок ставиться крапка і записується виклик методу format, де аргументом є значення, що виводиться.

7 Проаналізуємо вигляд числа, що виводиться, за різних значень параметра <кількість десяткових знаків>. Загальну кількість знаків, що відводиться для числа, можна не писати, тому у фігурних дужках після двокрапки поставимо крапку.

```
number = 23.8589578
print("{:.3f}".format(number))  # 23.859
print("{:.4f}".format(number))  # 23.8590
```

Питання для самоперевірки

1. Опишіть характеристики величин, що відповідають змінним:
   а) a = 12; б) b = True; в) c = 3.14; г) d = "8.67".
2. Обчисліть значення виразів:
   а) 123//100; б) 123%10; в) 123//10%10.
3. Запишіть наведені вирази за правилами Python:
   а) $10^5 d = 4\frac{3}{7}c^3 + \frac{2}{3}$; б) $\frac{35x-4y}{b_1} - \frac{8c+5}{4b_2} \cdot \frac{b_2}{x+y}$.
4. Запишіть команди присвоювання, які реалізують такі дії:
   а) змінній c присвоїти суму значень змінних a і b;
   б) значення змінної a збільшити на 10.
5. Поекспериментуйте у вікні консолі.
   Виведіть число 123456.123456789 у різних форматах:
   а) '{:.Yf}'.format() із різними значеннями параметра Y;
   б) '{:X.Y}'.format() без указання типу величини, що виводиться.

Вправа 29

Написати програму для розв'язування задачі.

Задача. Обчислити значення щомісячної виплати m за відомими значеннями суми позики s, відсотка p і кількості років n за формулою

$$m = \frac{sr(1+r)^n}{12((1+r)^n - 1)}, \text{ де } r = \frac{p}{100}.$$

В IDLE виберіть команду File → New File.

1. Створіть вікно програми розмірами 250×200 із заголовком Розрахунок виплати (рис. 29.5, с. 192).
   Завершіть програму оператором root.mainloop().
2. Розмістіть у вікні віджет Label із написом Сума позики і віджет Entry для задання значення змінної s:

```
lab1 = Label(root, text = 'Сума позики',
             width = 15, font = 'Arial 14', anchor = E)
lab1.grid(row = 0, column = 0)
ent1 = Entry(root, width = 8, font = 'Arial 14')
ent1.grid(row = 0, column = 1, padx = 10)
```

3. Згідно з рис. 29.5 додайте до вікна об'єкти ent2, ent3, ent4 класу Entry для введення значень p і n, і розмістіть поряд із текстовими полями відповідні написи lab2, lab3, lab4.
4. Додайте до вікна об'єкт btn класу Button:

```
btn = Button(root, text = 'Обчислити', font = 'Arial 14',
             command = btn_click)
btn.grid(row = 3, column = 0, columnspan = 2)
```

5. Опишіть обробник події Натискання на кнопку для об'єкта btn. Вхідними даними є дробові значення s, p, ціле значення n. Треба вивести значення виплати m.
У вікні програми запишіть опис функції.

```
def btn_click():
    s = float(ent1.get())
    p = float(ent2.get())
    n = int(ent3.get())
    r = p/100
    m = (s*r*(1+r)**n)/(12*((1+r)**n–1))
    ent4.insert(0, str(round(m, 2)))
```

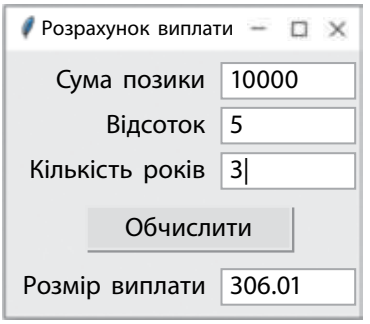


Рис. 29.5

6. Збережіть файл з іменем Vprava29.py. Перевірте роботу програми на тестовому наборі вхідних даних (див. рис. 29.5).

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/109356).

§ 30. МАТЕМАТИЧНІ ФУНКЦІЇ

У мові Python для розв'язування математичних задач існують убудовані стандартні функції. Їх можна використовувати без підключення модулів. Ознайоммося з таблицею:

Функція	Призначення	Приклад	Результат
abs(x)	Повертає |x|	y = abs(–2.5)	y = 2.5
divmod(a, b)	Повертає неповну частку й остачу від ділення a на b	c, d = divmod(7, 3)	c = 2 d = 1
round(num[,n])	Округлює число num до n знаків після коми; якщо аргумент n відсутній, повертає найближче ціле число	x = 1.5278 y = round(x) z = round(x, 2)	y = 2 z = 1.53

Для використання інших математичних функцій потрібно завантажити модуль math:

```
from math import*
```

Розгляньмо деякі функції модуля math. З іншими функціями ви ознайомитеся в міру вивчення математики.

Функція	Призначення	Приклад	Результат
sqrt(x)	Повертає квадратний корінь (square root) з x	a = sqrt(81)	a = 9.0
ceil(x)	Повертає найближче ціле число, яке дорівнює або перевищує x	a = ceil(3.1) b = ceil(7)	a = 4 b = 7
factorial(num)	Повертає добуток цілих чисел від 1 до num	a = factorial(5)	a = 120

Продовження таблиці

Функція	Призначення	Приклад	Результат
degrees(rad)	Конвертує значення кута rad із радіанів у градуси	a = degrees(3.14159)	a ≈ 180
radians(grad)	Конвертує значення кута grad із градусів у радіани	a = radians(180)	a = 3.1415926
cos(rad)	Повертає косинус x, де x виражений у радіанах	x = cos(radians(60))	x = 0.5
sin(rad)	Повертає синус x, де x виражений у радіанах	x = sin(radians(90))	x = 1.0
pi	Константа π = 3.1415...	c = 2*pi*5	c = 31.415926…

1 Нам потрібно розставити n різних книжок на полиці. Скільки варіантів упорядкування (перестановок) книжок існує?
Нам легко з'ясувати, що кількість перестановок n предметів дорівнює добутку цілих чисел від 1 до n. Такий добуток називають факторіалом числа n і позначають n! (читається ен-факторіал).
Код мовою Python може мати такий вигляд:

```
from math import*
n = int(input('Скільки книжок? '))
p = factorial(n)
print(p, ' варіантів')
```

```
Скільки книжок? 5
120 варіантів
```

Рис. 30.1

Результат наведено на рис. 30.1.

2 Обчислимо довжину гіпотенузи прямокутного трикутника з катетами a і b.

```
c = sqrt(a**2+b**2)
```

3 Обчислимо відстань між точками A(xa, ya) і B(xb, yb) на координатній площині (рис. 30.2). Довжина відрізка AB обчислюється за формулою

$AB = \sqrt{(x_b - x_a)^2 + (y_b - y_a)^2}$:

```
d = sqrt((xb–xa)**2+(yb–ya)**2)
```

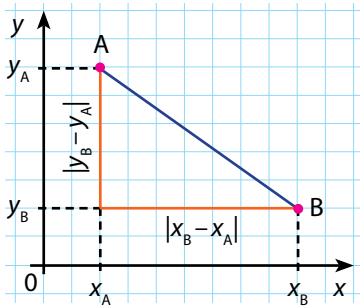


Рис. 30.2

Отже, модуль math відкриває доступ до набору математичних функцій. Це допомагає використовувати готові функції для виконання складних операцій, уникати помилок в обчисленнях.

Випадкові числа

Ви вже маєте досвід застосування випадкових чисел для створення ігрових або тестових програм. Згадаймо: щоб працювати з випадковими числами, слід завантажити модуль random:

```
from random import*
```

Функція	Призначення	Приклад

randint(x1,x2)	Повертає ціле випадкове число в діапазоні від x1 до x2 включно	>>> number = randint(1, 10)
		>>> print(number)
		2

random()	Повертає випадкове число з проміжку між 0.0 і 1.0	>>> number = random()
		>>> number
		0.0654992970998104

uniform (x1, x2)	Повертає випадкове дробове число в діапазоні від x1 до x2	>>> number = uniform(3,9)
		>>> number
		4.2654934970998104

choice (<послідовність>)	Повертає випадковий елемент із послідовності (рядка, списку)	>>> letter = choice('Python')
		>>> letter
		'o'

4 Нам потрібно випадковим чином вибрати зі списку gamers персонажа для гри.

```
gamers = ["Птах", "Заєць", "Бобер", "Кажан"]
gamer = choice(gamers)
```

Результат наведено на рис. 30.3.

```
>>> from random import*
>>> gamers = ['Птах', 'Заєць', 'Бобер', 'Кажан']
>>> gamer = choice(gamers)
>>> print(gamer)
Заєць
>>> gamer2 = choice(gamers)
>>> print(gamer2)
Птах
```

Рис. 30.3

Модуль random генерує псевдовипадкові числа, які насправді є результатом математичних алгоритмів. Це означає, що результати можуть виглядати випадковими, але залежати від певного початкового значення.

Питання для самоперевірки

1. Визначте результат виклику функції:
   а) abs(–6); б) divmod(123, 10); в) round(5.23739, 2).
2. Обчисліть значення, яке повертає функція модуля math:
   а) sqrt(25); в) factorial(6); ґ) 10*pi;
   б) ceil(5.2); г) degrees(1.57); д) radians(90).
3. Складіть програму для обчислення значення виразу $\sqrt{|x-1|+\sin x}$, якщо $x=12.7409$. Виведіть результат із двома десятковими знаками.
4. Складіть програму для обчислення довжини кола та площі круга за введеним радіусом.
5. Складіть програму для обчислення діаметра кола за введеною площею круга.
6. Складіть програму для обчислення довжини катета прямокутного трикутника, за введеними катетом та гіпотенузою.

Вправа 30

Написати програму для розв'язування задачі.

Задача. Туристи вийшли з точки A і пройшли маршрутом $ABCA$. Написати програму для визначення довжини маршруту, якщо відомі координати точок A, B, C (рис. 30.4).

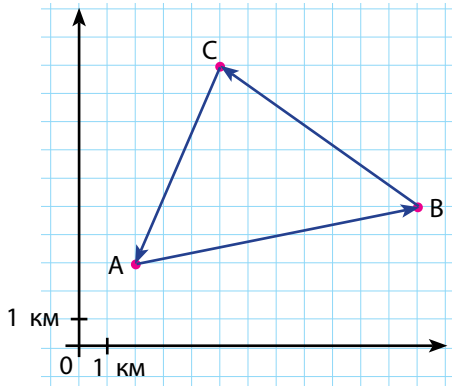


Рис. 30.4

В IDLE виберіть команду File → New File.

1. Завантажте модулі tkinter, math. Створіть вікно програми розмірами 600×300 із заголовком Маршрут (рис. 30.5). Завершіть програму оператором root.mainloop().
2. Додайте до вікна два об'єкти класу Label із написами 'XA =', 'YA =' та об'єкти entxa, entya класу Entry для введення координат точки A.

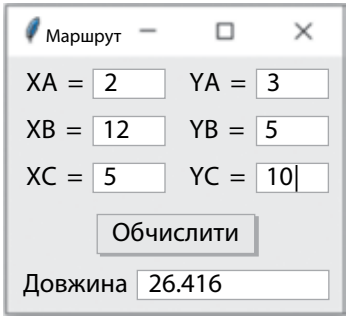


Рис. 30.5

```
Label(root, text = 'XA =').grid(row = 0, column = 0)
entxa = Entry(root, width = 5, font = 'Arial 14')
entxa.grid(row = 0, column = 1, padx = 5)
Label(root, text = 'YA =').grid(row = 0, column = 2)
entya = Entry(root, width = 5, font = 'Arial 14')
entya.grid(row = 0, column = 3, padx = 5)
```

Згідно з рис. 30.4 на с. 197 додайте до вікна об'єкти класу Entry для введення координат точок B і C, і для виведення довжини маршруту. Розмістіть поряд із текстовими полями відповідні написи.

3. Додайте до вікна об'єкт btn класу Button:
```
btn = Button(root, text = 'Обчислити', command = btn_click)
btn.grid(row = 3, column = 0, columnspan = 4)
```
Опишіть обробник події Натискання на кнопку для об'єкта btn. У тілі функції слід записати команди присвоєння змінним xa, ya, xb, yb, xc, yc значень, уведених відповідно в текстові поля entxa, entya, entxb, entyb, entxc, entyc.
Запишіть команду присвоєння змінній m суми довжин відрізків AB, BC, AC:
```
m = f(xa, ya, xb, yb)+f(xb, yb, xc, yc)+f(xc, yc, xa, ya)
```
4. Функція f(x1, y1, x2, y2) має обчислювати довжину відрізка за формулою $d=\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$.

Формальним параметрам x1, y1, x2, y2 передаються значення координат кінців відрізка. Опишіть функцію f():
```
def f(x1, y1, x2, y2):
    d = sqrt((x1–x2)**2+(y1–y2)**2)
    return d
```
5. Завершіть опис функції btn_click командою виведення значення змінної m із трьома десятковими знаками до відповідного текстового поля.
6. Збережіть програму у файлі з іменем Vprava30.py.
Створіть три тестових набори вхідних даних. Обчисліть очікувані результати за складеними тестами.
Перевірте роботу програми на ваших тестових даних.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/109392).

§ 31. РЯДКОВИЙ ТИП ДАНИХ

Для опрацювання текстових даних у мові програмування Python існує тип даних str (від англ. *string* — рядок). Значенням величини типу str є послідовність символів, яку у виразах беруть в одинарні або подвійні лапки: s = 'це рядок' або s = "це рядок".

Початок і кінець рядка позначають лапками одного виду. Рядок із апострофом слід брати в подвійні лапки, і навпаки:

```
s1 = "Я, сім'я, Україна"
s2 = 'Фільм "Земля" (реж. О. Довженко)'
```

Якщо рядок містить як апострофи, так і лапки, перед тими з них, що збігаються з обмежувачами, ставлять символ «\»:

```
s3 = "Сім'я дивиться фільм \"Земля\""
```

Величину будь-якого типу можна перетворити на рядок функцією str() (рис. 31.1).

```
>>> str(1.5e+5)
'150000.0'
>>> str(1/3)
'0.3333333333333333'
>>> str(15)
'15'
```

Рис. 31.1

Доступ до символів у рядку

Для опрацювання всіх символів рядка зручно скористатися циклом for <змінна> in <рядок>. Під час виконання такого циклу змінна почергово набуває значень окремих символів рядка.

Виведемо символи рядка, введеного з клавіатури, поставивши після кожного з них двокрапку.

Наприклад, країна → к:р:а:ї:н:а:

```
s = input()
for C in s:
    print(C, end = ':')
```

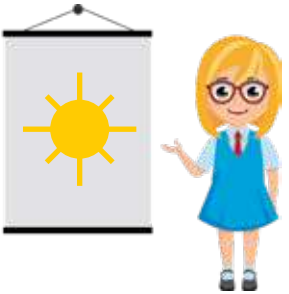
До певного символу в рядку можна звернутися за його індексом (номером у рядку). Нумерація зростає від 0 (від початку рядка) або спадає від −1 (від кінця рядка), що показано на рис. 31.2.

```
>>> s = 'Це рядок'
>>> s[3]
'р'
>>> s[-3]
'д'
```

Рис. 31.2

Отриманий за індексом символ можна присвоїти іншій змінній.

```
>>> s = 'Сонечко яскраво сяє'
>>> b = s[8]
>>> b
'к'
```



Якщо спробувати отримати доступ до індексу за межами діапазону або використовувати числа, відмінні від цілих (наприклад, числа з рухомою комою), Python видасть помилку.

Із рядка можна добувати зріз: копіювати деяку послідовність символів рядка (не обов'язково суміжних) за певним правилом.

Команда добування зрізу з рядка:

a = s[i: j: step],

де a — змінна, в яку зберігаються добуті символи;

i — індекс початку зрізу;

j — індекс межі кінця зрізу (не входить у зріз);

step — крок, із яким вибираються символи.

Розглянемо застосування операції добування зрізу на прикладі рядка s = 'Сонечко яскраво сяє'.

Приклад	Опис	Результат
a = s[8:15]	Зріз від символу з номером 8 рядка s включно до символу з номером 14	a = 'яскраво'
a = s[:7]	Зріз від початку рядка до символу з номером 6	a = 'Сонечко'
a = s[8:]	Зріз від символу з номером 8 рядка s до кінця рядка	a = 'яскраво сяє'
a = s[::2]	Копіювання символів з рядка s із кроком 2	a = 'Снчоякаосє'
a = s[::-1]	Запис символів рядка s у зворотному порядку	a = 'єяс оваркся окченоС'

Дії над рядковими величинами

Рядки в Python є незмінюваними послідовностями.

