

8

ІНФОРМАТИКА



УДК 004:37.016(075.3)

I-74

Авторський колектив:

Олена Бондаренко, Василь Ластовецький,
Олександр Пилипчук, Євген Шестопапов

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

Видано за рахунок державних коштів.
Продаж заборонено

I-74 **Інформатика** : підруч. для 8 кл. закл. загал. серед. освіти / [О. О. Бондаренко, В. В. Ластовецький, О. П. Пилипчук, Є. А. Шестопапов]. — Харків : Вид-во «Ранок», 2021. — 240 с. : іл.

ISBN 978-617-09-6959-0

УДК 004:37.016(075.3)



Інтернет-підтримка

ISBN 978-617-09-6959-0

© Бондаренко О. О., Ластовецький В. В.,
Пилипчук О. П., Шестопапов Є. А., 2021
© ТОВ Видавництво «Ранок», 2021

ДОРОГІ ВОСЬМИКЛАСНИКИ ТА ВОСЬМИКЛАСНИЦІ!

Ви тримаєте в руках підручник з інформатики, призначений саме для вас, учнів і учениць 8 класу. У 5–7 класах ви вивчали основи об'єктно-орієнтованого програмування та розробки застосунків мовою Python, ознайомилися з поштовими службами інтернету, етикетом електронного листування, онлайн-новинами, онлайн-перекладачами і хмарними сервісами, опанували табличний процесор — програму для опрацювання даних, поданих у вигляді електронних таблиць.

У цьому навчальному році на вас теж очікує чимало цікавого та корисного, а пропонований підручник буде вашим надійним помічником. Підручник складається з п'яти розділів. Пригадаймо, як із ним працювати.

На початку кожного розділу вміщено матеріали рубрики «*Повторюємо*». Вони допоможуть згадати відомості, які ви вивчали з цієї теми в попередніх класах і які будуть необхідні для засвоєння нового матеріалу.

Розділ складається з параграфів і практичних робіт за темами розділу. Кожен параграф містить теоретичні відомості, приклади практичного застосування знань, вправи, питання для самоперевірки.

Питання для самоперевірки допоможуть з'ясувати, чи достатньою мірою ви опанували вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт за комп'ютером.

Вправи складаються із завдань теоретичного і практичного спрямування. Вправи з позначкою ► містять набір окремих завдань, а вправи з позначкою ►► — завдання з описом покрокового виконання. Виконуючи завдання, ви набудете навичок роботи за комп'ютером.

Практичні роботи дають вам змогу розвинути дослідницькі навички. Щоб успішно виконати кожну з них, ви маєте повторити матеріал, вивчений на попередніх уроках, — і тоді зможете застосувати навички роботи з певним програмним забезпеченням.

Оцінити свої знання та вміння вам допоможе *комп'ютерне тестування* в онлайн-режимі з автоматичною перевіркою результату.

У тексті підручника використано такі позначення:



Запам'ятайте



Розгляньте приклад



Ознайомтеся з цікавою інформацією



Знайдіть відповідь в інтернеті



Виконайте завдання за комп'ютером



Виконайте завдання підвищеної складності



Виконайте завдання в парах



Дайте відповідь на питання для самоперевірки



Виконайте тестове завдання

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. Історія опрацювання інформаційних об'єктів
- § 5. Архітектура комп'ютера. Технічні характеристики складових комп'ютера
Практична робота 2. Визначення конфігурації комп'ютера під потребу

ПОВТОРЮЄМО



З уроків інформатики 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 блоку, монітора, миші та клавіатури. Усі пристрої комп'ютера поділяються на *пристрої введення, виведення, опрацювання, зберігання*.

Дії, які можна виконувати з інформацією: пошук, опрацювання, передавання, зберігання 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, а також зі способами взаємодії комп'ютера і користувача.

§ 1. Кодування та декодування повідомлень

Впродовж життя людина сприймає безліч повідомлень, опрацьовує інформацію, зберігає або передає набуті знання. Як це відбувається?

Опрацювання даних як інформаційний процес

З давніх-давен люди намагалися зберегти отриману з навколишнього світу інформацію, зокрема у вигляді наскельних малюнків (рис. 1.1). Зрозуміло, що передавати збережену таким чином інформацію від однієї людини до іншої було непросто. Проте до нас все ж дійшли відомості про життя наших предків.



Рис. 1.1

» В Україні поблизу Мелітополя є унікальне місце — Кам'яна Могила (рис. 1.2). На її каменях із прадавніх часів залишилося понад 3 тис. наскельних малюнків. На них зображено людей, диких і свійських тварин, а також різноманітні знаки — петрогліфи.



Рис. 1.2

Зібрану інформацію зазвичай опрацьовують; під опрацюванням розуміють такі дії з інформацією, унаслідок яких отримується нова інформація або змінюється форма її подання.

Кодування та декодування

Повідомлення передаються від джерела до приймача каналами зв'язку. Під час розмови по телефону звукові сигнали (голос) перетворюються на електричні (кодуються), які по мережі передаються до співрозмовника і знову перетворюються на звукові (декоднуються) (рис. 1.3).

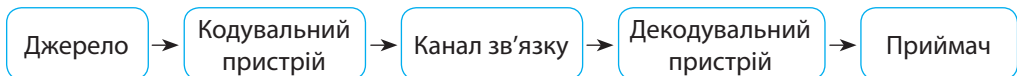


Рис. 1.3

Щоб передати повідомлення за допомогою технічних засобів, його необхідно закодувати відповідно до типу каналу зв'язку. Після передавання повідомлення набуває вигляду, зрозумілого для приймача.



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 тощо.

Окремими ієрогліфами позначаються певні об'єкти або поняття (рис. 1.4), літерами — окремі звуки усного мовлення, із яких складаються слова, речення тощо.



Рис. 1.4

Ті самі повідомлення можна кодувати в різний спосіб, використовуючи цифри, знаки, символи, жести, світлові, звукові та електричні сигнали тощо.

1 Звук [а] природної мови можна закодувати і літерою А, і жестом, і звуковими сигналами — коротким та довгим, які на папері позначають крапкою і тире (рис. 1.5).

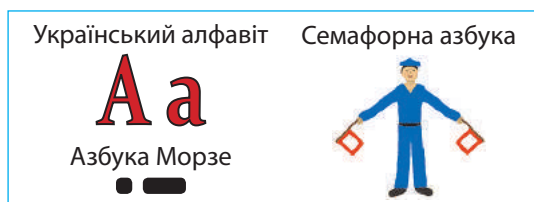


Рис. 1.5

Для передавання текстових повідомлень можна вигадати власний код, наприклад, замінюючи літери певними символами, малюнками або цифрами. Для декодування повідомлення потрібно знати, як воно закодоване, це дозволить відновити його зміст.

Слід пам'ятати, що те саме позначення може мати різні значення залежно від галузі його використання.

2 Велика латинська літера V може позначати об'єм тіла та римську цифру 5, а літера P — звук, периметр, масу тіла.

Приймати, декодувати й опрацьовувати повідомлення може людина, створений нею прилад, тварина.

- 3 На рис. 1.6. навчений службовий собака вправно виконує команди, які закодовані людиною у вигляді жестів.



Рис. 1.6

Способи кодування безперервно удосконалюються, адже вдало закодовані дані сприяють їх ефективнішому використанню у всіх галузях.

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



1. Розгляньте наскельний малюнок (див. рис. 1.1). Опишіть, яка інформація на ньому міститься.
2. Знайдіть в інтернеті азбуку Морзе і закодуйте слово «інформатика».
3. Знайдіть в інтернеті інформацію про абетку Брайля та підготуйте коротке повідомлення.
4. Вигадайте власну систему кодування тексту.
5. Закодуйте повідомлення та запишіть у зошиті.
6. Обміняйтесь з однокласником або однокласницею зошитами і вашим кодом (ключем) та декодуйте повідомлення.



Комп'ютерне тестування

Виконайте тестове завдання 1 із автоматичною перевіркою результату.



§ 2. Двійкове кодування

Пригадаймо, що дані — це повідомлення, в яких однотипну інформацію подано у формі, зручній для опрацювання, зазвичай за допомогою технічних засобів. Комп'ютер — універсальний пристрій для опрацювання даних різних типів. Розглянемо, як він сприймає 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 в 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 (числові, текстові, звукові, графічні тощо) для опрацювання комп'ютером кодують за допомогою лише 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 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			?
...			

1 8-бітовий код придатний для кодування до $N = 2^8 = 256$ різних символів, а 16-бітовий код — до $N = 2^{16} = 65\,536$ символів.

Під час кодування (декодування) тексту всі символи для зручності зводять у кодову таблицю, де поряд із двійковим кодом або замість нього іноді 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 є керувальними кодами, а решта — від 32 до 127 — 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.

2 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:;<=>?@ABCDEFGHIJKLMN
OPQRSTUVWXYZ[\]^_`abc
defghijklmnopqrstuvwxyz{|}~
```

► 8-бітовий код KOI8-U

Для кодування літер інших алфавітів (переважно 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 Інформації 8-бітовий Український*). Розглянемо фрагмент таблиці кодів KOI8-U:

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

Так, мала літера а українського алфавіту в таблиці кодів KOI8-U має десятковий код 193, що відповідає двійковому: 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 (*Unicode* — уніфіковане кодування). Поширений нині варіант Юнікоду — код UTF-8 — він дозволяє закодувати до 1 112 064 різних символів.

Одиниці вимірювання довжини двійкового коду

Двійкове кодування застосовують не тільки для текстових даних, але й для графічних, звукових тощо. При цьому обсяг даних вимірюється *довжиною двійкового коду*.

Послідовність із 8 двійкових розрядів (бітів) називають **байтом**. Наприклад, 10101101.

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

На практиці зручніше користуватися більшими одиницями вимірювання: кібібайт (КиБ), мебібайт (МиБ), гібібайт (ГіБ), тебібайт (ТіБ).

$$1 \text{ КиБ} = 2^{10} \text{ Б} = 1024 \text{ Б} \approx 10^3 \text{ Б}$$

$$1 \text{ МиБ} = 2^{10} \text{ КиБ} = 1024 \text{ КиБ} \approx 10^6 \text{ Б}$$

$$1 \text{ ГіБ} = 2^{10} \text{ МиБ} = 1024 \text{ МиБ} \approx 10^9 \text{ Б}$$

$$1 \text{ ТіБ} = 2^{10} \text{ ГіБ} = 1024 \text{ ГіБ} \approx 10^{12} \text{ Б}$$

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

4 Нехай сторінка книжки, поданої в електронному вигляді, містить, наприклад, 56 рядків по 64 символи в кожному рядку. Обчислимо обсяг даних (довжину двійкового коду) на сторінці.
Маємо: $56 \cdot 64 = 3584$ байти = 3,5 КиБ.

Щоб обчислити приблизний обсяг текстових даних у книжці, слід кількість символів на сторінці помножити на кількість сторінок. Розрахуємо обсяг даних у книжці, яка має 256 подібних сторінок:

$$3584 \cdot 256 = 917\,504 \text{ Б} = 896 \text{ КіБ} = 0,875 \text{ МіБ}.$$

Не слід ототожнювати довжину двійкового коду текстового повідомлення з обсягом інформації, який 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 видів, ви ознайомитеся далі.

Питання для самоперевірки



1. Як кодуються дані в комп'ютері?
2. Чому в комп'ютері використовують двійкове кодування?
3. Яким чином 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



1. Знайдіть в інтернеті таблицю кодів символів ASCII.
2. Розкодуйте повідомлення:
1001000 1100101 1101100 1101100 1101111 0100001.
3. Змініть програму з прикладу 1 так, щоб поряд із символами виводились їхні числові коди.
4. Збережіть програму у файл Вправа 2-2 та перевірте її роботу.
5. Визначте, вміст скількох книжок (приклад 4) в електронному вигляді можна зберегти на флешнакопичувачі обсягом 32 ГіБ.
6. Визначте приблизний обсяг текстових даних на одній зі сторінок вашого підручника з інформатики.



Комп'ютерне тестування



Виконайте тестове завдання 2 із автоматичною перевіркою результату.



§ 3. Кодування графічних даних

З уроків інформатики в 6 класі ви знаєте, що є два способи кодування графічних даних: растровий і векторний.

Растрові зображення

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

Принцип побудови растрового зображення легко зрозуміти, якщо відкрити в графічному редакторі фотографію і дуже 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.

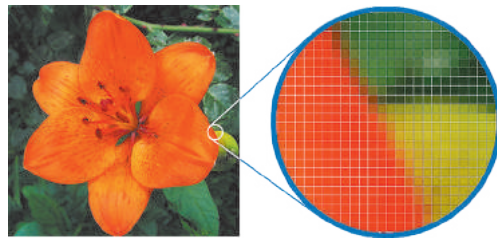


Рис. 3.1

! 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, тобто 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 числом 1 (*другий* колір). Для зберігання коду пікселя буде достатньо одного біта, тому говорять, що двоколірне зображення має глибину кольору 1 біт (рис. 3.2).

0	0	1	0	1	0	0
0	1	0	1	0	1	0
1	0	1	1	1	0	1
0	1	1	0	1	1	0
1	0	1	1	1	0	1
0	1	0	1	0	1	0
0	0	1	0	1	0	0

Рис. 3.2



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

- 1 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. Побудуємо таблицю десяткових і двійкових кодів цих кольорів:

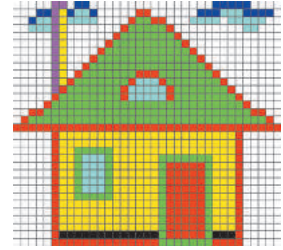


Рис. 3.3

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 будиночка послідовністю чисел, зазначивши його ширину й висоту (в пікселях) і коди кольорів пікселів, перелічені зліва направо зверху вниз: 35 32 7 7 7 7 7 5 5 7 1 1 7 і т. д. Таким чином, за глибини кольору 3 біти довжина двійкового коду кольорів пікселів цього зображення 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).

- ! Зі збільшенням числа кольорів глибина кольору зростає. За однакової кількості пікселів зображення це призводить до збільшення обсягу файлу.

Отже, у файлі растрового зображення зберігаються значення розмірів зображення та коди кольорів усіх його пікселів. Файли більшості растрових форматів містять також іншу інформацію про зображення: дані про прозорість пікселів, тип фотокамери, коментарі тощо.

- 2 Маючи код растрового зображення (див. приклад 1), можна скласти програму для виведення його на екран, зобразивши пікселі квадратами відповідних кольорів. Наприклад:

```
from tkinter import*
```

```
raster = input()           # Вводимо рядок із кодом зображення
pic = list(map(int, raster.split())) # Перетворюємо рядок на список чисел
W, H = pic.pop(0), pic.pop(0) # Відокремлюємо ширину W і висоту H
size = 20                  # Розмір квадратика-пікселя
```

```
root = Tk()
# Створюємо полотно необхідного розміру
canvas = Canvas(root, width = size*W, height = size*H)
canvas.pack()
# Список кольорів
color = ['black', 'blue', 'green', 'brown', 'red', 'violet', 'yellow', 'white']
```

```
for r in range(H):
    for c in range(W):
        canvas.create_rectangle(c*size, r*size,
                                (c+1)*size, (r+1)*size,
                                fill = color[pic[r*W+c]])
```

```
root.mainloop()
```

Векторні зображення

Пригадаємо, що **векторним** називають зображення, яке складається з графічних примітивів — геометричних об'єктів (відрізків, кіл, кривих тощо).

Таким чином, базовим елементом векторного зображення є лінія, описана математичною формулою (рис. 3.4).

У найпростішому випадку криволінійний контур графічного примітива можна розглядати як ланцюжок відрізків, тобто ламану.

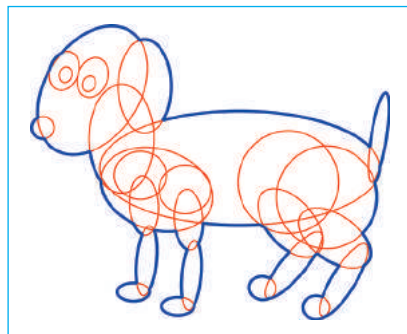


Рис. 3.4

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.

- 3 Контур літери К (рис. 3.5, а) тричі зобразили ланцюжками відрізків, щоразу збільшуючи їх кількість на криволінійних ділянках. Як бачимо, зі збільшенням кількості відрізків зростає точність відтворення криволінійного контуру (рис. 3.5, б, в, г).



Рис. 3.5

- 4 Розглянемо рис. 3.6. Векторний код зображеного контуру з 15 відрізків міститиме $1 + 2 \cdot (15 + 1) = 33$ числа 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.

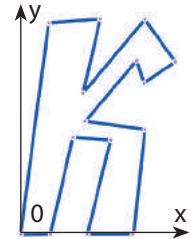


Рис. 3.6

За такого способу кодування можна послідовно записувати коди кількох контурів. Це не ускладнить розкодування, оскільки кількість чисел у коді одного контуру цілком визначається кількістю відрізків, зазначеною на його початку.

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

```
from tkinter import*
```

```
vector = input()
```

```
# Вводимо рядок із кодом контуру
```

```
x = list(map(int, vector.split()))
```

```
# Перетворюємо рядок на список чисел
```

```

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

size = 4         # Масштабний множник

root = Tk()
# Створюємо полотно потрібного розміру
canvas = Canvas(root, width = size*max(x), height = size*max(y))
canvas.pack()

for v in range(N):    # Будуємо відрізки
    canvas.create_line(size*x[v], size*y[v], size*x[v+1], size*y[v+1])

root.mainloop()

```

Запустивши програму, ми побачимо вікно з виведеним векторним зображенням (рис. 3.7). Це зображення виявилось віддзеркаленим відносно горизонтальної осі, оскільки вісь y на полотні спрямована донизу.

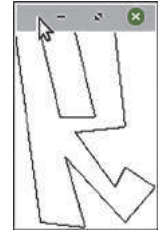


Рис. 3.7

У більшості випадків на практиці застосовуються значно складніші коди. Такі коди дозволяють точно описувати криволінійні ділянки контурів за допомогою кривих Безьє, колір і тип заповнення, прозорість та розмиття об'єктів, багат шарові зображення тощо.

Питання для самоперевірки



1. Які є способи кодування графічних даних?
2. Опишіть особливості растрового зображення; векторного зображення.
3. Василько показав Оксані фрагмент фотографії (рис. 3.8), збільшений настільки, що стало видно пікселі, але Оксана сказала, що це підробка. Хто з них правий?
4. Якою буде довжина двійкового коду зображення розміром 2048×1536 пікселів, якщо глибина кольору становить 24 біти?

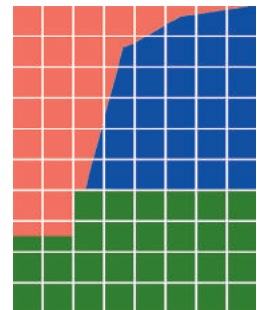


Рис. 3.8

5. Фотографія розміром 2048×1536 пікселів, збережена у форматі JPG, займає на диску 2,4 МіБ. Чому цей обсяг значно менший, ніж довжина коду, обчислена в питанні 4?
6. Поясніть, як виконується цикл `for v in range(N)` із прикладу 5.

Вправа 3



1. У середовищі програмування Python запустіть програму з прикладу 2. Збережіть у файл із назвою Вправа 3-1 і 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. 3.9.
3. Запишіть код кожного з растрових зображень, поданих на рис. 3.10. Перевірте правильність отриманих кодів за допомогою програми (див. п. 1).

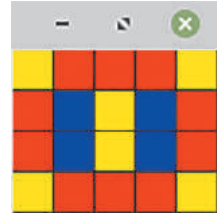
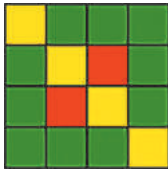
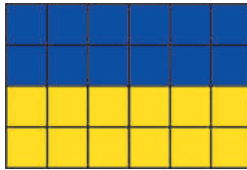


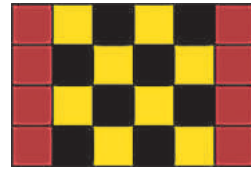
Рис. 3.9



а



б



в

Рис. 3.10

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 також вводився з клавіатури, і перевірте її роботу.
6. Змініть програму з прикладу 5 таким чином, щоб зображення виводилося не віддзеркаленим.



Збережіть у файл із назвою Вправа 3-3 і перевірте роботу програми.

Комп'ютерне тестування

Виконайте тестове завдання 3 із автоматичною перевіркою результату.





Практична робота 1

Визначення довжини двійкового коду текстових даних

Завдання: обчислити довжину двійкового коду текстових даних.

Обладнання: комп'ютер, простий текстовий редактор (наприклад, Блокнот), підручник, зошит.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запишіть у зошиті своє ім'я та прізвище. Визначте довжину двійкового коду записаного тексту при однобайтовому кодуванні символів.
2. Запустіть текстовий редактор Блокнот. Наберіть своє ім'я та прізвище через пропуск і збережіть файл із назвою Практична робота 1 у відповідній папці.
За допомогою контекстного меню файлу (команда Властивості) з'ясуйте розмір файлу в байтах і порівняйте зі значенням, отриманим у п. 1. Поясніть результат.
3. За допомогою фрагмента таблиці кодів KOI8-U (с. 11 підручника) закодуйте слово «байт» і запишіть його в зошиті за допомогою двійкових і десяткових кодів.
4. Користуючись програмою Калькулятор, визначте приблизний обсяг пам'яті, необхідний для зберігання тексту однієї сторінки вашого підручника з інформатики і всього підручника (без малюнків) в електронному вигляді.
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 за комп'ютером.

Зробіть висновок: як визначати довжину двійкового коду текстових даних.

§ 4. Історія опрацювання інформаційних об'єктів

Історія людства нерозривно пов'язана з інформаційними процесами. Люди здавна розробляли різноманітні засоби для полегшення опрацювання інформації, зокрема таких інформаційних об'єктів, як звук, зображення, текст, числа.

Етапи розвитку обчислювальних засобів

В історії розвитку обчислювальних засобів розрізняють кілька основних етапів (рис. 4.1).



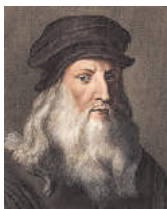
Рис. 4.1

На домеханічному етапі люди використовували для лічби різні підручні засоби: камінці, мотузки, зерна тощо. Згодом різні народи почали використовувати обчислювальні пристрої, найбільш розповсюдженими серед яких були абак і його різновиди.

На **механічному етапі**, з розвитком промисловості й торгівлі, створено низку механічних пристроїв для обчислень. Попри їх невеликі можливості, вони допомагали значно прискорити розрахунки.

» Біля витоків механічної обчислювальної техніки стояли відомі науковці, інженери і дослідники (рис. 4.2).

- Перший відомий ескіз механічного обчислювального пристрою близько 1500 року виконав італійський винахідник, інженер і художник *Леонардо да Вінчі*.
- Перший відомий діючий пристрій для додавання та віднімання чисел («паскаліну») розробив і виготовив у 1642 році *Блез Паскаль* — згодом відомий французький учений.
- У 1673–1710 роках *Готфрід Вільгельм Лейбніц* розробив механічний калькулятор, здатний швидко виконувати 4 арифметичні операції. Він також описав двійкову систему числення, на якій ґрунтується робота сучасних комп'ютерів.
- У 1833 році англійський математик *Чарлз Беббідж* розробив проєкт програмованої обчислювальної машини. Опис та програму для неї склала його учениця *Ада Лавлейс* — саме її вважають першим у світі програмістом.
- У 1938 році німецький інженер *Конрад Цузе* побудував першу електромеханічну програмовану цифрову машину.



Леонардо
да Вінчі



Блез
Паскаль



Готфрід
Лейбніц



Чарлз
Бebbідж



Ада
Лавлейс

Рис. 4.2

На **електронному етапі** було створено електронно-обчислювальні машини (комп'ютери), розвиток яких триває і в наш час. Значний внесок у вдосконалення обчислювальних пристроїв зробили провідні IT-корпорації світу (наприклад, такі як Intel, IBM, AMD, Apple та ін.). Саме на електронному етапі комп'ютер із приладу, цікавого лише науковцям, перетворився на цілком буденну річ.

Основні етапи розвитку комп'ютерів

Розглянемо історію розвитку комп'ютерів як універсального засобу для роботи з різними інформаційними об'єктами. За елементною базою комп'ютери поділяють на кілька поколінь (рис. 4.3).



Рис. 4.3

» Перші комп'ютери були створені в різних країнах у 40–50-ті роки XX ст. Це, наприклад, Z3 (Німеччина, 1941), Colossus (Велика Британія, 1943), Harvard Mark I (США, 1944), МЕСМ (СРСР, 1950).

Основоположником обчислювальної техніки в СРСР був академік Сергій Олексійович Лебедев (рис. 4.4). Під його керівництвом у Київському електротехнічному інституті у 1951 році було створено перший в СРСР і континентальній Європі комп'ютер з назвою «МЕСМ» (рос. МЭСМ — *Малая Электронная Счетная Машина*).



Рис. 4.4

» В Інституті кібернетики Академії наук України у 60-х роках ХХ ст. створено серію машин для інженерних розрахунків (скорочено МІР). Колектив учених очолював Віктор Михайлович Глушков.

» Один із перших персональних комп'ютерів створили у 1976 році Стів Джобс і Стів Возняк (рис. 4.5), засновники американської фірми Apple Computer.



Рис. 4.5

Можливо, комп'ютерами п'ятого покоління стануть штучно створені біологічні системи, біо-процесорами в яких слугуватимуть молекули ДНК, білкові молекули тощо. Перший комп'ютер на базі ДНК створили американські вчені ще в 1994 році. Нині в кількох країнах світу тривають дослідження в галузі біокомп'ютерних технологій.

У 2020 році дослідникам з університету Мейнца (Німеччина) вдалося «спіймати» і перемістити на незначну відстань фотони світла. Дослідження стало поштовхом у розвитку фотонної обчислювальної техніки.

Види сучасних комп'ютерів

Сучасні комп'ютери мають різні обчислювальні можливості.

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

» Українські центри суперкомп'ютерних обчислень у Національному технічному університеті «Київський політехнічний інститут імені Ігоря Сікорського» та Інституті кібернетики імені В. М. Глушкова НАН України мають найпотужніші в країні комп'ютери.

» Рейтинг найпотужніших суперкомп'ютерів світу постійно змінюється. Так, Summit, розроблений IBM, утримував першість два роки, а в 2020 році поступився японському суперкомп'ютеру Fugaku.

Мейнфрейми — високопродуктивні надійні сервери (рис. 4.6), призначені для використання в критично важливих системах (керування електростанціями, продаж квитків, хмарні технології тощо).

Мікрокомп'ютери призначені для реалізації простих задач керування, наприклад побутовою технікою (рис. 4.7).



Рис. 4.6



Рис. 4.7

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

Конструкції та розміри ПК досить різноманітні (рис. 4.8).



Рис. 4.8

Можливості ПК залежать від потреб людини, яка його використовує: офісному працівнику достатньо комп'ютера з мінімальними

можливостями, а кіберспортсмену для тренувань у сучасних комп'ютерних іграх знадобиться потужніше обладнання. Завдяки збільшенню швидкодії, зменшенню розмірів, зниженню ціни кількість комп'ютерів у світі вже давно перевищила 2 млрд і продовжує стрімко зростати.

Виробництво комп'ютерів постійно вдосконалюється, розширюється сфера їх застосування. Набувають поширення нові види комп'ютерів: роботизовані іграшки, роботи-помічники по господарству тощо (рис. 4.9). Домашня автоматика «розумного будинку» здатна діяти замість людини, спростити побутові задачі людям похилого віку, особам з інвалідністю.



Рис. 4.9

Питання для самоперевірки



1. Назвіть основні етапи розвитку обчислювальних засобів.
2. Хто створив перший механічний обчислювач?
3. Кого називають першим у світі програмістом?
4. Яка елементна база персональних комп'ютерів?
5. Хто є основоположником розробки комп'ютерів в Україні?
6. Стисло опишіть види сучасних комп'ютерів.

Вправа 4



1. Запустіть текстовий процесор і створіть новий документ.
2. Знайдіть у матеріалах параграфа дані про швидкодію комп'ютерів першого покоління і сучасного комп'ютера та запишіть їх у документ.



3. Знайдіть в інтернеті та скопіюйте в документ інформацію про внесок у створення перших комп'ютерів Джона Вінсента Атанасова і Михайла Пилиповича Кравчука.



4. Складіть повідомлення на тему «Комп'ютери майбутнього».
5. Уведіть текст повідомлення у створений документ. Збережіть файл із назвою Вправа 4 у відповідній папці.
6. Обговоріть перспективи розвитку комп'ютерів з однокласниками.

Комп'ютерне тестування



Виконайте тестове завдання 4 із автоматичною перевіркою результату.



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

Більшість сучасних комп'ютерів побудовано за принципами, які обґрунтував американський науковець Джон фон Нейман (рис. 5.1) ще у 1948 році.

Відтоді, попри зміни вигляду комп'ютерів та появу нових пристроїв, загальна схема роботи комп'ютера не змінилася (рис. 5.2).



Рис. 5.1



Рис. 5.2

Архітектура комп'ютера — це опис будови комп'ютера та принципів роботи його складових. Основні пристрої персонального комп'ютера містяться в системному блоці.

Системна (материнська) плата — пристрій комп'ютера, на якому розміщені процесор, внутрішня пам'ять, системна шина, порти (роз'єми, слоти). **Системна шина (магістраль)** — провідники і пристрої, призначені для обміну даними між усіма пристроями комп'ютера (див. схему обміну даними на рис. 5.3).

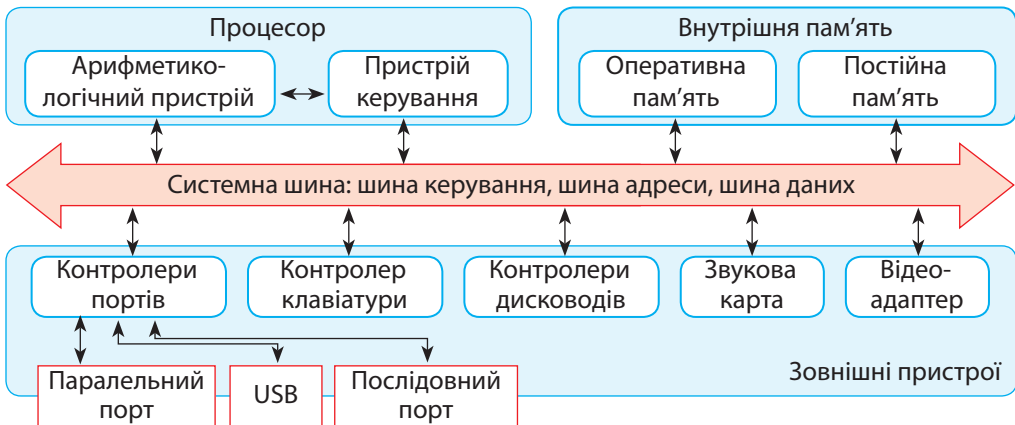


Рис. 5.3

Процесор

Розглянемо пристрої комп'ютера та їх технічні характеристики.

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

Процесор складається з двох основних складових:

- **пристрою керування**, який керує роботою та взаємодією складових комп'ютера;
- **арифметико-логічного пристрою (АЛП)**, який виконує арифметичні та логічні операції над даними.

Процесор має внутрішню пам'ять (кеш-пам'ять), обмін даними з якою відбувається значно швидше, ніж з оперативною пам'яттю.

Конструктивно сучасний процесор (мікропроцесор) являє собою велику інтегральну схему, вміщену в захисний корпус (рис. 5.4).

Швидкодія процесора є основним чинником, який впливає на продуктивність комп'ютера.

Розглянемо технічні характеристики процесора:



Рис. 5.4

Характеристика	Опис
Тактова частота	Кількість керувальних імпульсів (тактів) за секунду; вимірюється в герцах (Гц); визначає швидкодію процесора
Обсяг кеш-пам'яті	Обсяг кеш-пам'яті, яка використовується процесором 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 роз'єму для процесора на системній платі

- 1 Розглянемо стислий запис характеристик процесора:
Intel Core i9-10900K 3.7 GHz 20 MB (BX8070110900K) s1200
Маємо: Intel — виробник; Core i9-10900K — тип — 64-розрядний 10-ядерний; 3.7 ГГц — тактова частота; 20 МБ — обсяг кеш-пам'яті; (BX8070110900K) — модель; s1200 — сокет.

Сокет для процесора на системній платі має відповідати сокету процесора, інакше встановити процесор на цю плату буде неможливо.

Пам'ять комп'ютера

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

Пам'ять поділяють на внутрішню і зовнішню; енергозалежну й енергонезалежну (рис. 5.5).

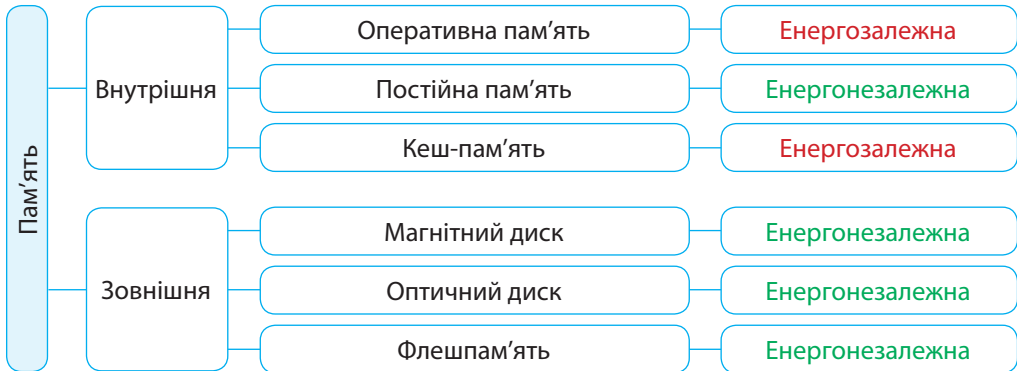


Рис. 5.5

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 комп'ютера.

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.


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

У 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.

Розглянемо деякі зовнішні запам'ятовувальні пристрої:

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

Пристрої введення-виведення даних

Ви вже знайомі з такими пристроями введення-виведення даних, як клавіатура, миша, монітор. Їх називають основними. Крім цих пристроїв, існують й інші. Робота пристроїв введення-виведення і комп'ютера потребує узгодження, наприклад за допомогою адаптера.

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

Існують різні адаптери: відеоадаптери, або відеокарти (рис. 5.6, а), — для роботи з відеокамерами і моніторами, аудіоадаптери, або звукові карти (рис. 5.6, б), — для роботи з аудіопристроями та ін.

Монітор — пристрій виведення, призначений для візуального відображення текстової, графічної та відеоінформації.

За *принципом дії* розрізняють такі типи моніторів: на електронно-променевої трубі (CRT); на рідких кристалах (LCD); на органічних світлодіодах (OLED); «електронне чорнило» (e-ink); проекційні та ін.

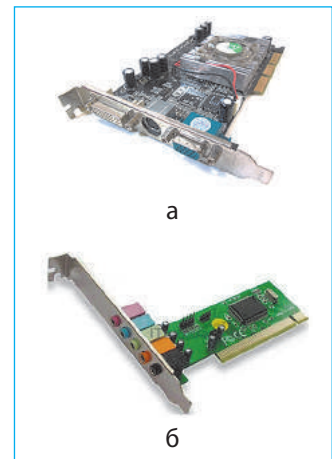


Рис. 5.6

Розмір монітора визначають довжиною діагоналі, поданою в дюймах ($1'' = 2,54$ см). Зображення на екрані моніторів формується з пікселів. Важливою характеристикою монітора є роздільна здатність екрана — кількість пікселів на дюйм (ppi — *pixels per inch*).

Принтер — пристрій виведення, призначений для виведення інформації на тверді носії (папір, пластик, скло тощо).

За технологією друку розрізняють такі типи принтерів:

Тип принтера	Опис технології друку
Матричний	Зображення формується з точок від ударів голок по фарбувальної стрічці
Струменевий	Зображення формується краплями чорнил різного кольору
Лазерний, світлодіодний	Зображення формується з порошкового тонера на наелектризованій поверхні, а потім переноситься на папір

Існують також принтери з іншими технологіями друку.

До основних технічних характеристик принтера належать *роздільна здатність* — найбільша кількість точок на дюйм (dpi — *dots per inch*); *швидкість друку* — кількість друкованих сторінок за хвилину; *формат паперу* — найбільший розмір паперу для друку; *параметри друку* — можливість друку без полів, двостороннього друку тощо; *кількість барвника* — на скільки сторінок вистачає барвника.

Дедалі більшого поширення набувають **3D-принтери** (рис. 5.7), які методом пошарового нанесення речовини формують об'ємні предмети. З розвитком технологій 3D-друку з'явилися *промислові принтери*, на яких, наприклад, з металевого порошку виготовляють деталі машин, а також *будівельні принтери*, що з будівельних сумішей «друкують» будинки (рис. 5.8), тощо.



Рис. 5.7



Рис. 5.8

Мультимедійні пристрої

Як відомо, комп'ютер опрацьовує повідомлення, подані в різний спосіб (текстові, графічні, звукові, комбіновані тощо). Комбіновані повідомлення називають мультимедійними.



Мультимедія — поєднання різних способів подання повідомлень та використання різних пристроїв для їх відтворення.

Для введення в комп'ютер мультимедійних даних приєднують такі зовнішні пристрої, як мікрофон, сканер, відеокамеру тощо, для виведення — монітор, проектор, навушники, принтер тощо. Для узгодження дій цих пристроїв і комп'ютера використовують відповідні адаптери.

Аудіоадаптер (звукова карта) — пристрій для відтворення та запису звуку. **Відеоадаптер** (відеокарта) — пристрій для підтримки перетворення даних на відеосигнал для монітора або проектора. Вони можуть бути як убудовані (інтегровані) в системну плату, так і встановлені в роз'єм (слот) на ній.

! Відеоадаптер, монітор і спеціальні програми для забезпечення їх підключення 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 відео, комп'ютерних ігор тощо.

Конфігурація комп'ютера

Конфігурацію комп'ютера визначає перелік його пристроїв та їхні технічні характеристики. Оскільки комп'ютерна техніка весь час розвивається і вдосконалюється, то й конфігурація швидко застаріває.

В описі конфігурації комп'ютера використовують позначення для характеристик монітора, процесора, оперативної пам'яті, накопичувача (на жорстких магнітних дисках або твердотільного), відеоадаптера, оптичного приводу, мережевої карти, операційної системи тощо.

2

Стислий запис конфігурації комп'ютера може виглядати так:
27" (2560x1440) LED/Intel Core i9-10900K 3.7 GHz 20 MB/RAM 32 Gb/SSD
500 GB/nVidia GeForce GTX 1060 MXM 6 GB GDDR5/DVD+RW/LAN/Ubuntu.

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 5



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 згадані характеристики, розшифруйте їх і запишіть у створений документ.

3. За каталогами інтернет-магазинів знайдіть опис одного з процесорів. Скопіюйте в документ посилання на сторінку та стислий запис характеристики процесора.



4. Визначте за цим записом і запишіть:

- розрядність;
- кількість ядер;
- тактову частоту;
- обсяг кеш-пам'яті;
- сокет.



5. Знайдіть в інтернеті та скопіюйте в документ опис таких характеристик ОЗП:

- пропускна здатність;
- тип;
- обсяг;
- частота.

6. Користуючись інтернетом, з'ясуйте і запишіть у документ, що означають окремі частини скороченого запису конфігурації комп'ютера (приклад 2, с. 32). Збережіть файл із назвою Вправа 5.

Комп'ютерне тестування



Виконайте тестове завдання 5 із автоматичною перевіркою результату.





Практична робота 2

Визначення конфігурації комп'ютера під потребу

Завдання: визначити конфігурацію комп'ютера під потребу комп'ютерної гри з такими системними вимогами:

- 30 ГіБ вільного місця на жорсткому диску;
- 8 ГіБ оперативної пам'яті;
- процесор Intel Core i9-10900K або AMD Ryzen 9 5900X.

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

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

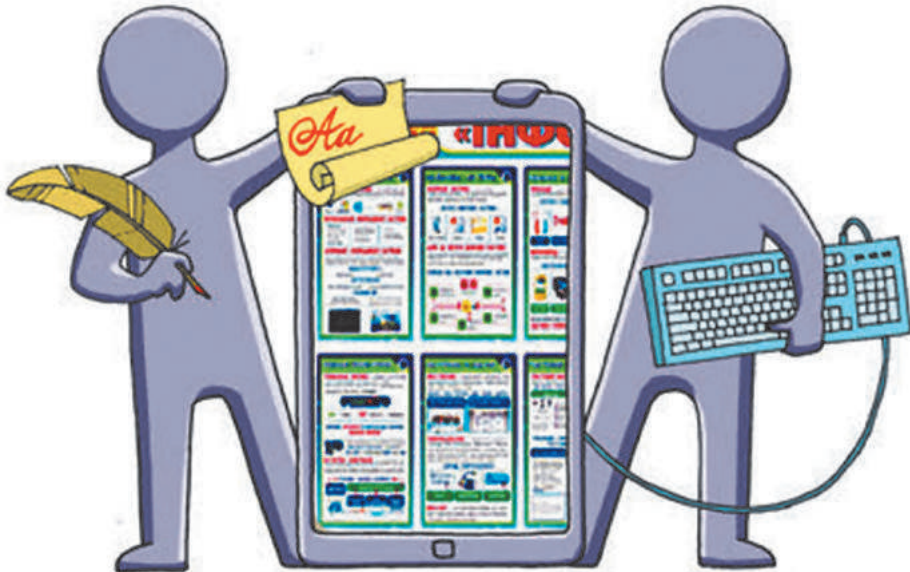
1. За каталогами інтернет-магазинів доберіть складові для комплектування комп'ютера з конфігурацією, потрібною для гри.
2. Запустіть текстовий процесор і створіть новий документ. Уведіть назви та вартість у гривнях таких пристроїв комп'ютера:

• корпус системного блока з блоком живлення;	• оптичний привід;
• системна плата;	• відеоадаптер (якщо він не вбудований у системну плату);
• процесор;	• мережева карта;
• монітор;	• клавіатура;
• оперативна пам'ять;	• миша.
• накопичувач SSD або HDD;	
3. Визначте, від яких пристроїв залежить відповідність конфігурації комп'ютера потребам комп'ютерної гри.
4. Скопіюйте назви пристроїв комп'ютера, стислі технічні характеристики та посилання на вебсторінки сайту інтернет-магазину.
5. Запишіть у документ, за якими характеристиками ви вибрали процесор; системну плату.
6. Збережіть файл із назвою Практична робота 2 у відповідній папці. Завершіть роботу за комп'ютером.

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

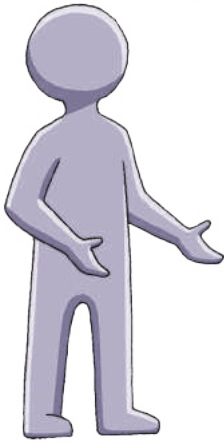
РОЗДІЛ 2

ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ



- § 6. Середовище текстового процесора
- § 7. Робота з фрагментами тексту
- § 8. Форматування об'єктів текстового документа
Практична робота 3. Створення текстового документа, що містить об'єкти різних типів
- § 9. Структура складного текстового документа
- § 10. Зміст документа
Практична робота 4. Автоматизоване створення змісту документа
- § 11. Опрацювання документів на хмарному диску
Практична робота 5. Створення документа на хмарному диску для спільного опрацювання

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* і *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 вмієте виконувати операції над *основними текстовими об'єктами* документа.

1. З 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 можливості стильового оформлення документів і спільної роботи з ними.

§ 6. Середовище текстового процесора

Минули часи, коли текстові документи створювались на друкарських машинках (рис. 6.1, а). Нині їх створюють і опрацьовують за допомогою текстових редакторів і роздруковують на принтері (рис 6.1, б).



Рис. 6.1

Від 2020 року з Національного класифікатора професій України зникла професія «Секретар-друкарка». Натомість з'явилась професія «Офісний службовець», основою якої має бути комп'ютерна й інформаційно-комунікаційна грамотність.

Текстові редактори

Пригадаємо, що текстовий редактор — це програмний засіб для створення та опрацювання текстових документів.

До найпростіших комп'ютерних програм належить текстовий редактор Блокнот (рис. 6.2), який входить до стандартного пакета програм операційної системи (ОС) Windows. Блокнот майже не має інструментів форматування тексту і призначений здебільшого для створення і редагування простих текстових документів.

Ще один текстовий редактор WordPad (рис. 6.3) теж, як і Блокнот, є стандартною прикладною програмою ОС Windows. Він призначений для опрацювання невеликих документів із можливістю форматування символів і абзаців, вставлення рисунків.



Рис. 6.2



Рис. 6.3

Існує багато текстових редакторів, що мають різне призначення і можливості. Так, для програмістів корисним є безкоштовний текстовий редактор Notepad++ (рис. 6.4). Він розпізнає синтаксис багатьох мов програмування, має можливість виокремлювати програмні блоки.



Рис. 6.4

Текстовий процесор Microsoft Word

Сьогодні найпоширенішим текстовим редактором є Microsoft Word (далі Word), більшість із вас працюють з ним ще з 5 класу (рис. 6.5). Word надає можливість здійснювати надзвичайно багато процесів опрацювання об'єктів текстового документа. Тому його називають *текстовим процесором*.



Рис. 6.5

Для запуску Word в ОС Windows потрібно вибрати команду Пуск → Всі програми → Microsoft Office → Microsoft Word.

Слід зазначити, що версії Word від 2010 до 2019 року випуску різняться незначно. Навчившись користуватися Word 2013, з якою ми продовжуємо знайомитися, ви зорієнтуєтеся і в решті версій програми.

Створення й збереження текстового документа

Отже, після запуску Word на екрані з'являється перелік останніх документів і вікно онлайн-пошуку шаблонів (рис. 6.6). Можна знайти потрібний документ і відкрити шаблон із певними даними.

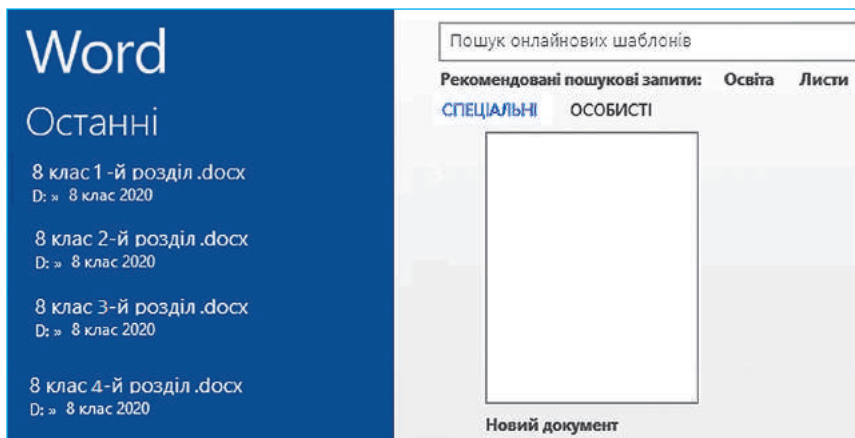
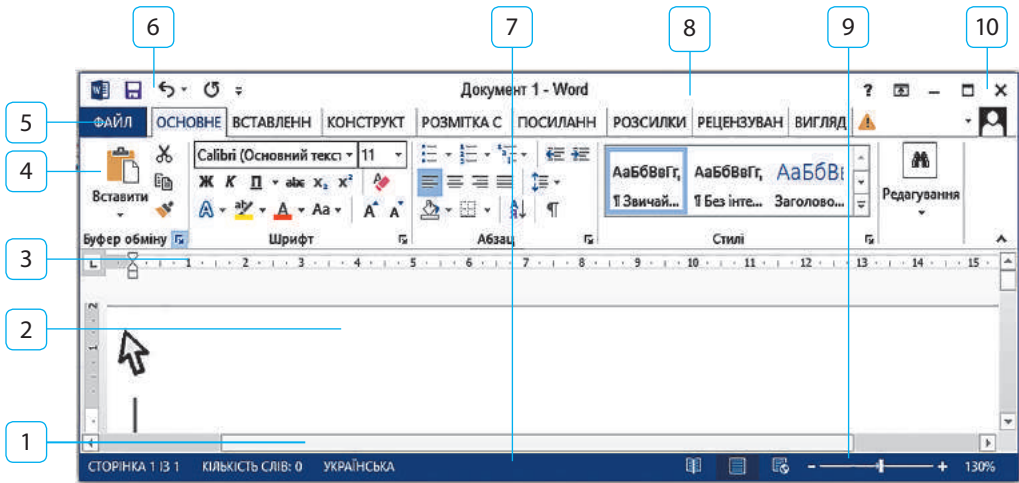


Рис. 6.6

► Створення текстового документа

Для **створення порожнього документа** слід клацнути зображення із написом **Новий документ**. Після цього відкриється вікно Word (рис. 6.7).




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


Рис. 6.7

► Збереження текстового документа

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

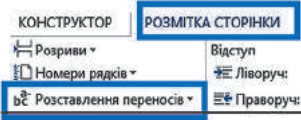
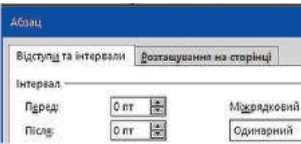
Щоб **зберегти текстовий документ**, потрібно виконати такі дії:

- 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) ввести назву файлу в поле **Ім'я файлу** і натиснути кнопку **Зберегти** (для збереження наявного документа в іншу папку та/або з іншим іменем слід вибрати команду **Зберегти як...**).


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

Правила набору тексту

Вміння швидко набирати текст на комп'ютері дуже важливе. Разом із тим текст потрібно набирати грамотно, а для цього необхідно дотримуватися певних правил. Розглянемо деякі з них:

Правило	Приклади	
	правильно	неправильно
Під час друку між словами ставте лише один пропуск (проміжок)	Правила набору текстів	Правила набору текстів
Розділовий знак (крапку, кому, двокрапку тощо) не відокремлюйте пропуском від попереднього слова	Привіт, друже!	Привіт, друже!
Після кожного розділового знака ставте пропуск	3 повагою, Марійка.	3 повагою,Марійка.
Текст, який береться в лапки або дужки, пропуском від них не відділяйте	(правила набору) «цифровий»	(правила набору) «цифровий»
Біля дефіса не робіть пропуски	Людина-робот	Людина-робот
Тире відділяйте пропусками	Книга — джерело знань	Книга-джерело знань
Не використовуйте клавішу Enter для переходу на новий рядок у межах одного абзацу (перехід відбувається автоматично)	Не використовуйте клавішу Enter для переходу на новий рядок в межах одного абзацу	Не використовуйте клавішу Enter для переходу на новий рядок в межах одного абзацу
Не ставте в кінці рядка знак переносу (це можна налаштувати)		Не ставте знаків переносу
Не вставляйте порожні рядки (для цього існують міжабзацні та міжрядкові інтервали)		Не вставляйте рядки. Існують інтервали.

Закінчення таблиці

Правило	Приклади	
	правильно	неправильно
Не нумеруйте вручну списки та сторінки (для цього є спеціальні методи)	 <p>1. → Січень¶ 2. → Лютий¶ 3. → Березень¶</p>	<p>1. Січень¶ 2. Лютий¶ 3. Березень¶</p>

Маємо такі позначення в таблиці: ¶ — кінець абзацу (натискання Enter); • — пропуск; → — знак табуляції (під час автонумерації).

Отже, комп'ютерний набір тексту, виконаний за всіма правилами, сприяє правильній автоматичній перевірці правопису, полегшує читання та сприймання тексту, спрощує подальшу роботу над документом.

► «Гарячі» клавіші Word

Для підвищення швидкості введення та опрацювання текстів багато команд у Word можна викликати за допомогою клавіш. Такі клавіші і їхнє сполучення називають «гарячими», або «швидкими». Деякі «гарячі» клавіші представлені на форзаці 1.

Перевірка правопису

Текстовий процесор оснащений засобом, який дозволяє перевірити й виправити написання слів у тексті. Якщо слово підкреслюється червоною хвилястою лінією, це зазвичай свідчить про можливу помилку.

► Автоматична перевірка правопису

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

Виклик контекстного меню для слова з помилкою (кляцання 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) (рис. 6.8).

Перевірка правопису вимикається автоматично, якщо кількість помилок у текстовому документі перевищує дві тисячі.

ПОМИЛКА :



помилка
помилко
помилкам
помилках
Пропустити все
Додати до словника
 Гіперпосилання...
 Примітка

Рис. 6.8

► Вибір мови перевірки правопису

Мову перевірки правопису можна змінити, якщо у рядку стану клацнути команду Мова, а в діалоговому вікні вибрати мову (рис. 6.9).

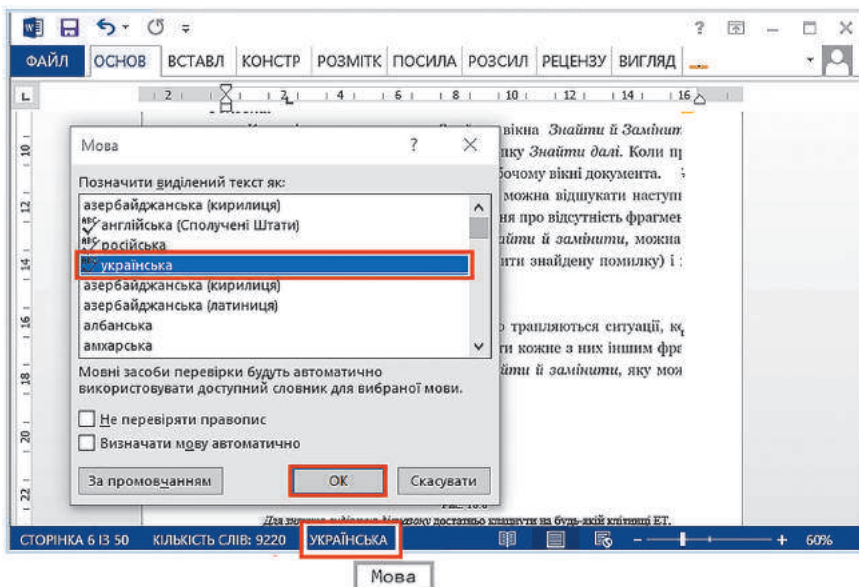


Рис. 6.9

У разі вимкненої автоматичної перевірки правопису його можна перевірити вручну. Для цього потрібно:

- 1) виділити фрагмент тексту;
- 2) клацнути кнопку Мова, що в рядку стану (рис. 6.9);
- 3) вибрати в діалоговому вікні Мова потрібну мову і клацнути кнопку ОК (див. рис. 6.9);
- 4) на вкладці Рецензування у групі інструментів Правопис натиснути кнопку Правопис і граматика (рис. 6.10).

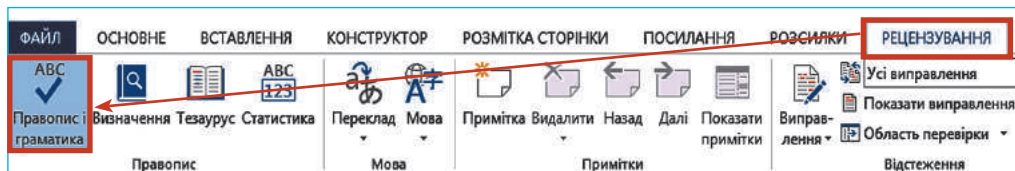


Рис. 6.10

Питання для самоперевірки



1. Що таке текстовий редактор; які текстові редактори вам відомі?
2. Чому текстовий редактор Word називають процесором?
3. Як запустити текстовий процесор і створити новий документ?
4. Назвіть кілька правил набору тексту.
5. Назвіть кілька «гарячих» клавіш або їх сполучень для редагування та форматування тексту.
6. Як увімкнути перевірку правопису у Word?

Вправа 6



1. Запустіть текстовий процесор і створіть новий документ.
2. Параметри сторінки і 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. Дотримуючись правил набору тексту, наберіть текст куплету Державного Гімну України. Куплет і приспів розмістіть в окремих абзацах (рис. 6.11). Приспів відформатуйте темно-жовтим кольором. За потреби виправте орфографічні помилки, слідкуйте, щоб не було підкреслення червоною хвилястою лінією.

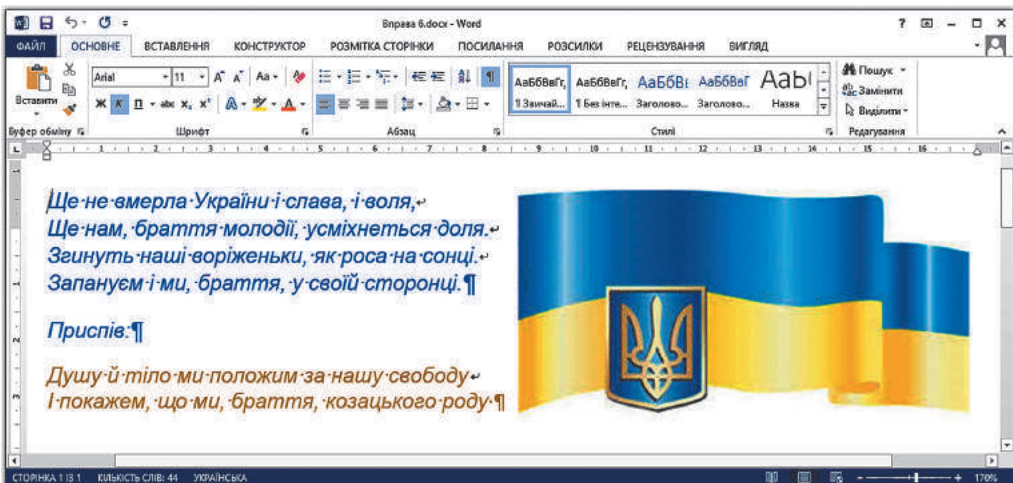


Рис. 6.11



5. Знайдіть в інтернеті зображення Державного Прапора України та Малого герба України, скопіюйте їх і вставте в документ. За потреби змініть розмір зображення. Розташуйте його праворуч від тексту (див. рис. 6.11).
6. Збережіть документ із назвою Вправа 6.

Комп'ютерне тестування

Виконайте тестове завдання 6 із автоматичною перевіркою результату.



§ 7. Робота з фрагментами тексту

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

Виділення фрагментів тексту

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

► Виділення за допомогою клавіатури

Для **виділення в документі всього тексту** можна скористатися сполученням клавіш Ctrl+A.

Щоб **виділити фрагмент тексту**, слід:

- 1) установити текстовий курсор на початок або кінець фрагмента;
- 2) утримуючи натиснутою клавішу Shift, переміщувати курсор клавішами керування курсором (рис. 7.1).

Виділений фрагмент тексту підсвічується кольором (рис. 7.2). Щоб **зняти виділення**, потрібно натиснути будь-яку клавішу керування курсором (див. рис. 7.1).

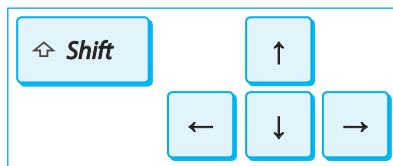


Рис. 7.1

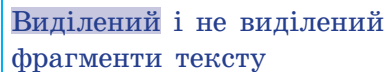


Рис. 7.2

► Виділення за допомогою миші

Для **виділення фрагмента тексту мишею** потрібно встановити її вказівник на початок фрагмента і, утримуючи натиснутою ліву кнопку миші, протягнути вказівник до кінцевого положення.

Для **виділення кількох фрагментів тексту** описані дії мишею повторюють з утриманням натиснутою клавіші Ctrl.

Існують швидкі способи виділення мишею. Розглянемо таблицю:

Дія	Результат
Клацнути на лівому полі документа	Виділиться рядок
Протягнути вказівник по лівому полю	Виділиться декілька рядків
Двічі клацнути на лівому полі	Виділиться абзац
Тричі клацнути на лівому полі	Виділиться весь текст документа
Двічі клацнути на символі	Виділиться слово
Тричі клацнути на символі	Виділиться абзац
Клацнути перед першим символом і, утримуючи клавішу Shift , клацнути за останнім символом фрагменту (рис. 7.3)	Виділиться текстовий фрагмент від першого до останнього символу

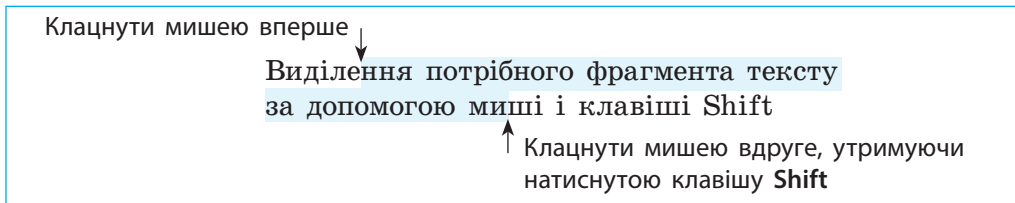


Рис. 7.3

Щоб **зняти виділення фрагмента тексту**, слід клацнути мишею в будь-якому місці робочого поля (на сторінці документа).

Копіювання та переміщення фрагментів тексту

До виділених фрагментів тексту можна застосовувати низку операцій, які виконуються за допомогою команд контекстного меню або сполучень клавіш (див. форзац 1). Наприклад, це може бути переміщення або вставлення, що передбачає копіювання в *буфер обміну* — ділянку пам'яті, призначену для тимчасового зберігання даних різних типів.

Таким чином, щоб перемістити виділений фрагмент з одного місця документа в інше, потрібно його вирізати (Ctrl+X), а потім уставити з буфера обміну (Ctrl+V) в інше місце.

- ! Не плутайте операції вирізання (Ctrl+X) і видалення (Delete), оскільки вирізаний об'єкт зберігається в буфері обміну, а видалений об'єкт — не зберігається.

На відміну від системного буфера обміну, який може містити лише один об'єкт, буфер обміну Microsoft Office може зберігати до 24 об'єктів, скопійованих із будь-яких джерел: із браузера, з різних офісних програм тощо. Якщо об'єктів більше, то скопійовані останніми залишаються в буфері обміну, а скопійовані першими — видаляються.

- 1 Існує спосіб переміщення текстового фрагмента без його зберігання в буфері обміну. Для цього достатньо виділити текстовий фрагмент і перетягнути вказівником миші на інше місце в документі (рис 7.4).

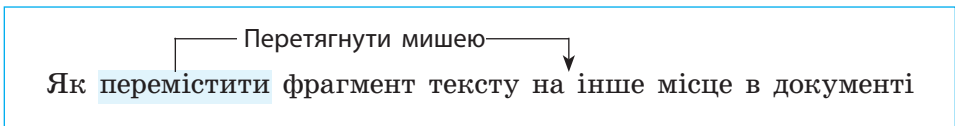


Рис. 7.4

Щоб переглянути вміст буфера обміну, потрібно вибрати на вкладці Основне кнопку відкриття діалогового вікна групи інструментів Буфер обміну (рис. 7.5).

А щоб уставити в документ довільний об'єкт, збережений у буфері обміну, достатньо двічі клацнути на вибраному елементі у вікні Буфер обміну.

Після того як усі програми пакета Microsoft Office закрито, у буфері обміну операційної системи Windows залишається доступним лише останній об'єкт з буфера обміну Microsoft Office (верхній у вікні Буфер обміну (рис. 7.5)).

Цей об'єкт зберігатиметься в системному буфері обміну включеного комп'ютера доти, доки замість нього до буфера обміну не буде скопійовано інший об'єкт.

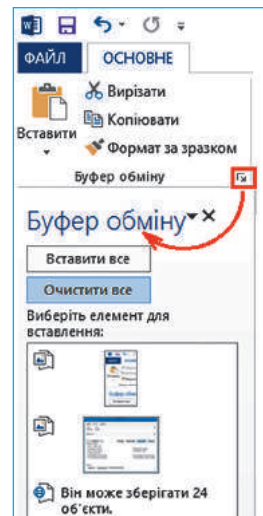


Рис. 7.5

Автоматичний пошук і заміна фрагментів тексту

Знайти в тексті необхідні фрагменти (наприклад, повторювані) вручну буває важко, особливо у великих за обсягом документах. Текстовий процесор оснащений засобом для автоматичного пошуку об'єктів документа (текстових фрагментів, рисунків, таблиць, формул та ін.).

► Пошук текстових фрагментів

Для **пошуку фрагмента тексту** потрібно:

- 1) на вкладці Основне в групі інструментів Редагування клацнути команду Пошук (рис. 7.6) або натиснути сполучення клавіш **Ctrl+F** — відкриється бокова вкладка Навігація;
- 2) у текстове поле пошуку ввести текст, який потрібно відшукати — на вкладці Навігація з'являться уривки текстового документа зі знайденим фрагментом тексту (рис. 7.6);
- 3) для переходу до відповідного місця в текстовому документі клацнути вибраний уривок, який виділено жовтим тлом (рис. 7.6).

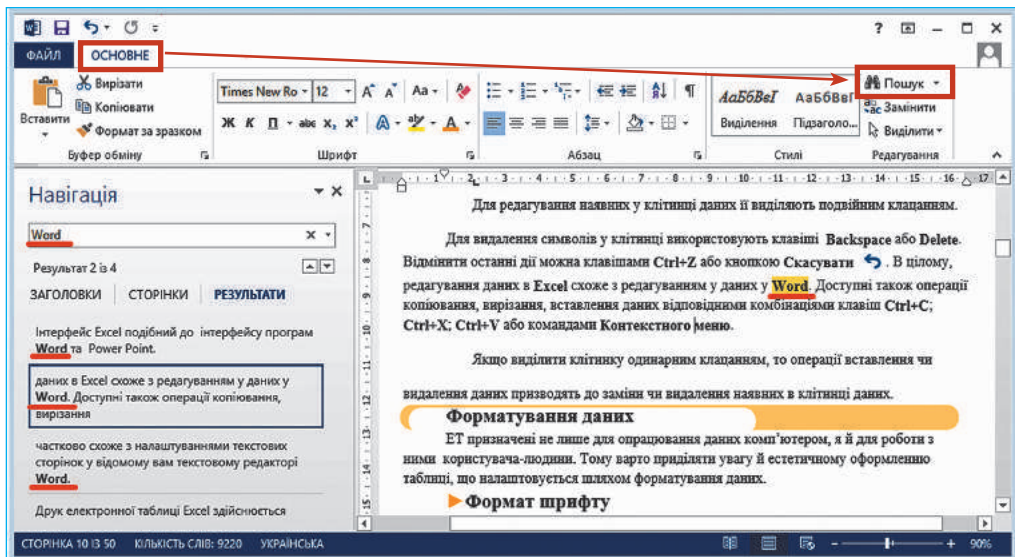


Рис. 7.6

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

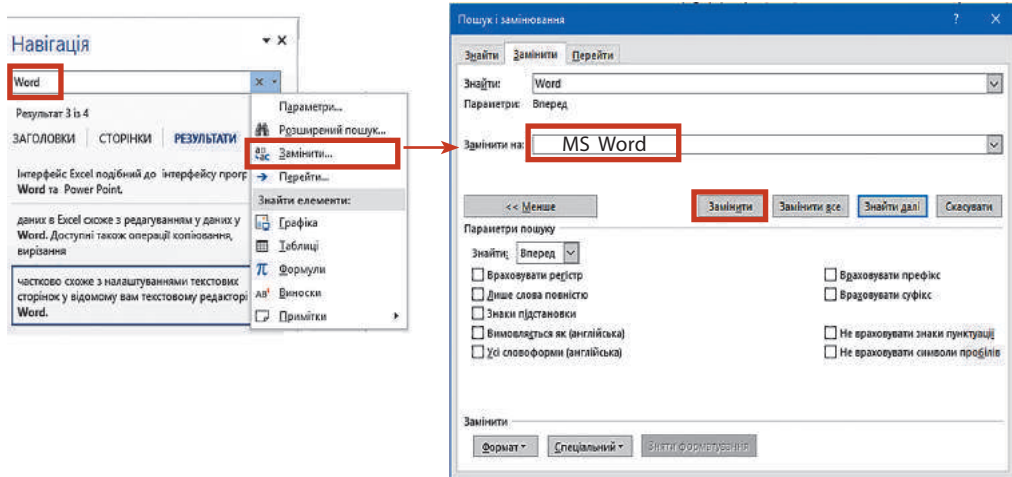
► Заміна текстових фрагментів

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

Щоб замінити фрагмент, слід виконати такі дії:

- 1) на боковій вкладці Навігація у списку, що розгортається, вибрати команду Замінити (рис. 7.7, а);
- 2) у вікно Пошук і замінювання, що відкриється, у полі Замінити на ввести текст, яким потрібно замінити знайдений текстовий фрагмент, наприклад, MS Word (рис. 7.7, б);
- 3) натиснути кнопку Замінити (або Замінити все) (рис. 7.7, б).

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



а

б

Рис. 7.7

Діалогове вікно Пошук і замінювання (див. рис. 7.7, б) можна відкрити також сполученням клавіш Ctrl+N.

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

Для пошуку інших об'єктів: рисунків, таблиць, формул тощо, потрібно вибрати відповідну команду зі списку, що розгортається біля текстового поля пошуку на вкладці Навігація (див. рис. 7.7, а).

Питання для самоперевірки



1. Як виділити в текстовому документі такі об'єкти, як весь текст, слово, рядок, речення, абзац?
2. Як скопіювати, вирізати, видалити, перемістити в текстовому документі виділений об'єкт?
3. Чим відрізняються операції Вирізати і Видалити?
4. Що таке буфер обміну і як переглянути об'єкти, збережені в буфері обміну Microsoft Office?
5. Поясніть, як здійснити автоматичний пошук текстового фрагмента в документі. Опишіть, як у текстовому документі можна виконати автоматичний пошук графічних зображень, таблиць, списків тощо.
6. Як виконати автоматичну заміну в документі одного текстового фрагмента на інший?



Вправа 7



1. Запустіть текстовий процесор і створіть новий документ.
2. Знайдіть в інтернеті стислий опис однієї з визначних пам'яток історії або природи тієї місцевості (області), де ви мешкаєте.
3. Скопіюйте в буфер обміну декілька абзаців знайденого опису й уставте у створений документ.
4. За потреби відредагуйте, відформатуйте текст, за потреби виправте помилки (слідкуйте, щоб не було червоного підкреслення). Чи для всіх слів вдалося це зробити?
5. Виберіть у тексті одне з повторюваних слів або словосполучень і замініть його у всьому 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 7.

Комп'ютерне тестування



Виконайте тестове завдання 7 із автоматичною перевіркою результату.



§ 8. Форматування об'єктів текстового документа

Набраний і відредагований текст часто потребує форматування. Форматування тексту — це процес надання йому потрібного зовнішнього вигляду. Для того, щоб відформатувати символи введеного тексту, їх необхідно попереньо виділити.

Властивості об'єктів текстового документа

Форматування документа у Word полягає у змінненні властивостей окремих його об'єктів. Розглянемо таблицю:

Об'єкт	Властивості
Символ	Шрифт, накреслення, розмір, колір тощо
Абзац	Вирівнювання (за лівим чи правим краєм, по центру або за шириною); відступи праворуч і ліворуч; відступ першого рядка; відступи до й після абзацу; міжрядковий інтервал тощо
Рисунок	Розташування, обтинання, різкість, яскравість і контрастність, обтікання текстом, межі тощо
Таблиця	Розташування, об'єднання комірок, напрямок тексту в комірках, товщина і колір меж, колір заливки комірок тощо
Список	Вигляд маркерів або номерів, число рівнів нумерації, початкове значення нумерації тощо
Формула	Вигляд індексів, дробів тощо, розташування формули, лінійний або професійний вигляд тощо
Сторінка	Розміри сторінки, ширина полів, орієнтація сторінки, рамка, колонтитули тощо
Розділ	Нумерація сторінок, кількість колонок тощо
Документ	Нумерація сторінок, зміст тощо

Після ознайомлення з форматуванням символів, абзаців, рисунків, таблиць розглянемо опрацювання інших об'єктів текстового документа: недрукованих і/або відсутніх на клавіатурі символів, формул, списків.

Робота із символами і формулами

Бувають випадки, коли до тексту потрібно додати символи, яких немає на клавіатурі: наприклад, грецькі літери, знак авторського права, знаки арифметичних дій, недруковані символи, 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 у 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.

1 Якщо увімкнути режим відображення недрукованих символів, наведений текст набуде такого вигляду (рис. 8.1):

Між·словами·з'являються·крапки,·а·також·знаки·абза-
цу·та·інші·символи,·які·раніше·не·виводилися.¶

Рис. 8.1

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

Вигляд недрукованого символу	Назва недрукованого символу	Клавіші
·	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 виникає потреба в додаванні символів, яких на клавіатурі немає (наприклад, символу §).

Для **вставлення в текст символів**, які відсутні на клавіатурі, потрібно на вкладці Вставлення в групі Символи розгорнути список Символ (рис. 8.2). Якщо потрібного символу в колекції немає, слід під списком вибрати команду Інші символи, а потім знайти і вставити потрібний.

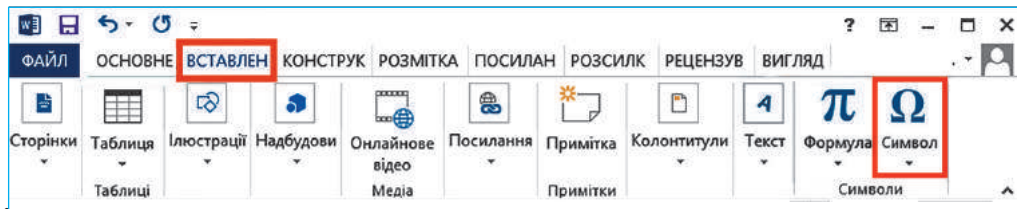


Рис. 8.2

► Формули в текстовому документі

Word має вбудовані засоби для роботи з формулами.

Для **вставлення в документ готової формули** потрібно на вкладці Вставлення в групі Символи розгорнути значком ▼ колекцію формул Формула, після чого вибрати потрібну формулу.

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

За потреби можна **створити власну формулу**, скориставшись інструментом вкладки Конструктор (Знаряддя для формул). Назва цієї вкладки з'являється у рядку назв інших вкладок, якщо клацнути віконець створюваної формули (рис. 8.3).

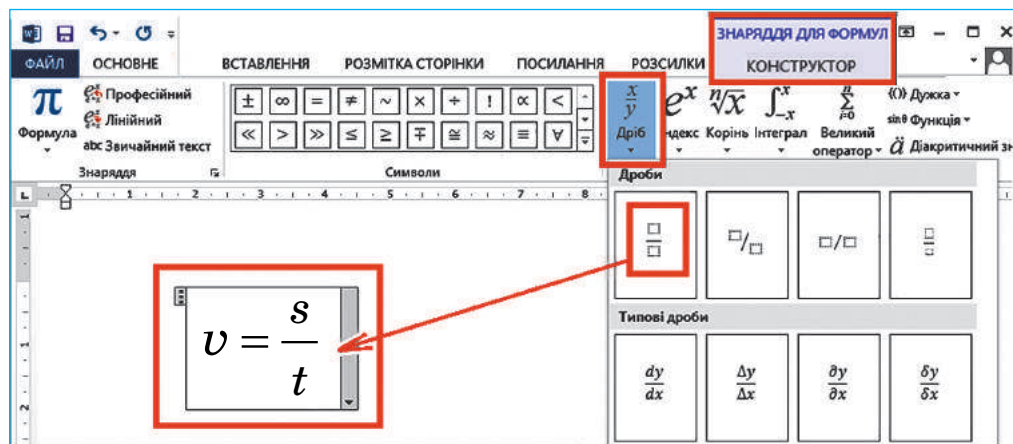


Рис. 8.3

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

- 2 Для зведення формули до однорядкового вигляду скористаймося інструментом Лінійний (рис. 8.4). Зверніть увагу на те, що при цьому формула, зображена на рисунку, перетвориться на таку: $v = s / t$.

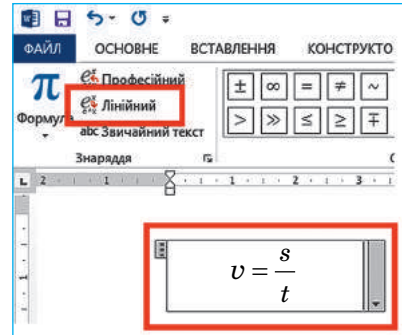


Рис. 8.4

Списки в текстовому документі

Упорядковану інформацію (наприклад переліки, інструкції, плани заходів тощо) зручно подавати у вигляді **списків** — послідовності абзаців, що мають спільну позначку (маркер або номер). Ви навчилися створювати однорівневі марковані і нумеровані списки. Але ж текстові процесори дають змогу створювати списки різних типів.

Ознайомимося з різними типами списків докладніше (рис. 8.5).

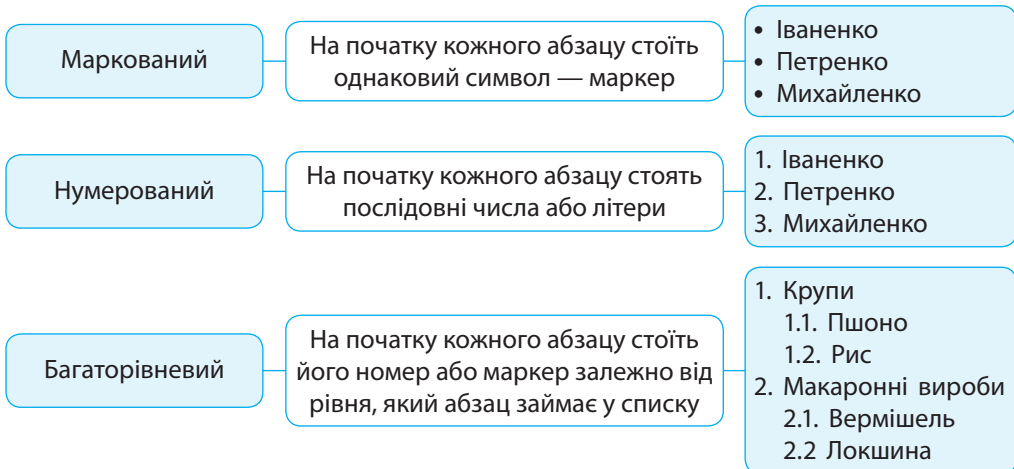


Рис. 8.5

▶ Створення списків

Існує кілька способів оформлення абзаців тексту у вигляді списків. Для **перетворення введеного тексту на список** потрібно:

- 1) виділити відповідні абзаци;
- 2) знайти на вкладці Основне у групі Абзац інструмент Маркери (рис. 8.6, а) або Нумерація (рис. 8.6, б);
- 3) відкрити колекцію кнопкою ▼ і вибрати потрібний тип маркера (номера) або створити власний маркер (номер) командою Визначити новий маркер (числовий формат).

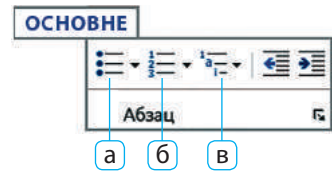


Рис. 8.6

Для **створення списку** перед уведенням тексту потрібно вибрати значок маркера або формат номера — з'явиться позначка першого абзацу. Після введення тексту і завершення абзацу натисканням клавіші Enter з'явиться позначка другого абзацу, третього і т. д.