4 Спроба замінити символ у рядку викликає помилку.

```
s = 'Сонечко'
s[2] = 'л'
TypeError: 'str' object does not support item assignment
```

Згадаймо прийоми опрацювання рядків. Розгляньмо таблицю:

Операція	Опис	Приклад	Результат
len(s)	Кількість символів у рядку	k = len('Сорока')	k = 6
subs in s	Визначення, чи є в рядку s підрядок subs	a = 'y' in 'Python'	a = True
s1+s2	Об'єднання рядків s1 і s2 в один рядок (конкатенація)	s = 'Сонце' + ' ' + 'сяє'	s = 'Сонце сяє'
n*s	Дублювання рядка s n разів	s1 = '+'*4	s1 = '++++'

Відповідні операції доступні й у вигляді методів рядкових об'єктів. У наведених прикладах методи викликаються для змінної s, яка містить опрацьовуваний рядок. Розгляньмо таблицю:

Назва методу	Опис дії
s.find(s1)	Шукає в рядку s підрядок s1 і повертає номер першого символу шуканого підрядка в рядку або –1, якщо в s немає підрядка s1
s.replace(sfind, sins)	У рядку s замінює підрядок sfind підрядком sins
s.count(subs)	Повертає кількість входжень підрядка subs у рядок s

Рядкові методи утворюють новий рядок. Для подальшого використання результату виконання, його слід зберегти в змінній:

```
S = 'шпала'
m = S.replace('па', 'ко')
print(m)                    # Буде виведено: школа
```

Багаторядкове текстове поле

Багаторядкове текстове поле є об'єктом класу Text. За замовчуванням створюється віджет висотою (height) 24 знакомісця і шириною (width) 80 знакомісць. Значення цих властивостей можна задати при створенні об'єкта.

Значення WORD властивості wrap вмикає перенесення слів у полі на новий рядок цілком, а не по буквах.

5 Створимо об'єкт text1 класу Text.

```
text1 = Text(width = 35, height = 6, wrap = WORD)
text1.pack()
```

Якщо в текстове поле введено більше рядків тексту, ніж висота поля, то під час перегляду текст можна прокручувати вгору й униз за допомогою колеса миші й клавіш керування курсором.

Методи Text

Основні методи в Text такі самі, як у класу Entry, — get(), insert(), delete(). Проте, викликаючи методи для багаторядкового поля, для символів слід зазначати два індекси, відокремлені крапкою, — номер рядка й номер символу в ньому.

Нумерація рядків починається з одиниці, а стовпців — із нуля.

6 Виведемо в поле об'єкта text1 вірш.

Щоб вірш виводився по рядках, початок кожного рядка слід позначити символом перенесення рядка \n:

```
s = "Встала весна, чорну землю\nСонну розбудила,\nУквітчала
     її рястом,\nБарвінком укрила.\nТ. Г. Шевченко"
text1.insert(1.0, s)
```

7 Отримаємо з поля віджета text1 символи 1 рядка з індексами від 0 до 12 (рис. 31.3).

```
subs = text1.get(1.0, 1.12)
                    # subs = "Встала весна"
```

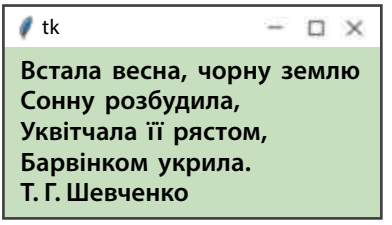


Рис. 31.3

8 Щоб отримати один рядок із поля Text, потрібно вказати допустимий індекс для початку рядка, а потім зазначити той самий індекс із додаванням "lineend":

```
subs = text1.get(1.0, "1.0 lineend")
                          # subs = "Встала весна, чорну землю"
```

9 Щоб отримати з поля Text його вміст, починаючи з певної позиції і до кінця, як другий параметр потрібно вказати END:

```
s = text1.get(1.0, END)
```

Питання для самоперевірки

1. Визначте, якого значення набуває змінна a після виконання команд, якщо s = 'Сонечко':
   а) a = s[2:5]; в) a = s[5:]; ґ) a = 5*s[2];
   б) a = s[:3]; г) a = s[::2]; д) a = s[::–1].
2. Запишіть команду, що реалізує таку дію:
   а) отримати підрядок рядка s із 2-го символу по 5-й включно;
   б) отримати підрядок із останніх п’яти символів рядка s.
3. Розгадайте фразу й запишіть оператор для її розшифрування:
   а) s = 'яа клеюгбмлщюй кіеннфгошрщмзазтхиткуур';
   б) s = 'мофинвафин пайфинтонфин'.
4. Складіть програму для визначення, чи є паліндромом уведений із клавіатури рядок s.
5. Поясніть, як створюється й використовується текстове поле.

Вправа 31

■ Скласти програму для шифрування тексту.
В IDLE виберіть команду File → New File.

1. Створіть файл із назвою Vprava31.
   Завантажте модуль tkinter. Створіть вікно програми із заголовком Шифрувальник.
2. Створіть об’єкт text класу Text. Усі віджети розміщуйте за допомогою методу grid().

```
text1 = Text(width = 35, height = 6, wrap = WORD)
text1.grid(row = 0, column = 0, columnspan = 2)
```

Уставте в поле віджета текст підказки:
text.insert(1.0,"Видаліть цей текст\ni напишіть свій.")

3. Створіть кнопки — об'єкти b_shifr, b_delete класу Button (рис. 31.4). Обробникам подій для кнопок надайте назви shifrText, deleteText.

```
b_shifr = Button(root, text = 'Зашифрувати', command = shifrText)
b_shifr.grid(row = 1, column = 0)
b_delete = Button(root, text = 'Видалити', command = deleteText)
b_delete.grid(row = 1, column = 1)
```

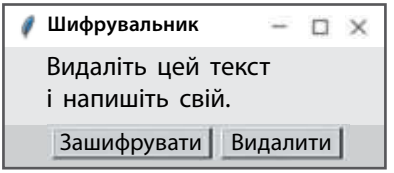


Рис. 31.4

4. Опишіть обробник події для кнопки b_delete:

```
def deleteText():
    text.delete(1.0, END)
```

5. Опишіть обробник події для кнопки b_shifr:

```
def shifrText():
    s = text.get(1.0, 'end–1c')
    text.delete(1.0, END)
    if (len(s)%2 != 0): s = s+' '       # Якщо кількість символів
    for i in range(0, len(s), 2):       # непарна, додати пропуск
        text.insert(END, s[i: i+2:][: :–1])
```

Уведіть текст і натисніть кнопку Зашифрувати.
Проаналізуйте, який спосіб шифрування використано.

6. Придумайте свій спосіб шифрування тексту й змініть опис функції shifrText(), щоб застосувати цей спосіб.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/109393).

§ 32. РОБОТА З ТЕКСТОВИМИ ФАЙЛАМИ

Файл може бути як джерелом, так і приймачем інформації.

Розгляньмо засоби Python для обміну даними між програмою і зовнішнім запам'ятовуючим пристроєм (диском). Програма може обробляти наявний файл або створювати новий. У будь-якому випадку робота з файлом складається з трьох етапів (рис. 32.1).

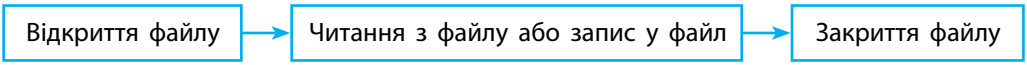


Рис. 32.1

Відкриття файлу

Для відкриття файлу призначена вбудована функція open():

f = open(<шлях до файлу>, <режим відкриття>),

де f — змінна для збереження файлового об'єкта, який використовується для читання і/або запису у ваш файл.

Якщо шлях до файлу складається лише з назви файлу, Python шукатиме цей файл у папці, в якій збережено вашу програму.

Параметр Режим відкриття визначає, які дії можна буде виконувати з файлом. Основні режими відкриття такі:

- "r" — відкриття тільки для читання;
- "w" — відкриття для записування: знищує вміст файлу, якщо він існує; створює новий файл для запису, якщо не існує.
- "a" — відкриття для дозаписування; інформація додається в кінець файлу; якщо файлу не існує, його буде створено.

Читання з файлу

Читання даних із файлу означає отримання доступу до них. Метод <файлова змінна>.read([n]) зчитує дані з файлу, а саме:

- зчитує всі дані з файлу, якщо аргумент n відсутній;
- зчитує n символів, якщо метод викликано з аргументом (цілим числом n).

1 У файлі 'text.txt' міститься рядок "Привітання від Python!". У консоль виводиться вміст файлу (рис. 32.2).

```
f = open("text.txt", "r")
print(f.read(1))    # Зчитати 1 символ з файлу
print(f.read())     # Зчитати всі інші символи
f.close()
```

```
=============== RESTART: C:/Users/Admin/Desktop/Python/1.py ===============
Привітання від Python!
>>>
```

Рис. 32.2

Метод <файлова змінна>.readline() зчитує із файлу один рядок. Після зчитування елемента (символу або рядка) з файлу поточним (доступним) стає наступний елемент (див. приклад 1).

Якщо файл містить декілька рядків тексту, можна прочитати файл порядково, скориставшись циклом for:

```
f = open("numbers.txt", "r")
for line in f:
    print(line, end = '')      # Якщо не вказати атрибут  end = '',
f.close()                      # виводяться зайві порожні рядки
```

Запис у файл

Щоб записати рядок s у файл, для файлового об'єкта можна викликати метод write():

<файлова змінна>.write(s)

Для того щоб зміни у файлі збереглися, потрібно обов'язково закрити файл, викликавши для файлового об'єкта метод close. Закриття файлу вивільняє ресурси, які були використані для роботи з ним.

2 Запишемо у файл два рядки.

```
f = open("f1.txt", "w")
f.write("111111")
f.write("222222")
f.close()
```

3 Файл f1.txt містить декілька чисел, розташованих по одному в рядку. Слід зчитати з файлу всі числа, знайти їх суму й дописати значення суми в кінець файлу (рис. 32.3, с. 207).

```
f = open("f1.txt", "r")
suma = 0
for line in f:
    suma = suma + int(line)
f.close()
f = open("f1.txt", "a")
s = 'сума = '+str(suma)
f.write(s)
f.close()
```

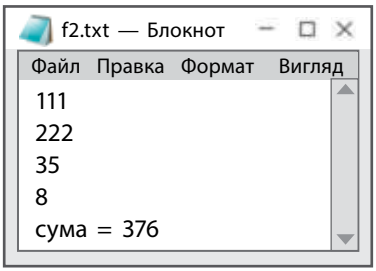


Рис. 32.3

Відкриття й збереження файлів

Модуль filedialog у Python є частиною бібліотеки tkinter і надає функції для роботи з діалоговими вікнами, пов'язаними з вибором файлів або каталогів. Це зручно для створення графічних інтерфейсів, де потрібно дозволити користувачам вибирати файли для обробки або збереження.

Для завантаження модуля слід записати команду:

from tkinter import filedialog as fd,

де fd — це псевдонім, яким ми, щоб скоротити код, замінятимемо назву модуля filedialog у разі виклику методів.

Функція askopenfilename() надає діалогове вікно для відкриття файлу, функція asksaveasfilename() — вікно для збереження файлу. Вони обидві повертають назву файлу, який має бути відкрито або збережено, але самі його не відкривають і не зберігають. Робити це має програміст у подальших командах.

4 Створимо графічне вікно з кнопками для відкриття й збереження файлу (рис. 32.4, с. 208).

```
from tkinter import*
from tkinter import filedialog as fd
def insertText():
    file_name = fd.askopenfilename(filetypes = (("TXT files", "*.txt"),
                ("HTML files", "*.html;*.htm"), ("All files", "*.*") ))
    f = open(file_name)
    s = f.read()
    text.insert(1.0, s)
```

```
        f.close()
    def extractText():
        file_name = fd.asksaveasfilename()
        f = open(file_name, 'w')
        s = text.get(1.0, END)
        f.write(s)
        f.close()
    root = Tk()
    text = Text(width = 25, height = 8)
    text.pack()
    b_open = Button(root, text = "Відкрити",
                command = insertText)
    b_open.pack()
    b_save = Button(root, text = "Зберегти",
                        command = extractText)
    b_save.pack()
    root.mainloop()
```

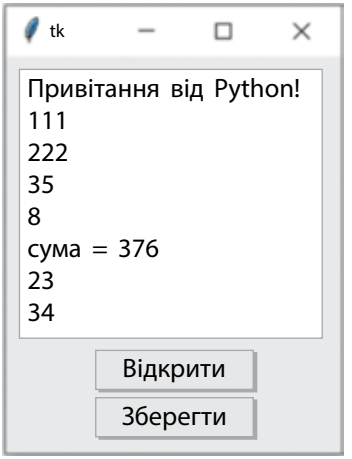


Рис. 32.4

Опція filetypes дає змогу перелічити типи файлів, які зберігатимуться або відкриватимуться, та їх розширення.

Питання для самоперевірки

1. Як організувати читання даних із текстового файлу?
2. Чим відрізняється робота функцій read(n), read() і readline()?
3. Як організувати виведення даних у текстовий файл?
4. Обговоріть у групі переваги використання діалогових вікон для відкриття та збереження файлів.
5. Файл t.txt містить декілька рядків символів (і цифр, і літер). Складіть програму, яка переписує цифри з файлу t.txt у файл cif.txt, усі інші — у файл lit.txt.

Вправа 32

Написати програму, що підраховує кількість заданих символів у фрагменті літературного твору.

1. Створіть у текстовому редакторі файл f_text.txt і скопіюйте фрагмент твору з інтернету (наприклад, «Собор» О. Гончара).

В IDLE виберіть команду File → New File.

2. Завантажте модулі tkinter, filedialog, messagebox.
3. Створіть вікно програми із заголовком Аналіз тексту.
   Розмістіть у вікні віджет Entry для задання шуканого символу та віджет Text для показу вмісту файлу f_text.txt.
4. Додайте до вікна два віджети Button: b_open для відкриття файлу, і b_count — для підрахунку числа входжень шуканого символу (рис. 32.5).
5. Опишіть функцію insertText() для кнопки b_open:

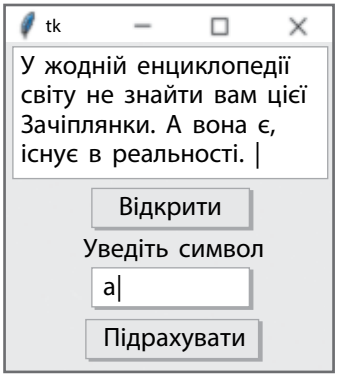


Рис. 32.5

```
def insertText():
    file_name = fd.askopenfilename(filetypes = (("TXT files", "*.txt"),
                                                ("All files", "*.*") ))
    f = open(file_name)
    s = f.read()
    text.insert(1.0, s)
    f.close()
```

6. Опишіть функцію count_C() для кнопки b_count:

```
def count_C():
    c = entry.get()
    s = text.get(1.0, END)
    k = s.count(c)
    messagebox.showinfo("Відповідь",
                        "k = "+str(k))
```

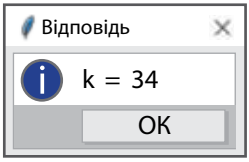


Рис. 32.6

Випробуйте програму для різних значень c (рис. 32.6).

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/109394).

§ 33. ЛОГІЧНИЙ ТИП ДАНИХ

До простих типів даних у Python належить логічний тип bool.

Змінна типу bool може набувати значення True (істина) або False (хибність).

▶ Прості та складені умови

Значень типу bool набувають вирази, побудовані за допомогою операцій порівняння (прості умови):

==	дорівнює	>	більше	>=	більше або дорівнює
!=	не дорівнює	<	менше	<=	менше або дорівнює

Проаналізуємо результати обчислення логічних виразів:

>>> print(2*2==4) True	Оскільки $2 \cdot 2 = 4$ — істинна рівність, то значення відповідного логічного виразу дорівнює True

>>> NUM = 10
>>> NUM%3 == 0
False
	Отримано відповідь на запитання «Чи ділиться значення змінної NUM на 3?»

Складніші логічні вирази (складені умови) будують, коли одночасно необхідно перевірити виконання кількох простих умов.

Для запису складених умов використовують логічні операції. Розгляньмо таблицю:

Операція	Назви	Приклад	Значення
or	або логічна сума	2*2 == 7 or 4 <= 5 3 > 8 or 12 < 5 or 9 == 1	True False
and	і логічний добуток	2*2 == 7 and 4 <= 5 3 <= 8 and 12 > 5 and 9 != 1	False True
not	не логічне заперечення	not 2*2 == 4 not 8 < 3	False True

2 Перевіримо, чи належить NUM проміжку [7; 20]:

```
>>> NUM = 12
>>> print (NUM >= 7 and NUM <= 20)
True
```

Якщо змінній присвоїти значення логічного виразу, вона належатиме до типу bool:

```
>>> a = 3>7
>>> a
False
```

Убудовані функції all і any

Одночасну перевірку декількох умов можна записати коротше за допомогою вбудованих функцій all (англ. *всі*) і any (англ. *хоч одна*). Функція all(<список логічних виразів>) повертає True, якщо істинні всі перелічені в списку умови.

3 З'ясуємо, чи є значення NUM парним числом із проміжку від 7 до 20. Для цього слід одночасно перевірити три умови:

```
>>> NUM = 8
>>> all([NUM >= 7, NUM <= 20, NUM%2 == 0])
True
```

Метод any(<список логічних виразів>) повертає True, якщо хоча б один елемент списку дорівнює True.

4 Перевіримо, чи дорівнює 0 хоча б одне зі значень x, y, z:

```
>>> x, y, z = 4, 0, 5
>>> any([x == 0, y == 0, z == 0])
True
```

Алгоритми з розгалуженнями

Як ви знаєте, **розгалуження** — це алгоритмічна структура, в якій, залежно від істинності деякої умови, виконується одна з двох дій.

Ви вже знайомі з синтаксисом та семантикою умовних операторів мови Python, призначених для програмування розгалужень.

Команда розгалуження if

Виконання команди if (англ. *якщо*) можна описати так: «якщо умова істинна, то виконати ці команди». Команда if реалізує алгоритмічну конструкцію неповне розгалуження (рис. 33.1).