Для **автоматичного створення списку** потрібно на початку абзацу ввести символи: зірочку «*» (для маркованого списку) або одиницю з крапкою чи дужкою: «1.» , «1)» (для нумерованого списку). Автоматичне створення списку можна скасувати, двічі натиснувши Enter.

Для **створення багаторівневих списків** застосовують інструмент Багаторівневий список (рис. 8.6, в). Багаторівнева нумерація дозволяє позначати різними маркерами або номерами різні рівні списку.

- 3 На рис. 8.7 перший, другий, третій рівні списку виділено червоним, зеленим і синім кольорами відповідно.

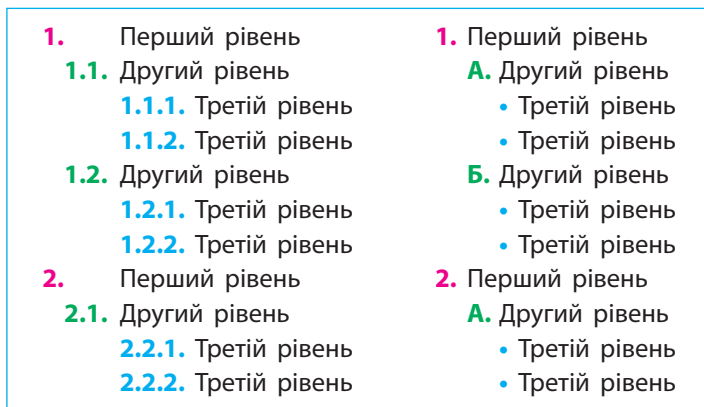


Рис. 8.7

► Редагування списків

У текстовому процесорі Word створені списки зручно редагувати.

Для **додавання абзацу до списку** потрібно встановити текстовий курсор на початок або в кінець відповідного абзацу тексту та натиснути клавішу Enter. Утворений абзац автоматично стане елементом списку, а нумерація наступних абзаців відповідно збільшиться.

Для **скасування нумерації одного абзацу** потрібно клацнути мишею в межах абзацу і вимкнути кнопку списку (див. рис. 8.6, *a–в*) або вилучити номер (маркер). Для цього на початку абзацу слід поставити курсор і натиснути клавішу Backspace — решта абзаців списку автоматично перенумерується.

Щоб **скасувати нумерацію кількох абзаців**, ці абзаци потрібно виділити і вимкнути кнопку списку. Якщо значення номера, утворене, наприклад, під час копіювання списку, не відповідає потребам, користувач може його змінити. Для цього в контекстному меню номера слід вибрати команду Почати заново або Продовжити нумерацію.

► Форматування списків

Форматування списку передбачає змінення вигляду значків маркерів і форматів номерів, відстані між маркерами і текстом тощо.

Для **замінення наявного маркера на інший** потрібно клацнути по ньому, відкрити Бібліотеку маркерів (рис. 8.8), вибрати потрібний маркер або, скориставшись командою Визначити новий маркер, призначити маркер, який відсутній у бібліотеці.

Для **встановлення нового формату номерів** треба виділити потрібні абзаци списку або клацнути якийсь номер (якщо є потреба змінити всі номери певного рівня) і змінити формат, скориставшись відповідними командами, наведеними на рис. 8.9.

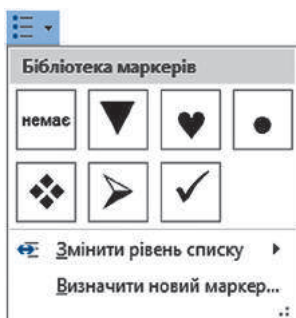


Рис. 8.8

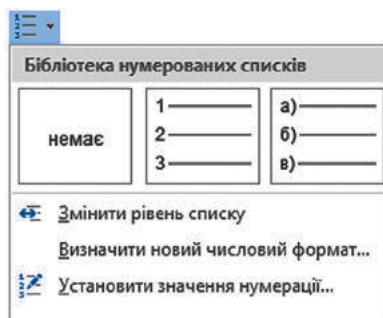


Рис. 8.9

Питання для самоперевірки



1. Назвіть кілька об'єктів текстового документа і поясніть, у чому полягає їх форматування.
2. Що таке недруковані символи і як їх відобразити на екрані?
3. Опишіть алгоритм вставлення в текстовий документ символа, відсутнього на клавіатурі.
4. Як створити власну формулу в текстовому документі?
5. Як створити багаторівневий список?

Вправа 8



1. Запустіть текстовий процесор і створіть новий документ.
2. Установіть такі параметри сторінки та шрифту:
 - розмір сторінки — А5;
 - 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 та створіть у текстовому документі (нижче, під списком) цю формулу.

Перегляньте в колекції формул, чи наявна в ній потрібна вам формула питомої теплоємності. За допомогою контекстного меню збережіть створену формулу в колекції Word.



5. Додайте до тестового документа під формулою новий абзац із написом:

§ 8. Форматування.

Встановіть для доданого абзацу розмір шрифту — 16, накреслення — жирне. Знак параграфа можна знайти у вікні вставлення Символ на вкладці Спеціальні символи.

6. Збережіть документ із назвою Вправа 8.

Комп'ютерне тестування



Виконайте тестове завдання 8 із автоматичною перевіркою результату.





Практична робота 3

Створення текстового документа, що містить об'єкти різних типів

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

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

Хід роботи


Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть текстовий процесор і створіть новий документ.
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, курсив.
2. Уведіть:
 - у *першому* абзаці знак і номер параграфа — § 1;
 - у *другому* абзаці — заголовок Рівномірний прямолінійний рух;
 - задайте вирівнювання — За серединою.
3. Знайдіть в інтернеті зображення гепарда, скопіюйте його та вставте в документ під заголовком.
4. Задайте формат обтікання малюнка текстом Зверху і знизу та обріжте малюнок так, щоб залишилося лише зображення гепарда. Пропорційно змініть розміри малюнка так, щоб за шириною він зайняв усю сторінку (від поля до поля).
5. Установіть для заголовка розмір шрифту — 31; відформатуйте текст так, щоб заголовок розмістився в одному рядку, змінивши за потреби міжсимвольний інтервал.

6. Знайдіть в інтернеті дані про швидкості руху пішохода, гепарда і сокола та введіть у документ в окремих абзацах (замість нулів, див. рисунок).

§ 1

Рівномірний прямолінійний рух



1. Швидкість пішохода 000 км/год
2. Швидкість гепарда 000 км/год
3. Швидкість сокола 000 км/год

Основні величини		
назва	позначення	формула
Пройдений шлях	s	$s = v \cdot t$
Швидкість руху	v	$v = \frac{s}{t}$
Час руху	t	$t = \frac{s}{v}$

7. Перетворіть абзаци з отриманими даними на нумерований список за зразком, наведеним на рисунку.
8. Додайте в документ таблицю, що складається із 5 рядків і 3 стовпців, і розташуйте під списком.
9. Об'єднайте клітинки верхнього рядка таблиці та введіть назву таблиці — Основні величини.
10. Заповніть таблицю за зразком, уведіть необхідні формули.
11. Оформіть (відформатуйте) таблицю за зразком, розташуйте написи по центру, перемістіть межі таблиці тощо.
12. Збережіть файл з іменем Практична робота 3. Завершіть роботу за комп'ютером.

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

§ 9. Структура складного текстового документа

Зазвичай великі за обсягом документи мають кілька розділів із наскрізною нумерацією сторінок. Для оформлення багатосторінкових документів дуже важлива сувора упорядкованість розміщення як розділів, так і заголовків, означень, приміток тощо.

Структурні елементи текстового документа

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



Структура документа — ієрархічна схема розміщення складових текстового документа.

Великі складні документи розбивають на частини, розділи, підрозділи, параграфи та інші структурні елементи (рис. 9.1), відформатовані з дотриманням певних правил, про які йдеться в наведеній далі таблиці.

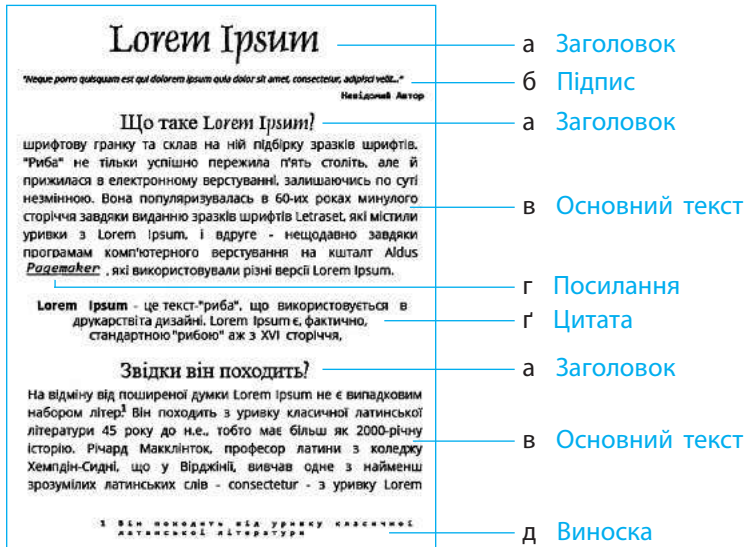


Рис. 9.1

Розглянемо цю таблицю:

Елемент	Опис	Форматування	
		шрифт	абзац
Заголовок (рис. 9.1, 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 документа. Заголовки вищого рівня оформляються вагоміше за нижчий (більші символи тощо)	Зазвичай вирівнювання За серединою
Підпис (або Автор) (рис. 9.1, б)	Супровідна інформація до назви розділу або цитати	Шрифт менш помітний, ніж заголовок, або текст цитати. Зазвичай накреслення — курсив. Якщо підпис стоїть угорі документа, то можливий більший шрифт, ніж в основному тексті	Зазвичай вирівнювання За правим краєм
Основний текст (рис. 9.1, в)	Найбільша за обсягом інформації частина документа	Шрифт нещільний і не фігурний, легкий для сприйняття	Зазвичай вирівнювання За шириною
Цитати, адреси, посилання на інші документи (рис. 9.1, г)	Виділена частина основного тексту	Зазвичай накреслення курсив	Фрагмент того самого абзацу
Важлива інформація (рис. 9.1, г)	Означення, висновки, попередження або інша інформація, що потребує особливої уваги	Шрифт більший і/або щільніший порівняно з основним текстом	Зазвичай вирівнювання За серединою
Допоміжна інформація (рис. 9.1, 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 у Word застосовують стильове оформлення.



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

Щоб задати стильове оформлення виділеного об'єкта, необхідно обрати певний стиль на вкладці Основне у групі Стилі (рис. 9.2).

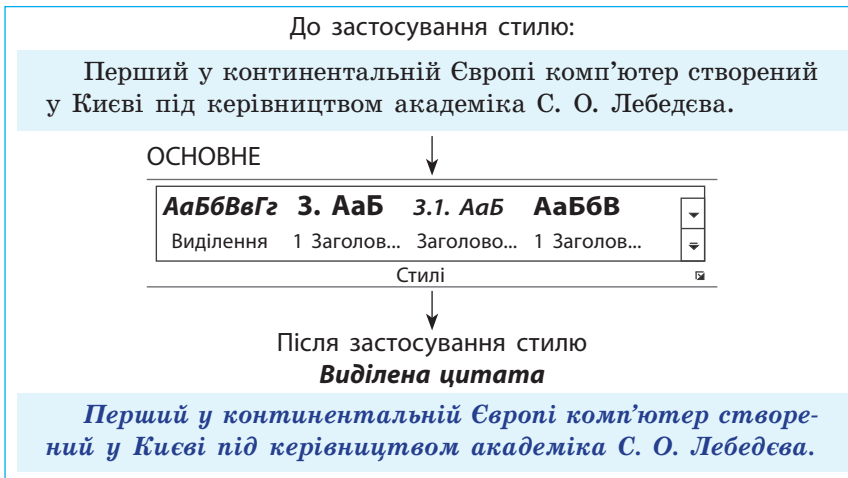


Рис. 9.2

Щоб змінити стиль, потрібно:

- 1) відкрити контекстне меню стилю (клацнути віконце правою кнопкою миші);
- 2) вибрати команду Змінити;
- 3) у вікні Зміна стилю клацнути кнопку Формат;
- 4) налаштувати параметри стилю.

Після цього зміниться вигляд усіх об'єктів, які мали такий стиль.

Шаблони документів

Погодьтеся, що досить часто нам доводиться створювати документи зі схожою структурою. У такому випадку доречно скористатися заздалегідь підготовленими та збереженими шаблонами документів.



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

У Word 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 вибрати тип файлу — Шаблон Word (*.dotx). Після цього з'являється можливість створити новий документ на основі цього 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 програмі Word створюється на основі шаблону **Normal.dotx**.

За наявності доступу до інтернету Word дозволяє шукати і використовувати різні шаблони документів. Це можна зробити зразу після запуску Word (див. рис. 6.6).

Щоб скористатись онлайн-шаблоном для створення документа, потрібно на вкладці Файл клацнути команду Створити і вибрати тематику онлайн-шаблонів, а потім вибрати бажаний шаблон (рис. 9.3).

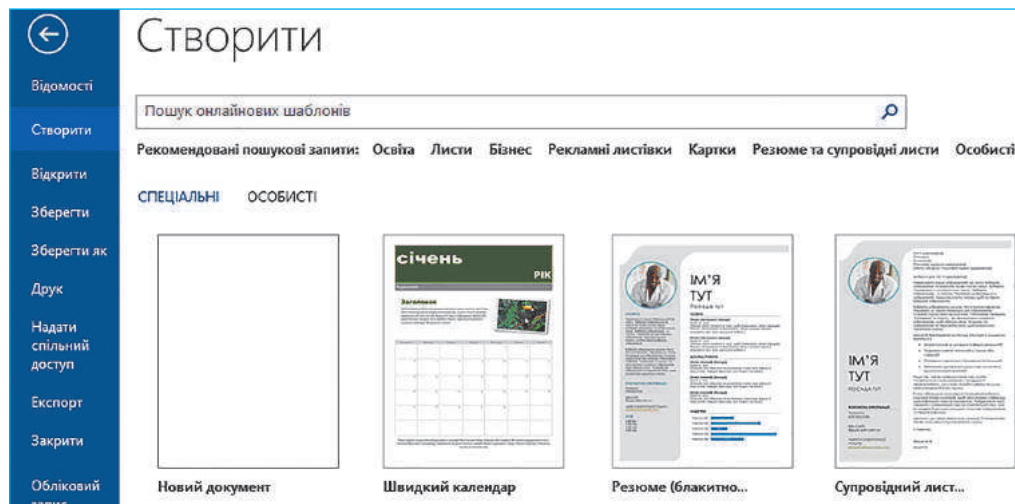


Рис. 9.3

Розділи у Word

Іноді для сторінок, які містяться в різних частинах документа, необхідно встановити різні формати. Для цього документ потрібно розділити на розділи, встановлюючи в потрібних місцях відповідні розриви (рис. 9.4).

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

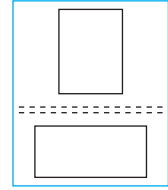


Рис. 9.4



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

► Розриви розділів

Розрив розділу зазначає місце, після якого можна задати нові параметри розмітки сторінки, притаманні лише цьому розділу (див. рис. 9.4).

Для **вставлення розриву розділу** потрібно:

- 1) клацнути на вкладці Розмітка сторінки у групі Параметри сторінки елемент Розриви;
- 2) вибрати потрібний тип розриву розділу (рис 9.5).

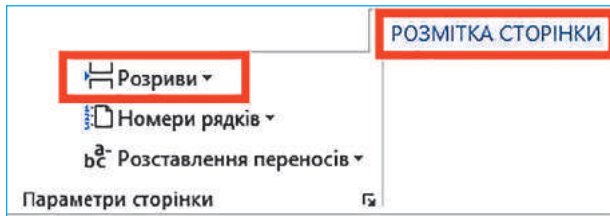


Рис. 9.5

Тепер у документі для окремих розділів можна встановити різні налаштування, наприклад, такі як розміри та орієнтацію сторінки, ширину полів, кількість колонок, джерело паперу для принтера тощо.

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

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

► Режим структури документа

Упорядковувати складові частини (розділи, параграфи) текстового документа великого обсягу (рис. 9.6), наприклад підручника, більш зручно в режимі структури документа.

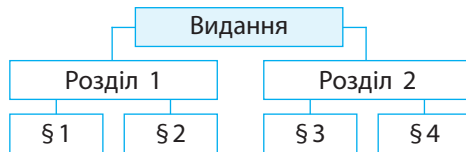


Рис. 9.6

Режим Структура у Word дозволяє приховати основний текст і побачити лише заголовки розділів, параграфів тощо. В цьому режимі можна змінювати рівень заголовків, копіювати і переміщувати параграфи та розділи, переміщуючи відповідні заголовки.

Щоб увімкнути режим структури, потрібно на вкладці Вигляд у групі Подання вибрати команду Структура (рис. 9.7).

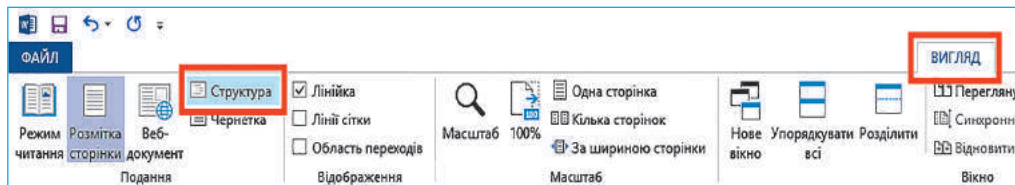


Рис. 9.7

Колонтитули

Текстовий процесор дозволяє додати на сторінки документа колонтитули, в яких міститься допоміжна інформація.



Колонтитули — повідомлення, які розміщуються у верхньому або нижньому полі сторінок текстового документа.

Інформація, подана в колонтитулі, відображається на всіх сторінках документа або на сторінках одного чи кількох розділів. Розрізняють верхній, нижній і бічні колонтитули.

Колонтитули можуть містити номери сторінок, назву документа, поточного розділу або параграфа, прізвище автора, зображення тощо.

Для встановлення колонтитула необхідно:

- 1) двічі клацнути в ділянці верхнього (або нижнього) краю поля або на вкладці Вставлення в групі Колонтитули вибрати відповідну команду;

- 2) в полі введення колонтитула ввести його текст або на вкладці Конструктор (Знаряддя для колонтитулів) у групі Вставлення вибрати потрібний інструмент (рис. 9.8);
- 3) налаштувати відображення колонтитула інструментами вкладки Конструктор (Знаряддя для колонтитулів) (рис. 9.8).

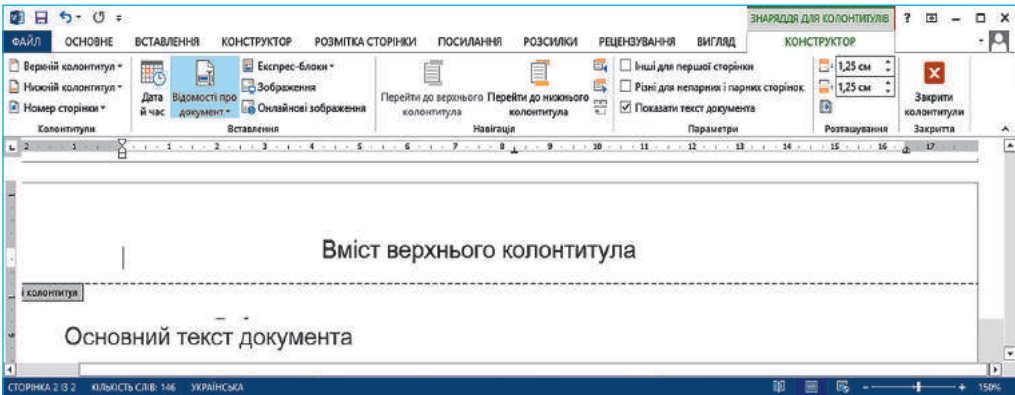




Рис. 9.8

Для повернення до основного тексту документа на ньому слід двічі клацнути або клацнути кнопку  на вкладці Конструктор (Знаряддя для колонтитулів) (див. рис. 9.8).

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

Для створення колонтитула для певного розділу документа слід:

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

Тепер уже можна змінювати вміст окремих колонтитулів, і це не впливатиме на колонтитули інших розділів документа.

Посилання

У програмі Word разом із підготовкою текстових документів, які призначено для друку, можна розробляти також електронні документи, які, зокрема, містять посилання.



Посилання (гіперпосилання) — об'єкт документа (малюнок, текст тощо), клацанням по якому здійснюється перехід у певне місце документа або до іншого файлу, розміщеного на комп'ютері користувача 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. 9.9);
- 3) вибрати Гіперпосилання і зазначити, на який саме об'єкт створюється посилання (рис. 9.9).

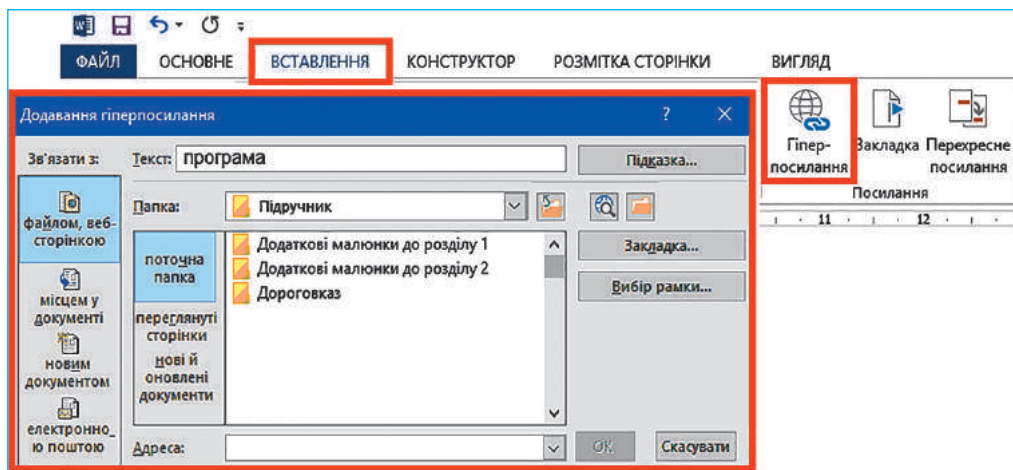



Рис. 9.9

Посилання також можна створити або змінити, якщо скористатися командою контекстного меню виділеного об'єкта Гіперпосилання.

Щоб створити посилання на певне місце в документі, в тексті цього документа необхідно створити закладку, на яку воно вказуватиме.

Для того щоб **перейти за посиланням**, необхідно клацнути на це посиланні лівою кнопкою миші, утримуючи натиснутою клавішу Ctrl.

Для створення закладки потрібно:

- 1) виділити об'єкт або встановити курсор у потрібне місце документа;
- 2) вибрати інструмент  (зкладка Вставлення, група Посилання);
- 3) увести назву нової закладки, клацнути 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. Що таке колонтитули і як їх налаштувати?
6. Як створити посилання (гіперпосилання) в тексті документа?

Вправа 9



►► Створити й оформити текстовий документ із використанням різних стилів.



- 1) Запустіть текстовий процесор і створіть новий документ.
- 2) Знайдіть в інтернеті одну з дій п'єси Івана Карпенка-Карого «Сто тисяч» і скопіюйте в документ.

3) Для основного тексту дії встановіть стиль Звичайний.

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 верхній колонтитул із поточною датою. Додайте в нижньому колонтитулі номери сторінок.
- 5) Зробіть назву твору посиланням на вебсторінку із цим твором.
- 6) Збережіть документ у форматі шаблону з назвою Вправа 9.

Комп'ютерне тестування

Виконайте тестове завдання 9 із автоматичною перевіркою результату.



§ 10. Зміст документа

Швидко орієнтуватися в багатосторінкових документах дає змогу зміст (рис. 10.1).



Зміст в електронному документі — це впорядкований список заголовків складових частин документа.

Зазвичай у змісті зазначаються номери сторінок, що дозволяє легко орієнтуватися в структурі великого за обсягом документа. Зміст розташовується на початку або в кінці документа, і часто перегляд починають саме з нього. Подивіться, де розташований зміст у вашому підручнику.

Наявність посилань на відповідні розділи в змісті електронного документа дозволяє швидко знаходити потрібний розділ та переміщуватися від одного розділу до іншого. Дуже зручно переміщуватися між розділами електронного текстового документа дозволяють інструменти вікна Навігація, яке можна ввімкнути прапорцем Область переходів на вкладці Видгляд (рис. 10.2).

Зміст можна створити в ручному або автоматичному режимах.

Зміст	
Передмова	3
Розділ 1. Алгоритми та програми	
§ 1. Основні поняття алгоритму та його записування	7
§ 2. Визначення алгоритму роботи з висхідними	11
§ 3. Створення алгоритму і програми із розв'язуванням на висхідних	16
§ 4. Різні як послідовності символів	21
§ 5. Оптимізація розв'язку алгоритму	27
Практична робота 1. Створення програми для обчислення різниці висхідних	
§ 6. Сторінка	186
§ 7. Оптимізація алгоритму швидкості	20
§ 8. Фінальний тест на складових частин	40
§ 9. Створення інтерфейсу користувача	47
Практична робота 2. Створення програми для обчислення висхідних	
§ 10. Створення меню модуля Windows	184
§ 11. Створення аналізів	190
§ 12. Обчислювальні програми	196
Практична робота 3. Використання функцій	
Розділ 2. Сучасні інструменти	
§ 13. Швидкі служби інструменту	77
§ 14. Робота з електронними таблицями	82
Практична робота 4. Створення електронної таблиці	
§ 15. Обчислювальні електронні таблиці	86
§ 16. Максимальна швидкість обчислень	95
§ 17. Використання інтернет-ресурсів	102
Практична робота 5. Використання змінних та функцій	

Рис. 10.1

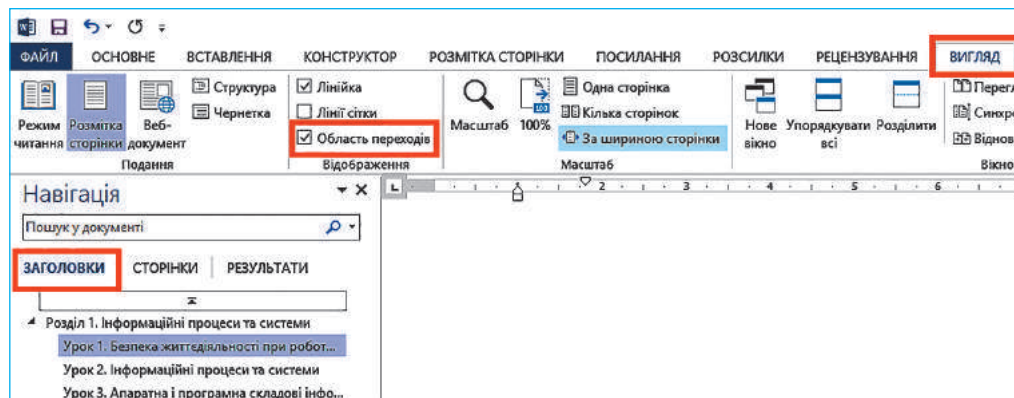


Рис. 10.2

Створення змісту в ручному режимі

Створення змісту в ручному режимі є доцільним для невеликих за обсягом документів, а також єдино можливим для документів, які налічують понад 9 рівнів заголовків (Word не підтримує більше ніж 9 рівнів заголовків).

Для створення змісту в ручному режимі потрібно:

- 1) пронумерувати сторінки документа (команда Вставлення → Номер сторінки);
- 2) створити сторінку із заголовками: назвами розділів, параграфів тощо.

Для цього слід або набрати назви заголовків вручну, або скопіювати з документа. Якщо заголовків не більше 24, то можна послідовно скопіювати їх всі до буфера обміну, наприклад, сполученням клавіш Ctrl+C, а потім вставити із буфера обміну на потрібну сторінку (див. рис. 7.5);

- 3) додати номери сторінок біля скопійованих заголовків — за необхідності;
- 4) у тексті документа до кожного заголовка створити закладку (див. матеріали § 9);
- 5) на створеній сторінці змісту кожену назву заголовка зробити посиланням на відповідну закладку в документі.

Створення змісту в автоматичному режимі

Автоматизована побудова змісту документа спрощується, якщо для заголовків і підзаголовків застосовуються стилі Заголовок 1, Заголовок 2, Заголовок 3 тощо.

Для автоматизованого створення змісту потрібно:

- 1) відформатувати назви розділів, параграфів тощо відповідними стилями (рис. 10.3);
- 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.

Для оновлення змісту потрібно:

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

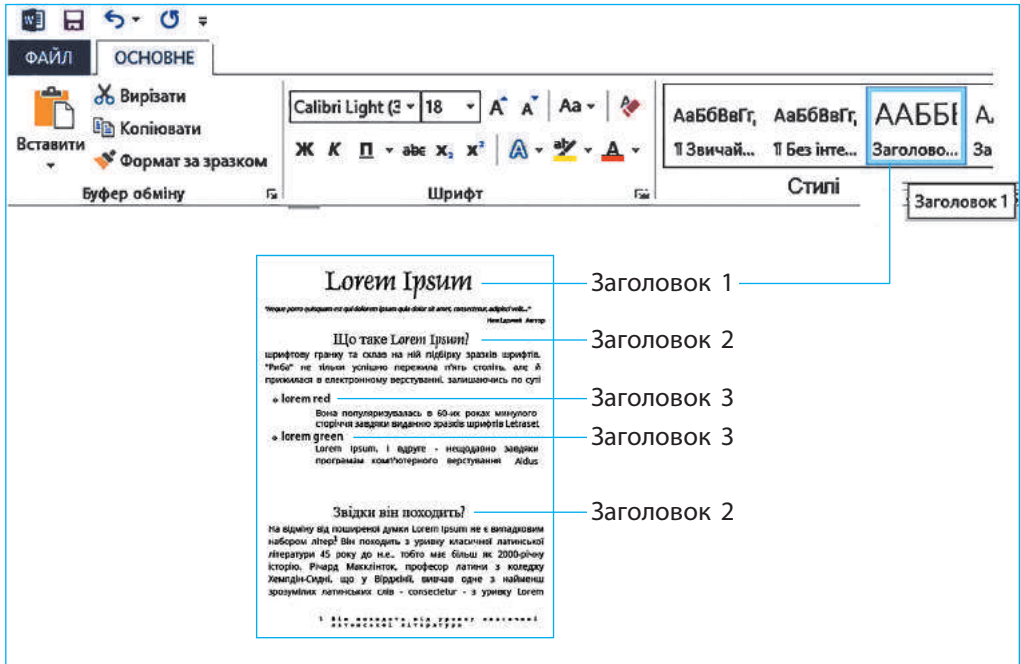


Рис. 10.3

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

Створення покажчика

Для пришвидшення пошуку потрібних термінів у кінці підручника зазвичай розміщують алфавітний покажчик.



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 в групі Покажчик вибрати команду Позначити елемент;
- 3) у вікні Визначення елемента покажчика налаштувати параметри:
 - а) змінити текст у полі Основний — за потреби;

- б) щоб елемент вказував на кілька термінів (наприклад, архівний файл; стиснений файл; архів), у полі додатковий вписати через двокрапку інші терміни;
 - в) натиснути кнопку Позначити або Позначити все (щоб знайти і позначити в документі всі терміни);
- 4) установити курсор у відповідне місце документа і на вкладці Посилання в групі Показчик вибрати команду Показчик.

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

Порядок опрацювання складних документів

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

Стисло розглянемо **основні етапи** опрацювання складних текстових документів.

- 1) **Набір і збереження** — введення 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: змінення властивостей об'єктів, налаштування стилів тощо.
- 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 у програмі Word робоче поле можна розділити на кілька частин на вкладці Вигляд → Розділити (група Вікно). Після цього в кожній частині документ можна прокручувати незалежно. Це зручно використовувати під час копіювання (або переміщення) фрагментів з одного місця в інше.

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 документа.

Вправа 10



- ▶ Створити зміст багатосторінкового структурованого документа.
- 1) Запустіть текстовий процесор Word і відкрийте документ Вправа 9. Або виконайте такі дії:
створіть новий документ; знайдіть в інтернеті першу дію п'єси Івана Карпенка-Карого «Сто тисяч» та скопіюйте в документ.
 - 2) Установіть стилі заголовків твору: для назви дії — Заголовок 1, для яв — Заголовок 2.
 - 3) Проставте номери сторінок (автоматично).
 - 4) Додайте вгорі порожню сторінку вставленням розриву сторінки.
 - 5) Створіть на доданій сторінці автоматизований зміст.
 - 6) Збережіть документ із назвою Вправа 10.



Комп'ютерне тестування



Виконайте тестове завдання 10 із автоматичною перевіркою результату.



Практична робота 4

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 зі встановленим текстовим процесором, під'єднаний до інтернету.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть текстовий процесор і створіть новий документ. Знайдіть в інтернеті три-чотири вірші одного з українських поетів або поетес і скопіюйте в документ.

Примітка. Якщо текст міститься в таблиці, то витягніть його. Для цього виділіть всю таблицю і 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.
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. За потреби відформатуйте малюнок: 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, а після змісту — 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 — полуторний. 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 4.
Завершіть роботу за комп'ютером.

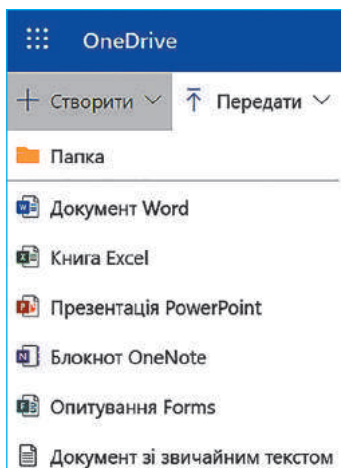
Зробіть висновок: якими є особливості роботи зі складним текстовим документом, як автоматизувати процес створення змісту.

§ 11. Опрацювання документів на хмарному диску

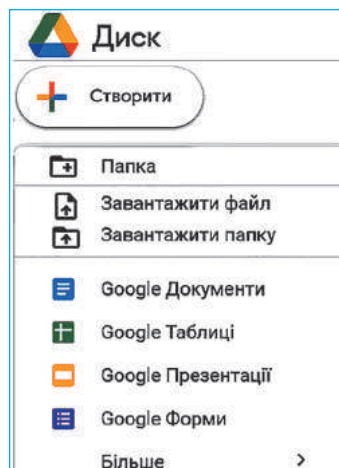
Останніми роками стрімко зростає впровадження хмарних технологій. Нині вони дозволяють працювати з документами в онлайн-режимі одночасно кільком користувачам.

Створення документів на хмарному диску

Створити текстові або інші документи й опрацювати їх за допомогою відповідних редакторів можна в хмарному сховищі OneDrive корпорації Microsoft (рис. 11.1, а) і/або на Google Диску корпорації Google (рис. 11.1, б).



а



б

Рис. 11.1






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

Для отримання можливості користуватися хмарними сервісами Microsoft і Google достатньо створити відповідні облікові записи.

На хмарному диску можна створити текстовий документ, електронну таблицю, презентацію, малюнок, карту, форму опитування тощо. Крім браузера, жодного додаткового програмного забезпечення користувачам встановлювати для цього не потрібно.

Слід наголосити, що користування текстовими онлайн-редакторами схоже на користування текстовим процесором Word.

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

- 1) увійти до свого облікового запису Microsoft (microsoft.com) або Google (google.com);
- 2) кнопкою  відкрити Запускач програм для Microsoft (рис. 11.2, а) або Додатки Google для Google (рис. 11.2, б);
- 3) відкрити хмарний диск, вибравши OneDrive  або Google Диск ;
- 4) натиснути кнопку Створити;
- 5) вибрати тип створюваного документа:



Документ Word (див. рис. 11.1, а) або



Google Документи (див. рис. 11.1, б).

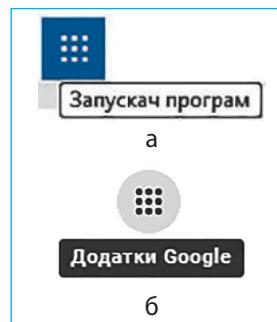



Рис. 11.2

Збереження файлів на хмарному диску

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

- ▶ 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. 11.3, а) і вибрати команду Зберегти як (рис. 11.3, б);
- 2) вибрати для збереження хмарний диск OneDrive  (рис. 11.3, в);
- 3) увійти до свого облікового запису Microsoft (рис. 11.3, г);
- 4) вибрати папку на хмарному диску, в якій слід зберегти файл документа;
- 5) зазначити (або змінити) назву та/або тип файлу;
- 6) натиснути кнопку Зберегти.

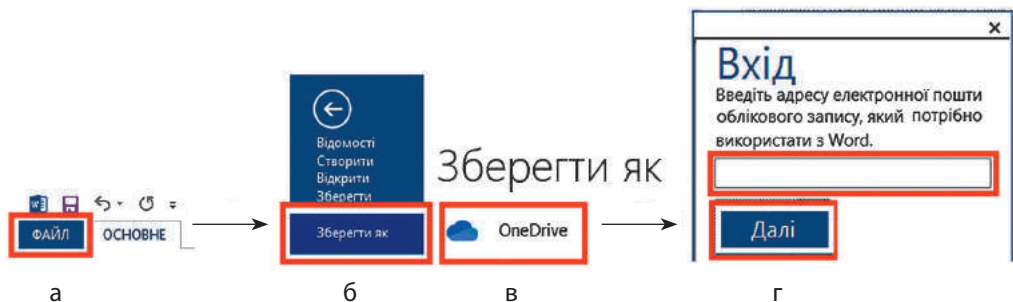


Рис. 11.3

Для надання кільком користувачам спільного доступу до документа, збереженого на OneDrive, слід вибрати на вкладці Файл команду Надати спільний доступ (детальніше про це дізнаєтеся далі).

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

► Збереження файлів на хмарному диску

Для організації спільної роботи над документом можна переслати файл іншим користувачам поштою. Автор документа (керівник проекту) 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 спільного опрацювання документів є робота з файлами, збереженими на хмарному диску за допомогою онлайн-редакторів.

» Корпорація Microsoft безкоштовно надає 5 Гб дискового простору на OneDrive для зберігання користувацьких файлів і папок, а також пакет сервісів Microsoft 365 різного призначення, зокрема й текстовий онлайн-редактор Word.

Для збереження файлу на OneDrive потрібно:

- 1) зайти до свого облікового запису Microsoft;
- 2) відкрити хмарне сховище OneDrive;
- 3) за необхідності відкрити папку на OneDrive для зберігання файлу;
- 4) натиснути кнопку Передати;
- 5) зазначити об'єкт для збереження: Файл або Папка;

- б) знайти у вікні провідника потрібний файл або папку і натиснути кнопку Відкрити — після цього вибраний файл або папку буде збережено на OneDrive.

» Власникам облікового запису Google для збереження поштової кореспонденції та інших файлів безкоштовно надається простір обсягом 15 Гб на Google Диску, а також низка онлайн-сервісів (додатків) Google разом з онлайн-редактором Google-документів.

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 (рис. 11.4).

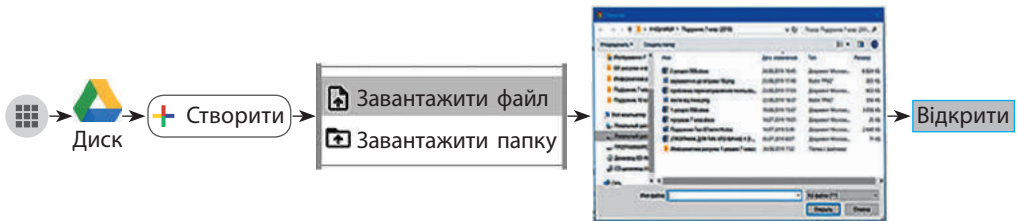


Рис. 11.4

► Синхронізація даних

Хмарні сховища можна синхронізувати з користувацьким комп'ютером, що дає змогу отримувати доступ до них із файлової системи комп'ютера, під'єданого 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.

Для синхронізації хмарного сховища з Windows 7 доведеться завантажити та встановити програму OneDrive і/або додаток Резервне копіювання та синхронізація для синхронізації з Google Диском.

Спільна робота з документом

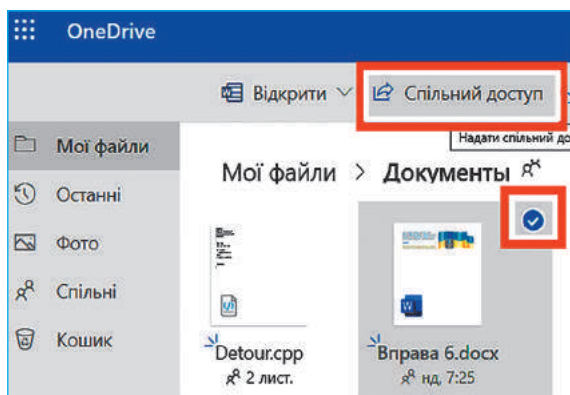
Працювати з документами, збереженими на хмарному диску, можуть тільки ті користувачі, які мають до них доступ. Потрібне лише під'єднання до інтернету та наявність браузера.

Доступ до файлів надає власник хмарного диску, де вони розміщені.

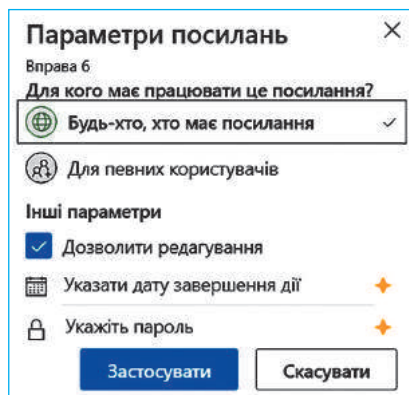
▶ Доступ на OneDrive

Для **надання доступу до файлу на OneDrive** слід позначити потрібний файл «галочкою» і натиснути кнопку Спільний доступ (рис. 11.5, а) 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 з певною адресою; дозволити редагування документа чи лише перегляд; обмежувати термін чи надати безстроковий доступ; встановлювати пароль доступу чи ні) і натиснути кнопку Застосувати (рис. 11.5, б).



а





б

Рис. 11.5

▶ Доступ на Google Диск

На Google Диск існують **два типи доступу до файлу**:

- 1) усім, хто має посилання на файл (увімкнено: , вимкнено: 
- 2) лише тим, кому надано доступ за їх іменами або електронними адресами.

Існують **три рівні доступу до документа**:

- 1) перегляд; 2) коментування; 3) 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 Дisku** необхідно в контекстному меню файлу вибрати команду Надати доступ (рис. 11.6).

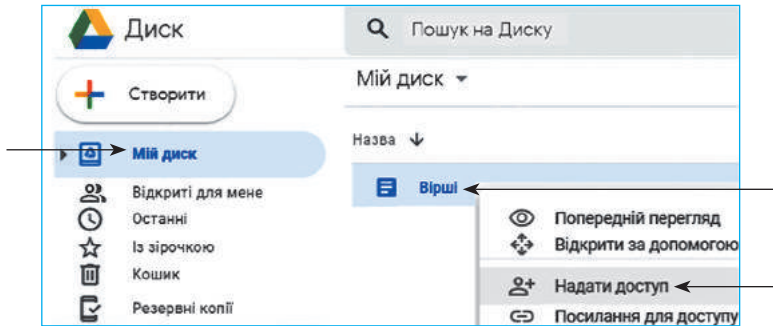


Рис. 11.6

Для надання доступу до файлу в режимі його онлайн-перегляду потрібно скористатись кнопкою Спільний Доступ (рис. 11.7).

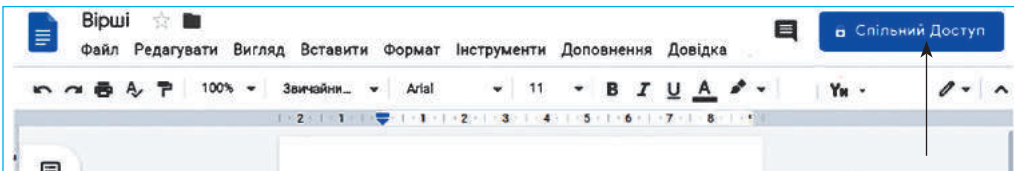


Рис. 11.7

Далі у формі, що відкриється, потрібно встановити **тип** і **рівень доступу** до файлу та клацнути кнопку Готово (рис. 11.8).

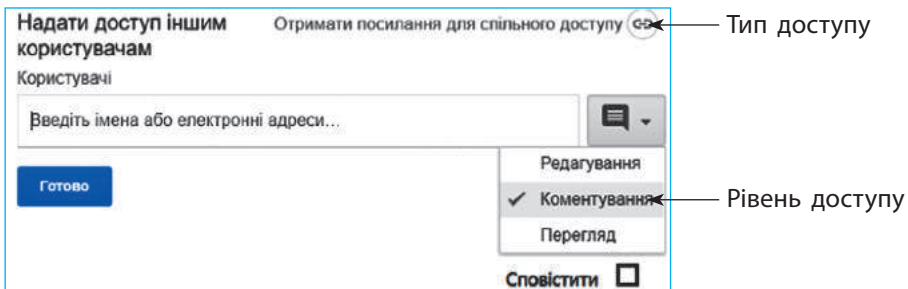


Рис. 11.8

Якщо в адресному рядку форми доступу зазначити електронні адреси, то адресати або отримають автоматичний лист-повідомлення про те, що їм надано доступ до файлу (за умови встановлення прапорця Сповістити), або дізнаються про це, увійшовши до свого Google-диска.

► Відстеження змін у документі

Іноді користувачам і користувачкам важливо відстежувати, хто і які зміни вносив у документ. Розглянемо, 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 внесено за відсутності користувача, потрібно в онлайн-редакторі документа Word на вкладці Рецензування увімкнути відстеження виправлень (рис. 11.9).

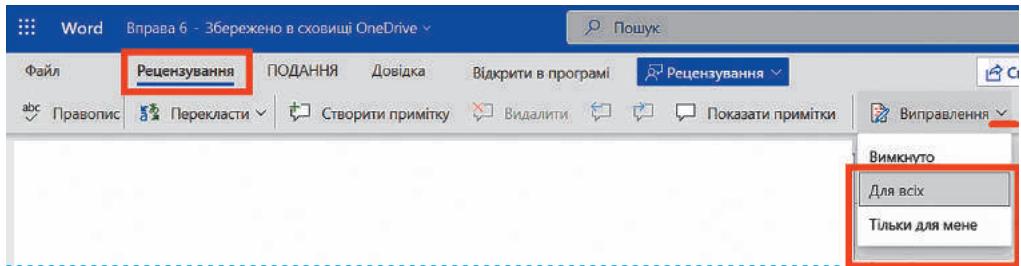




Рис. 11.9

У редакторі Google-документа виправлення й дати їх внесення можна побачити на вкладці Файл, скориставшись командою Історія версій.

Спілкування під час колективного опрацювання документів може відбуватись засобами текстових редакторів: за допомогою приміток — кнопка  у Word та коментарів в Google-документі — кнопка  у правому верхньому куті вікна.

Для **додавання приміток** до виділених фрагментів тексту в документі використовують команду Створити примітку, що міститься на вкладці Рецензування (рис. 11.10).

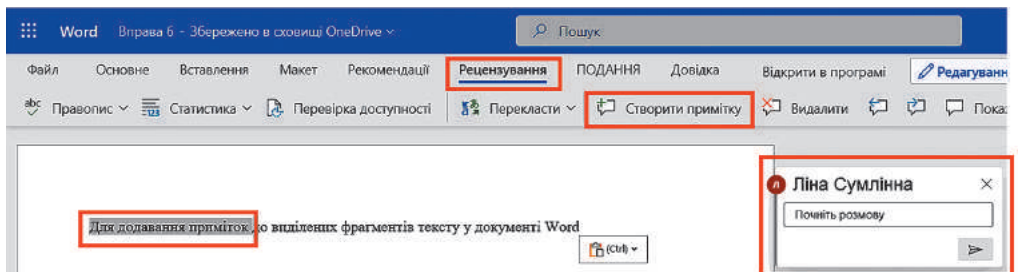



Рис. 11.10

Для додавання коментарів до фрагмента виділеного тексту в Google-документі використовують значок , який з'являється після виділення тексту (рис. 11.11).

Для додавання коментарів до фрагмента виділеного тексту в Google-документі



Рис. 11.11

Питання для самоперевірки



1. Що таке хмарний диск? Які переваги він надає в користуванні документами, збереженими на ньому?
2. Як створити текстовий документ на OneDrive або Google Диску?
3. Як зберегти документ на OneDrive під час роботи у Word?
4. Як зберегти файл на OneDrive або Google Диску?
5. Як надати доступ користувачам для перегляду файлів на OneDrive або Google Диску?
6. Як додати примітку або коментар до текстового фрагмента на хмарному диску і як відстежити зміни документа?

Вправа 11



- Опрацювання документів онлайн 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 документ Вправа 10.docx.
 - 4) Надайте доступ для перегляду збереженого документа вчителю та кільком однокласникам і однокласницям, автоматично сповістивши їх поштою.
 - 5) Попросіть однокласника чи однокласницю надати вам доступ до свого документа, надайте їм адресу своєї електронної пошти.
 - 6) Відкрийте свою поштову скриньку. Перевірте, чи надано доступ, відкрийте та перегляньте документ. Спробуйте відредагувати текст або залишити коментар (примітку). Поясніть, чи вдалося відредагувати документ.



Комп'ютерне тестування

Виконайте тестове завдання 11 із автоматичною перевіркою результату.





Практична робота 5

Створення документа на хмарному диску для спільного опрацювання

Завдання: колективно створити й опрацювати онлайн текстовий документ — реферат на певну тему.

Обладнання: комп'ютер, під'єднаний до інтернету.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

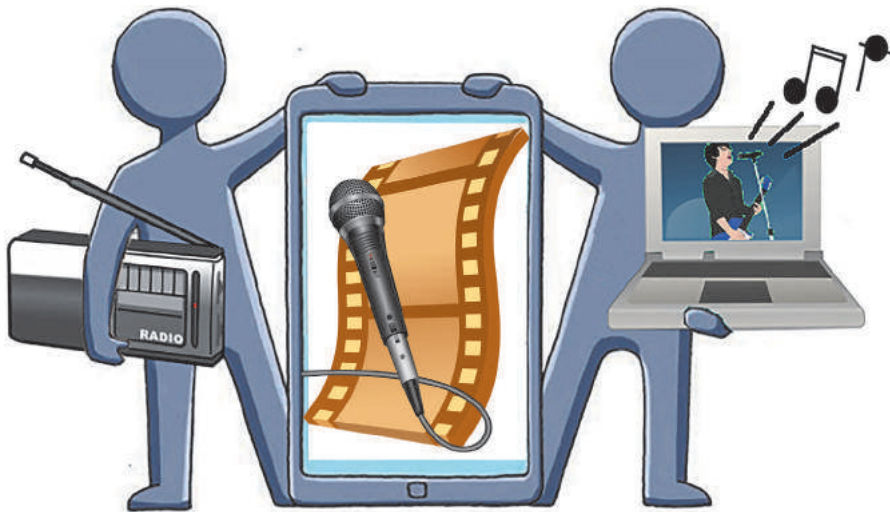
1. Об'єднайтеся в групи для роботи над створенням реферату.
2. Довільно або із запропонованого вчителем переліку виберіть тему.
3. Серед учнів і учениць своєї групи виберіть керівника.
4. Зайдіть до свого облікового запису Google або Microsoft.
5. Керівник створює текстовий документ на Google Диску або на OneDrive і надає доступ до його редагування іншим членам групи.
6. Назвіть (перейменуйте) документ відповідно до вибраної теми.
7. На титульній (*першій*) сторінці документа напишіть тему.
8. Знайдіть в інтернеті вимоги до учнівських (студентських) рефератів і ознайомтеся з ними. Колективно опрацюйте приблизний обсяг і зміст (за розділами) свого реферату та розподіліть завдання.
9. Під темою реферата зі свого робочого місця онлайн впишіть свої прізвище й ім'я, біля яких у дужках зазначте назву свого розділу.
10. Доберіть в інтернеті матеріали для свого розділу реферату та скопіюйте до спільно опрацьовуваного документа. Спостерігайте, щоб матеріали учасників і учасниць групи не збігалися з вашими.
11. Колективно відредагуйте та відформатуйте текстовий документ. До заголовка свого розділу додайте гіперпосилання на джерело, з якого взято інформацію. Використовуйте коментарі (або примітки) для онлайн-спілкування між собою.
12. Надайте вчителю доступ до документа із можливістю перегляду та коментування. Завершіть роботу за комп'ютером.

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

РОЗДІЛ 3

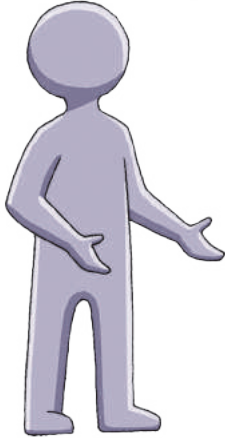
ОПРАЦЮВАННЯ

МУЛЬТИМЕДІЙНИХ ОБ'ЄКТІВ



- § 12. Поняття мультимедіа. Кодування аудіоданих
- § 13. Кодування відеоданих
Практична робота 6. Використання аудіо- й відеоданих у програмах мовою Python
- § 14. Програмне забезпечення для опрацювання звуку
- § 15. Програмне забезпечення для опрацювання відео
- § 16. Побудова відеоряду
- § 17. Опрацювання, зберігання та розміщення відеофільму в інтернеті
Практична робота 7. Створення відеофільму
Практична робота 8. Розміщення відеоматеріалів в інтернеті

ПОВТОРЮЄМО



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

До пристроїв, що входять до складу *мультимедійного обладнання*, належать мультимедійні дошки та проектори, відеокамери, акустичні системи тощо. З кожним роком кількість таких пристроїв збільшується.

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

Файли мультимедіа створюють та опрацьовують за допомогою різних комп'ютерних програм і зберігають у *спеціальних форматах*. Ви вже навчилися користуватися деякими з них, наприклад програмами для створення комп'ютерних презентацій, мультимедійним програвачем, графічним редактором 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 навчитесь створювати та опрацьовувати такі об'єкти мультимедіа, як звук і відео, опануєте процес створення відеофільму та способи його розміщення в інтернеті.

§ 12. Поняття мультимедіа. Кодування аудіоданих

Найдавніші 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 ледь більше ніж 100 років. Проте уявити без них сучасний 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 можливостей дозволило опрацювати звукові та відеодані — так комп'ютер став мультимедійним.



Мультимедіа (лат. *multum* + *medium*) — комбінування в одному інформаційному продукті різних форм подання інформації: текстової, графічної, звукової, відео, з можливістю її інтерактивного використання.

Кодування звуку

Людина отримала змогу зберігати і відтворювати звукові повідомлення, коли навчилася записувати звукові коливання на грамплатівки (рис. 12.1), магнітофонні стрічки тощо. Такий звукозапис називають **аналоговим**.

Ви вже знаєте, як кодують числові та графічні дані. Тепер ознайомимось із тим, як у комп'ютері кодується звук.



Рис. 12.1



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 дізнаєтесь на уроках фізики). Зобразимо ці коливання у вигляді графіка зміни рівня сигналу з часом (рис. 12.2).

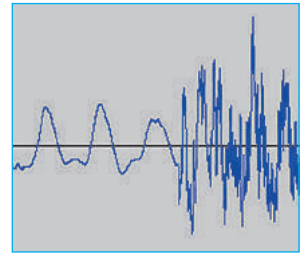


Рис. 12.2

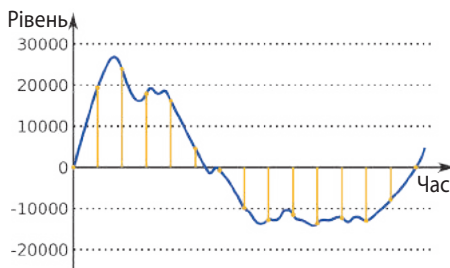
Щоб закодувати звук, потрібно описати числами форму отриманої лінії. Для цього виміряємо через однакові проміжки часу рівень сигналу, тобто відстані від горизонтальної осі до точок кривої (жовті відрізки на рис. 12.3).



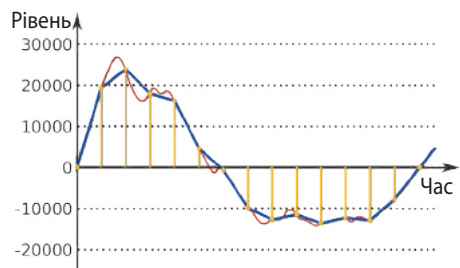
Процес кодування неперервного сигналу послідовністю окремих значень його рівня називають **дискретизацією сигналу**.

Результати вимірювання кодуються двійковим способом. Якщо для зберігання одного значення виділити 2 байти (16 бітів), то можна закодувати $N = 2^{16} = 65\,536$ різних значень рівня, від $-32\,768$ до $32\,767$.

1 Зображений на рис. 12.3, а, фрагмент звукової хвилі можна закодувати такою послідовністю чисел: 0, 19 630, 24 274, 18 180, 16 457, 4672, -735 , -9997 , -12928 , $-11\,681$, $-13\,815$, $-12\,444$, $-13\,128$, -7928 , 0. За цими значеннями можна приблизно відновити початкову форму хвилі (рис. 12.3, б).



а



б

Рис. 12.3

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 на рис. 12.3, б). Це означає, що збережений звук трохи відрізнятиметься від того, який записували.

Щоб точніше зберегти форму звукової хвилі, потрібно вимірювати рівень сигналу якомога частіше.



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

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

Під час розкодування за послідовністю чисел відтворюється форма звукової хвилі і надсилається на посилювач звуку і далі, наприклад, на звукові колонки.

» Іноді під час телефонної розмови можна почути, що якість звуку знижується. Це означає, що через перевантаження ліній, погіршення передавального каналу або з інших причин знижується частота дискретизації. Завдяки цьому вдається зменшити обсяг даних, які передаються, і не переривати розмову.

2 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 в аудіофайлах можуть використовуватися різні алгоритми стиснення.

Розглянемо деякі 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 з втратами якості через спотворення найвищих частот. Початковий код відновити неможливо

Відтворення звуку в програмах

Закодований звук із файлу можна відтворити у програмі мовою Python. Найпростіше це зробити за допомогою функції `playsound`, яка доступна в модулі з такою самою назвою.

3 Розглянемо програму, після запуску якої буде відтворено вміст файлу `Гарна музика.ogg`, якщо він є в одній папці з програмою:

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

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

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

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

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

Якщо запустити програму і ввести число, то для кожної цифри буде відтворено звук із відповідного файлу.

Як бачимо, кодування, збереження і подальше використання звукової інформації значно розширили можливості комп'ютерів.

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 користуєтесь?
5. Як запрограмувати відтворення звукового файлу?
6. Поясніть докладно, як виконується програма, розглянута в прикладі 4. За можливості випробуйте її.

Вправа 12

1. Відтворіть на комп'ютері музичний аудіофайл. Засобами медіаплеєра і/або операційної системи відшукайте і запишіть у зошит докладні дані про цей файл.



2. Складіть програму для озвучення введеного рядка великих англійських букв абеткою Морзе. Звукові файли можна завантажити з Вікісховища за посиланням <https://bit.ly/38sXzRs>.



3. Скориставшись наданими вчителем 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 12 із автоматичною перевіркою результату.



§ 13. Кодування відеоданих

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

Принципи кодування відео

Як ви вже знаєте, рухоме зображення — відео — отримують, демонструючи по чергово 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 зберігають в одному файлі, разом із закодованими даними звукового супроводу та іншими даними. Проте іноді кожен кадр відео зберігають у окремому файлі.

- 1 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 ≈ 120 ГіБ.
 Цікаво, як ці дані разом зі звуковою доріжкою поміщаються в значно меншому файлі обсягом 1,6 ГіБ?

Існують різні методи зменшення обсягу файлу до прийнятної для використання величини. Розглянемо два з них 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.

- 2 Під час *скринкастингу* передається через інтернет або записується до файлу те, що відбувається на екрані комп'ютера. Якщо ведучий при цьому 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 в різних форматах відеофайлів використовують ці та інші методи, а також їх різноманітні комбінації.

Формати відеофайлів

Відеофайл містить набір статичних (нерухомих) зображень, призначених для почергового виведення з певною частотою кадрів.

Для стиснення і відтворення відео використовують спеціальні програми — **кодеки** (від слів *кодування* — *декодування*).

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

Розглянемо деякі 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- й відеокодеки
MPG	Часто вживаний формат завдяки невеликим втратам під час стиснення та сумісності з багатьма програмами і пристроями
MP4	Формат для зберігання відео, аудіо та деяких видів анімації. Підтримує багатоканальний звук. Часто використовується в портативних пристроях, для передавання відеофайлів в інтернеті
ASF	Формат компанії Microsoft, використовується для передавання відео в інтернеті. Підтримує практично всі відеокодеки
MKV (Matroska)	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 веб-медіа. Спрощений аналог MKV з 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. Крім того, технології опрацювання мультимедійних даних розвиваються, це також спричиняє відмову від застарілих форматів файлів і появу 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.

У 7 класі ви вже навчилися створювати у вікні програми мовою Python анімацію з кадрів, збережених у файлах формату PNG. Програму можна вдосконалити таким чином, щоб в одному вікні виводилось декілька анімацій одночасно.

3 Опишемо клас, екземпляр якого за отриманим списком імен файлів і координатами виводитиме у заданому місці вікна анімацію:

```
class Videosprite:
    def __init__(self, frames, x, y):
        # Формуємо список зображень для об'єкта...
        self.images = []
        for name in frames:      # ...з файлів, імена яких є у списку frames
            self.images.append(PhotolImage(file = name))

        self.id = canvas.create_image(x, y, anchor = NW, image = self.images[0])
        self.cur_frame = 0      # Номер поточного кадру
        self.animate()         # Початок анімації

    def animate(self):
        root.update()          # Показуємо поточний кадр
        # Готуємо наступний кадр
        self.cur_frame += 1     # Коригуємо номер
        if self.cur_frame >= len(self.images):
            self.cur_frame = 0

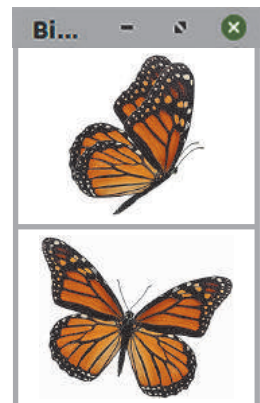
        # Замінюємо зображення наступним
        canvas.itemconfig(self.id, image = self.images[self.cur_frame])
        root.after(100, self.animate)    # Продовжуємо анімацію
```

Тепер у програмі можна створювати об'єкти класу Videosprite, кожний із яких виводитиме у потрібному місці анімацію із власного списку імен файлів (див. рисунок):

```
import time
from tkinter import*

root = Tk()
root.title('Відеоспрайти')

canvas = Canvas(root, width = 140, height = 280)
canvas.pack()
```



```
# Створюємо два об'єкти з різною анімацією
B = Videosprite(5, 5, ['1.gif', '2.gif', '3.gif'])
C = Videosprite(5, 140, ['4.gif', '5.gif', '6.gif', '7.gif', '8.gif'])
root.mainloop()
```

Як бачимо, розуміння принципів отримання рухомого зображення із сукупності нерухомих і цифрового кодування відеоданих дозволяє не тільки зберігати відеоматеріали для перегляду, але й складати 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?
5. Поясніть, як виконується метод `__init__` класу `Videosprite` (див. приклад 3).
6. Поясніть, як виконується метод `animate` класу `Videosprite` (див. приклад 3).

Вправа 13



1. Знайдіть в інтернеті відеофільм О. Довженка «Земля» для завантаження. Запишіть у зошиті формат і розмір відеофайлу; тривалість фільму.
2. Розрахуйте обсяг відеоданих без упакування (див. приклад 1).
3. Поясніть, чому саме цей формат відеофайлу використовується для передавання відео в інтернеті.
4. Випробуйте програму, описану в прикладі 3. Збережіть програму у файлі з назвою Вправа 13. Доповніть програму так, щоб при клацанні на анімаціях вони зникали (якщо складно, то пригадайте, як було реалізовано обробку клацання на об'єкті в 7 класі в практичній роботі 3).



Комп'ютерне тестування



Виконайте тестове завдання 13 із автоматичною перевіркою результату.

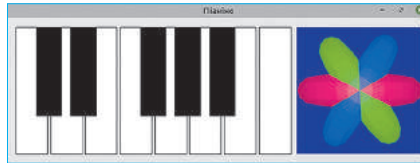




Практична робота 6

Використання аудіо- й відеоданих у програмах мовою Python

Завдання: створити програму Піаніно (див. рисунок). При натисканні на «клавіші» мають відтворюватись звуки, а в правій частині вікна — виводиться анімаційний фрагмент.



Обладнання: комп'ютер із системою програмування IDLE; папки notes (із файлами формату OGG зі звуками нот) і frames (із файлами формату PNG з кадрами анімації).

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть програму IDLE і відкрийте вікно нового файлу.
2. За зразком (див. приклад 3 в § 13) запрограмуйте створення вікна програми (назва об'єкта root) із полотном canvas 670×230 пікселів.
3. На початку програми опишіть клас PianoKey.

Екземпляром класу буде прямокутник-«клавіша», після клацання на якому звучатиме нота. У конструктор передаватимуться такі параметри: *x*, *y* — координати лівого верхнього кута; *color* — колір ('black' або 'white'); *soundfile* — назва звукового файлу:

```
class PianoKey:
```

```
    def __init__(self, x, y, color, soundfile):
        self.soundfile = soundfile
        w, h = 50, 200 # Розміри прямокутника-«клавіші»
        if color == 'black': w, h = 40, 140 # Якщо чорна — зменшуємо
        self.id = canvas.create_rectangle(x, y, x+w, y+h, fill = color)
        # Пов'язуємо подію з обробником
        canvas.tag_bind(self.id, '<1>', self.on_click)
```

4. Додайте до класу обробник клацання на прямокутнику лівою кнопкою миші:


```
def on_click(self, event):
    playsound(self.soundfile)
```
5. Для того щоб перевірити правильність опису класу PianoKey, додайте до основної програми команди:


```
PianoKey(15, 15, 'white', 'notes/do.ogg')
root.mainloop()
```

 Запустіть програму. Збережіть її у файлі Практична робота 6. При клацанні на прямокутнику має звучати нота до.
6. Прямокутники для 8 нот можна отримати, записавши окремі команди для кожного з них (назви файлів для білих клавіш: do, re, mi, fa, sol, la, si, do2). Зробіть це і перевірте роботу програми.
7. Запрограмуйте малювання чорних клавіш (назви файлів для них: do-d, mi-b, fa-d, sol-d, si-b). Перевірте роботу програми.
8. З файлу Вправа 13.py скопіюйте код класу Videosprite і додайте на початок програми. Вилучіть із нього рядки, які спричиняють зникнення об'єкта при клацанні.
9. Додайте до програми команду для виведення кадрів анімації з папки frames (замість трьох крапок допишіть імена решти графічних файлів):


```
B = Videosprite(460, 15, ['frames/f01.png', 'frames/f02.png',
                           'frames/f03.png', ...])
```
10. Випробуйте роботу програми, спробуйте зіграти мелодію.
11. Для малювання білих клавіш записано 8 команд. Зменште обсяг програми, замінивши їх таким фрагментом:


```
notes = ['do', 're', 'mi', 'fa', 'sol', 'la', 'si', 'do2']
for key in range(8):
    PianoKey(15+key*55, 15, 'white', 'notes/'+notes[key]+' .ogg')
```

 Перевірте роботу програми. Поясніть, як працюють ці команди.
12. Змініть програму так, щоб при клацанні ділянки вікна з анімацією анімація припинялась, а при наступному клацанні — продовжувалась. Перевірте роботу програми. Завершіть роботу за комп'ютером.

Зробіть висновок про те, які можливості відкриває винайдення способу кодування і зберігання звуку та відео.

§ 14. Програмне забезпечення для опрацювання звуку

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

Захоплення звуку

Щоб опрацювати на комп'ютері звукові дані, отримані із зовнішніх пристроїв, наприклад мікрофона, електрогітари, синтезатора тощо, їх потрібно перевести в цифровий формат (оцифрувати).



Процес перетворення звукового сигналу із зовнішніх джерел у цифровий формат і подальшого запису в аудіофайл називають **захопленням звуку**.

Для захоплення звуку слід:

- 1) приєднати до комп'ютера аудіопристрій, приміром мікрофон або електрогітару;
- 2) запустити програму для записування звуку;
- 3) виконати мелодію, прочитати перед мікрофоном текст тощо;
- 4) зберегти записані аудіодані у файл.

Програми для запису звуку

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

1 На рис. 14.1 наведено вигляд вікна програми Записувач звуку (ОС Linux Mint) під час записування звуку з мікрофона (тобто після натискання кнопки Record — Записати).

Щоб зупинити запис, слід натиснути кнопку Done — звукові дані після цього будуть автоматично збережені у файл. У нижній частині вікна виводиться список збережених записів.

Програма дозволяє здійснити деякі налаштування (рис. 14.2): вибрати формат звукового файлу, режим запису, гучність прослуховування записаних звуків і чутливість мікрофона.

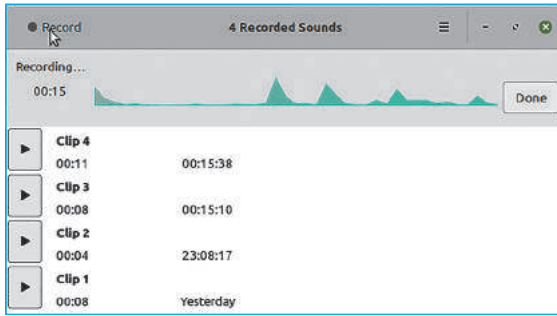


Рис. 14.1

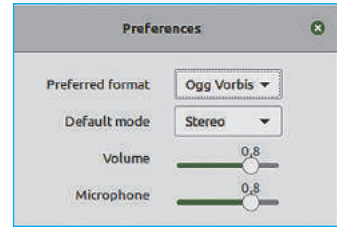


Рис. 14.2

Аудіоредактори

Якщо звук потрібно не лише записати у файл, а й опрацювати його (наприклад, вилучити зайві шуми, змінити гучність окремих фрагментів тощо), то варто скористатися програмою з ширшими можливостями — аудіоредактором.



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

За допомогою встановленого на комп'ютер аудіоредактора користувач має можливість записувати аудіо з різних джерел, накладати різноманітні ефекти, обрізати аудіофайли, монтувати запис із декількох файлів, створювати різні звуки тощо.

Серед популярних аудіоредакторів можна назвати Audacity, Sony Sound Forge, Adobe Audition та ін. (рис. 14.3). Для відтворення аудіоданих (прослуховування звуку) використовують аудіоплеєри, наприклад Winamp, AIMP, VLC (рис. 14.4).



Audacity



Sony Sound Forge



Adobe Audition



Winamp



AIMP



VLC

Рис. 14.3

Рис. 14.4

Аудіоредактор Audacity

Аудіоредактор Audacity (рис. 14.5) належить до вільного програмного забезпечення (ліцензія GNU GPL) і може бути завантажений із сайту програми <https://www.audacityteam.org/>.

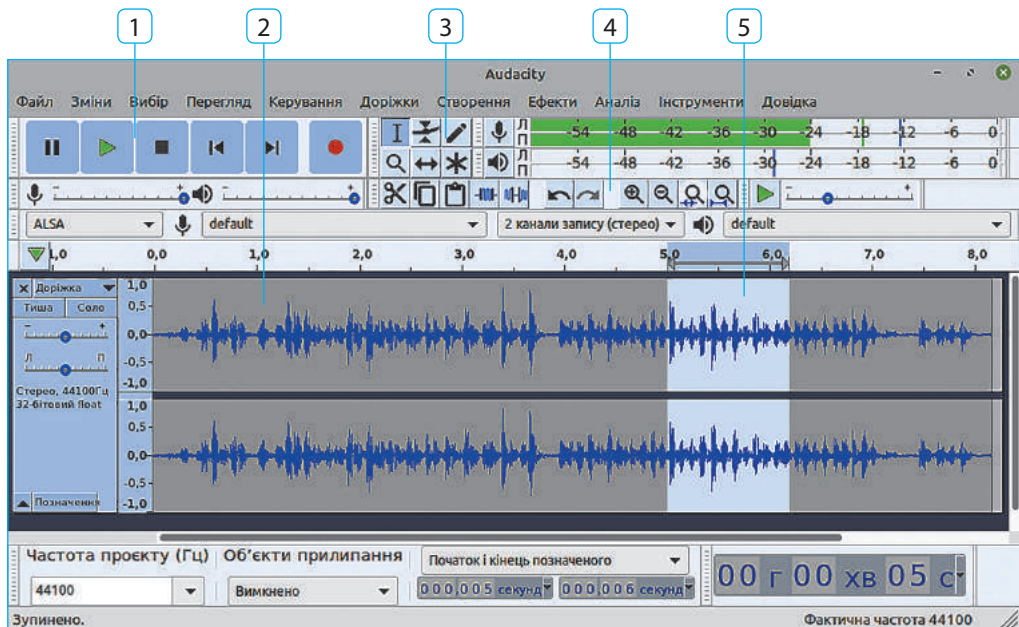


Рис. 14.5

Аудіоредактор Audacity надає такі можливості:

- 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 програми відкривається головне вікно (рис. 14.6). Воно містить елементи керування, необхідні для виконання основних операцій зі звуковими даними. Більш детальні налаштування та широке коло команд доступні в меню програми.








- 1 — панель **Керування**
 2 — звукова доріжка
 3 — панель **Інструменти**

- 4 — панель **Зміни**
 5 — вибраний аудіофрагмент
 (виділено світлим тлом)

Рис. 14.6

► 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

2 На рис. 14.6 подано алгоритм запису звуку за допомогою мікрофона, приєднаного до комп'ютера або вбудованого в нього.



Рис. 14.6

► Робота з фрагментами

Фрагмент, над яким потрібно виконати певні операції, виділяють перетягуванням безпосередньо на звуковій доріжці.

Якщо на початку або в кінці фрагмента рівень не дорівнює нулю, то в цьому місці після виконання операції може з'явитись різкий перепад рівня, який під час прослуховування сприйматиметься як шум.

Щоб цього уникнути, після виділення фрагмента слід натиснути клавішу Z — межі виділення змістяться так, щоб фрагмент починався та закінчувався в точках перетину хвилі з осью ліній, тобто з нульовим рівнем сигналу.

- 3 На рис. 14.7 показано розташування початку виділеного фрагмента і форму хвилі до та після зменшення рівня. У випадку *a* клавішу Z не натискали, тому з'явився різкий перепад рівня.

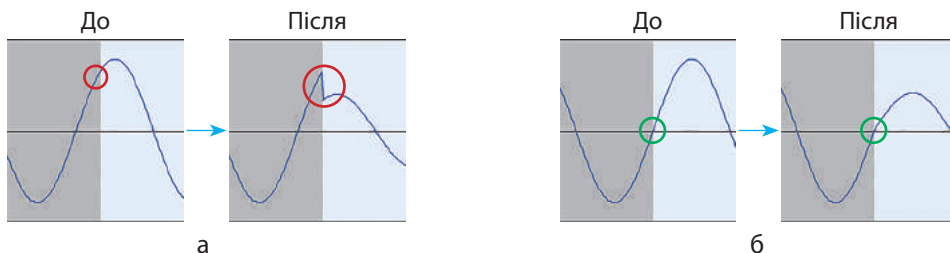






Рис. 14.7


Щоб краще роздивитись хвилю на виділеному фрагменті, потрібно на панелі Зміни клацнути кнопку  — масштаб відображення зміниться так, що фрагмент займе всю ширину вікна. Якщо ж клацнути 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 і 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 цього фрагмента, підсовується вліво.

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

- 1) клацнути на потрібному місці доріжки;
- 2) натиснути клавішу Z, щоб поточна позиція змістилась до найближчого перетину хвилі з віссю;
- 3) вибрати команду меню Вставити або натиснути кнопку .

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

► Збереження звукових даних

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

4 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. 14.8).

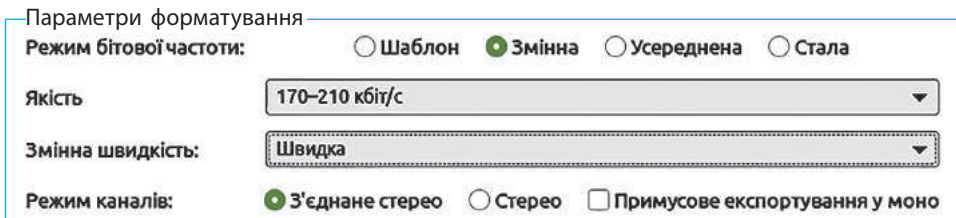


Рис. 14.8

Якість (бітова частота) 192 Кб/с вважається високою. За використання змінної бітової частоти різні ділянки звукової доріжки кодуються з різною частотою дискретизації. Тому звучання файлу в цілому сприймається якіснішим, ніж файлу такого ж розміру, але зі сталою бітовою частотою;

- 3) вибрати папку та назву файлу і натиснути Зберегти.

Залежно від формату, в який виконується експортування, можуть додатково з'явитися діалогові вікна для уточнення параметрів.

Отже, аудіоредактор дозволяє захопити звук, опрацювати аудіодані і експортувати результат у файл бажаного звукового формату, придатний для подальшого використання в інших програмах.

Питання для самоперевірки



1. В 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 14

- ▶ Створити аудіофайл за допомогою аудіоредактора Audacity.
- 1) Створіть у власній папці папку Медіа.
 - 2) Якщо потрібно, підключіть мікрофон та навушники до комп'ютера. 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).
 - 6) Експортуйте запис у файл з іменем Вправа 14.mp3 у папці Медіа. Завершіть роботу програми, вимкніть мікрофон. Прослухайте запис. Визначте його частоту дискретизації.

**Комп'ютерне тестування**

Виконайте тестове завдання 14 із автоматичною перевіркою результату.



§ 15. 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: захоплення, опрацювання, зберігання, перетворення формату тощо. Для цього використовують інше програмне забезпечення та відповідні пристрої.

Захоплення відео

Процес перетворення відеосигналу із зовнішнього пристрою в цифровий формат із подальшим записом у відеофайл називають **захопленням відео**.

За джерелом і типом відеосигналу розрізняють захоплення відео з аналогового пристрою, цифрових відеокamer, екрана монітора (рис. 15.1).

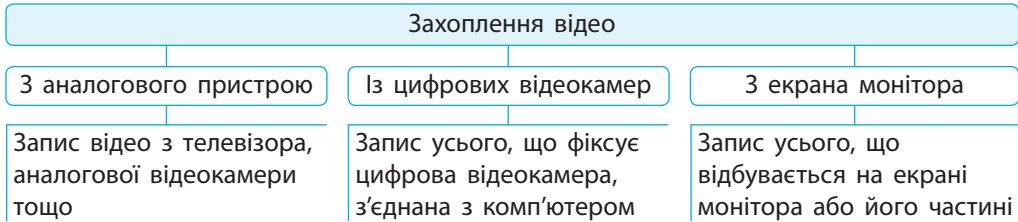


Рис. 15.1

Для оцифрування аналогового відеосигналу використовують спеціальні пристрої — **плати відеозахоплення**. Серед програм для захоплення відео можна назвати OBS Studio, Iccream Screen Recorder, CamStudio тощо. Вони дозволяють захоплювати не лише відео з екрана та інших джерел, але й звук з мікрофона, колонок, зберігати відеофайли у форматах, підтримуваних медіаплеерами. Редактор презентацій PowerPoint також має засоби для захоплення відео та звуку з екрана.

► Програма для захоплення відео OBS Studio

OBS Studio — вільна багатоплатформна програма з відкритим кодом для запису відео або трансляції з широкими можливостями (рис. 15.2).

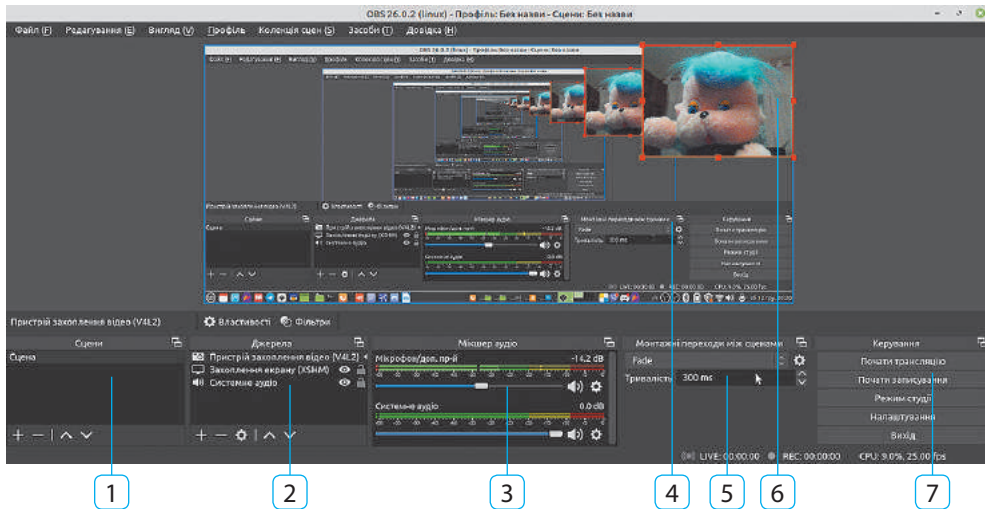


Рис. 15.2

Розглянемо елементи вікна, позначені на рис. 15.2:

- | | |
|---|--|
| 1 — перелік сцен | 4 — зразок захоплення з екрана |
| 2 — перелік джерел поточної сцени | 5 — налаштування переходів між сценами |
| 3 — регулятори гучності аудіовиходу і мікрофона | 6 — зразок захоплення з камери |
| | 7 — загальне керування програмою |

Програма OBS Studio дозволяє захоплювати одночасно відео з екрана та вебкамери; захоплювати звук з мікрофона та програм; довільно компоувати сцену, тобто розміщувати на екрані джерела відео; створювати декілька сцен і перемикатись 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.

Щоб отримати відео з таким макетом екрана, як на рис. 15.2, потрібно додати джерела Пристрій захоплення відео і Захоплення екрану. Для запису звуку програм і мікрофона слід додати джерело Захоплення аудіовиводу (рис. 15.3). Якщо ж потрібен лише звук мікрофона, то замість останнього джерела слід вибрати джерело Захоплення аудіовиходу.

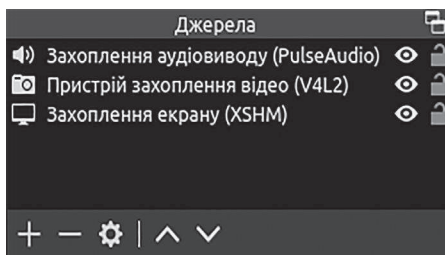






Рис. 15.3

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. Так, для перегляду відео на великому екрані телевізора бажано мати файл із великими розмірами кадру, а для екрана смартфона вистачило б значно менших розмірів кадру, а отже, й обсягу файлу.

Перетворити відеофайли з одного формату в інший допомагають спеціальні програми — конвертери.



Конвертер — програма, яка дозволяє перекодувати медіафайли з одного формату в інший.

► Програма для конвертування відеофайлів HandBrake

HandBrake — вільна багатоплатформна програма, що є потужним інструментом для перекодування відеофайлів з одного формату в інший. Розглянемо роботу з відеофайлом (рис. 15.4; використано кадр із анімаційного фільму Sintel, випущеного Blender Institute).

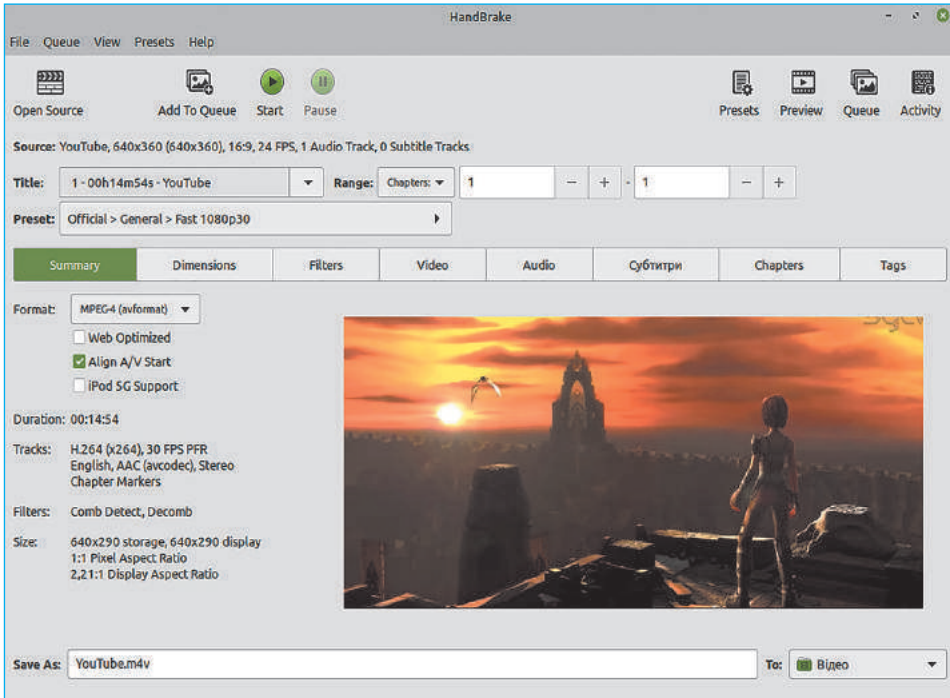


Рис. 15.4

Для **конвертування відеофайлу** за допомогою HandBrake потрібно:

- 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) ввести ім'я файлу, а в списку To: (У:) вибрати папку, куди буде збережено перекодоване відео;
- 4) натиснути кнопку Start (Почати).

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

З розвитком хмарних інтернет-технологій з'явилася можливість конвертування аудіо- та відеофайлів за допомогою низки онлайн-сервісів, які легко відшукати за допомогою пошукових систем.

Розглянуті програми дозволяють підготувати звукові та відеофрагменти, з яких можна змонтувати відеофільм і опублікувати в інтернеті.

Питання для самоперевірки



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

Вправа 15



- ▶ Створити відеофайл, захопивши відео з екрана комп'ютера, за допомогою програми захоплення відео OBS Studio.
- 1) Виберіть тему малюнка (місто, новорічна ялинка тощо). 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 і виконайте малюнок.
 - 5) Зупиніть запис і збережіть файл із назвою Вправа 15 у папці Медіа. Завершіть роботу програми.
 - 6) За допомогою медіаплеєра перегляньте збережений відеофільм і обговоріть його з однокласниками й однокласницями.



Комп'ютерне тестування

Виконайте тестове завдання 15 із автоматичною перевіркою результату.



§ 16. Побудова відеоряду

Відеофрагменти, отримані під час захоплення відео, зняті на цифрову камеру або записані з екрана, можна використати для створення відеоролика або навіть відеофільму. Для цього знадобиться спеціальна програма — відеоредактор.



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

Серед популярних відеоредакторів можна назвати такі програми, як Lightworks, Pinnacle Studio, OpenShot, Sony Vegas Pro тощо (рис. 16.1). У Windows 7–10 можна встановити відеоредактор Кіностудія, завантаживши його з інтернету.



Lightworks

Pinnacle
Studio

OpenShot

Sony Vegas
ProКіностудія
Windows

Рис. 16.1

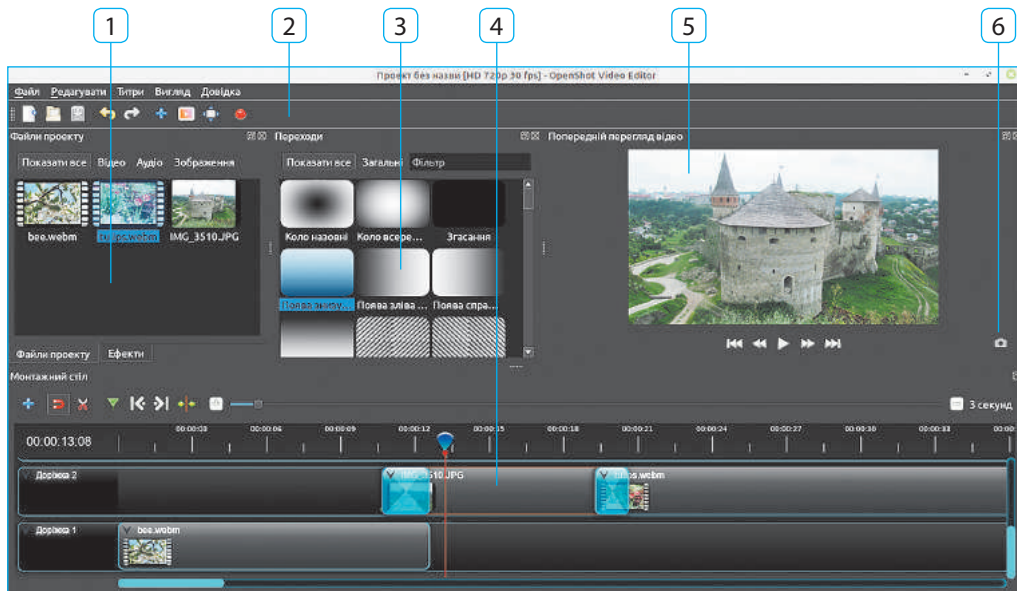
Відеоредактор OpenShot

Відеоредактор OpenShot (див. рис. 16.1) належить до вільного програмного забезпечення (ліцензія GNU GPL). Останню версію програми можна отримати на офіційному сайті (<https://www.openshot.org/uk/>).

Розглянемо основні елементи інтерфейсу програми OpenShot версії 2.5.1 (рис. 16.2).

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. На ньому формується послідовність кліпів (фрагментів відеоряду та звукового супроводу, титрів), змінюються їх тривалості та інші властивості, налаштовуються відеоєфекти та відеопереходи тощо.



- 1 — файли проекту
- 2 — панель інструментів
- 3 — переходи

- 4 — монтажний стіл
- 5 — екран перегляду
- 6 — кнопка збереження поточного кадру

Рис. 16.2

Процес створення відеофільму

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

- імпортування аудіо- та відеофрагментів у проект;
- редагування аудіо- та відеофрагментів;
- побудова відеоряду, налаштування переходів та ефектів;
- підготовка та додавання титрів;
- налаштування звукового супроводу;
- експорт відеофільму у відеофайл.

Більшість кроків необов'язково виконувати саме в такій послідовності, але в процесі роботи слід приділити увагу кожному з них.

▶ Імпортування аудіо- та відеофрагментів у проект

На цьому кроці формується добірка графічних 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 перетягуванням з вікна папки в ділянку вікна Файли проєкту.

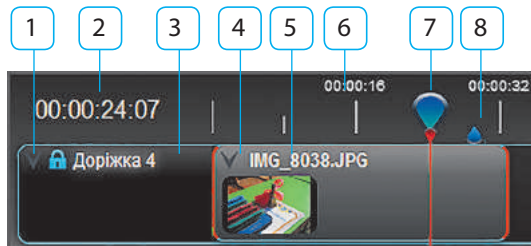
► Редагування аудіо- та відеофрагментів

Наступним кроком роботи над відеофільмом 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. 16.3 показано кліп на доріжці монтажного столу та елементи керування, пов'язані з доріжкою та кліпом.



- | | |
|--------------------------------|---------------------------------------|
| 1 — виклик меню доріжки | 5 — кліп |
| 2 — індикатор поточної позиції | 6 — часова шкала |
| 3 — назва доріжки | 7 — «відтворювальна головка» (курсор) |
| 4 — виклик меню кліпу | 8 — маркер |

Рис. 16.3

Відеоредактор OpenShot дозволяє розділити довгий кліп на два коротші, обрізати початок або кінець кліпу.

Для **розділення кліпу** на дві частини потрібно натиснути кнопку інструмента Лезо  і клацнути на тому місці кліпу, де має бути розріз. Щоб вимкнути інструмент Лезо, слід натиснути кнопку ще раз.

Кліпи деяких відеоформатів (наприклад, MKV) розрізаються не в будь-якому місці, а лише на найближчому ключовому кадрі (див. § 13). Якщо кліп потрібно розрізати точніше, слід перед імпортом перекодувати відеофрагмент у формат, який це дозволяє.

Щоб **обрізати початок або кінець кліпу**, потрібно перетягнути на доріжці відповідний край його рамки (червоні лінії на рис. 16.3) в потрібне місце, визначаючи його візуально на екрані перегляду або за часовими параметрами на шкалі часу вгорі монтажного столу.

Обрізаний кліп можна відновити перетягуванням країв рамки в протилежний бік.

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

Кнопка збереження поточного кадру (див. рис. 16.2, позначка 6) дозволяє зберегти у файл формату PNG той кадр відео, який видно на екрані перегляду. Потім цей файл можна додати до проекту і використати під час монтажу для оформлення відеофільму.

▶ Побудова відеоряду

Наступним кроком є побудова аудіо- та відеоряду.

Ознайомимося з призначенням кнопок монтажного столу:



— 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. 16.4).

Доріжку можна заблокувати від випадкових переміщень і змін тривалості кліпів, проте можливість розрізати кліпи та змінювати їхні властивості при цьому залишається.

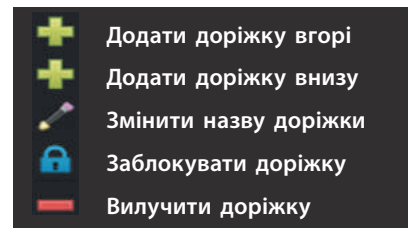


Рис. 16.4

► Підготовка та додавання титрів



Титри являють собою написи, що з'являються на початку або наприкінці відеофільму, на певному відеофрагменті або графічному зображенні тощо.

Відеоредактор OpenShot дозволяє додавати до відео титри різних типів: звичайні та анімовані.

Звичайні титри додають до проєкту командою меню Титри → Титри. При цьому відкривається діалогове вікно Титри (рис. 16.5), в якому потрібно виконати такі дії:

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

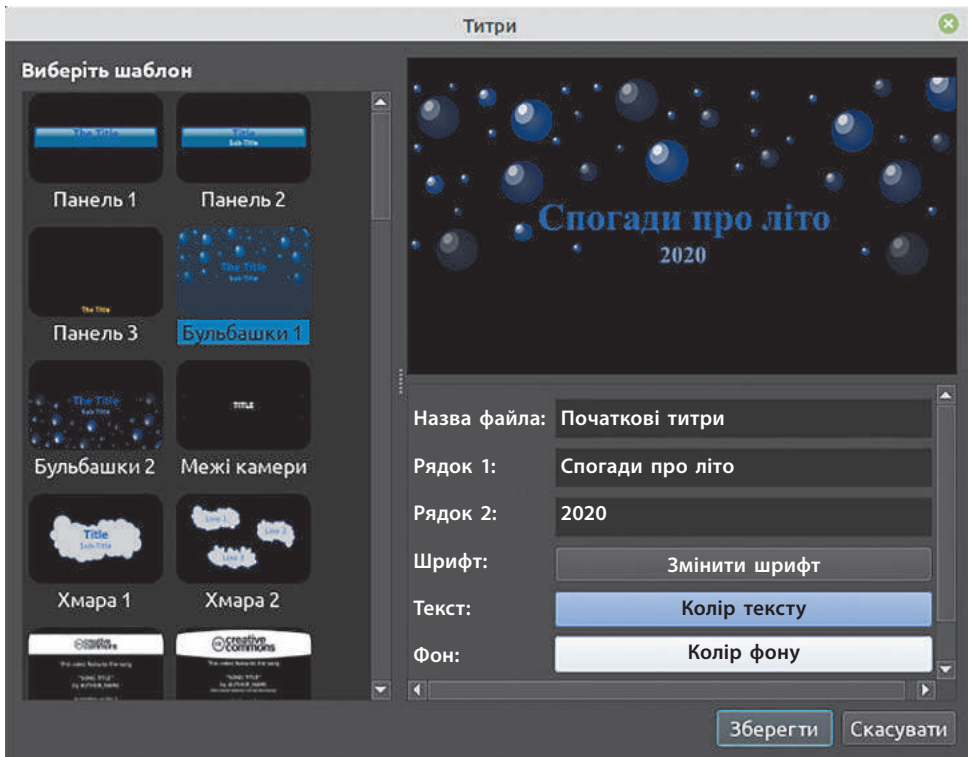


Рис. 16.5

Після цього у вікні Файли проекту 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 більш детального 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 проекту додається відеофрагмент з анімованими титрами.

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

Зберігання проекту відеофільму

Зазвичай робота над відеофільмом займає багато часу, тому незавершений проект потрібно зберегти для подальшої роботи з ним.

Під час зберігання проекту створюється файл із розширенням `.osp`, який містить інформацію про всі файли та проект у цілому, але не містить відео- та аудіоданих. Тому коли користувач відкриє файл проекту на іншому комп'ютері, то обов'язково виникнуть проблеми через відсутність потрібних аудіо- і відеофайлів, зображень тощо. Щоб уникнути такої ситуації, необхідно зберегти всі потрібні файли і файл проекту в окремій папці і вже з неї імпортувати їх в OpenShot.

Ви ознайомились із деякими основними кроками створення відеофільму. На цьому процес не завершується. Далі ви дізнаєтесь 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 проект?
3. Опишіть порядок роботи з кліпами на монтажному столі.
4. Як збільшити час відтворення зображення?
5. Опишіть послідовність додавання титрів.
6. Що містить файл проекту з розширенням `*.osp`?

Вправа 16

- ▶▶ Створити відеофільм зі звуковим супроводом.
- 1) Знайдіть в інтернеті й завантажте файл для звукового супроводу відеофільму в папку, зазначену вчителем. Збережіть у тій самій папці відеофайли, власні або підготовлені вчителем.
 - 2) Запустіть відеоредактор, 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 проект (Файл → Зберегти проект) із назвою Вправа 16 у відповідній папці.

**Комп'ютерне тестування**

Виконайте тестове завдання 16 із автоматичною перевіркою результату.



§ 17. Опрацювання, зберігання та розміщення відеофільму в інтернеті

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

▶ 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 проекті 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 накладались, то між ними буде автоматично додано перехід із поступовим збільшенням прозорості першого з наявних кліпів (рис. 17.1).



Рис. 17.1

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 із двох кліпів. При цьому кліпи необов'язково розміщувати на одній доріжці.

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. 17.2).



Рис. 17.2

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

У наведеному випадку необхідно змінити напрямок переходу, вибравши в його контекстному меню команду **Обернути перехід**.

► Налаштування звукового супроводу

Звукові файли для супроводу відеофільму потрібно імпортувати до проекту, а потім додавати на доріжки так само, як відеофрагменти.

Щоб змінити **тривалість звучання аудіокліпу**, слід перетягнути його правий або лівий край на часовій шкалі.

Команди керування звуком зібрано в контекстному меню кліпу. Розглянемо ці команди 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**, слід у його контекстному меню вибрати команду **Гучність** → **Увесь кліп** → **Рівень 0%**.

► Експорт відеофільму у відеофайл

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

Після натискання кнопки Екпортування відео починається опрацювання проєкту (тривалість залежить від його складності та тривалості відео), а після завершення в цільовій папці з'являється відеофайл.

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 тим, кому ви вважатимете за потрібне.

Існує й інший спосіб поширення відео — опублікування на спеціалізованих інтернет-сервісах або в соціальних мережах.

Нині одним із найпопулярніших інтернет-сервісів для поширення відео є YouTube. На ньому користувачі можуть коментувати й оцінювати відео, ділитися ними одне з одним. Існують також інші подібні сервіси: Instagram, Video Direct, Vimeo тощо.



YouTube



Instagram



Video Direct



Vimeo

Рис. 17.3

Для розміщення відеофільму в інтернеті на сервісі YouTube потрібно:

- 1) увійти в обліковий запис YouTube (Google);
- 2) клацнути кнопку Створити ;
- 3) вибрати операцію Завантажити відео (рис. 17.4);
- 4) вибрати відео для завантаження;
- 5) ввести інформацію про відео;

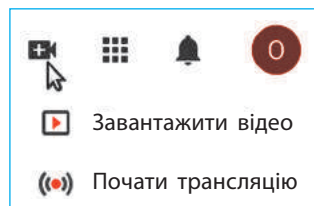


Рис. 17.4

- 6) вибрати рівень доступу (приватне; не для всіх; для всіх);
- 7) клацнути Зберегти або Опублікувати (залежно від рівня доступу).

Після розміщення відеофільму в інтернеті вам надійде електронний лист з посиланням на нього. Поширюючи це посилання (наприклад, публікуючи в соціальних мережах, розсилаючи електронною поштою), можна зробити відеофільм доступним усім зацікавленим.

Питання для самоперевірки



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

Вправа 17



- ▶ Налаштувати часові параметри відеофільму, додати відеопереходи між кліпами.
- 1) Запустіть OpenShot і відкрийте проєкт із назвою Вправа 16. 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) Екпортуйте відеофільм у файл із назвою Вправа 17-1, вибравши профіль Інтернет і не змінюючи параметрів експорту.
 - 5) Екпортуйте відеофільм, вибравши профіль із невеликим розміром кадру (наприклад, 720×576) у файл Вправа 17-2. Порівняйте розміри файлів.
 - 6) Завантажте файл Вправа 17-2 на хмарний диск і надішліть посилання вчителю.



Комп'ютерне тестування

Виконайте тестове завдання 17 із автоматичною перевіркою результату.





Практична робота 7

Створення відеофільму

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

Обладнання: комп'ютер зі встановленим відеоредактором; навушники або звукові колонки; проєкт Вправа 16.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть відеоредактор OpenShot. Відкрийте проєкт Вправа 16.
2. Перегляньте відеофрагменти, збережіть із кожного з них по одному кадру, який характеризує зміст фрагмента.
3. Додайте збережені зображення до проєкту.
4. На монтажному столі розмістіть додані зображення перед відповідними відеокліпами.
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 на початку відеофільму.
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 на половині кадру або слова тощо та виправте виявлені недоліки.
12. Збережіть проєкт із назвою Практична робота 7 у відповідній папці. Завершіть роботу за комп'ютером.

Зробіть висновок: як створити аудіо- та відеокліпи, додати переходи, налаштувати часові параметри аудіо- та відеоряду, як синхронізувати аудіо- та відеоряд відеофільму.



Практична робота 8

Розміщення відеоматеріалів в інтернеті

Завдання: опублікувати власний відеофільм у мережі інтернет та надати до нього доступ глядачам.

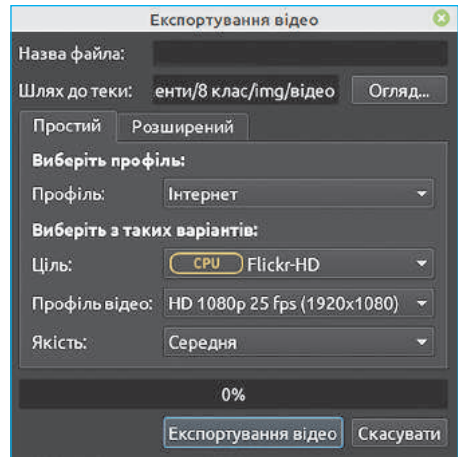
Обладнання: комп'ютер з браузером, відеоредактором, відеоконвертером, з'єднаний з інтернетом; проект Практична робота 7.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть відеоредактор і відкрийте проект з назвою Практична робота 7.
2. Експортуйте відеофільм із назвою Практична робота 8. У вікні експортування виберіть профіль Інтернет (див. рисунок).
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.
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. Завершіть роботу за комп'ютером.

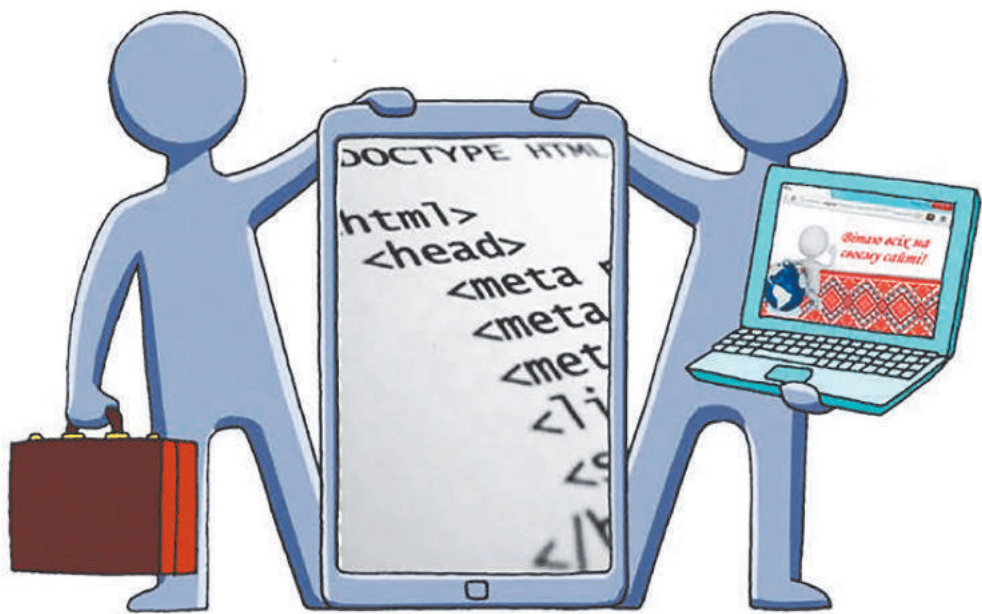
Зробіть висновок: як розмістити аудіо- та відеоматеріали в інтернеті, які переваги та недоліки мають різні способи розміщення.



РОЗДІЛ 4

СТВОРЕННЯ ТА ПУБЛІКАЦІЯ

ВЕБРЕСУРСІВ



§ 18. Обмін даними в інтернеті

§ 19. Адресація в інтернеті

§ 20. Створення сайтів

Практична робота 9. Конструювання сайтів

із використанням системи керування вмістом

Практична робота 10. Наповнення сайту вмістом

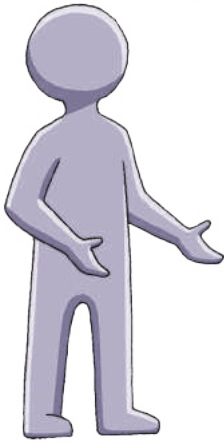
§ 21. Розмічання тексту засобами HTML

§ 22. Графічні зображення та гіперпосилання на вебсторінці

§ 23. Розмічання таблиць засобами HTML

Практична робота 11. 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 тощо.

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

Ви знаєте, що сайт складається з вебсторінок, поєднаних спільною темою. Існують сайти *різного призначення* — навчальні, соціальні, комунікаційні, розважальні тощо.

1. Які комп'ютери називають клієнтами; серверами?
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 також дізнаєтеся, що таке мова розмітки гіпертексту і як її застосовувати для форматування сторінок сайтів.

§ 18. Обмін даними в інтернеті

Доступ до інформаційних ресурсів мережі інтернет неможливий без організації швидкого та якісного передавання даних. Як це зробити?

Передавання даних в інтернеті

Переважна більшість мереж використовує пакетний принцип обміну повідомленнями. Спеціальні програми поділяють повідомлення на частини — **пакети даних**, долучають адреси відправника й одержувача та порядковий номер пакета, а потім послідовно передають мережею.

Потім комп'ютер-отримувач збирає пакети даних, об'єднує у правильному порядку й відновлює вихідні дані. Кожний пакет даних передається одним із мільйонів маршрутів. Навіть якщо деякі мережеві комп'ютери й канали зв'язку одночасно вийдуть з ладу, пакети даних дійдуть до одержувача іншим маршрутом.

Далі пакети збираються, вихідні дані відновлюються. Завдяки тому, що ці пакети пронумеровано, це легко зробити, навіть якщо вони дістались адресата в різний час і за різними маршрутами.

Кожен пакет містить інформацію про цілісність даних (контрольна сума). І якщо в пакеті виявляються помилки, комп'ютер-отримувач надсилає комп'ютеру-відправнику запит на повторне передавання саме цього пакета (рис. 18.1).

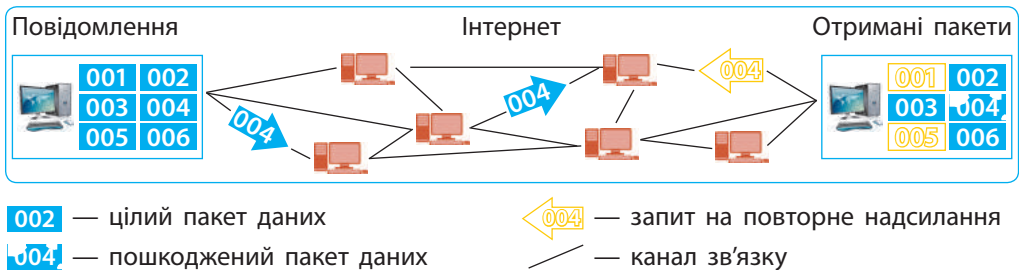


Рис. 18.1

Протоколи передавання даних

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



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

Розглянемо деякі найвідоміші протоколи та їхнє призначення:

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

Основними протоколами інтернету можна вважати TCP і IP, які забезпечують пакетний обмін даними. Розробляє та впроваджує технологічні стандарти міжнародна організація World Wide Web Consortium.

Провайдери

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



Провайдер (інтернет-провайдер) — це організація, яка надає доступ до інтернету, а також послуги, пов'язані з ним.

Сервер, який належить провайдеру, є початковою точкою доступу до інтернету. Саме з ним з'єднуються комп'ютери або локальні мережі.

Серед основних функцій провайдерів найважливішими є такі:

- забезпечення доступу до інтернету;
- виділення дискового простору для сайтів;
- підтримка роботи поштових скриньок;
- послуги зв'язку з передавання голосових повідомлень тощо.

Щоб почати роботу в інтернеті, потрібно:

- 1) одержати доступ до сервера одного з провайдерів;
- 2) установити й налаштувати програми-клієнти потрібних служб.

Способи підключення до інтернету

Існують різні способи підключення до інтернету (рис. 18.2).

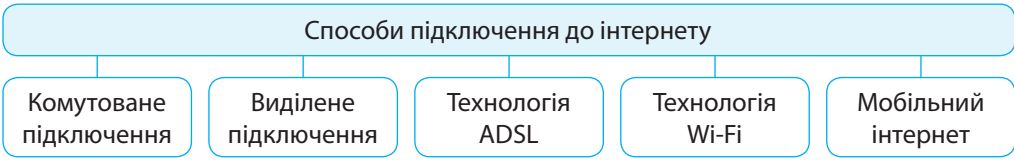


Рис. 18.2

Наведені способи підключення до інтернету відрізняються принципом роботи, швидкістю передавання даних тощо (рис. 18.3).

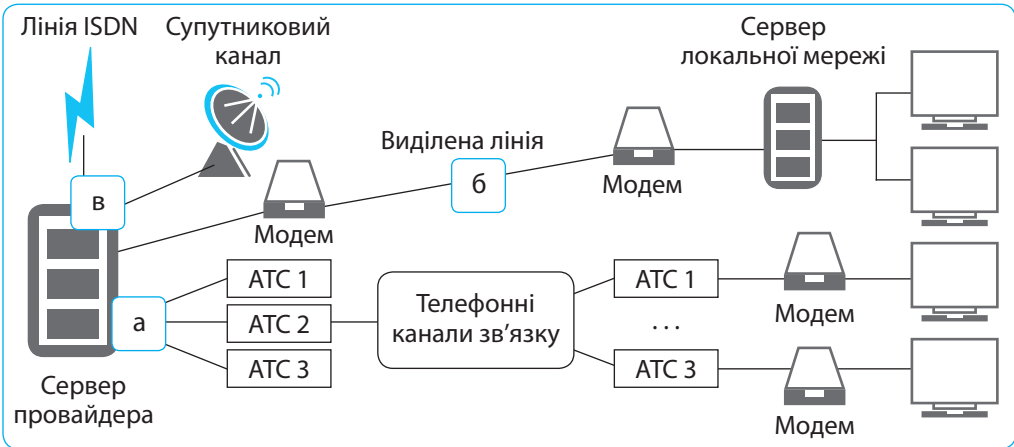


Рис. 18.3

Розглянемо наведені способи підключення докладніше.

► Комутоване підключення

Комутоване підключення здійснюється з використанням телефонної лінії та модема, що приєднується до комп'ютера. Цей спосіб є найдавнішим. На час встановлення комутованого з'єднання телефонна лінія блокується для голосового зв'язку (див. рис. 18.3, а).

► Виділене підключення

Виділене підключення здійснюється через виділений канал зв'язку, який забезпечує постійне з'єднання (кабель, оптичне волокно, радіоканал тощо) із сервером провайдера. Цей спосіб є найпопулярнішим.

Він недешевий, але завдяки великій швидкості передавання дозволяє користувачам обмінюватися великими обсягами даних (див. рис. 18.3, б).

▶ **Технологія ADSL**

Технологія ADSL (*A*symmetric *D*igital *S*ubscriber *L*ine) реалізується з використанням асиметричної цифрової передплатної лінії. Для обміну даними використовується телефонна лінія. Вона залишається доступною під час високошвидкісного доступу до інтернету.

▶ **Технологія Wi-Fi**

Технологія Wi-Fi — стандарт бездротового зв'язку. Її ядром є так звана точка доступу, підключена до мережі. Щоб з'єднатися з точкою доступу, власнику ноутбука або мобільного пристрою із Wi-Fi-адаптером потрібно просто потрапити в радіус її дії (50–100 метрів).

Існують й інші способи підключення до інтернету, наприклад за допомогою супутникової антени (див. рис. 18.3, в).

▶ **Підключення за допомогою мобільного пристрою**

Популярним є підключення за допомогою мобільного пристрою: телефону чи модема. Такий доступ обмежений територією, на якій провайдер створив покриття. Існує вже кілька поколінь мобільного зв'язку (рис. 18.4). Найпоширенішими є стандарти 3G і 4G.



Рис. 18.4

» Наприкінці 2020 року Китай запусив у космос перший у світі супутник для експериментів із технологією зв'язку 6G, яка використовуватиме високочастотні хвилі терагерцового діапазону.

Швидкість обміну даними мережею 5G в десятки разів перевищує можливості 4G. Стандарти, що розробляються для мережі 6G, у тисячі разів перевищуватимуть показники 5G (рис 18.5).



Рис. 18.5

Завантаження мережею 6G 100-годинного відеоматеріалу найвищої якості займатиме менше від секунди.

Питання для самоперевірки



1. Опишіть пакетний принцип передавання даних.
2. Що таке мережевий протокол?
3. Назвіть кілька мережевих протоколів. Яке їхнє призначення?
4. Назвіть основні функції провайдера.
5. Опишіть способи підключення до інтернету.
6. Як реалізується технологія ADSL; технологія Wi-Fi?

Вправа 18



- Визначити параметри підключення до інтернету: спосіб підключення, провайдера, IP-адресу та швидкість обміну даними.
- 1) Запустіть текстовий процесор і створіть новий документ.
 - 2) Визначте спосіб підключення вашої локальної мережі до інтернету та запишіть у створений документ.
 - 3) Відкрийте один із сайтів для визначення параметрів підключення до інтернету (наприклад, www.speedtest.net; 2ip.ua тощо). З'ясуйте назву вашого провайдера і запишіть у документ.
 - 4) Визначте IP-адресу вашого комп'ютера та запишіть у документ.
 - 5) Визначте швидкість приймання та передавання даних в інтернеті вашим комп'ютером та запишіть у документ.
 - 6) Збережіть файл із назвою Вправа 18. Завершіть роботу за комп'ютером.



Комп'ютерне тестування

Виконайте тестове завдання 18 із автоматичною перевіркою результату.



§ 19. Адресація в інтернеті

Призначенням будь-якої мережі є доступ до віддалених інформаційних, апаратних і програмних ресурсів і спільне їх використання.

Призначення і структура інтернету

Інтернет об'єднує окремі комп'ютери та локальні мережі в усьому світі. Розглянемо структуру інтернету, наведену на рис. 19.1.

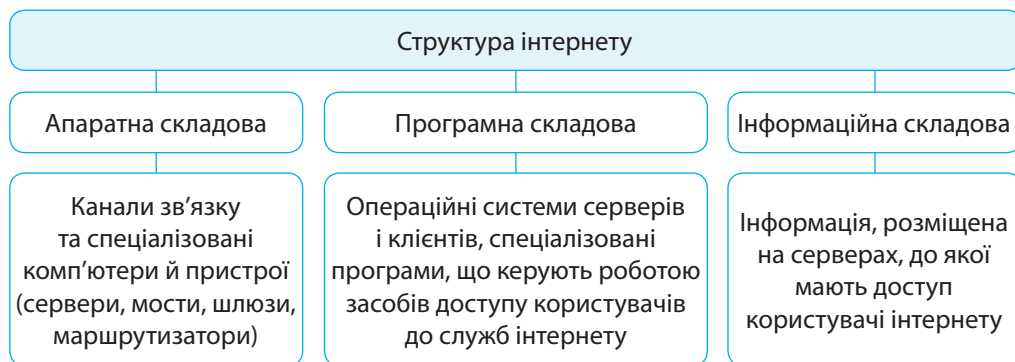


Рис. 19.1

Оснoву мережі інтернет складають сервери, з'єднані між собою лініями (каналами) зв'язку. Клієнти можуть перебувати на значній відстані від сервера і з'єднуватися з ним різними каналами зв'язку.

- ! Для узгодженої взаємодії усіх зазначених складових структур інтернету потрібна певна система адресації.

IP-адреса

Згадаємо, що 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.

З 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 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 $3,4 \cdot 10^{38}$ (близько 340 секстильйонів).

Комп'ютер у мережі може мати *статичну (постійну)* або *динамічну (тимчасову)* IP-адресу. Статичну IP-адресу мають усі сервери, щоб комп'ютери мережі «знали», де шукати інформацію, а динамічну IP-адресу комп'ютер отримує щоразу в процесі тимчасового з'єднання.

Доменне ім'я

IP-адреси складно запам'ятати, тому використовують доменні імена.



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

Домен (від англ. *domain* — територія, область, сфера) — це 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 кількох частин (назв доменів), розділених крапками. Домен, назву якого зазначено праворуч, називають доменом *першого* (або *верхнього*) *рівня* (рис. 19.2).

Домени першого рівня бувають *організаційні* (спільного користування) та *національні* (домени держав) (рис. 19.3).

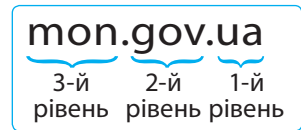


Рис. 19.2

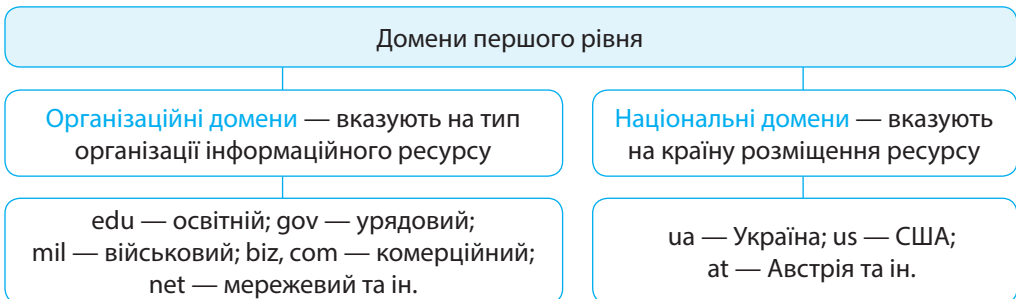


Рис. 19.3

Ліворуч від домену першого рівня зазначається домен *другого рівня* (він є ресурсом домену першого рівня) і так далі — до домену *четвертого* (або *нижнього*) *рівня*, який є ресурсом домену *третього рівня*.

Кожному доменному імені відповідає певна IP-адреса. Але комп'ютеру з певною IP-адресою 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, лише адреси сервера недостатньо. Потрібна також адреса ресурсу із зазначенням протоколу й унікального шляху до нього. Таку адресу називають уніфікованим вказівником на ресурс — **URL** (*Uniform Resource Locator*).



URL — це стандартизована адреса ресурсу в інтернеті (документа, відео, вебсторінки, зображення тощо).

URL зазвичай складається з трьох частин:

- 1) назви протоколу обміну даними (http, https, ftp та ін.);
- 2) доменного імені або IP-адреси сервера, де зберігається файл;
- 3) шляху до файлу на сервері.

1 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`, зазначати не обов'язково.

На багатьох сайтах вебсторінки генеруються динамічно за запитами на основі певних даних.

2 Динамічний URL може мати такий вигляд:

`http://www.example.com/index.php?do=static&page=str1`

ОС Windows має вбудовані засоби для перегляду вебсторінок. Якщо в адресному рядку вікна папки набрати URL веб-сторінки та натиснути клавішу `Enter`, то ця вебсторінка 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 використання доменних імен?
5. Назвіть кілька прикладів доменів першого рівня.
6. Що таке URL? Яке призначення URL?

Вправа 19



►► Визначити тип наданої провайдером IP-адреси (постійна чи тимчасова адреса) та складові URL інтернет-ресурсу.



- 1) Запустіть текстовий процесор і створіть новий документ.
- 2) Визначте IP-адресу вашого комп'ютера на спеціалізованому сайті (www.speedtest.net; 2ip.ua тощо).
Запишіть у створений документ.



- 3) Порівняйте щойно визначену IP-адресу з адресою, яку було визначено раніше на цьому комп'ютері у вправі 18.
Зробіть висновок щодо того, чи змінилася надана провайдером IP-адреса, чи є вона постійною, і запишіть у документ.



- 4) Знайдіть в інтернеті відомості про те, які символи є обов'язковими складовими великого Державного Герба України.
Знайдіть і скопіюйте один із ескізів Державного Герба України.
- 5) Відкрийте сторінку зі знайденим ескізом.
Запишіть до текстового документа URL ресурсу та поясніть його складові: протокол, ім'я сервера, шлях до файлу.
- 6) Збережіть файл із назвою Вправа 19 у відповідній папці.
Завершіть роботу за комп'ютером.

Комп'ютерне тестування



Виконайте тестове завдання 19 із автоматичною перевіркою результату.



§ 20. Створення сайтів

Розвиток інтернету й електронних технологій впливає на всі аспекти нашого життя. Так виникла необхідність шукати нові засоби комунікації. Популярністю користуються соціальні мережі, сайти тощо.

Вебсторінки і сайти

Інформаційний простір став мультимедійним; термін «гіпертекст» увів в інформатику американський учений Теодор Нельсон в 1963 році.



Сайт, або **вебсайт** (від англ. *web* — мережа, *site* — місце), — це об'єднання 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.

Вказівник миші, наведений на гіперпосилання, набуває вигляду руки (рис. 20.1). Гіпертекст є основою вебсторінки, вебсторінки у свою чергу об'єднуються у сайти, а сукупність сайтів утворює Всесвітню павутину.



Рис. 20.1

Розміщення сайтів у мережі

Інформацію із сайту зазвичай можна отримувати в будь-який час із будь-якого комп'ютера із доступом до інтернету. 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 розташовано більшість 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 створення сайтів, надання додаткового програмного забезпечення тощо. Хостинг може бути *платним* або *безкоштовним* (в останньому випадку встановлюються певні обмеження).

Етапи створення сайтів

Створення сайту — це трудомісткий і тривалий процес, який умовно можна розділити на декілька основних етапів (рис. 20.2).

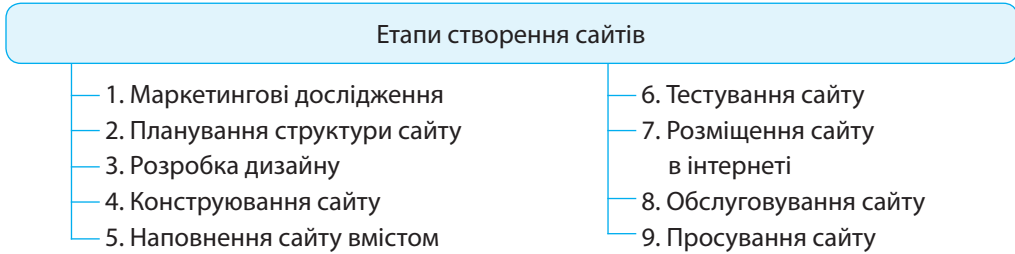


Рис. 20.2

Розглянемо наведені 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 конструювання та наповнення.

► 2. Планування структури сайту

Грамотна організація структури сайту підвищує зручність користування ним, його обслуговування тощо. Потрібно обміркувати структуру сайту: скільки сторінок він міститиме, яким чином користувач потраплятиме на сторінки, як знаходитиме потрібну інформацію тощо.

Для сайту з невеликою кількістю сторінок доцільно використовувати *ієрархічну структуру* (рис. 20.3), в якій перехід на потрібну сторінку можна здійснити за два-три клацання.

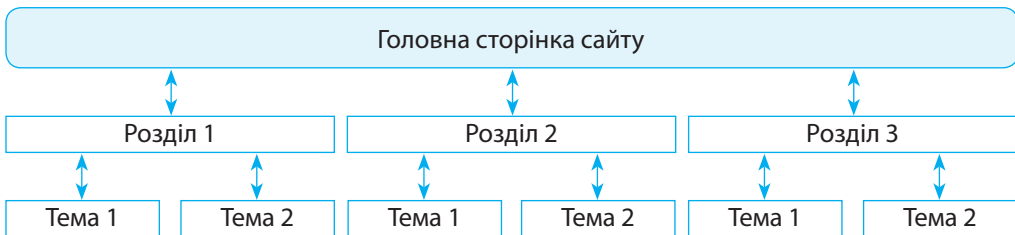


Рис. 20.3

Лінійну структуру варто використовувати для сайтів, перегляд сторінок на яких відбуватиметься послідовно (рис. 20.4).

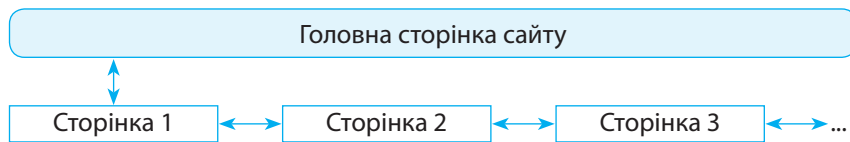


Рис. 20.4

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

► 3. Розробка дизайну

Вигляд вебсторінок сайту має відповідати певному стилю, бути привабливим і відрізнитися від аналогічних сайтів.

► 4. Конструювання сайту

Вебсторінки сайту повинні однаково добре відображатися й працювати в усіх браузерах. Сконструювати сайт можна шляхом написання HTML-коду або за допомогою спеціальних програм — **систем керування вмістом** — СКВ (від англ. *Content Management System* — CMS).

Існує багато СКВ, і кожна має певні особливості.

Найпопулярнішими є такі безкоштовні СКВ (рис. 20.5), як WordPress, Joomla, OpenCart, SocialEngine та ін.



Рис. 20.5

► 5. Наповнення сайту вмістом

Для наповнення сайту варто заздалегідь зібрати необхідні матеріали (тексти, зображення тощо), які становитимуть вміст (контент) сайту.

► 6. Тестування сайту

Створений і повністю або частково наповнений вмістом сайт доцільно випробувати на локальному сервері. Для створення локального сервера з метою розробки й тестування сайту можна використати безкоштовні програми, такі як Apache, PHP, MySQL, phpMyAdmin, або програму Open Server Panel, яка їх об'єднує.

► 7. Розміщення сайту в інтернеті

Для доступу користувачів інтернету до сайту він повинен мати доменне ім'я. Зазвичай послугу реєстрації доменних імен пропонують хостинг-провайдери. Вони й забезпечують цілодобовий доступ до сайту.

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

► 8. Обслуговування сайту

Обслуговування сайту полягає в його періодичному оновленні (тобто поповнення новими матеріалами), супроводі (адміністрування), обслуговуванні (удосконалення дизайну та структури сайту) і ін.

► 9. Просування сайту

Просування сайту полягає у здійсненні низки заходів (рекламування, оптимізація структури та змісту, організація опитувань та ін.) з метою збільшення його відвідуваності інтернет-користувачами.

Системи керування вмістом

Створити сайт можна за допомогою мов гіпертекстової розмітки та вебпрограмування. Але для вебмайстрів-початківців зручнішим є автоматизоване створення сайтів за допомогою онлайн-ових СКВ.

Вебсторінки зручно створювати в додатку Google Sites (рис. 20.6).

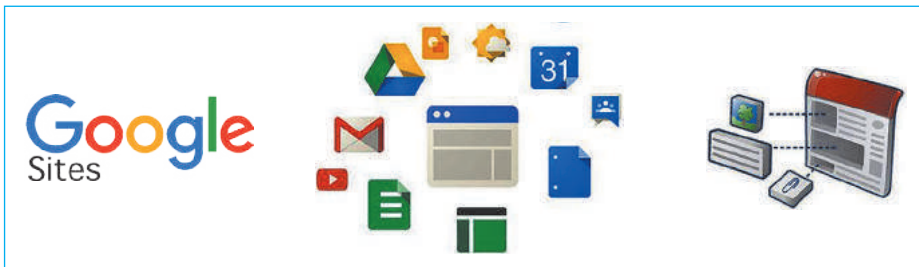


Рис. 20.6

Власний сайт можна швидко розробити за допомогою СКВ, скориставшись сайтами www.hostinger.com.ua; www.webkonstruktor.com та багатьма іншими. Ці сайти також надають послуги платного та безкоштовного хостингу, реєстрації доменних імен, електронної пошти та ін.

До оформлення сайту потрібно підходити виважено. Дизайн вебсторінок можна вибрати з колекції готових шаблонів і налаштувати відповідно до власних потреб.

Питання для самоперевірки



1. Дайте означення гіпертексту.
2. Що таке сайт?
3. Що таке хостинг?
4. Назвіть основні етапи створення сайту.
5. Опишіть особливості ієрархічної структури сайту; лінійної структури; мішаної структури.
6. Що таке система керування вмістом? Наведіть приклади СКВ.

Вправа 20



▶▶ Спланувати структуру власного сайту.

- 1) Запустіть текстовий процесор і створіть новий документ із назвою Вправа 20.
- 2) Запишіть у документ відомості про майбутній сайт: тематику, назву, призначення, мету створення, а також на кого орієнтовано сайт (цільову аудиторію).
- 3) Знайдіть в інтернеті сайти схожого призначення та визначте, чим ваш сайт відрізнятиметься від решти подібних. Продумайте структуру (кількість і розташування сторінок, меню, посилання, інші елементи) свого майбутнього сайту.
- 4) У графічному редакторі схематично зобразіть головну сторінку сайту, використовуючи як зразок головні сторінки наявних сайтів схожого призначення, або зробіть оригінальний ескіз. Вставте зображення головної сторінки майбутнього сайту в текстовий документ.
- 5) Продумайте та створіть вміст однієї з вебсторінок сайту: меню, посилання, тексти тощо. Збережіть вміст сторінки та її ескіз.
- 6) Збережіть документ. Завершіть роботу за комп'ютером.



Комп'ютерне тестування



Виконайте тестове завдання 20 із автоматичною перевіркою результату.





Практична робота 9

Конструювання сайтів із використанням системи керування вмістом

Завдання: створити і розмістити на хостингу сайт із використанням системи керування вмістом.

Обладнання: комп'ютер, підключений до інтернету.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Продумайте структуру свого майбутнього сайту або скористайтеся документом із назвою Вправа 20. Запустіть браузер і відкрийте вебсторінку webkonstruktor.com (рис. 1).

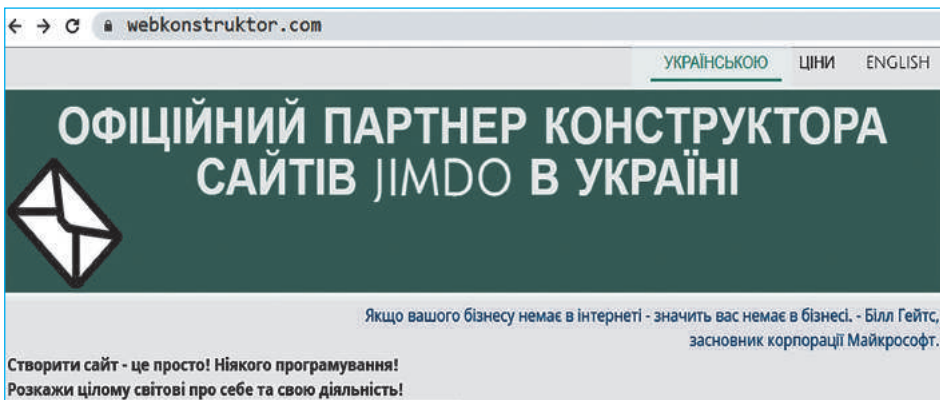


Рис. 1



2. Відкрийте на сайті сторінку безкоштовного створення сайту. Перейдіть за посиланням: Створити безкоштовно сайт.
3. Відкрийте вікно входу до системи конструктора сайтів: команда LOG IN (рис. 2).



Рис. 2

Зареєструйтесь на сайті webkonstruktor.com за допомогою своєї електронної пошти: кнопка **Sign Up**.

Відкрийте свою поштову скриньку та в отриманому електронному листі від webkonstruktor.com перейдіть за посиланням.

Примітка. Можна зайти до системи webkonstruktor.com за допомогою облікових записів Google  Google або Facebook  Facebook.

4. Натисніть кнопку Створіть вебсайт (рис. 3).

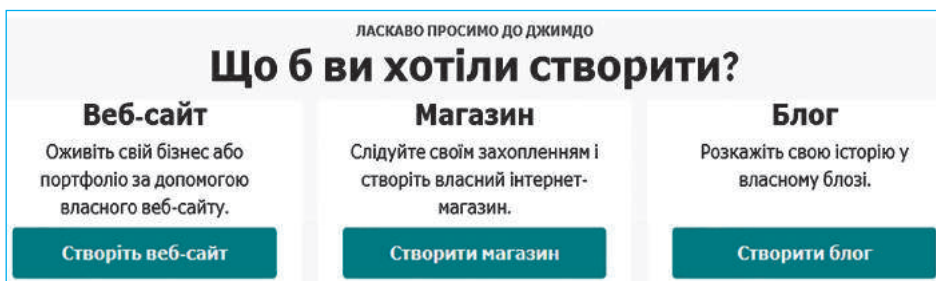


Рис. 3

Виберіть тип сайту. Клацніть шаблон сайту, який вам сподобався. Виберіть режим безкоштовного створення сайту.

Примітка. Переклад англomовних назв і команд можна здійснити засобами контекстного меню сторінки у браузері Google Chrome.

5. Придумайте назву безкоштовного домену сайту (рис. 4). Перевірте, чи не зайнята вже кимось назва, придумана вами.

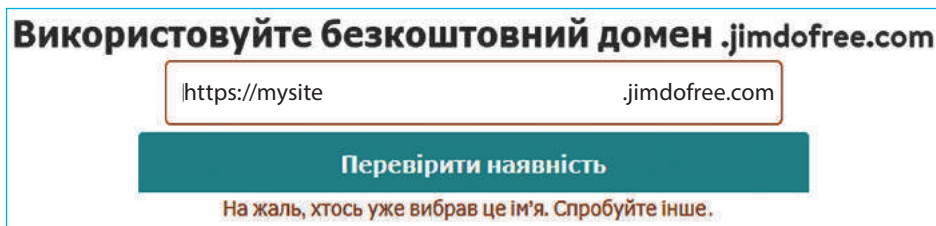




Рис. 4

За потреби придумайте нову назву та натисніть кнопку Створити безкоштовно сайт. Відкриється адміністративна панель з вибраним шаблоном.

6. Відкрийте панель інструментів кнопкою  Меню у верхньому лівому кутку сторінки.

Примітка. Щоб відкрити колекцію шаблонів для заміни вибраного виконайте команди:  Меню → Дизайн → Шаблони.

7. Увімкніть налаштування окремих елементів:  Меню → Дизайн → → Стиль (рис. 5).

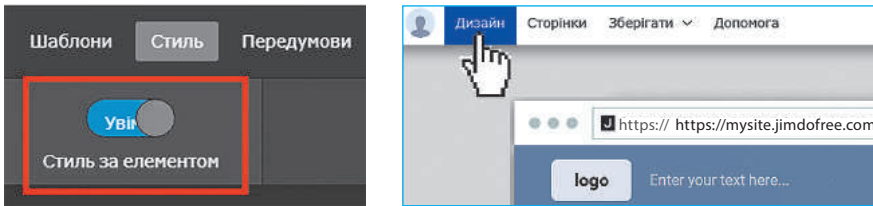



Рис. 5

8. Клацаючи окремі елементи шаблону у візуальному режимі конструктора, змініть назву сайту, опис, відредагуйте навігаційне меню та підменю згідно із задумом.

Змініть колірну схему і шрифт заголовка на власний розсуд.

9. Установіть фоновий малюнок відповідно до задуму. Закрийте меню налаштувань.
10. Вийдіть із режиму редагування сайту, виконавши команду  Меню → Панель приладів → Профіль → Вийти.

Примітка. Надалі для входу в режим адміністрування слід до адреси сайту додавати /login. Наприклад: mysite.jimdofree.com/login

11. В адресному рядку браузера введіть адресу свого сайту, наприклад mysite.jimdofree.com, і перегляньте, як він виглядає.
12. Надішліть адресу свого сайту вчителю.
Завершіть роботу за комп'ютером.

Зробіть висновок: які етапи розробки власного сайту з використанням онлайнві СКВ були найважчими, а які несуттєвими і чому.



Практична робота 10



Наповнення сайту вмістом

Завдання: наповнити кілька вебсторінок власного сайту вмістом з використанням онлайнної СКВ.

Обладнання: комп'ютер, підключений до інтернету.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Продумайте зміст і наповнення кількох сторінок вашого сайту.
2. Запустіть браузер, в адресному рядку введіть адресу свого сайту, створеного на практичній роботі 9, і перегляньте його.
3. Відкрийте вікно входу до системи адміністрування сайту, наприклад `mysite.jimdofree.com/login`, увійдіть до свого облікового запису.
4. Видаліть на головній сторінці зайві написи, меню, зображення.
5. Додайте (або змініть) на головній сторінці сайту написи, меню і зображення згідно із вашим задумом і відповідною тематикою.
6. В режимі візуального редактора створіть нову вебсторінку.
7. Заповніть створену вебсторінку текстами й малюнками відповідно до вашого задуму.
8. Створіть ще одну сторінку, заповнивши її текстами та малюнками відповідно до задуму.
9. Перегляньте, як виглядає ваш сайт. Для цього натисніть кнопку  Режим перегляду у правому верхньому кутку сторінки і виберіть команду Відкрити посилання.
10. За необхідності на вкладці адміністрування сайту внесіть зміни і доповнення до дизайну та вмісту вашого сайту.
11. Вийдіть із режиму редагування сайту, вибравши команду  Меню → Панель приладів → Профіль → Вийти.
12. Надішліть адресу свого сайту вчителю. Завершіть роботу за комп'ютером.

Зробіть висновок: які зміни в дизайні сайту та наповненні його вмістом вам вдалося виконати якнайкраще, а які — не дуже вдало і чому.

§ 21. Розмічання тексту засобами HTML

Текстові редактори мають інструменти для форматування. Як відбувається форматування гіпертексту для відображення у вікні браузера?

Поняття про мову HTML

Щоб браузер потрібним чином відображав дані на вебсторінках, застосовують спеціальні позначки — *теги*, сукупність яких разом із правилами їх використання називають **мовою розмітки гіпертексту**.



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

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

HTML-документ

HTML-документ (HTML-файл) є текстовим документом, тому створити його можна в будь-якому текстовому редакторі, з яким ви працювали.



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

Базову структуру HTML-документа визначають такі теги:

<code><html></code>	Початок документа
<code><head></code>	Початок заголовка — службової частини коду
<code><title></code>	Назва, що виводитиметься на вкладці браузера
<code></title></code>	Кінець назви
<code></head></code>	Кінець заголовка
<code><body></code>	Початок тіла документа, яке виводить браузер
<code></body></code>	Кінець тіла документа
<code></html></code>	Кінець документа

1 На рис. 21.1 наведено один із варіантів відображення вебсторінки.

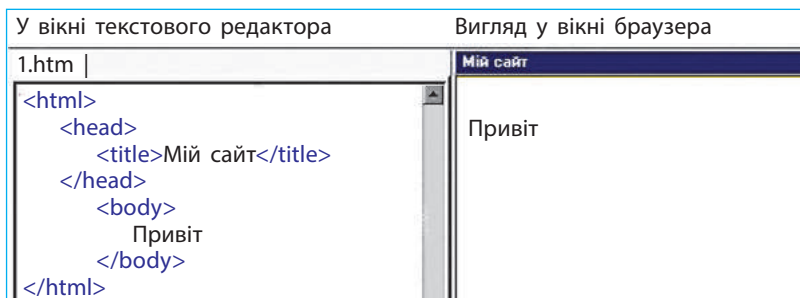


Рис. 21.1

Форматування тексту

Текст із тегами можна розташовувати довільно: розбивати на абзаци, виділяти відступи. У HTML-документі розмір і накреслення символів, які відображаються у вікні браузера, визначаються розміткою.

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

Теги бувають парними і непарними. **Парний тег** складається із відкривального тегу і закривального тегу, який починається символом «/». Теги з помилками або такі, що не підтримуються браузером, ігноруються.

2 `...` — парний тег накреслення жирний (рис. 21.2).
`
` або `
` — непарний тег розриву рядка.

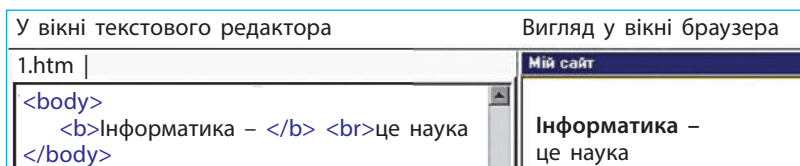


Рис. 21.2

Розглянемо деякі теги форматування тексту:

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

Тег може містити атрибути, значення яких впливають на його дію. Атрибути записуються у відкривальному тегу після його назви.

! Значення атрибутів, що містять пропуски, беруть в одинарні або подвійні лапки. Для решти значень це не обов'язково.

3 Застосуємо атрибути text (колір тексту) і bgcolor (колір тла) (рис. 21.3):

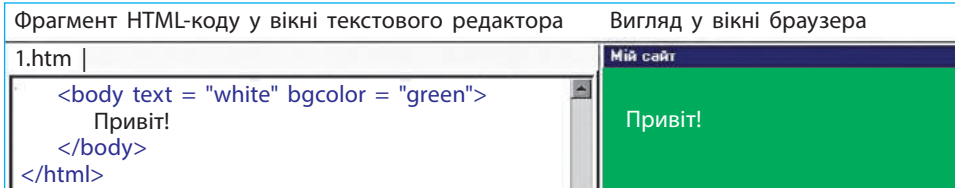


Рис. 21.3

4 Розглянемо приклад застосування атрибутів на рис. 21.4.

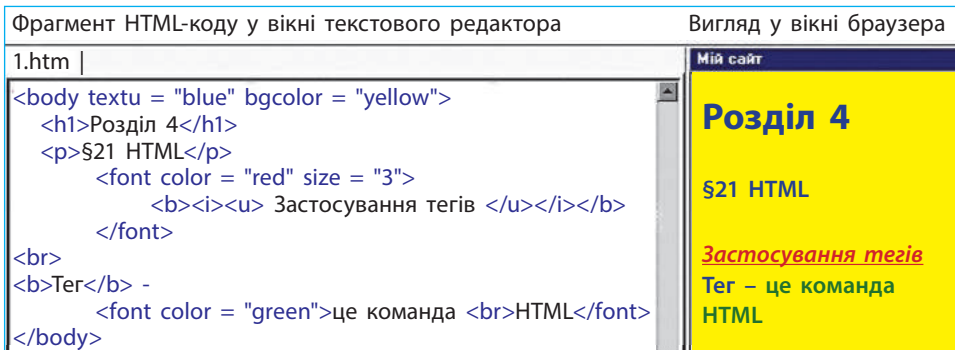


Рис. 21.4

Марковані та нумеровані списки

Для створення маркованого списку використовують тег `...` з атрибутом `type`, який задає вигляд маркера:

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

Для створення нумерованих списків використовують тег `...` з атрибутами `type` (тип нумерації: "1" — арабські цифри, "A" — латинські літери, "I" — римська нумерація тощо) і `start` (задання початкового номера в списку арабськими цифрами).

5 Елементи списку розмічають тегами ... (рис. 21.5).

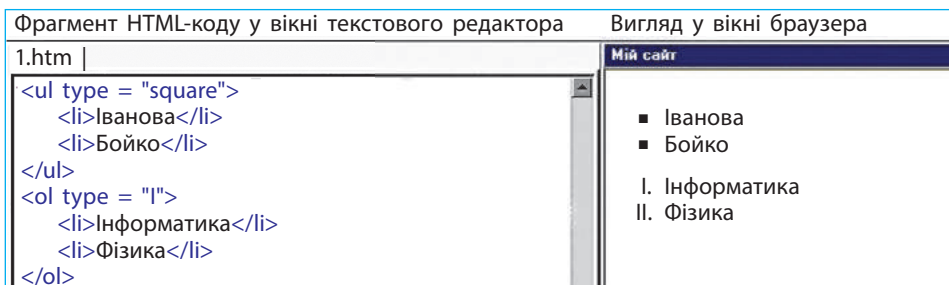


Рис. 21.5

Питання для самоперевірки



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

Вправа 21



- ▶ Створити HTML-документ із відформатованим текстом.
- 1) Запустіть текстовий редактор Блокнот і введіть у документ набір тегів базової структури HTML-документа (див. рис. 21.1).
 - 2) Збережіть створений документ як HTML-файл: Файл → Зберегти як → Тип файлу (всі файли) → Ім'я: Вправа 21.html.
 - 3) Відкрийте папку зі збереженим файлом і за допомогою браузера відкрийте HTML-файл Вправа 21.
 - 4) Змініть HTML-код так, щоб нумерований список відображався на синьому тлі, колір шрифту — жовтий, тип накреслення — жирний (рис. 21.6).
 - 5) Збережіть файл Вправа 21.html і оновіть сторінку браузера.
 - 6) За потреби відкоригуйте HTML-код відповідно до п. 4.

2. Віторок
3. Середа
4. Четвер

Рис. 21.6

Комп'ютерне тестування



Виконайте тестове завдання 21 із автоматичною перевіркою результату.



§ 22. Графічні зображення та гіперпосилання на вебсторінці

Сайт стає більш цікавим і наочним, якщо його вебсторінки доповнити графічними зображеннями, а також застосувати гіперпосилання.

Вставлення малюнків

На вебсторінку сайту можна додавати різноманітні зображення.

Щоб вставити на вебсторінку малюнок з файлу, необхідно скористатися тегом ``, де `img` — назва тегу, призначеного для вставлення зображення, `src = "my.jpg"` — атрибут, в якому вказується ім'я файлу з малюнком, що міститься в одній папці з HTML-файлом.

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

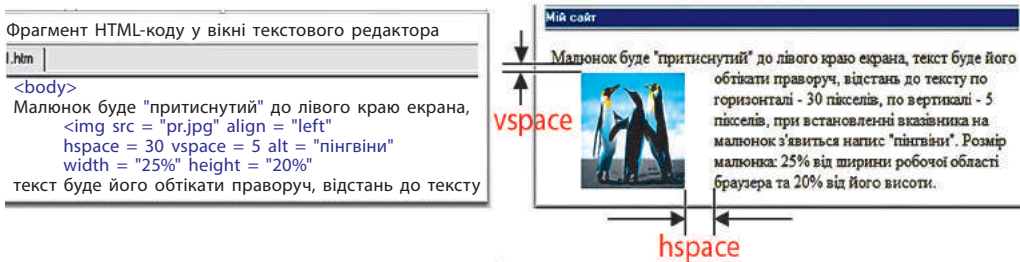
1 Якщо малюнок міститься у вкладеній папці `images`:

```
<img src = "images/my.jpg">
```

Якщо малюнок міститься у зовнішній папці `images`:

```
<img src = "../images/my.jpg">
```

Для визначення способу розташування малюнка на вебсторінці використовують певні атрибути тегу. Наведемо можливе розташування малюнка на вебсторінці (див. рисунок) та обтікання його текстом.



Проаналізуємо декілька тегів для вставлення малюнків:

- малюнок з лівого краю, обтікання текстом праворуч:
``

- відстань між текстом і малюнком по вертикалі 10 пікселів, по горизонталі — 30 пікселів:
``

- опис малюнка:
``

Примітка. Якщо навести вказівник на малюнок, з'явиться текст — Моє фото. Крім того, значення параметра alt буде виведене на екран замість малюнка, якщо у браузері вимкнено показ малюнків.

- ширина малюнка 100 пікселів, висота — 20 % від висоти робочої області вікна браузера:
``
- товщина рамки навколо малюнка — 5 пікселів:
``

Додавання гіперпосилань

Сайт складається з багатьох вебсторінок, пов'язаних гіперпосиланнями, одна з яких є головною. Ця сторінка сайту зазвичай міститься у файлі index.html і відкривається першою. Для створення гіперпосилань на вебсторінки сайту та зовнішні ресурси інтернету використовують тег `<a>...`.

- 2 Приклад додавання гіперпосилань до рисунків і фрагментів тексту. Нехай в одній папці з файлом index.html створено файл prf.html, який містить вашу фотографію. У файлі index.html до фрази Подивитися фото можна додати гіперпосилання, яке матиме такий вигляд:
` Подивитися фото `

Щоб гіперпосилання можна було додати до малюнка, необхідно вказати тег вставляння малюнка:

```
<a href = "prf.html"><img src = "foto1.jpg"></a>
```

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 (Друга сторінка, Третя сторінка) відбудеться перехід за гіперпосиланням на відповідну вебсторінку.

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 малюнок на вебсторінку?
2. У чому відмінність між значеннями атрибутів для вставлення малюнку із поточної, внутрішньої і зовнішньої папок?
3. Як вирівняти малюнок 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 22



►► Створити вебсторінку сайту з малюнками і гіперпосиланнями.

- 1) Створіть папку з назвою Вправа 22. Знайдіть в інтернеті малюнок на шкільну тематику та збережіть його у створеній папці.
- 2) Створіть HTML-документ (головну сторінку сайту) із заголовком Розклад уроків. Додайте малюнок, збережений у папці, 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) з розкладом уроків на кожний із цих днів тижня. До кожної сторінки додайте зверху її назву та знизу слово Головна, яке буде гіперпосиланням на головну сторінку. 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 і графічні зображення.
- 6) Перевірте роботу сайту. Завершіть роботу за комп'ютером.



Комп'ютерне тестування

Виконайте тестове завдання 22 із автоматичною перевіркою результату.



§ 23. Розмічання таблиць засобами HTML

Інформацію на вебсторінці сайту можна подати у вигляді таблиці.

Теги розмітки таблиці

Ознайомимося з основними тегами розмітки таблиці:

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

Розглянемо, як розмічати таблицю відповідними тегами (рис. 23.1):

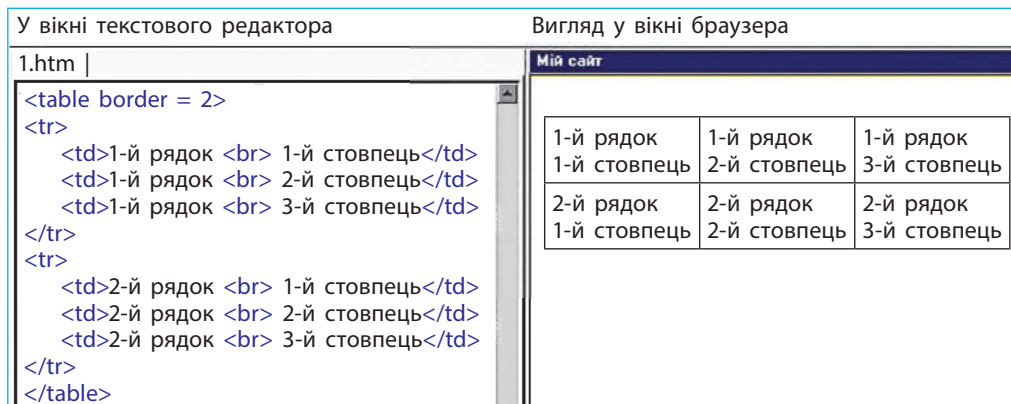


Рис. 23.1

Макет вебсторінки можна утворити за допомогою HTML-таблиці, об'єднуючи певні клітинки і визначаючи розміщення певних об'єктів. Але зараз для вебсторінок застосовують теги структурування документа (див. форзац 2) і CSS (від англ. *Cascading Style Sheets*) — каскадні таблиці стилів, які розміщують у HTML-файлі або в окремому файлі.

Атрибути тегів розмітки таблиці

Теги можуть містити атрибути. Атрибути в тегах розмітки таблиці впливають на вигляд таблиці, клітинок і написів у них:

Атрибут	Призначення
align	Вирівнювання таблиці у вікні браузера
background	Фоновий малюнок
bgcolor	Колір тла
border	Товщина рамки в пікселях
bordercolor	Колір рамки
cellspacing	Відстань між клітинками таблиці
cellpadding	Відстань між текстом клітинки та її межею
rules	Відображення рамок таблиці
title	Спливна підказка
width	Ширина таблиці (клітинки) у відсотках або пікселях
height	Висота таблиці (клітинки) у відсотках або пікселях

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.

Відступи

Для налаштування відстаней між клітинками, а також між текстом і межами використовують атрибути відступів (рис. 23.2):

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

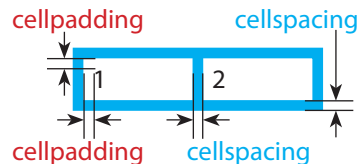


Рис. 23.2

Форматування вмісту клітинок

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

Атрибут align дозволяє задавати горизонтальне вирівнювання:

align = "left" — за лівим краєм;

align = "center" — по центру клітинки;

align = "right" — за правим краєм.

Атрибут valign дозволяє задавати вертикальне вирівнювання:

valign = "middle" — посередині клітинки;

valign = "top" — вгорі клітинки;

valign = "bottom" — внизу клітинки.

Варіанти вирівнювання тексту в клітинках показано на рис. 23.3.

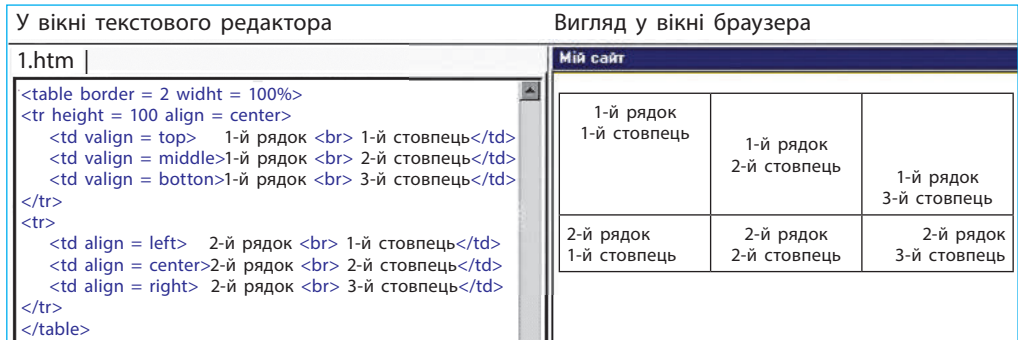
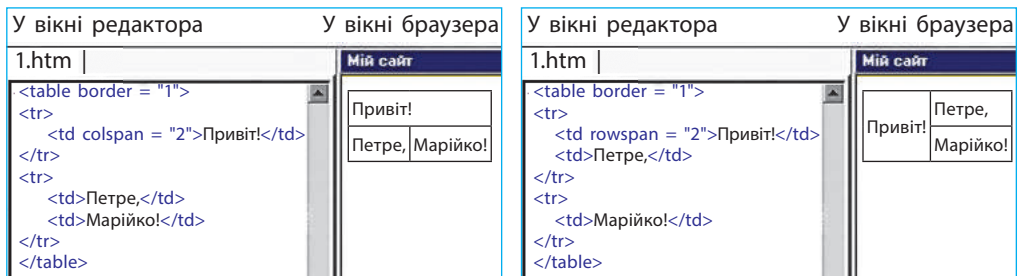


Рис. 23.3

При розмічанні таблиці виникає потреба об'єднати кілька клітинок. Для цього використовують атрибути colspan (злиття клітинок у рядку) (рис. 23.4, а) і rowspan (злиття клітинок у стовпці) (рис. 23.4, б).



а

Рис. 23.4

б

Питання для самоперевірки



1. Як додати таблицю на вебсторінку сайту?
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 23



►► Створити таблицю за зразком, наведеним на рис. 23.5.

- 1) Відкрийте текстовий редактор та створіть HTML-документ для відображення таблиці (рис. 23.6). Збережіть документ з іменем Вправа 23.html.
- 2) Змініть HTML-код, об'єднавши потрібні клітинки за зразком, і заповніть їх, як показано на рис. 23.7.
- 3) Задайте розміри таблиці в пікселях: ширина — 400, висота — 200.
- 4) Установіть параметри клітинок у відсотках від розміру таблиці: висота верхнього рядка — 35 %, ширина другого і третього стовпців — 60 % і 20 % відповідно.
- 5) Вирівняйте по вертикалі і горизонталі написи в клітинках (див. рис. 23.5).
- 6) Збережіть документ. Перегляньте таблицю в браузері. За необхідності відформатуйте клітинки таблиці і написи в них у відповідності до зразка (рис. 23.5).

Завершіть роботу за комп'ютером.

1111	2222	
	3333	4444

Рис. 23.5

1111	2222	3333
4444	5555	6666

Рис. 23.6

1111	2222	
	3333	4444

Рис. 23.7

Комп'ютерне тестування



Виконайте тестове завдання 23 із автоматичною перевіркою результату.





Практична робота 11

Створення головної сторінки сайту засобами HTML

Завдання: створити головну сторінку сайту засобами HTML, дотримуючись правил ергономічного розміщення матеріалів на вебсторінці.

Обладнання: комп'ютер, підключений до інтернету, комп'ютерний або онлайн-редактор сайтів.

Теоретична частина

Більшість шаблонів, які пропонують СКВ, створюються професіоналами; в них уже дотримано основні правила щодо розробки сайтів:

- вміст лаконічний, поданий у зрозумілій формі;
- посилання на вебсторінках помітні та однозначні;
- навігація вебсторінками сайту зручна і проста.

Вебсторінки сайту мають відповідати основним правилам розробки сайтів та критеріям ергономічного розміщення матеріалів на сторінках сайту.

Критерії ергономічного сайту

► Оформлення вебсторінок

- Дотримання єдиного стилю оформлення вебсторінок.
- Вибір кольорів тла та тексту згідно з правилами колористики.
- Читабельний шрифт, виділення заголовків кольором і більшим розміром шрифту.
- Вирівнювання тексту за шириною.
- Якісні графічні зображення з можливістю перегляду.
- Уникнення використання подразнюючої анімації.
- Текстові поля і зображення з відступами один від одного та від країв таблиць.

► Зміст наповнення

- Зазначення назви сайту на всіх вебсторінках.
- Наявність заголовків на вебсторінках.
- Розміщення найважливіших відомостей на початку вебсторінки.
- Розміщення тексту максимум у три колонки.
- Недопустимість використання горизонтальних смуг прокручування для перегляду сторінок.

- Дублювання в нижньому колонтитулі меню, розміщення контактних даних власника ресурсу, відомостей про авторські права.

► Розташування вмісту

- Можливість переходу з усіх вебсторінок на головну сторінку.
- Використання не більш ніж трьох рівнів вкладеності вебсторінок.
- Помітність і наочність меню і кнопок, уникнення їх двозначності.
- Відмінність гіперпосилань від решти тексту.
- Використання в одному блоці не більш ніж 7 гіперпосилань (краще використовувати вкладені меню або блоки).
- Використання засобів навігації всередині тексту (зміст із посиланнями тощо) на вебсторінках із великим обсягом тексту.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

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



а



б

Рис. 1

4. За вказівкою вчителя запустіть встановлений на вашому комп'ютері один із HTML-редакторів, наприклад Microsoft SharePoint Designer, hefs-редактор тощо, або один із онлайн-редакторів: Online

HTML Editor; html-instant; Real-Time HTML Editor; TimsFreeStuff HTML Editor; Online WYSIWYG HTML Editor тощо.

5. Створіть HTML-код вебсторінки із порожньою таблицею і доповніть його кодом таблиці з потрібною кількістю рядків і стовпців відповідно до вашого макета (п. 3, 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, щоб загальний вигляд таблиці відповідав запланованому розташуванню елементів на головній сторінці вашого сайту (рис. 2).

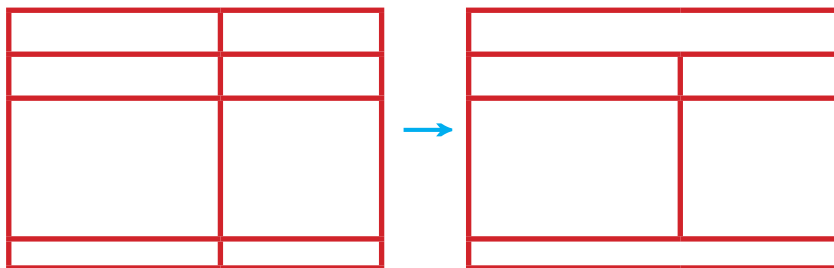


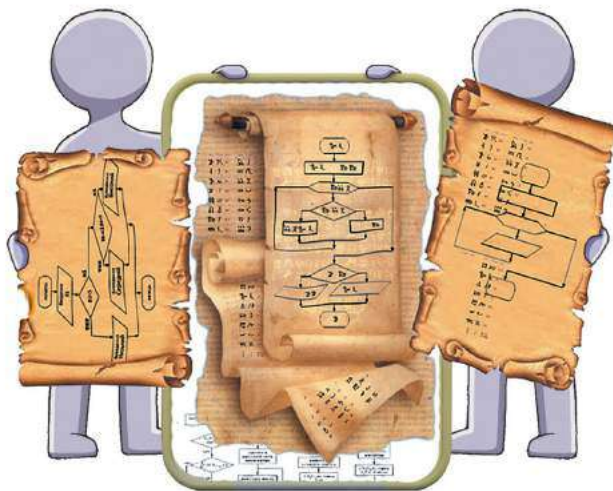
Рис. 2

7. Установіть потрібні розміри (відносні та/чи абсолютні) для комірок таблиці згідно із зображеним вами макетом (рис. 2).
8. Додайте до таблиці потрібні написи і рисунки, меню, кнопки тощо відповідно до ескізу головної сторінки (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 роботу посилань.
11. Вимкніть відображення меж таблиці. Збережіть HTML-файл. Перегляньте сторінку сайту у кількох браузерах, з різним масштабуванням. Перевірте роботу меню та/або кнопок.
12. Надішліть HTML-файл вчителю електронною поштою або збережіть файл на хмарному диску, надавши до нього доступ вчителю. Завершіть роботу за комп'ютером.

Зробіть висновок: як використовувати таблицю для макетування вебсторінки.

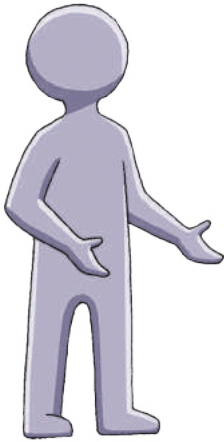
РОЗДІЛ 5

АЛГОРИТМИ ТА ПРОГРАМИ



- § 24. Сучасні мови програмування. Система програмування PyCharm
- § 25. Налаштування програмного коду
- § 26. Об'єкти, їх властивості та методи
- § 27. Графічний інтерфейс користувача
- § 28. Обробники подій
- § 29. Організація діалогу користувача з програмою
Практична робота 12. Введення і виведення даних
- § 30. Функції користувача
- § 31. Величини. Числові типи даних
- § 32. Математичні функції
- § 33. Рядковий тип даних
- § 34. Робота з текстовими файлами
- § 35. Логічний тип даних. Умовні оператори
- § 36. Елементи вибору
- § 37. Алгоритми з повтореннями. Цикли
- § 38. Словники. Віджет Combobox
Практична робота 13. Створення програми з використанням словників і списків

ПОВТОРЮЄМО



У 5–7 класах ви опанували *основи мови програмування Python*, знаєте, як реалізувати базові алгоритмічні структури і вмієте створювати програми з використанням розгалужень і повторень.

Ви розширили уявлення про *типи величин і алгоритми опрацювання даних різних типів* і розробляли програми з графічним інтерфейсом, створюючи об'єкти класів графічних компонентів. Ви також навчилися задавати властивості найуживаніших об'єктів графічного інтерфейсу та програмувати обробку подій для цих об'єктів.

Ви вмієте описувати об'єкти, використовувати *екземпляри класів* у програмах; оволоділи прийомами роботи в середовищі програмування IDLE, навчилися *вести діалог* в інтерактивному режимі в оболонці IDLE та ін.

1. Що таке тип величини?
2. З якими типами даних ви працювали?
3. Які базові алгоритмічні структури використовуються для конструювання алгоритмів?
4. Опишіть алгоритм створення програмного коду.
5. Який модуль містить набір компонентів для створення графічного інтерфейсу?



У цьому розділі ви продовжите вивчення основ програмування мовою Python і навчитесь працювати в середовищі PyChart. Воно включає текстовий редактор для введення програми з можливостями інтелектуального введення та автодоповнення коду, транслятор тощо.

§ 24. Сучасні мови програмування. Система програмування PyCharm

Історія мов програмування бере початок у XIX ст. і бурхливо розвивається з появою ЕОМ. Нині існує близько 8500 мов програмування.

Програма та мова програмування

Ви вже знайомі з поданням алгоритму розв'язування задачі у вигляді програмного коду. Пригадаємо деякі основні означення.

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

Мова програмування — це штучна мова, що являє собою 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 комп'ютером команд, записаних мовою програмування

Класифікація мов програмування

Всі мови можна класифікувати за різними критеріями (рис. 24.1).

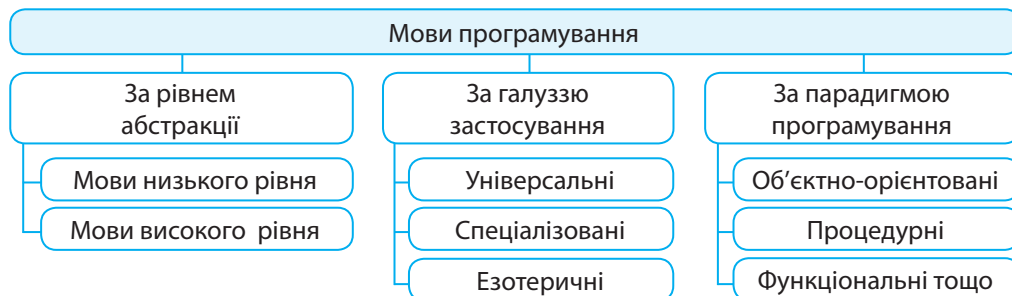


Рис. 24.1

Розглянемо зазначені критерії класифікування мов детальніше.

► За рівнем абстракції

Мови програмування низького рівня (наприклад, мова асемблера) базуються на машинних командах процесора. Їх використовують для розробки швидкодійних програм.

Мови програмування високого рівня оперують сутностями, зрозумілишими людині, — об'єктами, функціями тощо (див. таблицю).

Машинний код	Мова асемблера	Мова високого рівня
10100001	mov eax, a	a = int(input())
10001011	mov ebx, b	b = int(input())
00011011	add eax, ebx	print(a*b)

Програми, що розроблені мовами високого рівня, можна запускати на будь-яких комп'ютерах, для яких існують транслятори цих мов.

► За галузю застосування

Універсальні мови використовуються для розв'язування різних завдань, **спеціалізовані мови** — для розв'язування завдань певного виду.

1 До універсальних мов належать мови Python, C/C++, Java та ін. (рис. 24.2), до спеціалізованих — призначені для вебпрограмування мови Perl, VBScript, JavaScript та ін. (рис. 24.3).



Рис. 24.2



Рис. 24.3

Езотеричні мови, з одного боку, здебільшого не призначені для практичного застосування, з іншого — є непоганим тренуванням для досвідчених програмістів і створюються здебільшого для розваги.

► За парадигмами програмування

Парадигма програмування — це система ідей і понять, які визначають стиль написання програм, а також як програміст уявляє роботу програми. Існує низка парадигм; більшість мов (зокрема й Python) підтримують декілька з них. Ви використовували елементи чотирьох: структурної, процедурної, об'єктно-орієнтованої, подійно-орієнтованої.

За **структурної парадигми** програма розглядається як послідовність дій. Базовими поняттями структурного програмування є: *оператор* (команда) та *базові алгоритмічні структури*.

За **процедурної парадигми** програма складається з окремих блоків команд — *процедур* або *функцій*. Це дозволяє використовувати певний фрагмент коду, записавши його один раз і надавши йому 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. Якщо він може реагувати на події, які з ним відбуваються, і надсилати повідомлення іншим об'єктам, то це ознака **подійно-орієнтованої парадигми**.

» Доктор фізико-математичних наук Катерина Ющенко (рис. 24.4) стояла біля витоків української школи програмування. Вона авторка однієї з перших у світі мов програмування високого рівня, в якій застосовуються операції над адресами об'єктів у пам'яті комп'ютера. Ця мова використовувалась на комп'ютерах «Київ», «Дніпро», М20, БЕСМ, «Урал».



Рис. 24.4

Інтерпретація та компіляція програм

Для перекладання програм із мов програмування високого рівня на машинну мову застосовують спеціальні програми — **транслятори**. Вони поділяються на дві категорії: інтерпретатори і компілятори.

Інтерпретатор перетворює невеликий фрагмент програми на машинний код, який одразу виконується процесором. Далі інтерпретатор опрацьовує наступний фрагмент програмного коду. Машинний код для повторного виконання не зберігається (рис. 24.5).

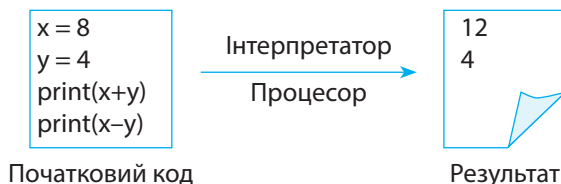


Рис. 24.5

Компілятор перетворює відразу всю програму на машинні коди і вміщує їх у пам'ять комп'ютера, не виконуючи (рис. 24.6). Скомпільовану програму вже можна зберегти для подальшого використання.



Рис. 24.6

Збережений результат компіляції називається **виконуваним файлом** (наприклад, із розширенням *.exe в ОС Windows).

Система програмування PyCharm

Система програмування **PyCharm** — це комплекс програмних засобів, до якого входять: текстовий редактор для введення й редагування програми, транслятор і налагоджувач та інші компоненти.

PyCharm надає користувачам **інтегроване середовище розробки (ІСР; англ. IDE — *I*ntegrated *D*evelopment *E*nvironment)** із такими функціями, як підсвічування синтаксису та помилок, навігація серед проєктів і в коді програми, відображення файлової структури проєкту, швидкий перехід між файлами, класами, методами тощо.

Програму PyCharm можна завантажити з офіційного сайту <https://www.jetbrains.com/>. Після інсталювання програми на комп'ютер у меню Програми з'являється рядок JetBrains PyCharm Community Edition 2019.1 x64, а на робочому столі — ярлик програми (рис. 24.7).



Рис. 24.7

Створення проєкту в PyCharm

Ознайомлення із середовищем програмування PyCharm розпочнемо зі створення проєкту. Для цього запустимо PyCharm.

Якщо відкритих проєктів немає, то з'явиться вікно привітання (рис. 24.8), в якому пропонуються варіанти входу до ІСР. Нас цікавлять два з них: Create New Project (Створити новий проєкт), Open (Відкрити проєкт).

Що таке проєкт? Програма, яка розробляється в PyCharm, складається з файлів, кожний із яких має певне призначення.



Рис. 24.8

Отже, **проект** — це набір файлів, які створені користувачем або автоматично, необхідних для роботи розроблюваної в ІСР програми.

Виберемо варіант Create New Project і в діалоговому вікні, що відкрилося, введемо назву проекту (рис. 24.9). Буде створено папку проекту, а в ній — папки і файли, необхідні для подальшої його розробки. Після цього відкриється головне вікно середовища розробки.



Рис. 24.9

Інтерфейс середовища розробки PyCharm

Розглянемо елементи головного вікна PyCharm (рис. 24.10).

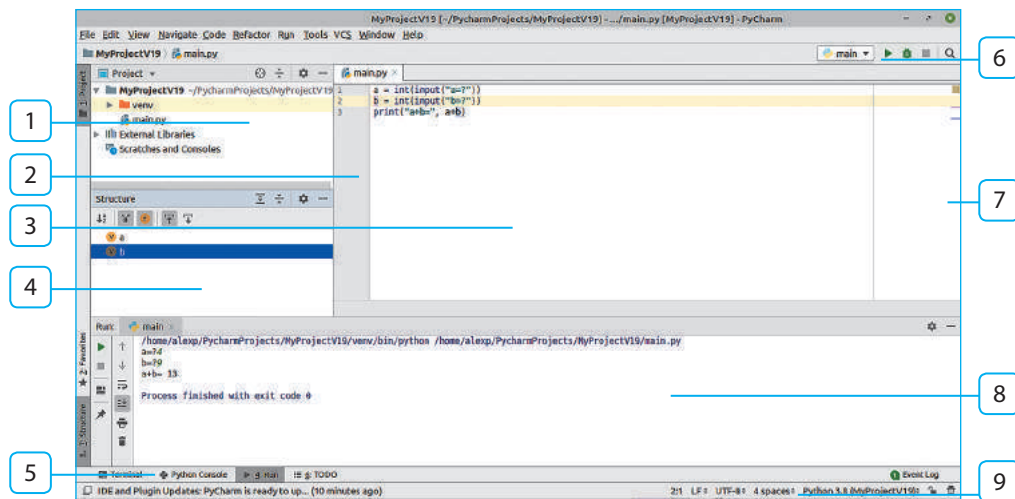


Рис. 24.10

- 1 — панель Project (Проект), яка відображає файли проекту;
- 2 — Left gutter (Ліва смужка) — вертикальна смуга зліва від редактора, що відображає номери рядків коду і точки зупинки;
- 3 — вікно текстового редактора, в якому пишеться код і яке містить вкладки для зручної навігації між відкритими файлами;

4 — панель Structure (Структура), яка містить список змінних і функцій, створених у програмному коді;


5 — вкладки панелей інструментів і кнопка для перемикання між панелями. Якщо навести вказівник на цю кнопку, з'явиться список доступних у цей момент панелей;

6 — Navigation Bar (Панель навігації), яка містить кнопки для запуску на виконання і завершення програми;

7 — Right gutter (Права смужка) — вертикальна смуга праворуч від редактора, що показує результати перевірки програмного коду;

8 — панель виведення для перегляду та введення повідомлень під час виконання програми;

9 — Status Bar (Рядок стану), він відображає деякі параметри ІСР.

Деякі панелі можна приховати, натиснувши кнопку  в їхніх заголовках. А щоб показати приховану панель, слід вибрати її назву в меню View → Tool Windows.

Створення програми

Коли проєкт створено, можна приступати до розробки програмного коду. Варто скористатися одним із таких способів:

- **створити новий файл із розширенням .py**

Для цього потрібно вибрати команду File → New і в переліку об'єктів New — Python file.

Потім у вікні New Python file (Новий файл Python) ввести назву файлу і натиснути кнопку ОК;

- **додати раніше створений файл із кодом**

Для цього слід перетягнути потрібний файл до вікна текстового редактора, натиснути F5 і в діалоговому вікні вибрати папку проєкту для копіювання файлу.

Отже, створимо новий файл і надамо йому ім'я sum.py. У вікні текстового редактора введемо код (рис. 24.11). Під час введення коду звернемо увагу на особливості середовища програмування та ознайомимося з ними.

Щоб запустити програму, в меню Run слід вибрати пункт Run <назва програми> або скористатися сполученням клавіш Shift+F10.

Результати роботи програми виводяться на панелі виведення.