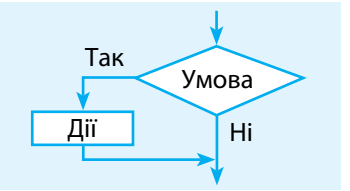


Рис. 33.1

Синтаксис команди:

```
if <умова>:
__<команди>
```

! Команди записують з обов'язковим однаковим відступом від лівого краю. Якщо команда лише одна, її можна записати в одному рядку з умовою.

5 Складемо програму для впорядкування уведених значень змінних a і b так, щоб a не перевищувало b.

```
a, b = map(int, input().split())
if a>b: a, b = b, a
print(a, b)
```

Команда розгалуження if...else

Алгоритмічній конструкції повне розгалуження відповідає команда if...else (англ. *якщо...інакше*) (рис. 33.2).

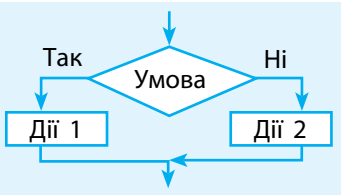


Рис. 33.2

Синтаксис команди:

```
if <умова>:
    <команди 1>
else:
    <команди 2>
```

Якщо умова істинна, тобто має значення True, то виконується блок дій <команди 1>. В іншому випадку виконується блок дій <команди 2>, записаний після службового слова else.

! Команди, вкладені в гілки if і else, записують з однаковим відступом.

6 Складемо програму для перевірки, чи три введені числа є довжинами сторін прямокутного трикутника. Для перевірки скористаємося теоремою Піфагора:

```
a, b, c = map(int, input().split())
if any([ a**2 == b**2+c**2 , b**2 == a**2+c**2 , c**2 == a**2+b**2]):
    print('Так')
else:
    print('Ні')
```

Вкладення розгалужень

Якщо під час виконання (або невиконання) деякої умови потрібно знову робити вибір, то застосовують так звані вкладені розгалуження: у відповідній гілці команди розгалуження знову використовують команду if.

7 Складемо програму, яка б за уведеним цілим числом виводила слово «рік» у правильному числі й відмінку (рис. 33.3).

1 рік	11 років	21 рік
2 роки	12 років	22 роки
3 роки	13 років	23 роки
4 роки	14 років	24 роки
5 років	15 років	25 років
6 років	16 років	26 років
7 років	17 років	27 років
8 років	18 років	28 років
9 років	19 років	29 років
10 років	20 років	30 років

Рис. 33.3

Проаналізувавши форми слова «рік», помічаємо, що для числа K вибір закінчення визначає його остання цифра. Проте значення від 11 до 14 слід опрацювати окремо (див. алгоритм на рис. 33.4, с. 214).

Цей алгоритм реалізує така програма:

```
K = int(input())
if K%10 == 0 or K%10>4 or K >= 11 and K <= 14: print('років')
else:
    if K%10 == 1: print('рік')
    else: print('роки')
```

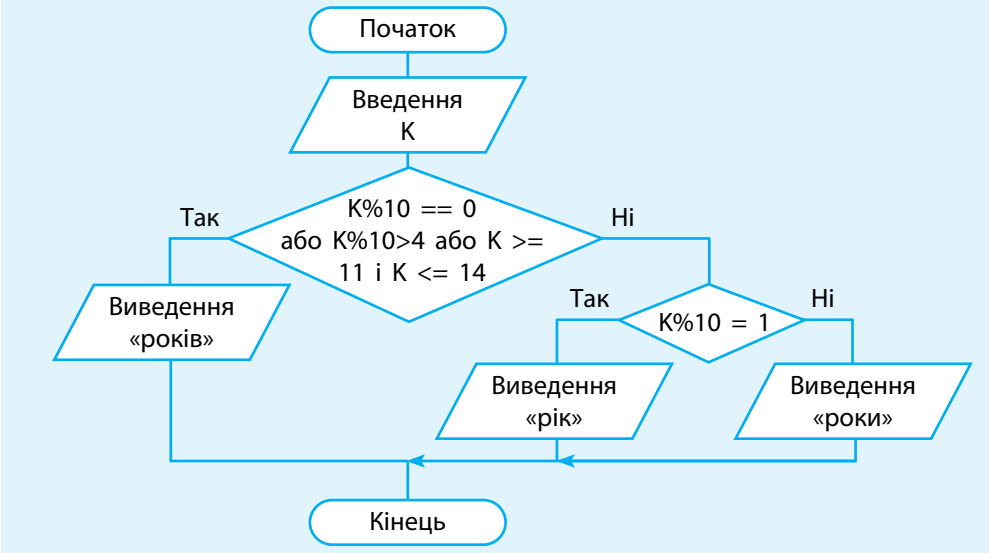


Рис. 33.4

Багатоваріантне розгалуження

Вкладені розгалуження можна записати у більш компактному вигляді. Перевірку умови в гілці else можна записати скорочено, скориставшись ключовим словом elif (англ. *else if* — інакше, якщо). Це дозволяє реалізувати багатоваріантне розгалуження (рис. 33.5).

Синтаксис команди:

```
if <умова1>:
        <команди 1>
elif <умова2>:
        <команди 2>
...
elif <умова N>:
        <команди N>
else:
        <команди N+1>
```

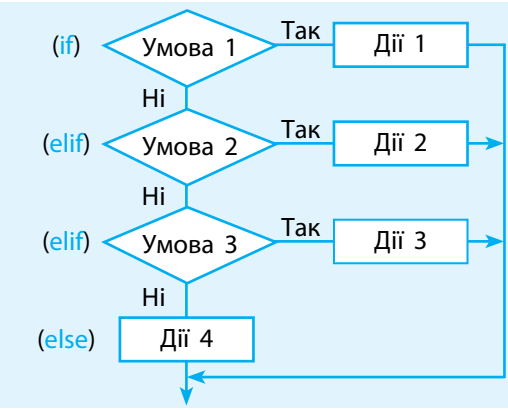


Рис. 33.5

Визначимо, у якій координатній чверті лежить точка з координатами x, y ($x \neq 0$, $y \neq 0$) (рис. 33.6).

```
x, y = map(int, input('x, y? ').split())
if x>0 and y>0: print('I чверть')
elif x<0 and y>0: print('II чверть')
elif x<0 and y<0: print('III чверть')
elif x>0 and y<0: print('IV чверть')
```

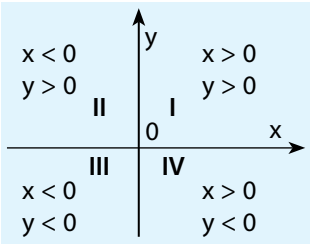


Рис. 33.6

У цьому прикладі складені умови, що перевіряються в гілках if...elif, охоплюють усі можливі варіанти розташування точки, що відповідають умові. Але якщо є ймовірність того, що хоча б одна з координат точок дорівнює 0, то слід додати гілку else: else: print('Точка лежить на осі').

Розглянуті команди розгалуження дозволяють реалізувати вибір дій у як завгодно складних випадках.

Питання для самоперевірки

1. Початкові значення змінних a = –3; b = 5. Визначте, чому дорівнюють a і b після виконання таких команд:

   а)
   ```
   if a>b: a = a–b
   else: b = b–a
   ```

   б)
   ```
   if a<0:
       a = –a
   ```

   в)
   ```
   if a != b:
       a = b
   ```

2. Початкові значення змінних a = 8; b = 5. Визначте, чому дорівнюють a і b після виконання таких команд:

   а)
   ```
   if a<0:
       if a>b: a = b
       else: b = a
   else:
       if a>b: b = a
       else: a = b
   ```

   б)
   ```
   if a<5:
       b = a+1
   else:
       if a<10:
           b = a*2
   ```

3. Запишіть команду розгалуження для обчислення значення y:

   а) $y=\begin{cases}1-\sin a, & \text{якщо } -1<a<1;\\ 0, & \text{для всіх інших;}\end{cases}$

   б) $y=\begin{cases}x, & \text{якщо } x<0;\\ x^2, & \text{якщо } 0\leqslant x\leqslant 5;\\ 2x, & \text{якщо } x>5.\end{cases}$

4. Наведіть приклад використання команди if...elif...else.

5. Запишіть команду розгалуження, яка виконує таку дію: менше з двох значень x і y замінити нулем, а у випадку їх рівності — замінити нулями обидва.
6. Складіть програму, яка за введеним значенням температури t виводить повідомлення про стан, у якому перебуває вода за такої температури (твердий, рідкий, газоподібний).

Вправа 33

Скласти програму для знаходження коренів квадратного рівняння вигляду $ax^2+bx+c=0$.

В IDLE виберіть команду File → New File.

1. Створіть файл програми з назвою Vprava33.
2. Завантажте модулі tkinter, messagebox (як mb), math.
3. Створіть вікно програми із заголовком Квадратне рівняння. У кінці програми напишіть команду root.mainloop().
4. Розробіть інтерфейс програми за зразком (рис. 33.7). Віджетам Entry дайте назви enA, enB, enC.

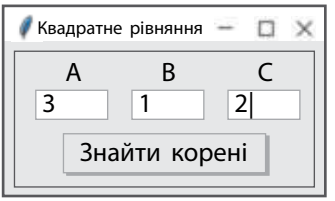


Рис. 33.7

5. Опишіть функцію btn_click() — обробник події для кнопки Знайти корені. Значення a, b, c можна зчитати з полів enA, enB, enC в такий спосіб:

```
a, b, c = map(float, [enA.get(), enB.get(), enC.get()])
```

Результати обчислення виводьте до діалогового вікна. Якщо рівняння має два корені, їх значення можна вивести так:

```
s = 'x1 =' +str(round(x1, 2))+'x2 = ' +str(round(x2, 2))
mb.showinfo('Два корені', s)
```

6. Доберіть такі набори значень a, b, c, щоб квадратне рівняння мало один корінь; два корені; не мало коренів.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/109395).

§ 34. ЕЛЕМЕНТИ ВИБОРУ

Вам, напевно, доводилося робити вибір певних значень із низки запропонованих у діалоговому вікні за допомогою таких елементів керування, як перемикачі та прапорці (рис. 34.1).

У Tkinter перемикачі складаються з об'єктів класу Radiobutton (радіокнопка), прапорці ж є об'єктами класу Checkbutton.

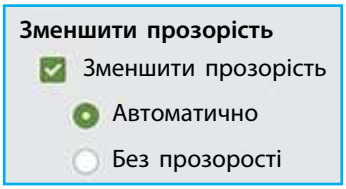


Рис. 34.1

Елемент керування Радіокнопка (Radiobutton)

Перемикачі, що складаються з двох або більше радіокнопок (віджетів класу Radiobutton), використовують, коли потрібно вибрати один із декількох варіантів.

! У разі вибору іншої радіокнопки попередня вимикається.

Перемикачі корисні, коли важливо уникнути вибору декількох значень одночасно. Наприклад, вибір способу оплати: "Кредитна картка", "Готівка", або вибір рівня складності гри: "Легкий", "Середній", "Складний".

Зв'язок між радіокнопками перемикача встановлюється через спільну змінну. Її назву зазначають в атрибуті variable кожної з радіокнопок. Ця змінна має бути екземпляром спеціального класу IntVar, описаного у Tkinter, і може набувати, відповідно, цілих значень. Щоб отримати значення змінної, слід викликати метод get(), а щоб його змінити, — метод set().

Якщо радіокнопку вибрано, спільна змінна отримує значення її атрибута value, і навпаки — надавши їй певного значення, можна увімкнути радіокнопку з відповідним значенням value.

Отже, для **створення перемикача** потрібно:

1) створити змінну var класу IntVar для керування станом перемикача;
2) створити потрібну кількість об'єктів класу Radiobutton із різними значеннями властивості value;
3) занести в атрибут variable усіх радіокнопок назву змінної var.

Створимо перемикач для вибору одного з трьох кольорів.

```
var = IntVar()                  # Змінна для зв'язку між радіокнопками
var.set(0)                      # Активуємо радіокнопку з value = 0
rbR = Radiobutton(text = "Червоний", variable = var, value = 0)
rbY = Radiobutton(text = "Жовтий", variable = var, value = 1)
rbG = Radiobutton(text = "Зелений", variable = var, value = 2)
rbR.pack(anchor = W)
rbY.pack(anchor = W)
rbG.pack(anchor = W)
```

Параметр anchor поміщає віджет у певній частині контейнера. Щоб вирівняти радіокнопки за лівим краєм, слід надати параметру значення W (West (захід) — ліва сторона). Під час запуску програми виявиться ввімкненою радіокнопка rbR, бо значення її атрибута value = 0 збігається з поточним значенням змінної var. Якщо вибрати радіокнопку rbY, то вона увімкнеться, а кнопка rbR вимкнеться, і змінна var набуде значення 1.

Щоб у відповідь на перемикання виконувалися певні дії, потрібно пов'язати радіокнопки з обробниками подій.

Пов'яжемо всі кнопки з функцією — обробником change(), так само, як ми це робили для кнопок (Button). В обробнику, залежно від того, яка з радіокнопок активна, змінюватимемо колір тла форми (рис. 34.2).

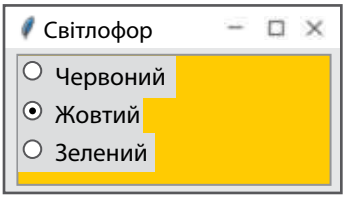


Рис. 34.2

```
def change():
    if var.get() == 0: root.config(bg = 'red')
    elif var.get() == 1: root.config(bg = 'yellow')
    elif var.get() == 2: root.config(bg = 'green')
```

Розробимо програму Конструктор піц, яка при замовленні дає змогу користувачеві вибирати розмір піци (рис. 34.3).

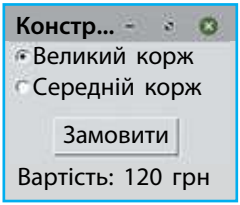


Рис. 34.3

```
from tkinter import*
def btn_click():
    lab.config(text = 'Вартість: ' +str(var.get()*60)
               + ' грн')
root = Tk()
root.title('Конструктор піц')
var = IntVar()
var.set(0)
Radiobutton(text = 'Великий корж', variable = var,
            value = 2).pack(anchor = W)
Radiobutton(text = 'Середній корж', variable = var,
            value = 1).pack(anchor = W)
Button(root, text = 'Замовити', command = btn_click).pack(pady = 10)
lab = Label(root, text = 'Вартість: ', width = 20, anchor = W)
lab.pack(padx = 5, pady = 10)
root.mainloop()
```

Елемент керування Прапорець (Checkbutton)

Прапорець — об'єкт класу Checkbutton — використовують у програмі для того, щоб умикати або вимикати певний параметр. Тобто для візуалізації станів увімкнено / вимкнено, коли серед параметрів необхідно вибрати потрібні.

Прапорці можна також розташувати групою, але кожна позначка в такій групі встановлюється або знімається незалежно від стану інших прапорців. Тобто в групі можна одночасно встановити декілька позначок.

Кожний об'єкт класу Checkbutton повинен мати свою змінну типу BooleanVar():

```
cb_var = BooleanVar()
cb_var.set(0)
```

За значенням цієї змінної, пов'язаної з об'єктом Checkbutton, можна визначити стан об'єкта (чи встановлено прапорець).

Об'єкт cb класу Checkbutton створюють такою командою:

```
cb = Checkbutton(text = "<напис>", variable = cb_var, onvalue = 1,
                 offvalue = 0)
```

За встановленого прапорця пов'язана з ним змінна набуває значення onvalue, а за знятого — offvalue.

4 Для перевірки стану прапорця та активізації пов'язаних із ним параметрів слід отримати значення змінної:

```
if cb_var.get() != 0: <активізація параметрів>
```

5 Додамо до програми Конструктор піц (див. приклад 3) можливість вибирати склад піци (рис. 34.4).

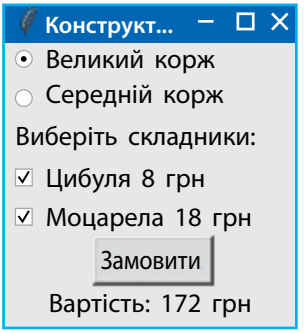


Рис. 34.4

Додамо перед рядком із викликом конструктора Button() команди для створення та налаштування прапорців:

```
Label(root, text = 'Виберіть складники:',
      width = 20, anchor = W).pack(pady = 10)
cvar1 = BooleanVar()
Checkbutton(text = "Цибуля 8 грн",
            variable = cvar1, onvalue = 1,
            offvalue = 0).pack(anchor = W)
cvar2 = BooleanVar()
Checkbutton(text = "Моцарела 18 грн", variable = cvar2, onvalue = 1,
            offvalue = 0).pack(anchor = W)
```

Змінимо код обробника btn_click(), щоб під час розрахунку враховувався стан прапорців:

```
def btn_click():
    suma = var.get()*(60+cvar1.get()*8+cvar2.get()*18)
    lab.config(text = 'Вартість: ' +str(suma)+ ' грн')
```

Віджети Radiobutton і Checkbutton підтримують більшість атрибутів зовнішнього вигляду, які є в інших елементів графічного інтерфейсу.

Питання для самоперевірки

1. У яких випадках використовують компонент Radiobutton?
2. Що забезпечує використання компонента Checkbutton?
3. Як створити зв'язок між радіокнопками?
4. Назвіть три відмінності між об'єктами класу Radiobutton та об'єктами класу Checkbutton.
5. Як організувати перевірку стану прапорця?
6. Як до програми Конструктор піц додати ще один прапорець (наприклад, для вибору складника "Бекон").

Вправа 34

Створити програму Вгадай колір.

Гравцеві потрібно вибрати колір напису, що з'являється у вікні програми. Текст напису й колір символів вибираються випадковим чином.

В IDLE виберіть команду File → New File.

1. Завантажте модулі tkinter, random.
   Запишіть команди для створення вікна програми tk із заголовком Вгадай колір (рис. 34.5).
   Завершіть програму командою tk.mainloop().

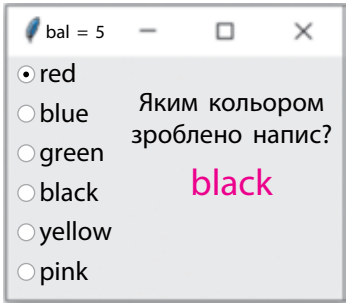


Рис. 34.5

2. Створіть список назв кольорів:
   colors = ["black", "red", "blue",
   "yellow", "green", "pink"]
   Створіть змінну bal для збереження кількості правильних відповідей:
   bal = 0
   Змінній k надайте випадкове значення зі списку colors:
   k = choice(colors)
3. Створіть шість радіокнопок із підписами, що відповідають назвам кольорів. Розмістіть віджети у вікні за допомогою пакувальника grid.
   Щоб вирівняти віджети за лівим краєм, задайте атрибуту sticky значення W.

```
var = IntVar()
Radiobutton(text = "black", variable = var, value = 0,
            command = change).grid(row = 0, column = 0, sticky = W)
```

Запишіть команди для створення п'яти інших перемикачів.

4. Створіть об'єкт lab1 класу Label.

```
lab1 = Label(text = "Яким кольором зроблено напис?")
lab1.grid(row = 1, column = 1)
```

Створіть об'єкт lab2 класу Label. Властивості text об'єкта надайте випадкове значення зі списку colors:

```
lab2 = Label(text = choice(colors), fg = k, font = "Times 30")
lab2.grid(row = 2, column = 1)
```

5. Заповніть пропуски в описі функції change():

```
def change():
    n = "GO"  # Ознака продовження гри
    global bal, k  # Дозвіл на змінення значень глобальних змінних
    if var.get() == 0 and k == 'black': bal = bal+1
    elif var.get() == 1 and k == 'red': bal = bal+1
    <Додайте решту рядків>
    else: n ="end"
    if n == "end": lab2['text'] = "GAME OVER!"
    else:
        tk.title('bal = '+str(bal))        # Виведення значення лічильника
        t = choice(colors)                 # відповідей у заголовок вікна
        lab2['text'] = t        # Виведення випадкового значення
                                # зі списку colors
        k = choice(colors)      # Надання k випадкового значення
                                # зі списку colors
        lab2['fg'] = k      # Відображення символів напису кольором k
```

6. Збережіть файл з іменем Vprava34.py.

Перевірте роботу програми.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/109396).

§ 35. АЛГОРИТМИ З ПОВТОРЕННЯМИ. ЦИКЛИ

Ви маєте досвід записування алгоритмів із повторенням (циклів) мовою Python і вмієте використовувати два види операторів циклу: із параметром та з умовою.

Як вам відомо, **повторення (цикл)** — це алгоритмічна структура, за допомогою якої та сама послідовність дій виконується кілька разів.

Цикли дають змогу запрограмувати повторення подібних обчислень, команди для створення однакових елементів зображення (наприклад, квадрати, лінії, кола), що зменшує кількість написаного коду і спрощує його змінювання.

Цикли незамінні при створенні програм з анімацією, моделюванні руху об'єктів на екрані та ін.

Згадаймо: команди, що повторюються під час виконання циклу, називають **тілом циклу**, а кожне виконання тіла циклу — **ітерацією**.

Цикл із параметром

Цикл із параметром (або **цикл** for) використовують, якщо кількість повторень визначається певною послідовністю значень (символи рядка, елементи списку тощо).

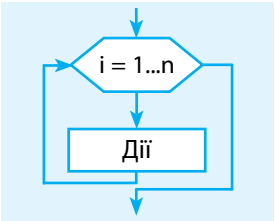


Рис. 35.1

Блок-схему циклу подано на рис. 35.1.

Синтаксис циклу for:

```
for x in <послідовність>:
    <тіло циклу>
```

Змінна x — параметр циклу — послідовно набуває значень кожного члена послідовності.

1 Посимвольна обробка рядка: виведемо рядок, пропустивши «зірочки».

```
s = 'AB*C*D*EF*G'
for C in s:
    if not C == '*': print(C, end = '')
```

2 Дано список днів тижня. Скільки з них припадають на вихідні (сб, нд)?

```
d = ['пн', 'нд', 'чт', 'сб', 'пт', 'сб', 'вт', 'ср', 'нд', 'чт', 'чт', 'нд', 'вт', 'нд', 'сб']
k = 0
for day in d:
    if day in ['сб', 'нд']: k += 1
print('Вихідних днів – ',k)
>>> Вихідних днів – 7
```

Числові послідовності можна задавати за допомогою вбудованої функції:

range([<start >,] <finish>[, < step >])

Вона створює послідовність цілих чисел від значення start до finish із заданим кроком. Зауважмо, що саме значення finish у послідовність не входить.

3 Надрукувати числа від 20 до 24. Якщо step = 1, цей параметр можна не зазначати.

```
for i in range(20, 25): print(i)
```

4 Надрукувати числа від 0 до 3. Якщо start = 0, цей параметр можна не зазначати:

```
for i in range(4): print(i)
```

5 Надрукувати числа від 5 до 1. Якщо потрібно вести відлік у зворотному порядку, step має бути від'ємним:

```
for i in range(5, 0, –1): print(i)
```

У мові програмування Python цикл із параметром відкриває багато можливостей. Так, використовуючи його, ми можемо запрограмувати рух об'єктів у вікні програми.

6 Запрограмувати пересування в програмному вікні об'єкта lab класу Label із координатами $(x, sin\ x)$ (рис. 35.2).

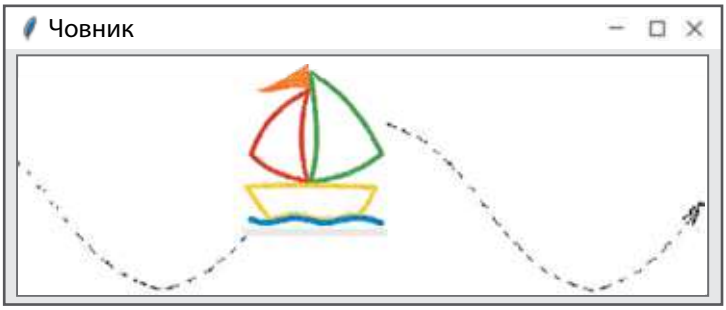


Рис. 35.2

```
from tkinter import *
import time, math
root = Tk()       # Створюємо вікно програми
root.title('Човник')
root.geometry('800x250')
im = PhotoImage(file = '1.png')
lab = Label(root, width = 70, height = 70, image = im)
lab.place(x = 1, y = 90)       # Початкове розташування об'єкта lab
for x1 in range(1, 250):
    lab.place(x = x1*2, y = 10*math.sin(x1)+100)
                        # Розташування об'єкта lab на x1-й ітерації
    root.update()         # Оновлення полотна
    time.sleep(0.1)       # Пауза на 0.1 секунди
root.mainloop()           # Запуск головного циклу обробки подій
```

Цикл із передумовою

Цикл із передумовою (або **цикл** while) буде повторюватися доти, поки істинна певна умова.

Синтаксис команди:

```
while <умова>:
    <тіло циклу>
```

Блок-схему команди while подано на рис. 35.3, с. 226.

Виконання команди циклу while починається з обчислення значення логічного виразу — умови повторення циклу.

Якщо умова *істинна*, то виконуються команди тіла циклу й керування повертається на перевірку умови.

Якщо умова *хибна*, то виконується команда, яка є наступною після команди while.

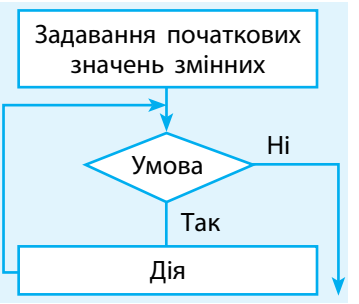


Рис. 35.3

! Слід мати на увазі, що тіло циклу while може не виконатися жодного разу. Це буває, коли при першій перевірці умова виявляється хибною.

Леонардо Фібоначчі — видатний математик середньовічної Європи. Числами Фібоначчі називають послідовність чисел, у якій кожне наступне число є сумою двох попередніх:
Fib[1] = Fib[2] = 1; Fib[n] = Fib[n–1]+Fib[n–2].
Для визначення номера першого числа Фібоначчі, яке більше за 1000, потрібен цикл із умовою:

```
a = b = 1
c = 2
n = 3
while c<1000:
    c = a+b
    a, b = b, c
    n += 1
print('число', c, 'має номер', n)
```

Переривання роботи циклу

Команда break використовується для дострокового завершення виконання циклу. Вона дає змогу зупинити роботу циклу після знаходження необхідного результату та уникнути виконання зайвих ітерацій. Звернімо увагу, що команда break здійснює вихід із циклу, навіть якщо виконано не всі ітерації.

У нескінченному циклі здійснюється додавання чисел, що вводяться з клавіатури. Якщо введене число 0, цикл припиняється (рис. 35.4).

```
suma = 0
k = 0
while True:
    a = int(input('a = ? '))
    if a == 0: break
    k += 1
    suma += a
print('Сума ', k, 'чисел дорівнює ', suma)
```

```
a = ? 5
a = ? 2
a = ? –3
a = ? 0
Сума 3 чисел дорівнює 4
```

Рис. 35.4

Вкладені цикли

Вкладені цикли використовують в алгоритмах розв'язування таких задач, як сортування списків, опрацювання прямокутних таблиць, обчислення всіх можливих комбінацій або перестановок, побудова орнаментів тощо.

Вкладений цикл — цикл, який міститься в тілі іншого циклу. Такий цикл також називають **внутрішнім**, а цикл, у якому він міститься, — **зовнішнім**.

Усередині вкладеного циклу може бути наступний вкладений цикл, і так далі.

Ознайомімося з принципом виконання вкладених циклів.

- При першій ітерації зовнішній цикл викликає внутрішній, який виконується до свого завершення, після чого керування передається в тіло зовнішнього циклу.
- При другій ітерації зовнішній цикл знову викликає внутрішній.

Такі дії тривають до завершення зовнішнього циклу (рис. 35.5).

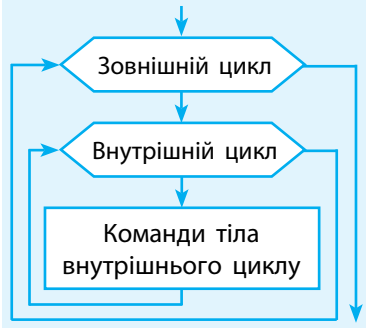


Рис. 35.5

9 Зобразити трикутник із чисел від 1 до 5, як наведено на рис. 35.6.

```
1
222
33333
4444444
555555555
```

Рис. 35.6

Зовнішній цикл for i відповідає за побудову кожного рівня (рядка) трикутника.
Внутрішній цикл for j призначений для визначення кількості пропусків перед числами, щоб правильно вирівняти трикутник по центру.
У внутрішньому циклі for k виводиться (2*–1) цифр.

```
rows = 5      # Задання кількості рядків у піраміді
for i in range(1, rows + 1):        # Зовнішній цикл для рядків
    for j in range(rows–i):         # Внутрішній цикл для пропусків
        print(" ", end="")
    for k in range(2*i–1):          # Внутрішній цикл для значення і
        print(i, end = "")
    print()                         # Перехід на новий рядок
```

10 Вивести в багаторядкове текстове поле таблицю множення.

```
from tkinter import *
root = Tk()                                  # Створюємо вікно програми
root.title('Таблиця Піфагора')
text1 = Text(width=70, height=10)   # Створюємо об'єкт класу Text
text1.pack()
for i in range(1, 10):
    s=''
    for j in range(1, 10):
        s=s+str(i*j)+"\t"        # Формуємо і-й рядок таблиці Піфагора
    k = float(i)          # Перетворюємо значення і на дробове, щоб
    text1.insert(k, s)  # отримати позицію початку виведення рядка s
root.mainloop()
```

Отже, маємо таке.
Зовнішній цикл for i in range(1, 10) спрацьовує 9 разів.
Внутрішній цикл for j in range(1, 10) спрацьовує 9 разів для однієї ітерації зовнішнього циклу й відповідно 81 раз для всіх ітерацій зовнішнього циклу.
У внутрішньому циклі формується рядок добутків чисел і та j (рис. 35.7).

Таблиця Піфагора

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

Рис. 35.7

Вивести прості числа, що належать проміжку [a; b].

За умовою задачі потрібно для кожного і від a до b перевірити, чи є число і простим. Для цього в циклі while перебираються всі числа k від 2 до i//2.

Якщо число k є дільником числа i (тобто і ділиться на k без остачі), то прапорцевій змінній prap присвоюється значення True, і за командою break перебір можливих дільників припиняється. Якщо після завершення роботи циклу while значення prap залишилося рівним False, це означає, що поточне значення і є простим числом (рис. 35.8).

```
a, b = 100, 150
for i in range(a, b+1):
    k = 2
    prap = False
    while k <= i//2:
        if i%k == 0:
            prap = True
            break
        k += 1
    if not prap: print(i)
```

```
101
103
107
109
113
127
131
137
139
149
```

Рис. 35.8

Далі ми використовуватимемо алгоритмічну структуру повторення для створення зображень і програмування руху об'єктів.

Питання для самоперевірки

1. У вікні консолі IDLE уведіть команду:
   for i in range(65, 90): print(chr(i))
   Обговоріть результати виконання команди for. Поекспериментуйте зі значеннями параметрів функції range().
2. Що буде надруковано в результаті виконання програми?
   а) for i in range(5): print(i*i);
   б) for x in 'рядок': print(2*x);
   в) for x in range(5, 1, –1): print(x).
3. Запишіть команду циклу для виконання таких дій:
   а) знайдіть суму та середнє арифметичне 10 довільних чисел, які вводять із клавіатури;
   б) із клавіатури вводять 10 довільних чисел. Знайдіть суму тільки від'ємних доданків.
4. Визначте значення змінної s після виконання операторів:

   а)
   ```
   s = 0
   i = 0
   while i<5:
       i = i+1
       s = s+i
   ```

   б)
   ```
   s = 0
   i = 5
   while i>1:
       s = s+i
       i = i–1
   ```

   в)
   ```
   s = 0
   i = 0
   while i<5:
       i = i+1
       s = s+10
   ```

5. Складіть програму для обчислення площі прямокутних трикутників, катети яких набувають значень:
   a = 10; 10.15; 10.3; 10.45; 10.6; 10.75 (м)
   b = 7; 7.4; 7.8; 8.2; 8.6; 9.0 (м)
   Скільки значень площі буде виведено?
6. Вкладник робить банківський внесок у розмірі a грн під 10 % річних (щороку сума внеску збільшується на 10 %). Напишіть програму для обчислення суми на рахунку вкладника через y років.

Вправа 35

Складіть програму для розв'язування задачі.
У поточному році обсяг деревини на території лісового масиву становив 120 000 м^3. При заготівлі деревини щорічно вирубується k м^3. Щорічний природний приріст 5,5 %.

Написати програму для визначення обсягу деревини на території через n років. Чи стане він меншим від мінімально допустимого значення (23 000 м^3)? За необхідності відкоригувати план лісозаготівлі з точністю до 100 м^3.

1. Проаналізуйте математичну модель задачі.

Дано: V_0 — початковий обсяг деревини, м^3;
k — план вирубування, м^3;
n — період часу, роки.
Що треба знайти? V_n — обсяг деревини через n років.
Який зв'язок між величинами? $V_i = V_{i-1} - k + V_{i-1} \cdot 0{,}055$, де $i = 1..n$.

2. Завантажте модуль tkinter. Розробіть інтерфейс програми згідно з рис. 35.9.

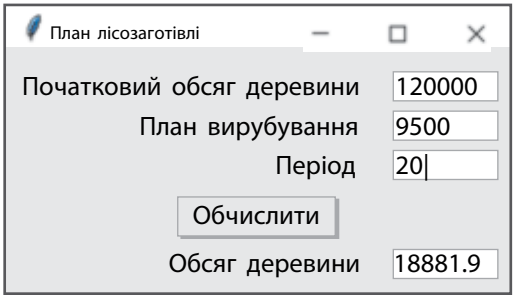


Рис. 35.9

3. Запишіть програмний код процедури обробки події для кнопки Обчислити.

```
def btn_click():
    v = float(ent1.get())
    k = float(ent2.get())
    n = int(ent3.get())
    for i in range(n):
        v = v–k+v*0.055
    ent4.delete(0, END)
    ent4.insert (0, str(round(v, 1)))
```

4. Збережіть програму у файлі з іменем Vprava35.py. Випробуйте програму для vn = 120 000, k = 9500, n = 20. Порівняйте результати з наведеними на рис. 35.9, с. 231.
   Екологічна інтерпретація отриманого розв'язку: через 20 років обсяг деревини стає меншим від допустимого мінімального значення. Додайте до програмного коду команди для виведення в консоль відповіді на питання: чи стає обсяг деревини меншим від мінімального.
5. Якщо не зменшувати план лісозаготівлі, через 20 років обсяг деревини становитиме V = 18 881,878 м^3, що менше від допустимого мінімального значення. Напишіть команду циклу для визначення кількості років без вирубування лісу, яка потрібна, щоб обсяг деревини відновився до допустимого мінімального значення (23 000 м^3).
6. Визначте максимальне значення k (план вирубування), за якого обсяг деревини за 20 років не стане меншим від допустимого мінімального значення (23 000 м^3).