```
a = int(input('a = ?'))
b = int(input('b = ?'))
sum = a+b
print('a+b = ', sum)
```

Рис. 24.11

Якщо потрібно зберегти всі зміни в проєкті, необхідно вибрати команду File → Save All, а якщо треба зберегти файл в іншій папці — вибрати команду File → Save As.

Автозаповнення коду

Ви, мабуть, помітили, що під час введення ключових слів розкривається список властивостей і методів, у назвах яких є набрані літери. ICP пропонує вибрати зі списку назву властивості або методу.

Система «інтелектуального введення» аналізує, що саме ви зараз робите, і вже на основі цього пропонує певні конструкції (рис. 24.12).

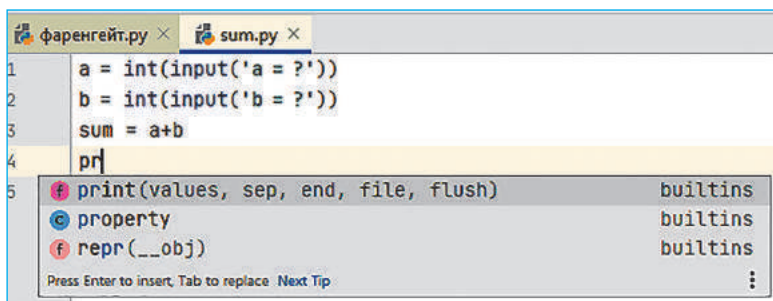



Рис. 24.12

PyCharm виявляє потенційні помилки й проблеми і пропонує швидкі виправлення (quick-fixes) для них. Коли ICP знаходить невикористаний код, нескінченний цикл тощо, з'являється «лампочка» . Потім треба клацнути її або натиснути Alt+Enter, щоб застосувати виправлення.

Отже, навіть перше знайомство з PyCharm дозволяє дійти висновку, що ICP робить розробку програмного коду більш продуктивною завдяки функціям автодоповнення й аналізу коду. Миттєве підсвічування помилок і швидкі виправлення допомагають ефективно редагувати код.

Питання для самоперевірки



1. Пригадайте й опишіть базові алгоритмічні структури, на яких ґрунтується парадигма структурного програмування.
2. Які парадигми програмування підтримує мова Python?
3. Знайдіть в інтернеті інформацію на тему: «Рейтинг мов програмування». Обговоріть у групах, які мови є найпопулярнішими в сучасному професійному програмуванні.

4. Назвіть основні складові головного вікна середовища програмування PyCharm.
5. Як створити та виконати проєкт у середовищі PyCharm?
6. Як переглянути список змінних, створених у програмному коді?

Вправа 24



- ▶ Створити в середовищі програмування PyCharm програму для розв'язування задачі.

Задача. Пиріжок у їдальні коштує a грн і b коп. Визначте, скільки гривень і копійок потрібно заплатити за n пиріжків.

- 1) Завантажте середовище PyCharm. Створіть проєкт у папці Vprava.
- 2) Створіть у проєкті Vprava файл типу Python file із назвою Вправа 24.
- 3) Програма отримує на вхід три цілі невід'ємні числа a , b , n . Потрібно вивести два числа (вартість у гривнях і копійках). У вікні текстового редактора введіть текст програми (рис. 24.13).
- 4) Запустіть програму на виконання. Для цього виберіть команду Run → Run..., а потім — файл Вправа 24. На панелі виведення введіть дані для обчислень (рис. 24.14).

```
n = int(input("Скільки пиріжків?"))
a = int(input("Скільки гривень?"))
b = int(input("Скільки копійок?"))
vart = (a*100+b)*n
grn = vart//100
kop = vart%100
print("Вартість покупки", grn, "грн.", kop, "коп.")
```

Рис. 24.13

```
Run: Vartist x
C:\PycharmProjects\Vprava\venv\Sc
Скільки пиріжків?5
Скільки гривень?3
Скільки копійок?20
Вартість покупки 16 грн. 0 коп.
Process finished with exit code 0
```

Рис. 24.14

- 5) Відкрийте вкладку Structure (Структура) і перегляньте список змінних, що створені в програмі.
- 6) Вставте в операторах введення декілька зайвих пропусків між дужкою та ключовим словом. Дослідіть, як ISCP допомагає зробити код зрозумілішим. Зробіть висновки. Збережіть файл.



Комп'ютерне тестування

Виконайте тестове завдання 24 із автоматичною перевіркою результату.



§ 25. Налagodження програмного коду

Одне із завдань програміста під час розробки програми полягає у відшукуванні й усунуванні помилок у програмному коді. Цей процес називають **налагодженням** програмного коду.

Інтерпретатор Python виявляє два типи помилок — синтаксичні помилки і винятки.

Синтаксичні помилки

Синтаксична помилка виникає, якщо порушено правила запису операторів, наприклад пропущено двокрапку в заголовку циклу. Про синтаксичні помилки ІСР попереджає під час набирання коду: команда, записана з порушенням правил синтаксису, підкреслюється червоною хвилястою лінією.

1 Програміст пропустив лапки в кінці текстової константи. Якщо навести вказівник на підкреслений фрагмент оператора, з'явиться підказка, як виправити помилку (рис. 25.1).

Якщо ж запустити код на виконання, то на панелі повідомлень буде виведено номер рядка коду з помилкою, оператор, запис якого містить помилку, і опис помилки (рис. 25.2).

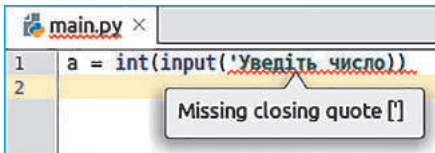


Рис. 25.1

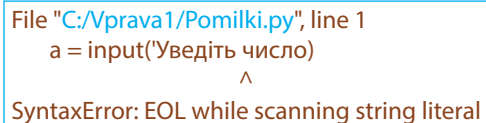


Рис. 25.2

Винятки

Іншим типом помилок є винятки (виняткові ситуації).



Виняток (англ. *exception*) — це помилка, що виникає у процесі виконання програми; вона не є синтаксичною і спричиняє аварійне припинення виконання програми.

Розглянемо причини виникнення винятків на прикладах.

- 2 Виняток `NameError` виникає за спроби використати змінну, якій не присвоєно значення. З точки зору інтерпретатора це означає, що такої змінної не існує (рис. 25.3, а). Назву змінної в тексті програми буде підкреслено червоною хвилястою лінією (рис. 25.3, б).


```
File "C:/vprava1/Pomilki.py", line 2
rez = val1+val2
NameError: name 'val2' is not defined
```

а

```
1 val1 = 5
2 rez = val1+val2
```

б

Рис. 25.3

Якщо встановити курсор на назву змінної, виводиться лампочка . Якщо її клацнути, буде запропоновано варіанти виправлення помилки. На панелі виведення буде виведено тип винятку і повідомлення про те, що назву змінної `b` не визначено.

- 3 `ZeroDivisionError` — виняток, який виникає, якщо в програмі робиться спроба ділення на нуль (рис. 25.4).

```
File "C:/vprava1/Pomilki.py", line 3
c = a/b
ZeroDivisionError: division by zero
```

```
Pomilki.py ×
1 a = 5
2 b = 0
3 c = a/b
```

Рис. 25.4

- 4 За спроби виконати оператор `a = int("Сонце")` виникає виняток `ValueError` — помилка значення: рядок "Сонце" не можна перетворити на ціле число.
- 5 За спроби виконати оператор `a = 10+"3"` виникає виняток `TypeError` — помилка типу. Число 10 і рядок "3" належать до різних типів даних, над якими не можна виконати операцію додавання.

Налагоджувач

Найскладніше виявити та виправити помилки, яких припустилися під час розробки алгоритму. Такі помилки називають **логічними**. Якщо, маючи на меті збільшити значення змінної `Z` на 1, написати `Z = Z+2`, то програма буде успішно опрацьована інтерпретатором. Проте результат її виконання виявиться хибним.

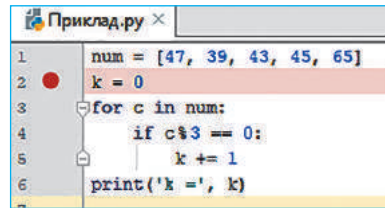
Відшукати логічні помилки в програмі вручну буває складно. Спрощує пошуки програма-налагоджувач (англ. — *debugger*). PyCharm має вбудований налагоджувач PyDev.Debugger.

Налагоджувач дозволяє встановити на певні рядки програми **точки зупинки** (англ. — *breakpoint*). При досягненні точки зупинки виконання програми зупиняється, а програміст може:

- вивчити стан програми в цей момент (зокрема, значення змінних);
- керувати подальшим виконанням програми по одній команді, слідкуючи, як змінюються значення змінних.

Щоб **створити точку зупинки**, слід клацнути на лівій смужці праворуч від номера рядка коду, на якому потрібно зупинити програму. Щоб **почати налагодження**, слід вибрати пункт Debug у меню Run.

6 Учень склав програму для підрахунку в списку чисел, які не діляться на 3 (рис. 25.5), але отримав неправильний результат. Скориставшись налагоджувачем, він поставив точку зупинки на другому рядку, щоб дослідити роботу програми після введення даних.

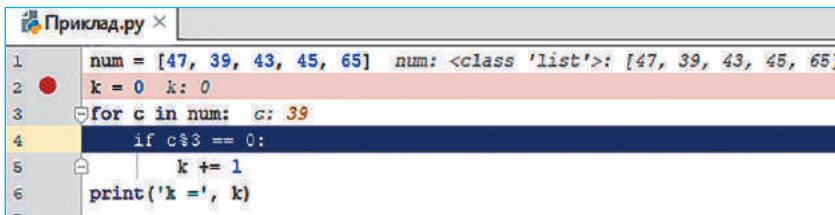


```

Приклад.ру ×
1 num = [47, 39, 43, 45, 65]
2 k = 0
3 for c in num:
4     if c%3 == 0:
5         k += 1
6     print('k =', k)
  
```

Рис. 25.5


Після запуску налагоджувача відкривається панель Console (Консоль), а після досягнення точки зупинки виконання призупиняється. Відкривається вкладка Debugger (Налагоджувач), у виконаних рядках із оголошеннями змінних з'являються їхні поточні значення.



```

Приклад.ру ×
1 num = [47, 39, 43, 45, 65] num: <class 'list': [47, 39, 43, 45, 65]
2 k = 0 k: 0
3 for c in num: c: 39
4 if c%3 == 0:
5     k += 1
6 print('k =', k)
  
```

Рис. 25.6

7 Після зупинки в другому рядку учень почав натискати кнопку  і спостерігати за значеннями змінних. При $c = 47$ змінна-лічильник k не збільшилась, хоча мала б, адже 47 на 3 не ділиться. Помилку виявлено в умові оператора `if c%3 == 0`. При зміні умови на протилежну (`if c%3 != 0`), програма запрацювала правильно.

Розглянемо панель виведення в режимі налагодження (рис. 25.7).

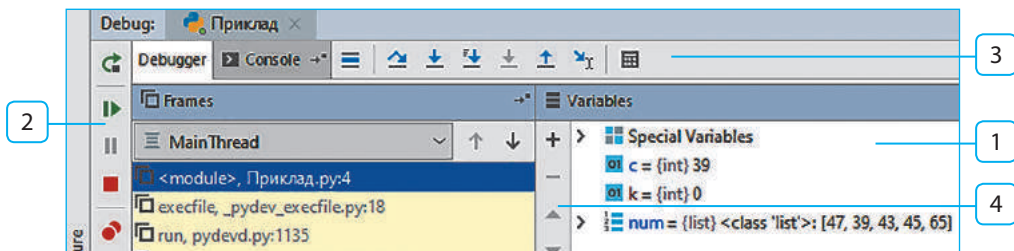


Рис. 25.7

- 1 — вікно Variables (Змінні) для спостереження за значеннями змінних;
- 2 — кнопки для загального керування (перезапустити проєкт, завершити виконання, нехтувати точки зупинки тощо);
- 3 — панель для перемикання між вкладками Debugger і Console; містить кнопки керування покроковим режимом виконання програми;
- 4 — містить кнопки для керування виведенням інформації у вікні Variables.

Щоб додати інформацію про змінну, слід натиснути **+** і вписати назву змінної, значення якої потрібно відстежувати. Для видалення змінної зі списку використовується кнопка **-**.

Питання для самоперевірки



1. Поясніть помилки:
а) `>>> 2a = 10;` б) `>>> print(a + b;` в) `>>> 8 + "3"`
2. Програма для розрахунку кількості тракторів, потрібних для обробки поля, вивела результат: 2,4 трактора. Визначте, до якого виду належить помилка.
3. Поясніть порядок використання точок зупинки.
4. Дослідіть, як ІСР повідомляє користувача про помилки. Для цього змініть запис деяких операторів у файлі Вправа 21, наприклад: видаліть знак арифметичної операції «*»; видаліть лапки, що закривають текстову константу, тощо.
5. Норма площі озеленення міст, встановлена Всесвітньою організацією охорони здоров'я, дорівнює 21 м² зелених насаджень на одного жителя. Напишіть програму для визначення площі зелених насаджень, потрібної вашому населеному пункту.

Вправа 25



- Виконати налагодження програми для розв'язування задачі.
Задача. Відомі назва планети, радіус її орбіти, швидкість руху по орбіті. Обчислити тривалість року на планеті (період її обертання навколо зорі). Перевірити роботу програми.
 Заповнити таблицю:

Назва планети	Радіус орбіти, км	Швидкість руху, км/год	Період обертання, доби
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	

- 1) Завантажте середовище PyCharm. Створіть файл типу Python file із назвою Вправа 25.
- 2) Уведіть текст програми у вікні текстового редактора (програмний код містить помилки!):


```
Nazva = int(input("Назва планети?"))
R = int(input("Радіус орбіти?"))
V = input("Швидкість руху? ")
Period = 2*3.14*R/V*24
Print Period
Print ('Рік на планеті ', 'Nazva', ' триває ', round(Period, 3), ' діб');
```
- 3) виправте синтаксичні помилки та перевірте роботу програми для даних планети Земля.
- 4) Чи збігається відповідь із вказаною в таблиці? Якщо ні — проаналізуйте вираз, за яким обчислюється значення Period:

$$\text{період обертання} = \frac{\text{довжина орбіти}}{\text{швидкість руху} \cdot 24}$$
- 5) Виконайте обчислення для Венери, Нептуна, Марса.
- 6) Змініть програму так, щоб результат обчислень виводився в земних роках.



Комп'ютерне тестування

Виконайте тестове завдання 25 із автоматичною перевіркою результату.



§ 26. Об'єкти, їх властивості та методи

Мова Python є об'єктно-орієнтованою, тобто кожна величина є об'єктом певного класу. Окрім вбудованих класів (типів даних), програміст може описувати і використовувати в програмі власні класи. Добре спроектовані класи — це «будівельні блоки», з яких легше будувати складні програми.

Клас — це опис об'єктів певного типу, **об'єкт** — це екземпляр деякого класу. Опис класу містить **атрибути (змінні)**, які відповідають властивостям об'єктів, і описи **методів класу** — дій, які можуть виконувати об'єкти цього класу.

1 Якщо опис класу `Animal` (Тварина) містить атрибут `маса`, то кожному екземпляру цього класу можна надати певне значення маси.

Опис класів і створення об'єктів

Згадаємо, як описують класи в програмі. Можна вважати, що клас — це своєрідна інструкція зі створення екземплярів. Об'єкт, створений на основі класу, називають екземпляром цього класу.

Опис класу зручно розташовувати на початку коду програми.

Синтаксис опису класу:

```
class <назва класу>(<базовий клас>):
    <атрибут класу> = <значення>
    def __init__(self, <інші параметри>):
        self.<атрибут екземпляра> = значення
```

Атрибути класу мають однакове значення для всіх екземплярів класу, тоді як атрибути екземплярів — окреме значення для кожного екземпляра.

Метод `__init__` називається **конструктором класу**. За наявності, він автоматично виконується під час створення кожного нового екземпляра класу для початкового налаштування властивостей об'єкта. Перший із параметрів в описі конструктора (здебільшого його позначають `self`) зберігатиме посилання на створений об'єкт.

Синтаксис команди створення об'єкта:

```
[змінна = ] <назва класу>([<перелік значень параметрів>])
```

Доступ до створеного об'єкта можна отримати через змінну. Якщо такий доступ не потрібен, назву змінної не вказують.

2 Опишемо клас, що моделює транспортний засіб (рис. 26.1).

```
class Transport():
    def __init__(self, type_vehicle, motor):
        self.type_vehicle = type_vehicle # Тип транспортного засобу
        self.motor = motor               # Тип двигуна
tr1 = Transport('вантажівка', 'дизельний') # Створення екземпляра класу
```

Рис. 26.1

В останньому рядку коду викликається конструктор класу `Transport` і створюється екземпляр класу, відповідні атрибути якого мають значення `вантажівка` і `дизельний`. Посилання на цей об'єкт зберігається в змінну `tr1`.

Методи класу

Під час виклику методів можуть змінюватися властивості (значення атрибутів) об'єкта, а також виконуватися інші дії.

Синтаксис заголовка методу класу:

```
def <назва методу>(self[, параметри]):
```

Код методу виконується у відповідь на виклик методу для конкретного об'єкта.

Виклик методу для об'єкта має такий **синтаксис**:

```
<об'єкт>.<метод>([значення параметрів])
```

У разі виклику методу до нього передається посилання на той об'єкт, для якого викликається метод (параметр `self`).

3 Додамо до опису класу `Transport()` опис методу `fuel()` для виведення повідомлення про вид пального (рис. 26.2):

```
class Transport():
    <...>
    def fuel(self): # Визначення виду пального
        print(self.type_vehicle, ':', self.motor, 'двигун')
tr1 = Transport('вантажівка', 'дизельний')
tr1.fuel()
```

Рис. 26.2

Буде надруковано: вантажівка: дизельний двигун.

Створення класів-нащадків базового класу

Об'єктно-орієнтоване програмування дозволяє прискорити розробку програми шляхом створення нових класів на основі створених раніше.



Успадкування — це можливість створення класу-нащадка на основі наявного (базового) класу.

- 4 Транспортний засіб має такі властивості: тип засобу, тип двигуна, вартість. Автобус, крім цих властивостей, має ще такі: кількість місць, призначення, маршрут. З точки зору успадкування клас Автобус є нащадком класу Транспортний засіб.

Клас-нащадок містить усі атрибути та методи базового класу, проте його можна розширити, додавши нові. У класі-нащадку можна не описувати атрибути та методи, успадковані від базового класу.

- 5 Створимо на основі базового класу Transport два класи-нащадки: Truck і Bus, що мають власні методи (рис. 26.3).

```

transport.py ×
7 class Truck(Transport):
8     def cargo(self, crg):
9         print(self.type_vehicle, ' перевозить', crg)
10 class Bus(Transport):
11     def passenger(self, destination):
12         print(self.type_vehicle, ' везе учнів', destination)
13 tr1 = Truck('вантажівка', 'дизельний')
14 tr2 = Truck('автоцистерна', 'бензиновий')
15 tr1.cargo('гравій')
16 tr2.cargo('воду')
17 tr3 = Bus('міжміський', 'дизельний')
18 tr3.passenger('на екскурсію')
  
```

Рис. 26.3

Використання панелі Структура

У панелі Структура відображається структура опису класів і перелік змінних екземплярів класів. Зверніть увагу на позначки заголовків класів **C**, методів **m**, атрибутів **f** і екземплярів класів **V** (рис. 26.4).

Якщо набрати назву екземпляра класу, автодоповнення запропонує перелік методів, які можна викликати для цього об'єкта із зазначенням класу, в якому метод описано (рис. 26.5).

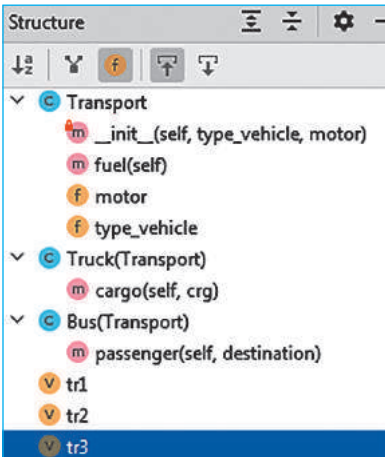


Рис. 26.4

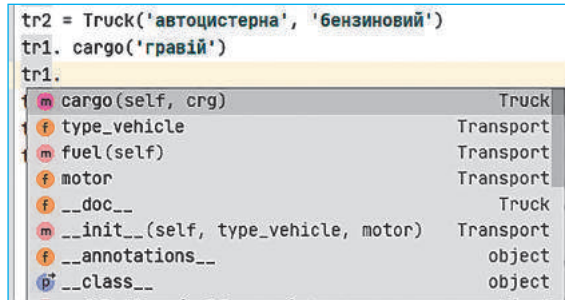


Рис. 26.5

Отже, програміст може описати свій тип даних (клас), визначити в класі певні методи. Разом із тим використання готових класів, описаних у файлах модулів (наприклад, у модулі tkinter або easogui) значно прискорює розробку програми.

Питання для самоперевірки



1. Поясніть поняття класу, об'єкта, атрибута класу, методу класу.
2. Поясніть сутність успадкування.
3. Створіть клас Element — модель хімічного елемента. У конструкторі класу `__init__` опишіть атрибути `name`, `symbol` і `number`. Створіть екземпляр `elem` класу Element зі значеннями атрибутів 'Augum', 'Au', 79. Виведіть значення атрибутів об'єкта `elem`.
4. Створіть модель країни — клас Country з атрибутами `name` (назва країни), `currency` (національна валюта). Опишіть метод класу `print_currency`, який виводить значення атрибутів об'єкта.
5. Створіть екземпляр `my_country` класу Country зі значеннями атрибутів 'Україна', 'гривня'. Викличте для об'єкта `my_country` метод `print_currency`.
6. Створіть екземпляри класу Country, що описують Польщу, Німеччину, Мексику, Австралію, Танзанію.

Вправа 26



- ▶ Створити програму, що моделює облік користувачів на сайті.
- 1) Завантажте середовище PyCharm. Створіть файл типу Python file із назвою Вправа 26.
 - 2) Створіть клас із назвою User і атрибутом `privileges = []` (порожній список; буде використано в класах-нащадках). Конструктор класу має містити атрибути `first_name`, `last_name`, `age`. Опишіть методи класу User:
 - `describe_user`, який виводить повне ім'я користувача;
 - `greeting_user()` для виведення вітання для користувача;
 - `show_privileges()` для виведення списку привілеїв користувача.

```
def show_privileges(self):
    print(self.first_name, self.last_name)
    print(self.privileges)
```
 - 3) Напишіть клас Admin, що успадковує від класу User. У список `privileges` помістіть рядки з описом привілеїв адміністратора (Дозвіл на блокування користувачів, Дозвіл на додавання повідомлень тощо):


```
class Admin(User):
    privileges = ['Дозвіл на блокування користувачів', 'Дозвіл на
                 додавання повідомлень']
```
 - 4) Напишіть клас Visitor, що успадковує від класу User. Задайте привілеї для користувачів цього класу (Дозвіл на перегляд вмісту, Заборона додавання повідомлень тощо).
 - 5) Створіть екземпляр класу User. Викличте для цього об'єкта методи `describe_user()`, `greeting_user()`:


```
user = User('Іванна', 'Репецька', 25)
user.describe_user()
user.greeting_user()
```
 - 6) Створіть екземпляри `admin` класу Admin, `visit` класу Visitor. Викличте для цих об'єктів методи `describe_user()`, `show_privileges()`. Запустіть і випробуйте програму Вправа 26.py.



Комп'ютерне тестування



Виконайте тестове завдання 26 із автоматичною перевіркою результату.



§ 27. Графічний інтерфейс користувача

Більшість сучасних програм використовують графічний інтерфейс, тобто організацію взаємодії людини і комп'ютера з використанням графічних елементів керування — віджетів.

Класи Tkinter

Ви вже знаєте, що для створення графічних програм у мові Python застосовується модуль `tkinter` — бібліотека класів, призначених для побудови інтерфейсу користувача.

Завантаження модуля `tkinter`:

```
import tkinter або from tkinter import*
```

Оскільки модуль `tkinter` містить описи класів графічних компонентів, то кожен елемент керування буде екземпляром одного з цих класів.

► Клас `Tk`

`Tk` є базовим класом будь-якого застосунка на основі `Tkinter`.

Створимо об'єкт `root` класу `Tk` — порожнє вікно програми:

```
root = Tk()
```

Через змінну `root` ми можемо керувати атрибутами вікна.

Завершити код графічної програми потрібно командою запуску головного циклу обробки подій:

```
root.mainloop()
```

Отже, мінімальний обсяг коду застосунку є таким:

```
from tkinter import*
```

```
root = Tk()
```

```
root.mainloop()
```

Далі у вікно додаються інші елементи керування.

1 Створимо порожнє графічне вікно:

```
from tkinter import*
```

```
root = Tk()
```

```
root.title("Графічна програма") # Встановити заголовок вікна
```

```
root.geometry("250x150") # Встановити розміри вікна (в пікселях)
```

```
root.mainloop()
```

Загальні властивості віджетів

Властивості елемента керування визначають, як він виглядає на екрані (наприклад, ширину і висоту, колір тощо). Початкові значення властивостей встановлюються під час створення об'єкта.

Віджети створюються викликом конструктора відповідного класу:
`[<змінна> =]<клас>([<батьківський_віджет>,][<атрибут> = <значення>])`



Батьківський віджет — це елемент керування, до якого буде поміщено створений віджет.

Якщо батьківський віджет не задано, то віджет буде розміщено в головному вікні. Далі можуть задаватися значення атрибутів.

Більшість віджетів у Tkinter мають такі властивості, як висота (`height`), ширина (`width`), шрифт (`font`), колір віджета (`bg` — від англ. *background*), колір шрифту (`fg` — від англ. *foreground*) тощо.

Для задання розташування віджета у вікні використовуються відомі вам методи:

- `place(x, y)`, де параметри `x` і `y` задають зміщення елемента по горизонталі й вертикалі відносно верхнього лівого кута вікна;
- `pack()`, виклик якого розташовує віджети щільно один біля одного.

Про ще один метод для розташування віджетів — `grid()` — ви дізнаєтесь пізніше. А зараз розглянемо створення елемента керування за його допомогою на прикладі напису — об'єкта класу `Label`.

Напис містить рядок (або декілька рядків) тексту й застосовується для виведення заголовків, підписів інших елементів інтерфейсу тощо. Отже, головна властивість об'єкта класу `Label` — `text`.

2 Додамо у вікно напис `Hello World!`.

```
lab = Label(root, text = 'Hello World!', font = 'Arial 18', bg = 'blue', fg = 'yellow')
lab.place(x = 50, y = 30)
```

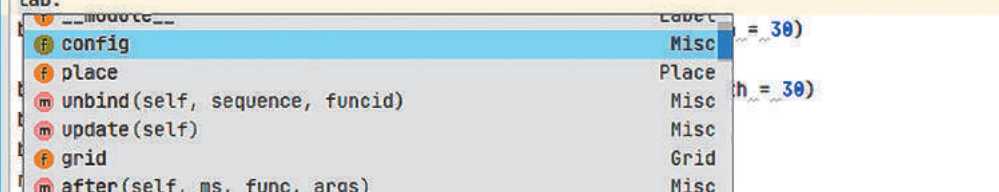
Методи віджетів

Класи графічних компонентів мають методи, які дозволяють змінювати властивості об'єктів. У `PyCharm` назву методу можна або ввести з клавіатури, або вибрати зі списку, який відкривається.

Щоб скористатися списком, слід:

- 1) записати назву об'єкта і поставити крапку;
- 2) дочекатись появи списку і клацнути назву методу (рис. 27.1).