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/109397).

ПРАКТИЧНА РОБОТА 12
СТВОРЕННЯ ПРОГРАМИ З ГРАФІЧНИМ ІНТЕРФЕЙСОМ

Завдання: розробити програму Міста України; залежно від стану перемикача вивести відомості та зображення.

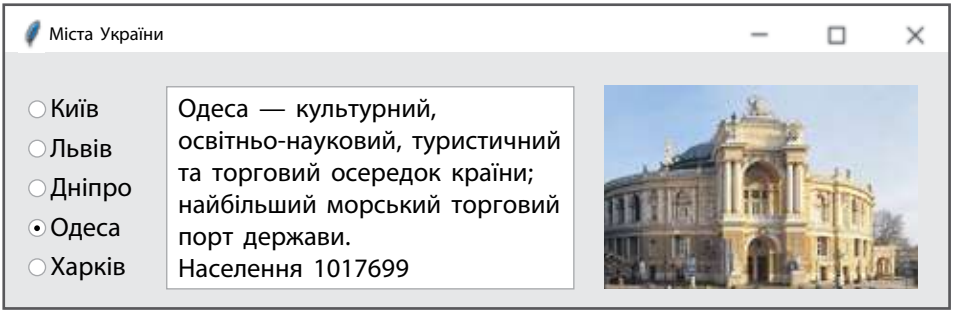
Обладнання: комп'ютер із середовищем програмування IDLE.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

В IDLE виберіть команду File → New File.

1. У текстовому редакторі підготуйте декілька файлів із короткою інформацією про міста України. Збережіть файли Київ.txt, Львів.txt тощо до папки Міста України.
2. Створіть вікно програми із заголовком Міста України. Завершіть програму командою root.mainloop(). Збережіть програму у файлі з назвою Практична12 у папку Міста України.
3. Створіть список назв міст:
   cities = ['Київ', 'Львів', 'Дніпро', 'Одеса', 'Харків']
4. Додайте у вікно віджет text1 класу Text, призначений для виведення інформації з текстових файлів:
   text1 = Text(width = 24, height = 10, wrap = WORD)
   text1.grid(row = 0, column = 1, rowspan = 5)
5. Створіть п'ять радіокнопок із підписами, що відповідають назвам міст (див. рисунок).



Оскільки значення параметрів різних радіокнопок подібні, створити їх можна в циклі за списком назв міст.

```
var = IntVar()                    # Спільна змінна для радіокнопок
var.set(0)
for i in range(len(cities)):
    Radiobutton(text = cities[i], variable = var, value = i,
                command = change).grid(row = i, column = 0, sticky='w')
```

6. Опишіть спільний обробник події для перемикачів — функцію change().
   Залежно від значення var.get зі списку cities вибирається назва міста, з якої формується ім'я файлу з інформацією про вибране місто:

```
def change():
    f = open(cities[var.get()]+'.txt')
    text1.delete(1.0, END)
    text1.insert(1.0, f.read())
    f.close()
```

7. Помістіть у папку з програмою графічні файли зі світлинами міст (імена файлів: Київ.png, Львів.png тощо).
8. Додайте до вікна об'єкт lab класу Label для виведення зображення міста.

```
image1 = None # Початкове "порожнє" значення
lab = Label(root, width = 200, height = 150)
lab.grid(row = 0, column = 2, rowspan = len(cities), padx = 20)
```

9. У тілі функції change() додайте команди для зміни властивості image об'єкта lab:

```
    global image1
    image1 = PhotoImage(file = cities[var.get()]+'.png')
    lab.config(image = image1)
```

10. Додайте в список cities ще кілька назв міст України.
11. Підготуйте текстові файли з цікавою інформацією про ці міста, графічні файли зі світлинами цих міст. Випробуйте програму.
12. Запропонуйте ідеї щодо покращення створеної програми. Спробуйте їх реалізувати.

Зробіть висновок: як використовувати елементи вибору у програмах із графічним інтерфейсом.

§ 36. ГРАФІЧНІ МЕТОДИ МОДУЛЯ TKINTER

У мові Python існують різні способи роботи з графікою. Так, ви вже знайомі з «черепашачою» графікою, що реалізується засобами модуля turtle. Також ви виводили на екран фотографії, додаючи їх на елемент керування Label.

Графічні методи модуля tkinter дають змогу гнучкіше працювати із зображеннями, а також створювати малюнки, програмуючи їх побудову з окремих геометричних фігур.

Розгляньмо, як використовуються ці засоби модуля tkinter.

Полотно для малювання

У tkinter зображення створюється в межах полотна — об'єкта класу Canvas, який теж входить до модуля tkinter. Тому, перш ніж записувати оператори для малювання, потрібно створити полотно.

Синтаксис команди створення об'єкта класу Canvas:

```
<змінна> = Canvas(<батьківський віджет>, width = <ширина>,
                  height = <висота>)
```

де width — ширина полотна; height — його висота, виражені в пікселях.

1 Створимо полотно розміром 500×500 пікселів:

```
from tkinter import*
root = Tk()
canvas = Canvas(root, width = 500, height = 500)
canvas.pack()
```

Для задавання положення точок на полотні використовують координати. Будь-яку точку можна задати парою чисел (X, Y), де X — відстань від точки до лівого краю полотна; Y — відстань від точки до верхнього краю полотна (рис. 36.1).

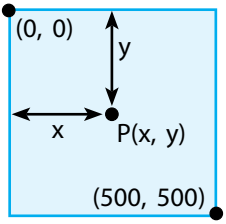


Рис. 36.1

Колір полотна, який воно отримує з налаштувань системи, можна змінити за допомогою оператора:

```
canvas.config(bg = 'blue')  # blue замініть назвою потрібного кольору
```

Таким чином, об'єкт Canvas має низку методів, призначених для малювання відрізків, прямокутників, кіл, дуг, виведення тексту тощо.

Методи для малювання фігур

Розгляньмо детальніше використання графічних методів полотна Canvas.

Відрізок

Метод create_line креслить відрізок від точки з координатами (x1, y1) до точки з координатами (x2, y2):

```
canvas.create_line(x1, y1, x2, y2)
```

2 Намалюємо відрізок від верхнього лівого кута полотна до правого нижнього кута:

```
canvas.create_line(0, 0, 500, 500)
```

Прямокутник

Метод create_rectangle малює незаповнений прямокутник, у якого протилежні кути збігаються з точками (x1, y1) і (x2, y2). Сторони прямокутника паралельні сторонам екрана:

```
canvas.create_rectangle(x1, y1, x2, y2)
```

3 Намалюємо квадрат зі стороною завдовжки 100 пікселів:

```
canvas.create_rectangle(220, 30, 319, 129)
```

Еліпс

Метод canvas.create_oval малює еліпс, вписаний у прямокутник, який задано координатами двох протилежних вершин (x1, y1) і (x2, y2) (рис. 36.2):

```
canvas.create_oval(x1, y1, x2, y2)
```

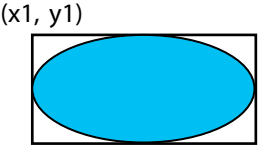


Рис. 36.2

4 Намалюємо коло діаметром 100 пікселів:

```
canvas.create_oval(220, 30, 319, 129)
```

Многокутник

Метод canvas.create_polygon малює многокутник за координатами його вершин (рис. 36.3):

```
canvas.create_polygon(x1, y1, x2, y2, …,
                      xn, yn)
```

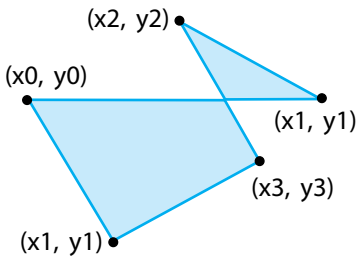


Рис. 36.3

5 Накреслимо трикутник.

```
points = [10, 210, 100, 210, 100, 300]
canvas.create_polygon(points)
```

За замовчуванням контури всіх об'єктів і заливка многокутника мають чорний колір.

Малювання тексту

Метод create_text() використовується для малювання тексту на полотні:

```
canvas.create_text(x, y, text)
```

За замовчуванням у точці (x, y) розташовується центр текстового напису. Щоб змінити це і, наприклад, розмістити за заданими координатами ліву межу тексту, використовують параметр anchor зі значенням W (від англ. *West* — захід).

Для оформлення тексту можна використовувати різні параметри форматування — колір тексту (fill), розмір символів і шрифт (font) тощо. Текст на полотні також є графічним об'єктом, тому можна задати кут повороту тексту (параметр angle).

Найчастіше текст на полотні використовують для підписів до геометричних фігур, схем, діаграм тощо.

6 Потрібно вивести текст на полотно, як наведено на рис. 36.4.

```
canvas.create_text(250, 100, text = "Hello
                   World", font = "Arial 30")
canvas.create_text(150, 200, text = "Hello",
     font = "Arial 25", fill = 'red', angle = 45)
canvas.create_text(300, 200, text = "World",
 font = "Arial 25", fill = 'green', angle = 120)
```

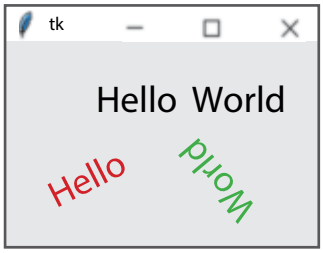


Рис. 36.4

Щоб **створити кольорові рисунки**, до переліку параметрів графічних методів потрібно додати параметри fill (колір заливки) і outline (колір контуру). Можна також збільшити товщину лінії, задавши значення параметра width. Щоб отримати багатокутник без заливки, слід обов'язково задати значення параметра fill = ''.

7 Намалюємо червоний круг із синім контуром завтовшки 3 пікселі:

```
canvas.create_oval(20, 30, 120, 130, fill = 'red', outline = 'blue', width = 3)
```

Виведення зображень із графічних файлів

Засобами tkinter можна **вивести на полотно зображення з графічного файлу**, наприклад, фотографію. Для цього потрібно:

1) графічні дані із файлу завантажити до змінної класу PhotoImage, зазначивши як параметр ім'я файлу (file = '<ім'я файлу>');
2) викликати метод create_image.

8 Виведемо на полотно рисунок picture1.gif, який міститься в кореневому каталозі диска D:

```
my_image = PhotoImage(file = 'd:\\picture1.gif')
canvas.create_image(0, 0, anchor = NW, image = my_image)
```

Параметр anchor визначає розташування рисунка на полотні. Значення NW вказує на верхній лівий кут полотна (рис. 36.5).

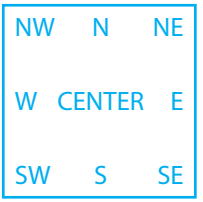


Рис. 36.5

Засобами tkinter можна завантажувати зображення з файлів розширеннями .gif і .png.

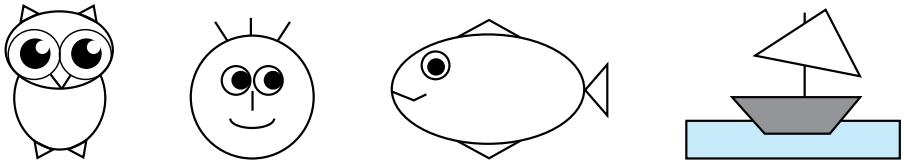
! Методи canvas.create_... повертають числове значення — ідентифікатор створеного об'єкта.

Ідентифікатор об'єкта зручно зберігати в змінній для подальших дій із об'єктом: ball = canvas.create_oval(10, 10, 30, 30) — змінна ball зберігатиме ідентифікатор круга.

Отже, за допомогою графічних методів Canvas у мові Python можна створювати рисунки як комбінацію геометричних фігур.

Питання для самоперевірки

1. Опишіть систему координат полотна Canvas.
2. Опишіть алгоритм створення полотна Canvas.
3. Як вивести на полотно зображення з графічного файлу?
4. Як побудувати круг зеленого кольору радіусом 100, центр якого збігається з центром полотна?
5. Запишіть фрагмент програми для побудови трикутника з вершинами в точках (100, 100), (150, 100), (80, 70); колір ліній — червоний, заливка відсутня.
6. Складіть програму для побудови зображення (рис. 36.6).



а б в г

Рис. 36.6

Вправа 36

Скласти програму для заповнення полотна випадковими прямокутниками.

В IDLE виберіть команду File → New File.

1. Імпортуйте необхідні модулі: tkinter, random.
2. Створіть графічне вікно й додайте полотно розміром 200 × 200.
3. Опишіть функцію draw_rectangle() для накреслення прямокутника, координатами вершин якого є випадкові числа.

   Функція draw_rectangle() має два аргументи width і height, і будує прямокутник із випадковими координатами лівого верхнього кута в діапазонах (0, width) і (0, height).

```
def draw_rectangle(width, height):
    x1 = randint(0, width); y1 = randint(0, height)
    x2 = randint(0, width); y2 = randint(0, height)
    canvas.create_rectangle(x1, y1, x2, y2)
```

4. З основної програми викличте в циклі for 30 разів функцію draw_rectangle():
```
for x in range(0, 30): draw_rectangle(200, 200)
```
Запустіть програму — полотно заповниться чорними прямокутниками з випадковими координатами вершин (рис. 36.7).

5. Далі змініть функцію draw_rectangle, додавши аргумент fill_color — колір заливки прямокутника. До переліку параметрів методу create_rectangle додайте параметр fill і присвойте йому значення fill_color. Після запуску полотно буде заповнене кольоровими прямокутниками (рис. 36.8).
```
def draw_rectangle(width, height, fill_color):
    x1 = randint(0, width); y1 = randint(0, height)
    x2 = randint(0, width); y2 = randint(0, height)
    canvas.create_rectangle(x1, y1, x2, y2, fill = fill_color)
c = ['red', 'green', 'blue', 'yellow']
for x in range(0, 30):
    draw_rectangle(200, 200, choice(c))
```

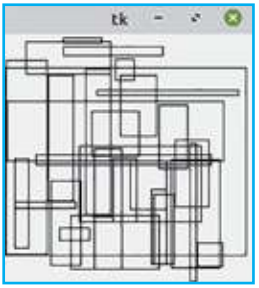


Рис. 36.7

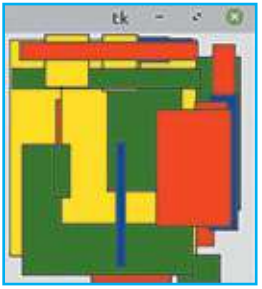


Рис. 36.8

6. Змініть функцію draw_rectangle так, щоб малювалися кола з випадковими координатами.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/109398).

§ 37. СТВОРЕННЯ АНІМАЦІЇ

Як ви вже знаєте, ілюзія руху об'єктів в анімації виникає завдяки швидкому зміненню зображень на екрані. Кожна з фігур, які ми будували на полотні, є окремим об'єктом. Це дає змогу програмувати їх рух і взаємодію, створюючи анімаційний ефект.

Рух графічних об'єктів

Ознайомимося з особливостями методів, призначених для переміщення об'єкта полотном.

- Метод canvas.move переміщає графічний об'єкт, змінюючи значення його координат на задані величини:
  canvas.move(item, x, y),
  де item — ідентифікатор, який повертає конструктор графічного об'єкта; x, y — значення зміщення по горизонталі та вертикалі (у пікселях).
- Метод tk.update() примусово оновлює зображення.
- Метод time.sleep() затримує виконання програми:
  time.sleep(t),
  де t — величина затримки (у секундах).

Організуємо горизонтальний рух круга полотном. Ідентифікатор об'єкта збережемо в змінній a:

```
import time
from tkinter import*
tk = Tk()
canvas = Canvas(tk, width = 400, height = 400)
canvas.pack()
a = canvas.create_oval(10, 10, 30, 30, fill = 'red')
for x in range(0, 60):
    canvas.move(a, 5, 0)          # Переміщення об'єкта
                                  # з ідентифікатором a
    tk.update()
    time.sleep(0.05)
```

Блок коду в циклі for призначено для пересування круга полотном. Так, команда canvas.move(a, 5, 0) пересуне об'єкт, ідентифікатор якого в змінній a (це і є круг) на 5 пікселів праворуч і 0 пікселів по вертикалі. Для пересування ліворуч знадобилася б команда canvas.move(a, –5, 0). Для руху по діагоналі слід змінювати одночасно координати x і y: canvas.move(a, 5, 5).

Програмування реакції об'єктів на події

Можна зробити так, щоб об'єкт рухався у відповідь на подію, яка відбулася під час виконання програми, — пересування миші, натискання певної клавіші. Для обробки події потрібно описати функцію (обробник події) і пов'язати об'єкт із подією й функцією, викликавши метод bind().

Змусимо круг рухатися праворуч у разі натискання клавіші зі стрілкою «→». Програмний код виглядатиме так:

```
from tkinter import*
def move_oval(event):          # Заголовок функції – обробника події
    canvas.move(1, 5, 0)
tk = Tk()
canvas = Canvas(tk, width = 400, height = 400)
canvas.pack()
canvas.create_oval(10, 10, 30, 30, fill = 'red')
tk.bind('<KeyPress-Right>', move_oval)
```

Функція має єдиний аргумент event, через який передається інформація про подію. Об'єкт event має властивість keysym, яка набуває значення символу натиснутої клавіші. Якщо натиснуто клавішу «→», це означає, що відбулася подія '<KeyPress-Right>', і властивість event.keysym отримує значення 'Right' (рис. 37.1).

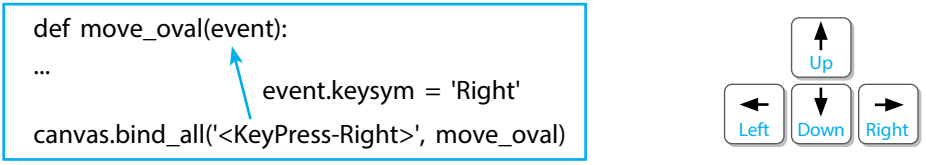


Рис. 37.1

Буде цікавіше змінювати напрямок руху круга, застосовуючи всі чотири клавіші зі стрілками. В тілі функції move_oval() запрограмуємо зміну координат об'єкта залежно від того, яке значення передано для аргументу event:

```
def move_oval(event):
    if event.keysym == 'Up': canvas.move(1, 0, –3)
    elif event.keysym == 'Down': canvas.move(1, 0, 3)
    elif event.keysym == 'Left': canvas.move(1, –3, 0)
    elif event.keysym == 'Right': canvas.move(1, 3, 0)
```

У функції move_oval() перевіряється, чи містить змінна keysym значення 'Up' (з англ. *вгору*). Якщо так, викликається метод canvas.move(1, 0, –3) і круг зміщується вгору. Якщо ні, то перевіряється, чи містить змінна keysym значення 'Down' (з англ. *вниз*), якщо так, робиться крок униз. Так само опрацьовується значення 'Left' (з англ. *вліво*) і 'Right'. Таким чином, при натисканні на клавішу об'єкт пересувається в заданому напрямку.

Далі повідомимо tkinter, що функцію move_oval() треба застосовувати для обробки натискання будь-якої клавіші:

```
canvas.bind('<Key>', move_oval)
```

Створення анімації з використанням графічних файлів

Використовуючи принцип анімації, можна **запрограмувати рух об'єкта на полотні** в такий спосіб:

1) підготувати декілька графічних файлів із зображеннями фаз руху деякого об'єкта;
2) запрограмувати послідовне завантаження на полотно зображень із цих файлів;
3) між завантаженнями слід затримати виконання програми для того, щоб користувач устиг побачити чергове зображення.

Створимо анімацію Політ метелика, яка складається з трьох кадрів. Розглянемо кроки створення анімації.

1. Підготуємо три файли з зображеннями фаз руху метелика (рис. 37.2). Імена файлів зберігатимуться в списку anim.

1.gif

2.gif

3.gif

Рис. 37.2

2. Анімацію реалізує така програма:

```
import time
from tkinter import *
def play():
    global cur_frame
    canvas.itemconfig(pic, image = frames[cur_frame])
    cur_frame = (cur_frame+1)%3
    root.update()                  # Оновлення полотна
    root.after(100, play)          # Повторний виклик функції play()
root = Tk()
root.title('Політ метелика')
root.geometry('200x250')
canvas = Canvas(root, width = 200, height = 200)
canvas.place(x = 1, y = 1)
anim=['1.gif', '2.gif', '3.gif']   # Список імен файлів
frames = []                        # Читаємо дані зображень із файлів,
for name in anim:                  # імена яких у списку anim
    frames.append(PhotoImage(file = name))      # у список frames
cur_frame = 0                      # Номер поточного зображення
pic = canvas.create_image(0, 0, anchor = NW)
play()
root.mainloop()
```

Проаналізуємо опис функції play(), яку призначено для зміни зображень. Спочатку на екран виводиться зображення, номер якого зберігає змінна cur_frame. Потім значення cur_frame коригується, щоб указувати на наступний кадр. Ураховується довжина списку зображень, тому їх кількість у списку можна збільшити й це не викличе помилки в програмі.

Далі викликом методу root.after із аргументами (100; play) планується повторний виклик функції play() через 100 мс, тобто наступна зміна зображення.

3. Збережемо файл і випробуємо програму з анімацією.

Отже, засоби бібліотеки tkinter дають можливість створювати програми з анімаційними ефектами, а також додавати реакцію на дії користувача шляхом програмування обробки події.

Питання для самоперевірки

1. Які функції призначено для переміщення об'єкта по полотну?
2. Як запрограмувати реакцію об'єкта на подію?
3. Поясніть призначення аргументів методу:
   canvas.bind('<KeyPress-Up>', step)
4. Як створити анімацію шляхом послідовної зміни зображень?
5. Як зміниться Політ метелика (див. приклад 4), якщо у виклику root.after() замість 100 написати 1000?
6. Створіть програму, що показує, як зоряним небом рухається космічний зонд Кассіні.

Вправа 37

Створити програму, в якій буде реалізовано рух круга по полотну.
Об'єкт має змінювати положення у відповідь на натискання клавіш керування курсором, залишаючи слід на полотні (рис. 37.3).
В IDLE виберіть команду File → New File.

Рис. 37.3

1. Підключіть модуль tkinter. Створіть вікно програми й додайте полотно розміром 400×400:
```
from tkinter import*
tk = Tk()
canvas = Canvas(tk, width = 400, height = 400)
canvas.pack()
```
2. Намалюйте в лівому верхньому куті червоний круг. Ідентифікатор створеного круга зберігатиметься в змінній ball:
```
ball = canvas.create_oval(10, 10, 30, 30, fill = 'red')
```

Слід круга утворюватимуть відрізки, що сполучають центри послідовних його положень.

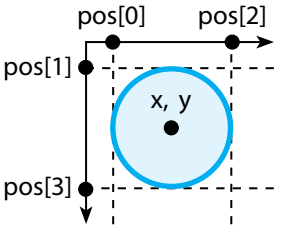


Рис. 37.4

3. Для обчислення координат центра круга (рис. 37.4) опишіть функцію midpoint:

```
def midpoint(pos):
    # pos — список координат x1, y1, x2, y2
    return[(pos[0]+pos[2])/2, (pos[1]+pos[3])/2]
```

4. Опишіть функцію move_oval(). У ній будуть змінюватися координати круга залежно від того, яку клавішу натиснуто (параметр event), а також малюватиметься відрізок між центрами круга в початковому і кінцевому положеннях:

```
def move_oval (event):
    step = 3                                  # Величина зміщення
    P1 = midpoint(canvas.coords(ball))        # Зберігаємо початкові
    if event.keysym == 'Up':                  # координати
        canvas.move(ball, 0, –step)
    elif event.keysym == 'Down':
        canvas.move(ball, 0, step)
    elif event.keysym == 'Left':
        canvas.move(ball, –step, 0)
    elif event.keysym == 'Right':
        canvas.move(ball, step, 0)
    canvas.create_line(P1, midpoint(canvas.coords(ball)))
```

5. Запишіть виклик методу tk.bind для встановлення зв'язку між подією Натискання на клавішу і функцією move_oval: tk.bind('<Key>', move_oval).

Збережіть файл із назвою Vprava37 і випробуйте програму.

6. Змініть товщину й колір сліду, який залишає круг; додайте діагональні переміщення за допомогою клавіш A, S, Z, X.

Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/109399).

ПРАКТИЧНА РОБОТА 13
СТВОРЕННЯ ІГРОВОЇ ПРОГРАМИ

Завдання: скласти ігрову програму Відбий кульку.
Кулька з'являється на полотні у випадковому місці й починає рух із відбиттям від меж вікна. Гравець не дає кульці торкнутися нижньої межі полотна. Гра триває до 5 промахів.

Обладнання: комп'ютер із середовищем програмування IDLE.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

В IDLE виберіть команду File → New File.

1. Імпортуйте модулі tkinter, random, time.
2. Створіть вікно програми tk із заголовком Рух кульки. Завершіть програму оператором tk.mainloop(). Збережіть файл із назвою Pract13.
3. Додайте об'єкт canvas класу Canvas розміром 500×400.
4. Створіть список колірних констант:
colors = ['red', 'orange', 'yellow', 'green', 'blue']
5. Створіть об'єкт b за допомогою методу create_oval. Колір заповнення кульки вибирається випадково зі списку colors:
b = canvas.create_oval(10, 150, 60, 200, fill = choice(colors))
6. Створіть змінну k — лічильник «життів»: k = 5
7. Рух кульки починається після натискання на ліву клавішу миші: canvas.bind('<1>', click)
8. Створіть об'єкт r класу Label — «ракетку».
r = Label(tk, width = 12, bg = 'blue')
r.place(x = 10, y = 320)
9. Ракетка рухатиметься вправо і вліво після натискання на клавіші керування курсором.
r.focus_set()
r.bind('<Key>', r_move)

10. Опишіть обробник події Натискання на клавіші керування курсором:

```
def r_move(event):
    if event.keysym == 'Left': step = –10
    elif event.keysym == 'Right': step = 10
    else: return
    event.widget.place(x = event.widget.winfo_x()+step)
```

11. Опишіть обробник події Натискання на ліву клавішу миші, у якому запускається рух кульки в нескінченному циклі. *Якщо кулька торкається межі полотна*, вона змінює напрямок руху. Метод move пересуває об'єкт b. *Якщо кулька торкнулася нижньої межі полотна*, кількість «життів» зменшується на 1. Після 5 промахів гра закінчується.

```
def click(event):
    global k
    dx = dy = 5                    # Задання кроку руху
    while True:                    # Початок нескінченного циклу
      # Отримуємо координати кульки
      x, y = cnv.coords(b)[0], cnv.coords(b)[1]
      if x>450 or x<10: dx = –dx       # Якщо об'єкт досягає меж
      if y>350:                        # полотна, ...
        k –= 1
        dy = –dy       # напрям руху змінюється на протилежний
      if y<10: dy = –dy
      if x>r.winfo_x()and x<r.winfo_x()+70 and y>270: # Якщо кульку
        dy = –dy                                      # відбито ракеткою
      canvas.move(b, dx, dy)       # Пересування кульки
      tk.title('Залишилось життів – '+ str(k))
      if k == 0:                   # Якщо допущено 5 промахів
        canvas.create_text(250, 100, text = "GAME OVER!")
        break
      tk.update()
      sleep(0.05)            # Затримка виконання програми
```

12. Поекспериментуйте з програмою.

Зробіть висновок: як реалізувати рух об'єктів на полотні.

РОЗДІЛ 6
ПРАКТИКУМ З ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Практична робота 14. Створення текстового документа, що містить об'єкти різних типів

Практична робота 15. Створення багатосторінкового електронного документа

Практична робота 16. Розробка застосунку «Піаніно»

Практична робота 17. Створення програми з графічним інтерфейсом

Практична робота 18. Створення програми «Калькулятор»

Практична робота 19. Моделювання руху об'єкта

ПОВТОРЮЄМО

Ви вже здобули певні знання та навички:

- навчилися виконувати кодування й декодування даних, детальніше ознайомилися з апаратним забезпеченням комп'ютера;
- дізналися, як працювати з об'єктами текстового документа; опанували опрацювання текстових документів онлайн на хмарному диску під час спільної роботи;
- навчилися створювати й опрацьовувати такі об'єкти мультимедіа, як звук і відео, опанували процес створення відеофільму та способи його розміщення в інтернеті;
- продовжили вивчення основ програмування мовою Python і навчилися розробляти програми з графічним інтерфейсом; знаєте, як задавати властивості найуживаніших об'єктів графічного інтерфейсу та програмувати обробку подій.

1. Як додати до документа автоматизований зміст?
2. Який порядок використання стилів у текстовому документі?
3. Як мовою Python запрограмувати відтворення звуку з файлу?
4. Опишіть відмінності в призначенні пакувальників pack, place і grid.
5. Які властивості спільні для багатьох типів віджетів tkinter?

У цьому розділі ви створите різні інформаційні продукти, користуючись інструментами програмних середовищ, із якими ознайомились у 8 класі.

ПРАКТИЧНА РОБОТА 14
СТВОРЕННЯ ТЕКСТОВОГО ДОКУМЕНТА, ЩО МІСТИТЬ ОБ'ЄКТИ РІЗНИХ ТИПІВ

Завдання: створити мінігазету — документ, який містить об'єкти різних типів, здійснити форматування його об'єктів, створити автоматизований зміст.

Обладнання: комп'ютер із текстовим процесором, під'єднаний до інтернету.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Запустіть текстовий процесор і створіть новий документ.
2. Створіть титульну сторінку мінігазети, на якій розмістіть назву, логотип, гасло або підзаголовок. Зазначте дату випуску й номер. Додайте своє прізвище, ім'я й клас у нижній частині сторінки як відомості про відповідальну особу.
3. Підготуйте статтю (3–4 абзаци) на певну тему (наприклад, опис шкільного проєкту, огляд благодійної виставки).
4. Розмістіть текст на наступній сторінці й відформатуйте.
5. Вставте відповідне зображення та додайте під ним підпис.
6. Додайте таблицю, наприклад, для розкладу подій або шкільних заходів. Налаштуйте в ній колір меж, стиль тексту, розмір клітинок.
7. Додайте нумерований або маркований список фактів, пов'язаних із темою статті або шкільними новинами.
8. Оформте колонтитули: у верхньому або нижньому колонтитулі додайте назву газети чи номер сторінки.
9. Для першої сторінки встановіть у верхньому колонтитулі свої прізвище, ім'я, клас.
10. Вставте текстове поле, створіть оголошення (наприклад, про гурток або захід у школі). Для привернення уваги використайте різні шрифти, накреслення та яскраві кольори.

11. Для назв статті й таблиці встановіть стиль Заголовок 1. Створіть автоматизований зміст газети.
12. Збережіть файл на хмарному диску. Надайте доступ до файлу вчителю / вчительці та кільком однокласникам / однокласницям.

Зробіть висновок: які особливості форматування текстових об'єктів було використано при оформленні документа.

ПРАКТИЧНА РОБОТА 15
СТВОРЕННЯ БАГАТОСТОРІНКОВОГО ЕЛЕКТРОННОГО ДОКУМЕНТА

Завдання: створити текстовий документ, який складається з кількох розділів, здійснити форматування його об'єктів, створити зміст і покажчик в автоматизованому режимі.

Обладнання: комп'ютер із текстовим процесором, під'єднаний до інтернету.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

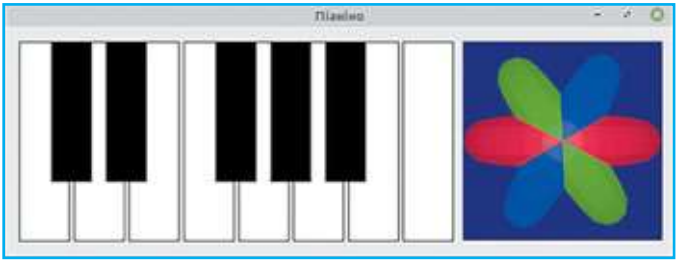
1. Запустіть текстовий процесор і створіть новий документ.
2. Створіть першу сторінку як обкладинку електронної книги. Додайте назву книги «Путівник по моєму місту» («Путівник улюблених місць» або на свій вибір), своє ім'я й клас як автора / авторки.
   Використайте великий шрифт, градієнтний фон, рамку або інші візуальні ефекти.
3. Додайте на обкладинку зображення, що символізує ваше місто або населений пункт, де ви зараз перебуваєте, чи хотіли б побувати (наприклад, фото відомого місця або пам'ятки).

4. Продумайте й створіть структуру книги: розділи про історію міста, цікаві місця, культурні заходи, визначні пам'ятки, а також корисні поради для відвідувачів.
5. Створіть Розділ 1 «Історія міста». До заголовка розділу застосуйте стиль Заголовок 1.
   Напишіть короткий текст про історію міста. Використайте заголовки різних рівнів (наприклад, «Стародавня історія», «Сучасність», «Ушанування Героїв»). Доберіть відповідне зображення.
6. Створіть Розділ 2 «Визначні місця». Створіть кілька підрозділів з описом популярних місць (наприклад, Алея Слави, парк, заповідник, музей, архітектурна пам'ятка). Додайте фото кожного місця й надайте до них стислі підписи. Використайте марковані списки для подання інформації (наприклад, адреса, час роботи, вхідна плата тощо).
7. Створіть Розділ 3 «Культурні заходи». Опишіть щорічні події у вашому місті (День Незалежності України, День захисників і захисниць України, спортивні заходи, інше — на вибір). Використайте таблицю для представлення дат цих заходів, їх локацій і стислих описів.
8. Створіть Розділ 4 «Поради для туристів» із порадами для гостей міста (наприклад, де можна зупинитися, що скуштувати, як краще пересуватися). Додайте цитати з відгуків туристів або місцевих жителів, використовуючи окремий стиль.
9. Додайте колонтитули з назвою книги, номером сторінки або прізвищем та ім'ям автора / авторки. Використайте різні колонтитули для парних і непарних сторінок (наприклад, ліворуч — назва книги, праворуч — номер сторінки).
10. Додайте покажчик.
11. Додайте зміст, використовуючи функцію автоматичного створення змісту.
12. Додайте список використаних джерел і корисних ресурсів.

Зробіть висновок: які вміння потрібні для роботи зі складним текстовим документом.

ПРАКТИЧНА РОБОТА 16
РОЗРОБКА ЗАСТОСУНКУ «ПІАНІНО»

Завдання: створити застосунок Піаніно (див. рисунок). У правій частині вікна має виводитись анімаційний фрагмент, а при натисканні на «клавіші» — відтворюватися звуки.



Обладнання: комп'ютер із середовищем програмування мовою Python; папки notes (із файлами формату MP3 зі звуками нот) і frames (із файлами формату PNG із кадрами анімації)*.

Теоретичні відомості

Один і той самий обробник події можуть викликати декілька різних об'єктів. З'ясуймо, як зробити, щоб результат виклику для кожного об'єкта був іншим. Наприклад, щоб клацання на різних прямокутниках відтворювало різні звуки.

Під час виклику обробника в нього передається значення (позначмо його event). Воно містить опис події, зокрема й ідентифікатор об'єкта, який спричинив виклик. Дізнатися ідентифікатор об'єкта можна так:

```
def on_click(event):
    ...
    key_id = event.widget.find_withtag("current")[0]
    ...
```

Тут ідентифікатор об'єкта збережено в змінній key_id. Його можна використати для подальшого опрацювання події.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. В IDLE виберіть команду File → New File.
2. Запрограмуйте створення вікна програми (назва об'єкта root) із заголовком Піаніно і полотном canvas шириною 670 і висотою 230 пікселів.
3. Підготуйте дані та структури даних для малювання клавіш:
   - порожні списки, які буде сформовано пізніше:

```
keys = []      # Для ідентифікаторів клавіш
sounds = []    # Для імен файлів з відповідними клавішам звуками
```

4. Створіть функцію для опрацювання натискання клавіші:

```
def on_click(event):
    # Дізнаємося ідентифікатор натиснутої клавіші
    key_id = event.widget.find_withtag("current")[0]
    # За ідентифікатором знаходимо ім'я файлу й відтворюємо звук
    playsound(sounds[keys.index(key_id)])
```

5. Створіть функцію для малювання прямокутника відповідного кольору та налаштування його для відтворення звуку при клацанні:

```
def PianoKey(x, y, color, soundfile):
    if color == 'black': w, h = 40, 140      # Розміри чорної клавіші
    else: w, h = 50, 200                     # Розміри білої клавіші

    key_id = canvas.create_rectangle(x, y, x+w, y+h, fill = color)
    keys.append(key_id)          # "Запам'ятовуємо" ідентифікатор...
    sounds.append(soundfile)     # та ім'я файлу зі звуком

    canvas.tag_bind(key_id, '<1>', on_click)
```

6. Для перевірки створеного коду додайте до основної програми команди:

```
PianoKey(15, 15, 'white', 'do.mp3')          # Пробний виклик
root.mainloop()
```

Збережіть програму у файлі Практична робота 16 у папці, що містить папки notes і frames. При клацанні прямокутника має звучати нота до.

7. Замість рядка з пробним викликом запишіть списки параметрів білих клавіш:
   - список координат x лівих верхніх кутів клавіш на полотні (координати y — однакові):
     coords = [15, 70, 125, 180, 235, 290, 345, 400]
   - список імен файлів зі звуками:
     slst = ['do.mp3', 're.mp3', 'mi.mp3', 'fa.mp3', 'sol.mp3', 'la.mp3', 'si.mp3', 'do2.mp3']
8. Запишіть цикл для малювання клавіш, описаних у списках coords і slst (від англ. *sound list* — список звуків):
   ```
   for key in range(len(coords)):
       PianoKey(coords[key], 15, 'white' if key < 8 else 'black',
                './notes/'+slst[key])
   ```
   Перевірте роботу програми.
   У вікні має з'явитися рядок білих клавіш, кожна з яких «звучить», якщо її натиснути. Поясніть, як виконується цей цикл.
   Доповніть список coords координатами чорних клавіш (47, 102, 212, 267, 322), а список slst — іменами відповідних файлів ('do-d.mp3', 'mi-b.mp3', 'fa-d.mp3', 'sol-d.mp3', 'si-b.mp3'). Перевірте роботу програми.
9. Створіть список імен файлів із кадрами анімації з папки frames (допишіть імена решти графічних файлів):
   ```
   plst = ['f01.png', 'f02.png', 'f03.png', ... ,'f24.png', 'f25.png']
   pfld = './frames/'                  # Шлях до папки з зображеннями
   ```
   Запрограмуйте читання графічних даних у список frms:
   ```
   frms = []                           # Читаємо дані зображень із файлів, ...
   for name in plst:                   # імена яких у списку plst, ...
       frms.append(PhotoImage(file = pfld+name))    # у список frms
   ```
10. Створіть на полотні зображення на основі графічних даних першого кадру:
    ```
    cur_frame = 0                          # Номер поточного кадру
    img = canvas.create_image(560, 115, image = frms[cur_frame])
    ```
11. Створіть функцію, яка неперервно виводитиме кадри анімації з інтервалом 100 мс:

```
def show_video():
    global cur_frame
    cur_frame  = (cur_frame+1)%len(frms)
    canvas.itemconfig(img, image = frms[cur_frame])
    # Змінюємо малюнок
    root.update()                    # Оновлюємо зображення полотна
    root.after(100, show_video)    # Повторний виклик функції play()
```

12. Додайте до програми виклик функції:
show_video()
Випробуйте роботу програми, спробуйте зіграти мелодію. Які особливості роботи програми ви помітили?
Запропонуйте варіанти вдосконалення програми, спробуйте їх реалізувати.

Зробіть висновок: які можливості відкриває винайдення способів кодування та зберігання звуку й відео.

ПРАКТИЧНА РОБОТА 17
СТВОРЕННЯ ПРОГРАМИ З ГРАФІЧНИМ ІНТЕРФЕЙСОМ КОРИСТУВАЧА

Завдання: розробити програму для розв'язування задачі.
Організації часто доводиться фарбувати баки циліндричної форми. Фарбувати потрібно лише верхню частину й зовнішню поверхню стінок. Створіть програму для визначення, скільки банок фарби потрібно для фарбування певного бака.

Обладнання: комп'ютер із середовищем програмування мовою Python.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Проаналізуйте математичну модель задачі.

Дано: d — діаметр бака, м;
h — висота бака, м;
b — витрати фарби (площа поверхні, яку можна пофарбувати однією банкою фарби), м^2;

Що треба знайти? k — кількість банок.

Який зв'язок між величинами?

$s_{\text{дах}} = \frac{\pi d^2}{4}$ — площа даху бака (круга);

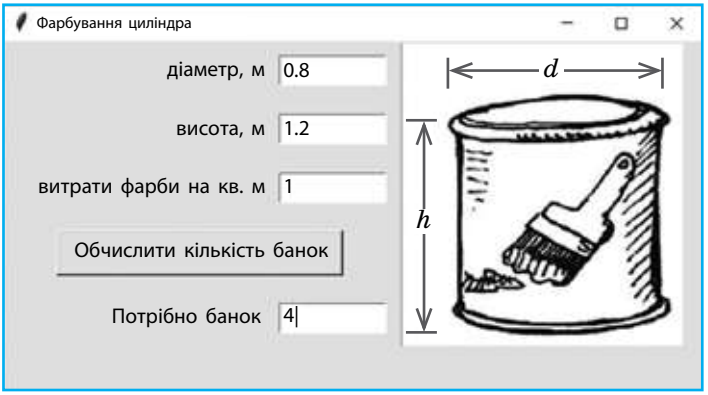
$s_{\text{бічна}} = \pi d h$ — площа бічної поверхні;

$k = \frac{(s_{\text{дах}} + s_{\text{бічна}})}{b}$

Які є обмеження на дані? $d > 0$, $h > 0$, $b > 0$.

В IDLE виберіть команду File → New File.

2. Завантажте модулі tkinter, math.
3. Створіть вікно програми розмірами 600×300 із заголовком Фарбування циліндра (див. рисунок). Завершіть програму командою root.mainloop().



4. Розмістіть у вікні віджет Label із написом діаметр, м:
```
lab1 = Label(root, text = 'діаметр, м', width = 20, font = 'Arial 14',
             anchor = E)
lab1.grid(row = 0, column = 0)
```
5. Розмістіть у вікні віджет Entry, щоб задати значення діаметра:
```
ent1 = Entry(root, width = 8, font = 'Arial 14')
ent1.grid(row = 0, column = 1, padx = 10)
```
6. Згідно з рисунком додайте до вікна об'єкти класу Label для виведення написів.
7. Згідно з рисунком додайте до вікна об'єкти класу Entry для введення значень висоти бака та площі поверхні, яку можна зафарбувати однією банкою фарби, і для виведення кількості банок для фарбування всієї поверхні.
8. Додайте до вікна об'єкт btn класу Button:
```
btn = Button(root, text = 'Обчислити кількість банок',
             font = 'Arial 14', command = btn_click)
btn.grid(row = 3, column = 0, columnspan = 2)
```
9. Опишіть обробник події Натискання кнопки для об'єкта btn. Виконання обробника події повинно привести до виведення шуканої кількості банок до поля ent4. Дробове значення числа банок потрібно округлити до більшого цілого.
```
d = float(ent1.get())
h = float(ent2.get())
b = float (ent3.get())
tp = pi*d**2/4
wall = pi*d*h
k = ceil((tp+wall)/b)
ent4.insert(0, str(k))
```
10. Щоб додати до вікна програми зображення, розмістіть у вікні об'єкт класу Label, запишіть у змінну mal назву вибраного вами файлу і надайте атрибуту image об'єкта значення змінної mal:
```
mal = PhotoImage(file = 'bak.png')
Label(image = mal).grid(row = 0, column = 2, rowspan = 5)
```
11. Збережіть програму у файлі з назвою Pract17.

12. Створіть три тестових набори вхідних даних. Перевірте роботу програми на ваших тестових даних.

Зробіть висновок: як організувати діалог із користувачем у програмі з графічним інтерфейсом користувача.

ПРАКТИЧНА РОБОТА 18
СТВОРЕННЯ ПРОГРАМИ «КАЛЬКУЛЯТОР»

Завдання: скласти програму Калькулятор для конвертування десяткового числа у двійкове і навпаки.

Обладнання: комп'ютер із середовищем програмування мовою Python.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

В IDLE виберіть команду File → New File.

1. Завантажте модуль tkinter.
2. Створіть вікно програми із написом Калькулятор (рис. 1).

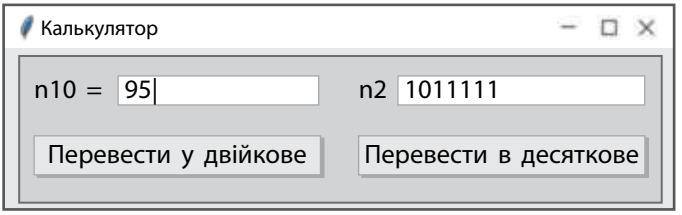


Рис. 1

Завершіть програму командою root.mainloop().
Поміркуйте, який пакувальник буде найзручнішим.

3. Додайте у вікно програми віджети lab1, lab2 класу Label, entry1, entry2 класу Entry, згідно з рис. 1.
4. Створіть віджет b1 класу Button із написом Перевести у двійкове.
5. Створіть обробник події для кнопки Перевести у двійкове. Для переведення десяткового числа у двійкову систему числення використайте алгоритм (рис. 2):
   1) створити порожній рядок dv;
   2) зчитати десяткове число dc із поля віджета entry1;
   3) поки dc>0, повторювати:
      — знайти r — остачу від ділення dc на 2;
      — перетворити r на рядок і додати в початок рядка dv;
      — поділити dc цілочисельно на 2;
   4) вивести значення dv у поле віджета entry2.

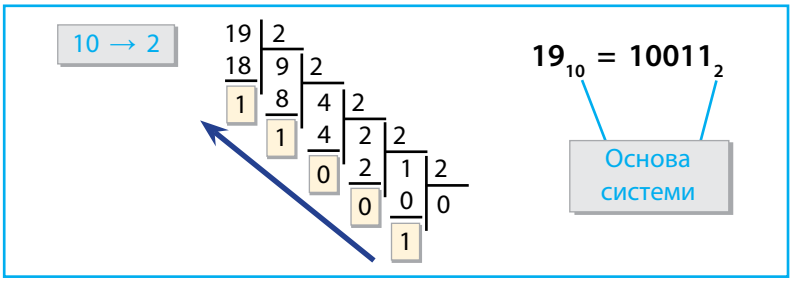


Рис. 2

6. Створіть віджет b2 класу Button із написом Перевести в десяткове.
7. Створіть обробник події для кнопки Перевести в десяткове. Для переведення двійкового числа в десяткову систему числення використайте алгоритм (див. рис. 2):
   1) зчитати рядок dv із поля віджета entry2;
   2) створити змінну для накопичення суми dc = 0;
   3) обчислити суму всіх цифр рядка dv, помножених на число 2, піднесене до степеня, що дорівнює номеру розряду відповідної цифри (рис. 3, с. 262);
   4) вивести значення dc у поле віджета entry1.

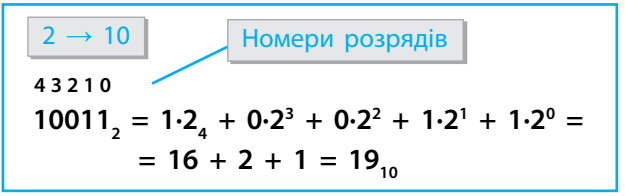


Рис. 3

8. Для переведення запишіть команду циклу:
```
dc = 0
for i in range(len(dv)):
    dc = dc+int(dv[i])*2**(len(dv)–i–1)
```
9. Завантажте модуль messagebox.
10. Додайте до опису обробника події для кнопки b1 виклик функції messagebox.showinfo для виведення повідомлень на зразок: Десяткове 19 дорівнює двійковому 10011.
```
vidp = 'Десяткове '+ str(dc)+ ' дорівнює двійковому '+dv
messagebox.showinfo('Переведення у двійкове', vidp)
```
11. Додайте до опису обробника події для кнопки b2 виклик функції messagebox.showinfo для виведення повідомлень на зразок: Двійкове 10011 дорівнює десятковому 19.
12. Перевірте роботу програми для 95_{10}, 25_{10}, 1024_{10}.

Зробіть висновок: як створити обробники подій для елементів керування в програмі з графічним інтерфейсом.

Барбара Лісков — американська дослідниця в галузі інформатики, лауреатка премії Тюрінга 2008 року, перша жінка в США, що отримала ступінь доктора з інформатики. У 1970–1980-х роках керувала розробкою мов програмування CLU і Argus, об'єктно-орієнтованої СУБД Thor [17].

ПРАКТИЧНА РОБОТА 19
МОДЕЛЮВАННЯ РУХУ ОБ'ЄКТА

Завдання: Складіть програму для розв'язування задачі.
Камінь кинуто горизонтально зі 100-метрової вежі зі швидкістю v м/с. Скласти математичну модель польоту каменя.
Написати програму, яка буде обчислювати й друкувати значення відстані від вежі по горизонталі та висоту над землею кожні 0,2 с польоту каменя.

Обладнання: комп'ютер із середовищем програмування IDLE.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Проаналізуйте математичну модель задачі.

Дано: $h_0 = 100$ — висота вежі, м;
v — швидкість, із якою кинуто камінь, м/с.
Що треба знайти?
s — відстань від вежі по горизонталі, м;
h — висоту над землею кожні 0,2 с польоту каменя, м.
Який зв'язок між величинами?
$s = vt$, де t — час польоту каменя в секундах;
$dh = 9.81t^2/2$ — вертикальне зміщення каменя.
Які є обмеження на дані? $v > 0$, $dh < h$.

2. В IDLE виберіть команду File → New File.
3. Підготуйте зображення вежі. Для цього у графічному редакторі встановіть розмір малюнка 400×450 пікселів.
Створіть папку Pract19 і збережіть у ній графічний файл і файл програми вежа.py.

4. Завантажте модулі tkinter і time.
Створіть вікно програми із заголовком Камінь падає з вежі (див. рисунок, с. 265).

5. Додайте до програмного коду константи, потрібні для подальших розрахунків і побудов:

```
h0 = 100                # Висота вежі
scale = 4               # Масштаб малюнка (в пікселях на 1 м)
dk = 20                 # Діаметр круга, який зображає камінь
x0, y0 = 100, 20        # Початкові координати круга на полотні
```

6. Додайте у вікно програми віджети класу Label і класу Entry згідно з рисунком.

```
Label(text = 'V = ', font = 'Arial 14').grid(row = 1, column = 0)
en = Entry(width = 8, font = 'Arial 14')
en.grid(row = 1, column = 1)
Label(text = ' s\th', font = 'Arial 14').grid(row = 2, column = 0,
      columnspan = 2, sticky = W)
```

7. Додайте у вікно програми віджет txt класу Text для виведення координат каменя під час падіння.

```
# Створюємо об'єкт класу Text
txt = Text(width = 20, height = 22, font = 'Arial 12')
txt.grid(row = 3, column = 0, columnspan = 2)
```

8. Створіть віджет btn класу Button із написом Обчислити.

9. Додайте об'єкт cnv класу Canvas розміром 400×450. Додайте на полотно малюнок із зображенням вежі.

```
cnv = Canvas(width = 400, height = 450)
cnv.grid(row = 1, column = 2, rowspan = 4)
im = PhotoImage(file = '1.gif')
cnv.create_image(0, 0, anchor = NW, image = im)
```

10. Створіть обробник події для кнопки Обчислити. Обчислення тривають доти, поки dh<h (поки камінь не впав на землю).

```
x, y = x0, y0           # Поточні координати круга на полотні
ov = cnv.create_oval(x, y, x+dk, y+dk, fill = 'red') # Створення круга
v = int(en.get())       # Введена початкова швидкість
dh = 0          # Вертикальне зміщення каменя (0 – на вершині)
```

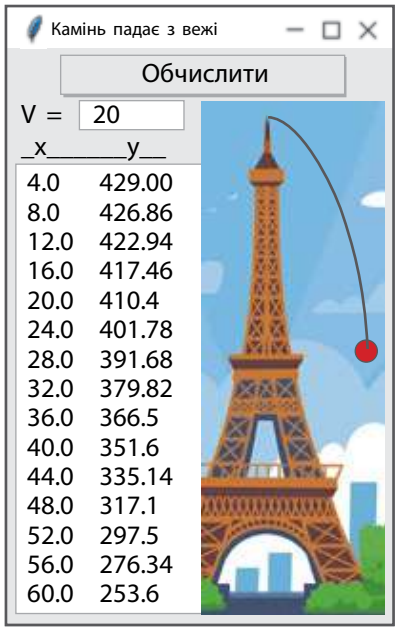
```
t = 0                  # Початок відліку часу
txt.delete(1.0, END)
while dh<h0:           # Поки камінь не впав
    t += 0.2           # Обчислюємо поточний час
    dh = 9.8*t**2/2        # Нове вертикальне зміщення каменя
    if dh<h0:          # Якщо камінь не впав
        dl = v*t           # Нове горизонтальне зміщення каменя
        # Обчислення нових координат круга на полотні
        # з урахуванням початкового положення й зміщень
        x1 = x0+scale*dl
        y1 = y0+scale*dh
        cnv.move(ov, x1–x, y1–y)          # Пересуваємо круг
        x, y = x1, y1                  # Оновлюємо поточні координати
        # Формуємо рядок
        s = str(round(dl, 2))+'\t'+
              str(round(h0–dh, 2))+'\n'
        txt.insert(END, s)    # Додаємо
              # рядок у текстове поле
        root.update()
              # Оновлення полотна
        sleep(0.1)
              # Затримка на 0.1 с
```

11. Додайте до програмного коду малювання відрізків, що відображають траєкторію руху каменя.

```
cnv.create_line(x1, y1, x, y)
```

12. Збережіть файл і випробуйте програму для v = 15.

Поекспериментуйте з різними значеннями швидкості руху.



Зробіть висновок: як змоделювати рух об'єкта в програмі з графічним інтерфейсом.

КОМП'ЮТЕРНИЙ СЛОВНИК

Адаптер — пристрій, призначений для приєднання до комп'ютера іншого пристрою та керування його роботою.

Архітектура комп'ютера — опис будови комп'ютера та принципів роботи його складових.

Батьківський віджет — елемент керування, до якого буде поміщено віджет.

Величина — інформаційний об'єкт, основними характеристиками якого є назва, вид, тип і значення.

Виняток — помилка, яка не є синтаксичною і спричиняє аварійне припинення виконання програми.

Вкладений цикл — цикл, який міститься в тілі іншого циклу.

Гіперпосилання — адреса ресурсу, пов'язана з елементом сторінки, клацання якого приводить до переходу до цього ресурсу.

Гіпертекст — текст, що містить гіперпосилання — зв'язки з іншими частинами документа або іншими документами чи файлами.

Графічний інтерфейс користувача — сукупність зображених на екрані віджетів (вікна, меню, кнопки тощо), за допомогою яких людина взаємодіє з комп'ютером.

Декодування — перетворення закодованого повідомлення у форму, прийнятну для приймача.

Дискретизація сигналу — процес кодування неперервного сигналу послідовністю окремих значень його рівня.

Доменне ім'я — текстова адреса комп'ютера в інтернеті.

Захоплення відео — процес перетворення відеосигналу із зовнішнього пристрою в цифрову форму і запису у відеофайл.

Захоплення звуку — процес перетворення звукового сигналу із зовнішніх джерел у цифрову форму і запису в аудіофайл.

Змінна — величина, значення якої може змінюватися в ході виконання програми.

Значення величини — це конкретні дані, які вона містить у момент звернення до неї.

Кодування — процес перетворення повідомлення в зручну для передавання, опрацювання та зберігання форму.

Колонтитул — повідомлення, яке розміщується у верхньому або нижньому полі кожної сторінки документа.

Конвертер — програма, яка дозволяє перекодувати медіафайли з одного формату в інший.

Мережевий протокол — сукупність правил (стандартів) для обміну даними між комп'ютерами.

Метод — фрагмент програмного коду, який вбудований в об'єкт і передбачає виконання деяких дій із ним.

Мультимедіа — комбінування в одному інформаційному продукті інформації, поданої різними способами з можливістю її інтерактивного використання.

Обробник події — блок програмного коду, який виконується в разі настання події, з якою він пов'язаний.

Подія — зовнішній вплив на елемент керування, на який цей елемент може реагувати.

Покажчик — список слів із зазначенням номерів сторінок, на яких вони згадуються.

Провайдер — організація, яка надає послуги, пов'язані з доступом до глобальної мережі «Інтернет».

Сайт (вебсайт) — об'єднання тематично пов'язаних вебсторінок і документів під одним доменним іменем.

Стиль — іменована сукупність параметрів форматування об'єктів документа, які визначають його зовнішній вигляд.

Структура документа — ієрархічна схема розміщення складових документа.

Тег — позначка мови HTML, записана в кутових дужках <>, яку опрацьовує браузер.

Титри — написи, що з'являються на початку або наприкінці відеофільму, на певному фрагменті або зображенні тощо.

Транслятор — програма, яка перекладає програму з мови програмування високого рівня на мову машинних кодів.

Хмарний диск — сховище даних, розташоване на віддаленому сервері й доступне через інтернет.

Штучний інтелект — розділ інформатики, присвячений розробці машин, які моделюють інтелект людини.

IP-адреса — ідентифікаційний номер комп'ютера в мережі.

АЛФАВІТНИЙ ПОКАЖЧИК

А
Адаптер 25
Архітектура комп'ютера 23
Атрибут об'єкта 167
Аудіоредактор 125
Б
Батьківський віджет 167
В
Величина 187
Виняток 159
Віджет 166
Відеоредактор 136
Вкладений цикл 227
Г
Гіперпосилання 92
Гіпертекст 47
Графічний інтерфейс користувача 165
Д
Дані 8
Декодування 8
Дискретизація сигналу 114
Доменне ім'я 40
З
Захоплення відео 131
Захоплення звуку 125
Зміст документа 94
Значення 188
К
Код 8
Кодування 8
Колонтитул 90
Командна кнопка 170
Конвертер 133
Л
Логічна помилка 162
М
Мережевий протокол 44
Мова програмування 151
Мультимедіа 113
О
Обробник події 171
П
Повторення (цикл) 223
Подія 171
Покажчик 96
Провайдер 44
Програма 151
Р
Розділ 88
С
Сайт (вебсайт) 47
Синтаксична помилка 158
Стиль 87
Структура документа 85
Т
Тег 53
Титри 140
Транслятор 154
Х
Хмарний диск 101
Ц
Цифровий звукозапис 114
Ш
Шаблон 88
Штучний інтелект 134
Н
HTML 52

ЗМІСТ

Передмова 3

Розділ 1. КОДУВАННЯ ДАНИХ ТА АПАРАТНЕ ЗАБЕЗПЕЧЕННЯ

§ 1. Кодування та декодування повідомлень 7
§ 2. Двійкове кодування 11
§ 3. Кодування графічних даних 16
Практична робота 1. Визначення довжини двійкового коду текстових даних 22
§ 4. Архітектура комп'ютера. Технічні характеристики складових комп'ютера 23
§ 5. Історія опрацювання інформаційних об'єктів 30
Практична робота 2. Визначення конфігурації комп'ютера під потребу 36

Розділ 2. СТВОРЕННЯ ТА ПУБЛІКАЦІЯ ВЕБРЕСУРСІВ

§ 6. Адресація в інтернеті 39
§ 7. Обмін даними в інтернеті 43
§ 8. Створення сайтів 47
§ 9. Розмічання тексту засобами HTML 52
§ 10. Графічні зображення та гіперпосилання на вебсторінці 57
§ 11. Розмічання таблиць засобами HTML 60
Практична робота 3. Створення головної сторінки сайту засобами HTML 65

Розділ 3. ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ

§ 12. Основні етапи створення текстового документа 69
§ 13. Робота з фрагментами тексту 74
§ 14. Форматування об'єктів текстового документа 78
Практична робота 4. Створення текстового документа, що містить об'єкти різних типів 83
§ 15. Структура документа. Колонтитули. Гіперпосилання 85
§ 16. Автоматизоване створення змісту документа 94
Практична робота 5. Автоматизоване створення змісту документа 99

§ 17. Користування хмарним диском. Спільна робота з документом 101
Практична робота 6. Створення документа на хмарному диску для спільного опрацювання 109
Практична робота 7. Створення мінігазети на хмарному диску 110

Розділ 4. ОПРАЦЮВАННЯ МУЛЬТИМЕДІЙНИХ ОБ'ЄКТІВ

§ 18. Поняття мультимедіа. Кодування аудіоданих. Формати аудіофайлів 113
§ 19. Кодування відеоданих. Формати відеофайлів 118
Практична робота 8. Використання аудіоданих у програмах мовою Python 122
§ 20. Програмне забезпечення для опрацювання звуку 125
§ 21. Програмне забезпечення для опрацювання відео 131
§ 22. Побудова аудіо- й відеоряду 136
§ 23. Опрацювання, зберігання та розміщення відеофільму в інтернеті 143
Практична робота 9. Створення відеофільму 147
Практична робота 10. Розміщення відеоматеріалів в інтернеті 148

Розділ 5. АЛГОРИТМИ ТА ПРОГРАМИ

§ 24. Сучасні мови програмування 151
§ 25. Налагодження програмного коду 158
§ 26. Графічний інтерфейс користувача 165
§ 27. Опрацювання подій 170
§ 28. Організація діалогу з користувачем 178
Практична робота 11. Введення і виведення даних 185
§ 29. Величини. Числові типи даних 187
§ 30. Математичні функції 193
§ 31. Рядковий тип даних 199
§ 32. Робота з текстовими файлами 205
§ 33. Логічний тип даних 210
§ 34. Елементи вибору 217
§ 35. Алгоритми з повтореннями. Цикли 223

Практична робота 12. Створення програми з графічним інтерфейсом 232
§ 36. Графічні методи модуля tkinter 235
§ 37. Створення анімації 241
Практична робота 13. Створення ігрової програми 247

Розділ 6. ПРАКТИКУМ ІЗ ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Практична робота 14. Створення текстового документа, що містить об'єкти різних типів 251
Практична робота 15. Створення багатосторінкового електронного документа 252
Практична робота 16. Розробка застосунку «Піаніно» 254
Практична робота 17. Створення програми з графічним інтерфейсом 257
Практична робота 18. Створення програми «Калькулятор» 260
Практична робота 19. Моделювання руху об'єкта 263
Комп'ютерний словник 266
Алфавітний покажчик 268

Відомості про користування підручником

№ з/п	Прізвище та ім'я учня / учениці	Навчальний рік	Стан підручника	
			на початку року	у кінці року
1				
2				
3				
4				
5				

Навчальне видання

БОНДАРЕНКО Олена Олександрівна
ЛАСТОВЕЦЬКИЙ Василь Васильович
ПИЛИПЧУК Олександр Павлович
ШЕСТОПАЛОВ Євген Анатолійович

«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»

Редакторка *Л. І. Каюда*. Технічний редактор *А. В. Пліско*.
Художнє оформлення *В. І. Труфена*, *Т. В. Задорожної*.
Комп'ютерна верстка *О. Л. Піковець*. Коректорка *Н. В. Красна*.

Підписано до друку 19.05.2025. Формат 70×90/16. Папір офсетний.
Гарнітура Шкільна. Друк офсетний. Ум. друк. арк. 19,83. Обл.-вид. арк. 19,73.
Наклад 51 081 пр. Зам. № 9504-2025.

ТОВ Видавництво «Ранок», пр. Незалежності, 5, Харків, 61022;
вул. Деревлянська, 13, Київ, 04119. E-mail: office@ranok.com.ua

Свідоцтво суб'єкта видавничої справи ДК № 7548 від 16.12.2021.

Надруковано у друкарні ТОВ «ТРІАДА-ПАК»,
пров. Сімферопольський, 6, Харків 61052.
Свідоцтво суб'єкта видавничої справи ДК № 5340 від 15.05.2017.
Тел. +38 (057) 712-20-00. E-mail: triadakniga@ukr.net

Служба підтримки:
rnk.com.ua/107946

ДЕЯКІ «ГАРЯЧІ»

РЕДАГУВАННЯ І ФОРМАТУВАННЯ ТЕКСТУ

Ctrl + O Щ	Відкриття документа
Ctrl + S І	Збереження документа
Ctrl + Z Я	Скасування попередньої дії
Ctrl + B И	Встановлення або скасування **жирного** накреслення
Ctrl + I Ш	Встановлення або скасування *курсивного* накреслення
Ctrl + U Г	Встановлення або скасування підкресленого накреслення
Ctrl + F3	Оформлення тексту ВЕЛИКИМИ, малими літерами або З Великої Літери
Ctrl + J О	Вирівнювання вибраних абзаців за шириною
Ctrl + E У	Вирівнювання вибраних абзаців по центру
Ctrl + L Д	Вирівнювання вибраних абзаців за лівим краєм
Ctrl + R К	Вирівнювання вибраних абзаців за правим краєм
Enter	Завершення абзацу
Shift + Enter	Розривання рядка без створення абзацу

КЛАВІШІ WORD

ПЕРЕМІЩЕННЯ ДОКУМЕНТОМ

Page Up / Page Down	На сторінку (екранну) вгору / вниз
Ctrl + →	До наступного слова або кінця області виділення
Ctrl + ←	До попереднього слова або початку області виділення
Ctrl + ↑	На початок поточного абзацу
Ctrl + ↓	На початок наступного абзацу
Ctrl + Home	На початок документа
Ctrl + End	У кінець документа
Home / End	На початок / у кінець поточного рядка

КОПІЮВАННЯ, ПЕРЕМІЩЕННЯ, ВИДАЛЕННЯ

Delete / Backspace	Видалення без збереження в буфері обміну
Ctrl + X Ч	Видалення зі збереженням у буфері обміну
Ctrl + C С	Копіювання в буфер обміну
Ctrl + V М	Вставлення з буфера обміну

ДЕЯКІ ТЕГИ

ОФОРМЛЕННЯ (СХЕМА) ДОКУМЕНТА		
<doctype>	Версія HTML	
<html>	HTML-документ	
<head>	Службова інформація	
	<base>	URL документа
	<meta>	Супутні дані
	<title>	Назва вкладки
	<link>	Зв'язок із документом
	<style>	Стиль
	<script>	Скрипт
<body>	Вміст документа	
<!-- текст -->	Коментар (не відображається браузером)	

СТРУКТУРУВАННЯ ДОКУМЕНТА	
<article>	Новина, стаття, запис (вміст)
<aside>	Блок збоку від вмісту
 	Розрив рядка (без відступів)
	Блок, частина документа
<h1> до <h6>	Заголовок (від 1-го до 6-го рівня)
<hr>	Горизонтальна розділювальна лінія
<hgroup>	Групування заголовків
<header>	Верхній блок (шапка) документа
<footer>	Нижній блок документа
<p>	Абзац
<section>	Частина, секція, розділ документа

СТВОРЕННЯ ПОСИЛАННЯ	
<a>	Посилання на сторінку, адресу, закладку
<audio>	Посилання (відтворення) аудіофайлу
<img>	Посилання (показ) зображення
<video>	Посилання (відтворення) відеофайлу

HTML 4–5

ФОРМАТУВАННЯ ТЕКСТУ

<b> або	Жирний шрифт
<i> або	Курсивний шрифт
<u>	Підкреслений шрифт
<mark>	Виділений текст
<sub>	Нижній індекс
<sup>	Верхній індекс
<cite>	Цитата (виноска)
<blockquote>	Довга цитата
<q>	Цитата
<code>	Програмний код
<pre>	Блок відформатованого тексту

ОФОРМЛЕННЯ СПИСКУ

<oi>	Нумерований список	
	Маркований список	
	Елемент (рядок) списку	
<dl>	Список визначень	
	<dt>	Термін
	<dd>	Визначення

ОФОРМЛЕННЯ ТАБЛИЦІ

<table>	Таблиця	
	<caption>	Заголовок до таблиці
	<thead>	Верх таблиці
	<tbody>	Тіло таблиці
	<colgroup>	Група стовпців таблиці
	<col>	Стовпець таблиці
	<tr>	Рядок таблиці
	<th>	Клітинка із заголовком
	<td>	Клітинка таблиці

Підручник для 8 класу
закладів загальної середньої освіти

Інформатика

Особливості підручника:

- зорієнтованість матеріалу на власний досвід дітей, підготовку до життя в інформаційному суспільстві
- покроковий опис виконання практичних робіт
- приклади з різних галузей людської діяльності

Інтерактивний електронний додаток містить:

- інтерактивні тести за всіма темами курсу
- підсумкові роботи до кожного розділу
- додаткові матеріали до практичних робіт
- навчальні відео, 3D-моделі, інтерактивні вправи

ISBN 978-617-09-9597-1
2548 5530
9 786170 995971

Інтерактивний електронний додаток до підручника доступний за QR-кодом або посиланням **rnk.com.ua/107276**