```
lab = Label(root, text = 'Hello World!', font = 'Arial 18', bg = 'blue', fg = 'yellow')
lab.place(x = 5, y = 5)
lab.
```



__module__	Label	
config	Misc	width = 30)
place	Place	
unbind(self, sequence, funcid)	Misc	width = 30)
update(self)	Misc	
grid	Grid	
after(self, ms, func, args)	Misc	

Рис. 27.1

Кожен клас графічних компонентів має свій набір методів. Разом із тим деякі методи, наприклад такі як pack(), place(), config(), destroy(), є спільними для всіх віджетів.

Конфігурація віджетів задається при створенні, проте й під час виконання програми може виникнути потреба змінити властивості віджета. Це можна зробити, викликавши метод config(). У дужках при цьому зазначають, яким саме атрибутам які значення слід присвоїти.

- 3 Змінимо текст та колір тла напису lab:
lab.config(text = 'Новий рядок', bg = 'red')

Метод destroy() викликається для усунення віджета.

- 4 Видалимо об'єкт, на який вказує змінна lab:
lab.destroy()

З'ясувати, чи видимий об'єкт у вікні, можна за допомогою виклику методу wininfo_viewable(). Метод повертає 1, якщо об'єкт видно у вікні, і 0, якщо ні.

Питання для самоперевірки



1. Поясніть, у чому полягає об'єктно-орієнтований підхід при розробці програм із графічним інтерфейсом.
2. Опишіть послідовність дій під час створення вікна програми.
3. Як виконується перший рядок програми у прикладі 2?
4. Поясніть, для чого використовується метод config().
5. Змініть у вже створеного віджета класу Label значення атрибута text на 'Нове значення'.
6. Поясніть, як усунути зайвий віджет із вікна.

Вправа 27



▶ Створити програму Годинник.

- 1) Завантажте середовище PyCharm. Створіть Python file із назвою Вправа 27. Створіть вікно програми розмірами 200×100 із заголовком Годинник.
- 2) Розмістіть у вікні віджет lab класу Label. Задайте атрибуту text значення 'Hello World!', атрибутам font, bg, fg — на свій розсуд.
- 3) Додайте команду завантаження модуля datetime для роботи з датою і часом: `from datetime import datetime`
- 4) Опишіть функцію `clock()`, призначену для відображення поточного часу у віджеті `lab`:


```
def clock():
```

```
    now = datetime.now()
```

```
    lab.config(text=str(now.hour)+':'+str(now.minute)+':'+str(now.second))
```

```
    root.after(1000, clock) # Повторний виклик функції clock() через 1 с
```

Змінна `now` після виклику методу `datetime.now()` вказує на об'єкт з атрибутами `hour` (години), `minute` (хвилини), `second` (секунди), значення яких відповідають поточному часу. У тілі функції `clock()` заголовок напису змінюється рядком, утвореним із цих значень.

В основній програмі запишіть команду виклику функції `clock`: `clock()`.

Запустіть програму на виконання, переконайтеся, що годинник працює (рис. 27.2).

- 5) Запрограмуйте зміну кольору фону вікна щосекунди. До тіла функції `clock()` додайте оператори:

```
color = ['red', 'green', 'yellow', 'white', 'grey'] # Список кольорів
```

```
root.config(bg = color[now.second%5]) # Зміна властивості bg
```

```
# об'єкта root
```

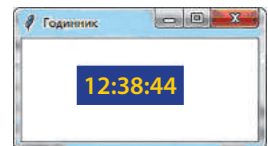


Рис. 27.2



- 6) Додайте до вікна об'єкт `lab1` класу `Label`. Запрограмуйте виведення на нього назви кольору фону вікна. Збережіть проект. Випробуйте програму.

Комп'ютерне тестування

Виконайте тестове завдання 27 із автоматичною перевіркою результату.



§ 28. Обробники подій

Основна відмінність програм із графічним інтерфейсом користувача — інтерактивність. Хід виконання такої програми залежить від втручання користувача, який постійно взаємодіє з програмою (натискає кнопки, вводить дані тощо, тобто створює події).



Подія — це зовнішній вплив на об'єкт, на який цей об'єкт може реагувати певним чином.

Для того щоб об'єкт міг реагувати на певну подію, слід створити **обробник події** — блок програмного коду, який виконується в разі настання події, з якою він пов'язаний.

Створення обробників подій

У мові Python обробником події є певна функція.

Функція — іменований блок коду. Кожна функція має назву, тіло і може мати перелік параметрів. У Python функції визначаються за допомогою зарезервованого слова `def`.

Як ви вже знаєте, **синтаксис опису функції** такий:

```
def <назва_функції>([перелік параметрів])
    <тіло функції>
```

Тіло функції являє собою послідовність операторів, які будуть виконані після викликання функції.

Згадаймо, як створити обробник події для віджета класу `Button` — командної кнопки. Командна кнопка застосовується для того, щоб запустити або завершити певний процес опрацювання даних.

Щоб **прикріпити до віджета обробник події Натискання на кнопку**, необхідно під час створення цього об'єкта в переліку параметрів присвоїти параметру `command` посилання на функцію, яка виконуватиметься в разі натискання (рис. 28.1).

```
def btn_click():
    ...
btn = Button(root, text = 'ok', command = btn_click)
```

Рис. 28.1

- 1 Створимо кнопку із заголовком Натисни мене!, яка виводить повідомлення Клік! (рис. 28.2). У цьому коді функція `button_clicked` викликається щоразу, коли користувач натискає кнопку `button1`:

```
from tkinter import*
def button_clicked():
    print("Клік!")
root = Tk()
button1 = Button(root, bg = 'red',
                 text = 'Натисни мене!',
                 command = button_clicked)
button1.pack()
root.mainloop()
```

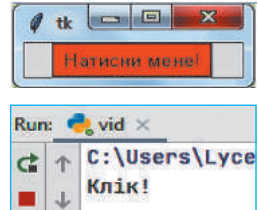


Рис. 28.2

Як ви знаєте, обробником події може бути один із методів класу.

- 2 Створимо клас `SmartButton` — кнопку, яка підраховує кількість натискань і виводить результат у заголовку форми, на якій вона розміщена. Як базовий використаємо знайомий вам клас `Button`:

```
from tkinter import*
class SmartButton(Button):
    def __init__(self, text, width, height):
        super().__init__(text = text, width = width, height = height,
                        command = self.btn_click)
        self.Clicks = 0 # Початкове значення лічильника натискань
    def btn_click(self):
        self.Clicks += 1 # Збільшення лічильника натискань
        self.master.title('Натиснуто '+str(self.Clicks))
```

Як бачимо, клас `SmartButton` має конструктор `__init__`, у першому рядку якого за допомогою спеціального об'єкта `super()` викликається конструктор базового класу (`super().__init__`). Це потрібно для того, щоб повністю налаштувати кнопку `Button`, яка буде основою для `SmartButton`. Як обробник події їй передається метод `self.btn_click`.

В останньому рядку методу-обробника `btn_click` кнопка через атрибут `master` звертається до форми, яка містить цю кнопку, щоб змінити її заголовок (`title`). Тепер можна скласти програму з використанням класу `SmartButton`:

```
root = Tk()
SmartButton(text = "Натискай!", width = 25, height = 3).pack()
root.mainloop()
```

Програмування реакції об'єктів на інші події

Можна зробити так, щоб об'єкт змінював свої властивості у відповідь на натискання певної клавіші клавіатури або миші.

Віджет із подією та функцією-обробником пов'язують викликом методу `bind()` (рис. 28.3):

```
<віджет>.bind(<подія>, <функція>),
```

де `<подія>` — рядок із системною назвою події; `<функція>` — назва функції-обробника.

У такий спосіб один віджет можна пов'язати з кількома подіями.

Який із віджетів має відреагувати на натискання клавіші, якщо у вікні їх декілька? Щоб це задати, для потрібного віджета слід викликати метод `focus_set()`. Дізнатися ж, який із віджетів має фокус, можна, викликавши для будь-якого з них метод `focus_get()`.

Якщо подію пов'язати з певним вікном, то серед віджетів, розташованих у вікні віджетів, на неї реагуватиме той, який має фокус.

3

Запрограмуємо віджети у вікні `root` так, щоб той, на якому щойно клацнули, при натисканні клавіші зі стрілкою `→` рухався вправо:

```
def move_wid_R(event):
    event.widget.place(x = event.widget.winfo_x()+5)
def focus_widget(event):
    event.widget.focus_set()
root.bind("<Button-1>", focus_widget)
root.bind('<KeyPress-Right>', move_wid_R)
... <Тут додаємо віджети>
```

Тут в обробниках подій з об'єкта `event` за допомогою атрибута `widget` здійснюється *доступ до віджета, для якого обробляється подія*. Методи `winfo_x()` і `winfo_y()` цього об'єкта повертають його координати `x` і `y` відносно вікна.

»

Макс Левчин (рис. 28.4) — американський підприємець українського походження, співзасновник PayPal і автор низки стартапів. Він вважає, що вивчення програмування стане величезною ракеткою-носієм для вашого майбутнього незалежно від професійних планів.



Рис. 28.4

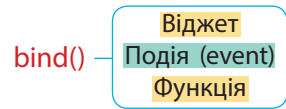


Рис. 28.3

Ознайомимося з формами назв подій.

- Один символ — натискання клавіші клавіатури:

```
lab.bind('z', action)
```

Функцію action буде викликано кожного разу, коли віджет lab у фокусі і натиснуто клавішу z.

- '<Return>' — натискання клавіші Enter.
- '<KeyPress-Up>', '<KeyPress-Down>', '<KeyPress-Left>', '<KeyPress-Right>' — натискання клавіш керування курсором.

Функція, яку буде викликано в разі настання події, повинна приймати один аргумент. Це об'єкт класу Event, що має, серед інших, атрибути, які стануть вам у нагоді при створенні програм — наприклад, щоб змусити об'єкти рухатися по вікну:

- x і y — координати вказівника миші на віджеті;
- char — набраний на клавіатурі символ;
- keysum_num — код набраного на клавіатурі символу.

Питання для самоперевірки



1. Визначте, як виглядає кнопка, створена з такими значеннями атрибутів:

```
btn = Button(root, text = 'Натисни мене', width = 20, height = 2,
             bg = 'blue', fg = 'white')
```
2. Опишіть алгоритм розробки програми з графічним інтерфейсом на Python.
3. У коді конструктора класу поясніть дію команди:

```
super().__init__(width = width, height = height).
```
4. Обговоріть відмінності у застосуванні обробника натискання на кнопку та обробника натискання зазначеної клавіші клавіатури або миші.
5. Запишіть оператор виклику для об'єкта lab методу bind() для обробки натискання клавіші зі стрілкою ↓.
6. Опишіть функцію — обробник події Натискання на літеру A, яка пересуває об'єкт, що викликав обробник, на 5 пікселів угору.

Вправа 28



- ▶ Створити програму з елементами керування, властивості яких змінюються під час виконання програми.
- 1) Завантажте середовище розробки PyCharm.

Створіть Python file із назвою Цікаві кнопки. Створіть вікно програми розмірами 250×200 із заголовком Цікаві кнопки (рис. 28.5).

Завершіть програму оператором `root.mainloop()`.

- 2) Розмістіть вгорі вікна кнопку із заголовком Червоний колір, яка займає всю ширину вікна.

Для цього викличте пакувальник `pack()` з параметрами `side = TOP`, `fill = X`:

```
b_red = Button(text = 'Червоний колір', command = button_red)
b_red.pack(side = TOP, fill = X)
```

Аналогічно створіть кнопки із заголовками Синій колір, Збільшити шрифт, Зменшити шрифт, Привітатися.

- 3) Додайте внизу вікна віджет `label_text` класу `Label`.

При створенні об'єкта додайте у виклик конструктора класу задання значень атрибутів `fg`, `bg` (колір фону, колір шрифту).
`label_text = Label(text = 'Hello, World!', font = 'Arial 12')`

```
label_text.pack(side = BOTTOM, fill = X)
```

- 4) Створіть обробник події Натискання на кнопку для об'єкта `b_red`:
`def button_red():`

```
    root.config(bg = 'red')
```

За зразком створіть обробник події для кнопки Синій колір.

- 5) Створіть обробник події для кнопки Збільшити шрифт:

```
def button_plus():
```

```
    label_text.config(font = 'Arial 18')
```

За зразком створіть обробник події для кнопки Зменшити шрифт.

- 6) Створіть обробник події для кнопки Привітатися, який змінює заголовок напису на Привіт від Python!.

Запустіть програму, перевірте роботу кнопок. Змініть розміри вікна, поясніть поведінку кнопок і напису.

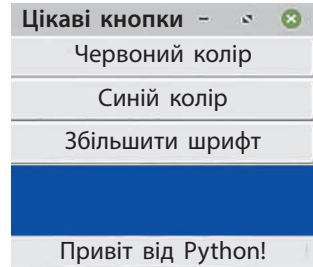


Рис. 28.5



Комп'ютерне тестування

Виконайте тестове завдання 28 із автоматичною перевіркою результату.



§ 29. Організація діалогу користувача з програмою

Більшість задач із програмування передбачають введення даних користувачем, після чого над ними виконуються деякі операції. Потім результат опрацювання виводиться у заданому вигляді (рис. 29.1).

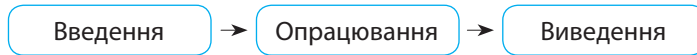


Рис. 29.1

Введення та виведення даних у консольному режимі

Ви вже користувалися функціями `input()` і `print()` для введення і виведення даних. Функція `input()` призначена для введення даних із клавіатури.

Згадаємо послідовність виконання вказівки введення даних на прикладі введення цілого значення для змінної `x`. Розглянемо рис. 29.2:

- 1 — виводиться підказка щодо очікуваного значення;
- 2 — програма призупиняє роботу й очікує введення;
- 3 — уведений рядок перетворюється на ціле число;
- 4 — отримане значення присвоюється змінній `x`.

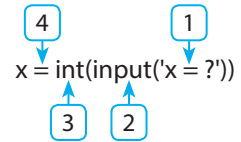


Рис. 29.2

! Функція `input()` повертає об'єкт рядкового типу.

Щоб **увести число**, слід до результату введення застосовувати функцію `int()` або `float()`, яка за рядковим поданням числа створює об'єкт відповідного числового типу (ціле або дробове число). Якщо потрібно **увести декілька значень** в одному рядку, можна скористатися функцією `map()`.

- 1 Введемо значення для змінних `a`, `b`, `c` одним оператором:


```
a, b, c = map(float, input('a, b, c? ').split())
```

 Проаналізуємо виконання цього оператора.

Функція `split()` перетворює рядок, який повернула функція `input('a, b, c? ')`, на список, використовуючи пропуск як роздільник.

Функція `map()` застосовує функцію `float()` до кожного елемента отриманого списку й повертає послідовність дробових чисел.

Функція `print()` потрібна, щоб вивести текст у вікно консолі:
`print(<список виведення>)`

У списку виведення можуть бути константи, змінні, вирази.

2 Обчислимо та виведемо значення виразу:

```
print('Результат = ', (5-x)/(2*y)+1/x)
```

У програмі з графічним інтерфейсом ми можемо також використовувати функції `input()` і `print()`. Введення та виведення відбуватиметься в панелі виведення головного вікна PyCharm.

Компоненти Tkinter для введення та виведення даних

Результати обчислень, повідомлення програми про стан об'єктів тощо можна вивести за допомогою написів — віджетів класу `Label`. Для введення даних зручно використовувати текстові поля — відомі вам віджети класу `Entry`.

3 Створимо об'єкт `entry1` класу `Entry` шириною 14 знакомиць зі шрифтом `Arial` розміром 18:

```
entry1 = Entry(root, width = 14, font = "Arial 18")
```

Згадаємо основні методи `Entry` на прикладі об'єкта `entry1`:

Метод	Призначення	Приклад виклику
<code>get()</code>	Отримати значення, що міститься в текстовому полі	Змінній <code>a</code> цілого типу присвоїти значення, що міститься в полі <code>entry1</code> : <code>a = int(entry1.get())</code>
<code>insert(index, str)</code>	Вивести в текстове поле рядок, починаючи зі знакомища з номером <code>index</code>	Вивести в поле <code>entry1</code> значення змінної <code>x</code> : <code>entry1.insert(0, x)</code>
<code>delete(first, last)</code>	Вилучити символи, починаючи зі знакомища з номером <code>first</code> до знакомища з номером <code>last</code> (нумерація символів з 0)	Очистити текстове поле перед виведенням. Щоб вилучити весь текст, як другий параметр потрібно указати <code>END</code> : <code>entry1.delete(0, END)</code>

► Метод grid()

Вам вже відомі методи pack() і place(), призначені для розміщування віджетів на формі. Якщо розташування віджетів нагадує таблицю, то швидко і правильно розмістити їх допоможе метод grid(). Тепер пронумеруємо уявні рядки і стовпці (рис. 29.3).

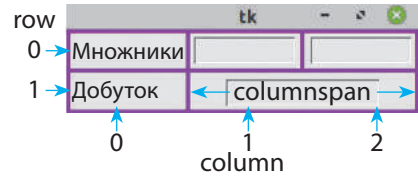


Рис. 29.3

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, sticky = W)
entry3 = Entry()
entry3.grid(row = 1, column = 1, colspan = 2, padx = 5, pady = 5)
```

Атрибути методу grid() мають такий зміст:

- row, column — номери рядка і стовпця відповідно;
- padx, pady — ширина (у пікселях) вільного місця навколо віджета по горизонталі і по вертикалі відповідно;
- sticky — напрямок «прилипання» віджета (можливі значення відповідають сторонам світу: N, S, W, E, NW, NE, SW, SE).
- colspan, rowspan — об'єднання декількох клітинок у рядку або стовпці відповідно.

4

Увести число до поля об'єкта entry1 і вивести квадрат числа до поля об'єкта entry2. Зчитування значення з текстового поля, піднесення до квадрата і виведення результату в поле entry2 виконує функція kvadrat() — обробник події Натискання клавіші Enter:

```
from tkinter import*
def kvadrat(event):
    a = float(entry1.get())
    entry2.delete(0, END)
    entry2.insert(0, str(a**2))
root = Tk()
Label(text = 'a = ').grid(row = 0, column = 0, padx = 8, pady = 8)
entry1 = Entry(width = 8)
entry1.grid(row = 0, column = 1, padx = 8, pady = 8)
```

```

entry1.bind('<Return>', kvadrat)
Label(text = 'a**2 = ').grid(row = 1, column = 0, padx = 8, pady = 8)
entry2 = Entry(width = 8)
entry2.grid(row = 1, column = 1, padx = 8, pady = 8)
root.mainloop()

```

Якщо для полів entry1 і entry2 виклик методу grid записати в одній рядку з викликом конструктора, то у функції kvadrat виникне помилка. Адже об'єкти entry1 і entry2 не будуть екземплярами класу Label.

Діалогові вікна

Щоб згенерувати вікно повідомлення з кнопкою ОК, слід імпортувати модуль messagebox: `from tkinter import messagebox`. Для об'єкта messagebox можна викликати методи `showerror()`, `showinfo()`, `showwarning()`.

Синтаксис виклику методу:

```
messagebox.showinfo(<заголовок>, <текст>)
```

5 Опишемо функцію `button_click()`, яка зчитує значення змінних із текстових полів віджетів `entry1`, `entry2` класу `Entry`.

Функція генерує вікно з повідомленням про результат обчислення частки від ділення a/b (рис. 29.4):

```

from tkinter import messagebox
def button_click(event):
    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))

```

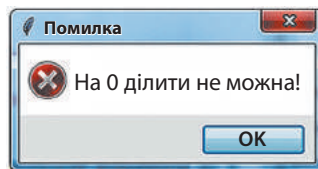


Рис. 29.4

Питання для самоперевірки



1. Поясніть, чому виникає помилка при спробі виконання коду:


```

a = input('Уведіть значення а: ')
cost = a//10 + a%10

```

 Як її виправити?
2. Збережіть ім'я користувача у змінній `name`. Виведіть у консоль рядок, сформований зі слова Привіт і значення змінної `name`.
3. Поясніть, як створити об'єкт класу `Entry`.
4. Як зчитати рядки з поля віджета `Entry` за натисканням `Enter`?
5. Поясніть, як використовується метод `grid()`.

Вправа 29



- ▶ Написати програму для розв'язування задачі.

Задача. Створити програму Збережи дерево (рис. 29.5), яка за масою макулатури визначає, скільки дерев і яку площу лісу її переробка допоможе зберегти від знищення. Відомо, що переробка 60 кг макулатури зберігає 1 дерево і 0,0005 га лісу.

- 1) Завантажте середовище PyCharm. Створіть Python file з назвою Вправа 29. Створіть вікно програми із заголовком Збережи дерево. Завершіть програму оператором `root.mainloop()`.
- 2) Створіть віджети класу `Label` і `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)\
      .grid(row = 1, column = 0, columnspan = 2, padx = 8, pady = 8)
```
- 4) Додайте решту написів та поля `entry1`, `entry2` згідно з рис. 29.5.
- 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) Запустіть програму для різних початкових значень. Проаналізуйте її роботу.

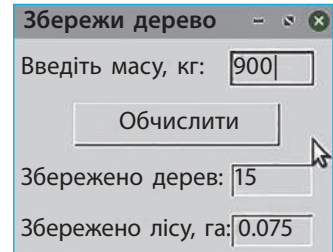


Рис. 29.5



Комп'ютерне тестування

Виконайте тестове завдання 29 із автоматичною перевіркою результату.





Практична робота 12

Введення і виведення даних

Завдання: скласти програму Касир, при реалізації якої комп'ютер буде виконувати такі дії:

- повідомляти вартість одиниці товару (рис. 1);
- запитувати про кількість товару, що купують;
- обчислювати вартість покупки;
- повідомляти вартість покупки;
- запитувати, скільки грошей дає покупець;
- повідомляти решту.



Рис. 1

Обладнання: комп'ютер із середовищем програмування PyCharm.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- Завантажте середовище PyCharm. Створіть Python file із назвою Практична робота 12. Завантажте модулі tkinter, messagebox.
- Створіть вікно програми із заголовком Діалог з касиром.
- Завершіть програму оператором root.mainloop().

- 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)
```

- Додайте ще два написи відповідно до рис. 2.

- Додайте віджети entry1, entry2 класу Entry відповідно до рис. 2.

- Розмістіть у вікні віджет btn класу Button:

```
btn = Button(root, text = 'Сплатити',
command = btn_click)
```

```
btn.grid(row = 3, column = 0, columnspan = 2)
```

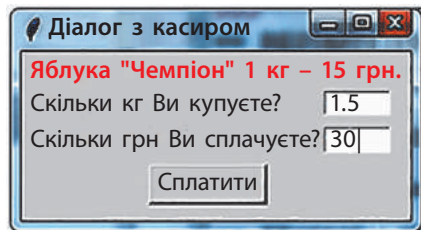


Рис. 2

8. Організуйте діалог із касиром за описом:
- 1) у поля вводяться дані для розрахунку;
 - 2) обчислюється вартість товару і різниця між сумою грошей і вартістю товару;
 - 3) 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. 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 виводилась сума у гривнях і копійках (рис. 3).

12. Помістіть у папку з файлом Касир графічний файл із зображенням яблука. 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.



Рис. 3

Зробіть висновок: як організувати діалог користувача з програмою з використанням модуля `tkinter`.

§ 30. Функції користувача

Нам постійно доводиться оперувати поняттям «функція». Проаналізуємо, які види функцій ми використовуємо при написанні програм.

Пригадаємо, що **функція** — це поймає блок програмного коду, до якого можна звернутися з іншого місця програми. Python надає у використанні вбудовані функції на зразок `len()`, `print()`, `input()` тощо, описані в різних модулях.

Створюючи класи, ви описували **методи класів** — функції, які визначають поведінку екземплярів цих класів. Програмуючи графічний інтерфейс, ви створюєте **обробники подій** — функції, які виконуються внаслідок певних подій із віджетами.

У 6 класі ви ознайомились із методом поділу складної задачі на підзадачі. Алгоритм розв'язування кожної підзадачі можна оформити у вигляді функції, створивши **функції користувача**. Це дає можливість багаторазового використання коду, спрощує його читання, написання, тестування, а також виправлення помилок.

Опис функцій у Python

Пригадаймо синтаксис опису функцій у Python:

```
def <назва_функції>([<перелік формальних параметрів>]):
    <тіло функції>
```

Під час опису функцій слід пам'ятати про кілька правил.

- Заголовок функції складається з ключового слова `def`, назви функції та переліку формальних параметрів, узятих у круглі дужки. Якщо у функцію не треба передавати значення, слід записати порожні дужки.
- Після дужок ставиться двокрапка (`:`) і з нового рядка з відступом починається тіло функції.

Якщо функція не повертає значення на місце виклику або повернуте значення не потрібне, **виклик функції** записують окремим оператором основної програми:

```
<назва_функції>([<перелік фактичних параметрів>])
```



Опис функції має міститися вище від виклику функції.

Коли програма зустрічає виклик функції, керування передається у функцію із зазначеною назвою. Значення *фактичних параметрів* (аргументів) передаються *формальним параметрам*, тому слід дотримуватися відповідності між списками формальних і фактичних параметрів щодо їх кількості, порядку, типів.

Після виконання операторів тіла функції програма продовжить виконуватися з оператора, наступного за оператором виклику функції.

- 1 Прослідкуйте за послідовністю виконання операторів програми (починаючи з рядка 3 на рис. 30.1). Зверніть увагу на значення, які отримують формальні параметри функції x і y .

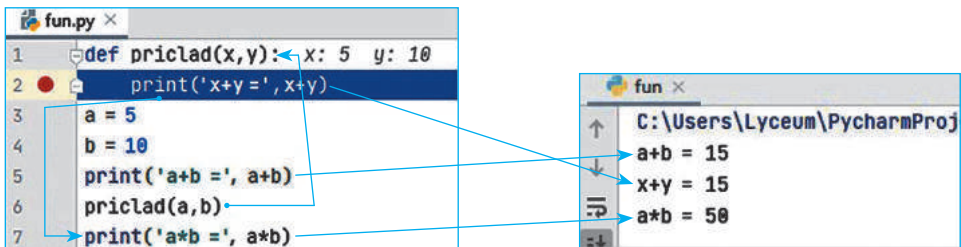


Рис. 30.1

Повернення результату виконання функції

Функція після виклику (рис. 30.2, 1) може повернути на місце виклику обчислене значення. Для цього в тілі функції записують оператор `return`. Він перериває виконання функції і повертає значення виразу (рис. 30.2, 2), яке можна присвоїти змінній, вивести тощо.

Якщо функція не використовує оператор `return` явно, то повертає результат `None`, що означає відсутність значення, тобто *ніщо*, або *порожнє місце*.

- 2 Пригадайте функцію для знаходження найбільшого спільного дільника двох чисел, знайому вам із 6 класу:

```
def nsd(m, n):
    while m != n:
        if m > n: m = m - n
        else: n = n - m
    return m
```

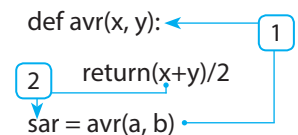


Рис. 30.2

3 Складемо програму для розв'язування задачі.

Задача. Дано дві шестірні, що мають a і b зубців (рис. 30.3). Складемо програму для підрахунку, скільки обертів має здійснити кожна шестірня, щоб вони повернулись у початкове положення.

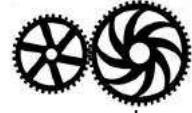


Рис. 30.3

Під час обертання зубці шестерень проходять однакові шляхи. Для повернення в початкове положення потрібно, щоб кожна з них виконала ціле число обертів. Тобто шлях L , пройдений за весь час одним зубцем, має бути спільним кратним кількостей зубців a і b , а кількості обертів — відповідно L/a і L/b . Нам відомо, що $\text{НСК}(a, b) = a \cdot b / \text{НСД}(a, b)$, де НСК — найменше спільне кратне, НСД — найбільший спільний дільник.

Складемо програму з використанням функції з прикладу 2:

```
a, b = map(int, input('a, b? ').split())
nsk = a*b//nsd(a, b)
print('Шестірні зроблять ', nsk//a, ' і ', nsk//b, 'обертів')
```

Результат виконання програми: для $a=18$, $b=24$ виводиться:
Шестірні зроблять 4 і 3 оберти.

Область видимості змінної

При створенні функції слід пам'ятати про області видимості змінних (рис. 30.4).

Тут змінну A оголошено у функції `my_function()`, тому за межами функції вона недоступна. Таку змінну називають *локальною*, і її область видимості показано білим прямокутником.

До змінної C , оголошеної в основній програмі, можна звернутись і в тілі функції `my_function()`, викликаній після оголошення змінної, тобто C для функції є *глобальною* змінною.

Однак, якщо значення глобальної змінної потрібно змінити, її слід додатково оголосити із використанням ключового слова `global`.

```
def my_function():
    A = 5    # Локальна змінна
    global C
    C = C+A

root = Tk()
C = 0      # Глобальна змінна
my_function()
print(C)
```

Рис. 30.4



Область видимості змінної — це та частина програмного коду, в якій змінна доступна для використання.

Отже, функції поділяють програму на частини, кожна з яких виконує певне завдання. Їх можна багаторазово викликати із різних місць програми і переносити в інші програми.

Питання для самоперевірки



- Як передаються значення параметрів при виклику функції?
- Знайдіть і поясніть помилки в програмному коді:

<pre>a) def f1(): a += 1 a = 10 f1() print(a)</pre>	<pre>б) def f2(y): y = y//10 x = 32 a = f2(x) print(a)</pre>	<pre>в) def f3(y): y = y%10 return y x = 25 f3(x) print(x)</pre>
---	--	--
- Що надрукує програма, якщо: а) $x=38, y=4$; б) $x=25, y=10$?


```
def m(a, b):
    return a//b, a%b
x = int(input('x = ?'))
y = int(input('y = ?'))
n, k = m(x, y)
print(n, k)
```
- Поясніть роботу функції `nsd(m, n)` з прикладу 2.
- Як змінити програму з прикладу 3, якщо механізм містить три шестірні (рис. 30.5)?

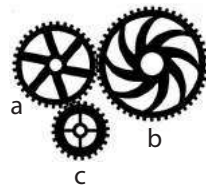


Рис. 30.5

Вправа 30



- Скласти програму для розв'язування задачі.
- Задача.* На стадіоні місця класу *A* коштують *a* грн, місця класу *B* — *b* грн, а місця класу *C* — *c* грн. Обчислити загальну суму прибутку від продажу квитків.
- Завантажте середовище PyCharm. Створіть Python file із назвою Вправа 30.
 - Запишіть оператори для введення значень змінних *a*, *b*, *c*.
 - Опишіть функцію `kvitok(a, b, c)`, яка запитує, скільки продано квитків класу *A*, і обчислює суму отриманого прибутку від продажу квитків цього класу.


```
def kvitok(a, b, c):
    k_a = int(input('Скільки продано квитків класу A? '))
```

```
sum_a = a*k_a
return sum_a
```

- 4) Додайте до опису функції `kvitok(a, b, c)` оператори для введення кількості проданих квитків класів *B* і *C* й обчислення суми, отриманої від продажу квитків кожного класу окремо. Після цього оператор `return` слід змінити так, щоб функція повертала три значення:


```
return sum_a, sum_b, sum_c
```
- 5) В основній програмі обчисліть загальну суму прибутку від продажу квитків:


```
s_a, s_b, s_c = kvitok(a, b, c)
suma = s_a+s_b+s_c
```
- 6) Запишіть оператор виведення значення загальної суми. Випробуйте роботу програми для різних початкових значень.

Комп'ютерне тестування



Виконайте тестове завдання 30 із автоматичною перевіркою результату.



§ 31. Величини. Числові типи даних

Будь-яка комп'ютерна програма оперує певними величинами. Так, у програмі обчислення шляху, подоланого автомобілем за певний час, будуть використані такі величини, як швидкість руху, час, шлях.

Характеристики величини

Величина — це окремий інформаційний об'єкт (число, символ, рядок тощо), який характеризується типом і значенням, розміром пам'яті, потрібної для зберігання, назвою (ідентифікатором).

Правила вибору назв величин такі самі, як і для інших об'єктів. Кожна величина належить до певного типу даних.

Тип даних визначає множину допустимих значень величини й операції, які можна виконувати над цими значеннями.

Ви вже знайомі з такими типами даних, як `int` (цілі числа), `float` (дробові числа), `bool` (логічні значення), `str` (рядки), `list` (списки).

Python визначає тип даних змінної 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.

Визначити тип величини, тобто до якого класу вона належить, можна за допомогою функції `type` (рис. 31.1).

```
>>> a = 5
>>> type(a)
<class 'int'>
```

Рис. 31.1

Об'єкти та змінні

У Python усі величини є об'єктами певного класу. Для доступу до об'єктів створюються змінні.

! Змінна в Python зберігає посилання на об'єкт (іноді кажуть, змінна вказує на об'єкт).

Значення змінної — це характеристика, яка може багаторазово змінюватися в процесі опрацювання інформації. Щоб **створити змінну** в Python, необхідно дати їй назву й присвоїти певне значення (рис. 31.2).

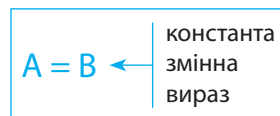


Рис. 31.2

Синтаксис команди присвоєння:

```
<назва змінної> = <константа>|<змінна>|<вираз>
```

Змінну іноді порівнюють зі скринькою, в якій зберігається значення величини. Проте в Python змінні більше схожі на ярлики, що прикріплюються до об'єктів.

1 Створимо змінну `a`: `a = 7`.

Змінна `a` — це посилання на об'єкт, який містить значення 7.

2 Можна створити декілька змінних, що посилаються на один і той самий об'єкт (рис. 31.3):

```
a = b = c = 7
```

Під час присвоєння одній зі змінних іншого значення створюється новий об'єкт, на який вона після цього буде посилатися: `b = 23`.



Рис. 31.3

Опрацювання даних числових типів

Цілі числа належать до класу `int`. У Python вони можуть бути як завгодно великими і обмежуються лише доступною пам'яттю.

Дробове число — величина типу `float` — може мати до 18 значущих цифр. Дізнатися діапазон можливих значень для конкретного комп'ютера допоможе команда `sys.float_info` з модуля `sys`.

Запишемо дробові числа у вигляді, який нагадує стандартний вигляд числа: `<число>e<показник степеня 10>`.

Так для числа задають показник степеня 10, на який воно множить:

<code>x = 2.9e3</code>	<code>x = 2.9e-3</code>
<code>print(x) # 2900.0 = 2,9 · 10³</code>	<code>print(x) # 0.0029 = 2,9 · 10⁻³</code>

Під час розробки програми може виникнути необхідність перетворення значення одного типу даних на значення іншого типу.

Наведемо відомі нам функції для перетворення типів:

- `float()` — створює дробове число з рядка або цілого числа;
- `int()` — створює ціле число з рядка або дробового числа; при перетворенні дробового числа лише відкидається дробова частина;
- `str()` — створює рядок із числа.

Згадаємо математичні оператори Python. Розглянемо таблицю:

Операція	Символ оператора	Приклад	Результат
Додавання	+	<code>a = 15+3</code>	<code>a = 18</code>
Віднімання	-	<code>b = 15-3</code>	<code>b = 12</code>
Множення	*	<code>c = 8*2</code>	<code>c = 16</code>
Ділення	/	<code>d = 7/2</code>	<code>d = 3.5</code>
Обчислення неповної частки від ділення	//	<code>e = 7//2</code>	<code>e = 3</code>
Обчислення остачі від ділення	%	<code>f = 7%2</code>	<code>f = 1</code>
Піднесення до степеня	**	<code>j = 7**2</code>	<code>j = 49</code>

Всі вирази записують в один рядок, а порядок дій, зокрема для дробів, позначають за допомогою дужок.

3 Присвоєння змінній `a` значення виразу $\frac{2x-5}{3+x} + 7x$ записують так:
`a = (2*x-5)/(3+x)+7*x`

Виведення дробових чисел

Для більшого унаочнення даних, які виводяться програмою, за допомогою методу `format()` можна задавати кількість знаків після десяткової крапки:

```
"{:<загальна кількість знаків>.<кількість десяткових знаків>f}".format(вираз)
```

Проаналізуємо **синтаксис форматування значення**:

- у лапках між фігурними дужками стоїть двокрапка, після якої записуються два цілих числа, розділені крапкою, — загальна кількість знаків і кількість знаків у дробовій частині числа;
- літера `f` вказує на тип значення (`float`), що виводиться;
- після лапок ставиться крапка і записується виклик методу `format`, якому передається вираз, значення якого слід відформатувати.

4

Проаналізуємо вигляд числа, що виводиться, за різних значень параметра `<кількість десяткових знаків>`. Загальну кількість знаків можна не вказувати, але тоді одразу після двокрапки слід поставити крапку:

```
number = 23.8589578
```

```
print("{:8.3f}".format(number))          # 23.859
```

```
print("{:.4f}".format(number))          # 23.8590
```

Питання для самоперевірки



- Обговоріть характеристики величин, що відповідають змінним:
 - $a = 12$;
 - $b = \text{True}$;
 - $c = 3.14$;
 - $d = "8.67"$.
- Обчисліть значення виразів:
 - $123//100$;
 - $123\%10$;
 - $123//10\%10$.
- Запишіть наведені вирази за правилами Python:
 - $10^5 d - 4 \frac{3}{7} c^3 + \frac{2}{3}$;
 - $\frac{35x - 4y}{b_1} - \frac{8c + 5}{4b_2} \cdot \frac{b_3}{x + y}$.
- Запишіть оператори присвоєння, які реалізують такі дії:
 - змінній c присвоїти суму значень змінних a і b ;
 - подвоїти значення змінної a ;
 - значення змінної a збільшити на 10.
- Як виконується така команда: `print("{:8.2f}".format(Z))`.
- Напишіть програму Калькулятор відстаней, яка за заданим значенням відстані в метрах виводить значення відстані у дюймах, футах, милях, ярдах тощо з двома десятковими знаками.

Вправа 31



- ▶▶ Напишіть програму Конвертер миль для переведення значення відстані, заданої в кілометрах, у милі.

Миля (від лат. *mille passuum* — тисяча подвійних кроків) — одиниця відстані, введена в Стародавньому Римі. Розрізняють *сухопутну* (1,609 км) і *морську* (1,852 км) милі. В Україні до метричної системи були відомі такі милі: *коротка*, або турецька (1,67 км), та *довга*, або козацька (8,35 км).

- 1) Завантажте PyCharm. Створіть Python file із назвою Вправа 31. Створіть вікно програми із заголовком Конвертер миль.
- 2) Створіть написи (об'єкти класу Label) (рис. 31.4). Віджети розмістіть за допомогою методу `grid()`.
- 3) Створіть об'єкт `entry` класу `Entry` для введення значення відстані в кілометрах:

```
entry = Entry(width = 8)
entry.grid(row = 1, columnspan = 2)
Створіть об'єкти entry1–entry4 класу Entry.
```

- 4) Створіть кнопку класу `Button` із заголовком Конвертувати і функцією-обробником `btn_click`.

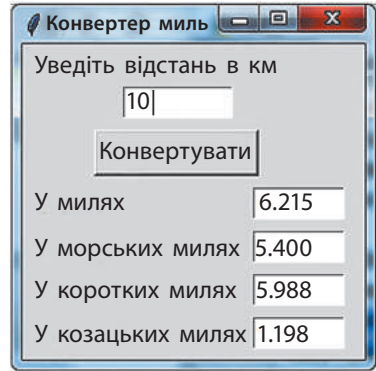


Рис. 31.4



- 5) Опишіть функцію `btn_click()`, у тілі якої з віджета `entry` зчитується значення відстані в кілометрах і зберігається у змінній `km`. Значення відстані у милях виводиться з трьома десятковими знаками до віджета `entry1`:


```
entry1.delete(0, END)
entry1.insert(0, "{:.3f}".format(km/1.609))
```

 Виведіть до віджетів `entry2–entry4` відповідні значення.
- 6) Додайте навколо віджетів вільне місце (5–10 пікселів). Випробуйте програму, переконайтесь у правильності обчислень.

Комп'ютерне тестування



Виконайте тестове завдання 31 із автоматичною перевіркою результату.



§ 32. Математичні функції

У мові Python для розв'язування математичних задач існують вбудовані (стандартні) функції. Їх можна використовувати без підключення модулів. Розглянемо таблицю:

Функція	Що повертає	Приклад	Результат
<code>abs(x)</code>	$ x $ — модуль x	<code>y = abs(-2.5)</code>	<code>y = 2.5</code>
<code>divmod(a, b)</code>	Неповну частку і остачу від ділення a на b	<code>c, d = divmod(7, 3)</code>	<code>c = 2</code> <code>d = 1</code>
<code>round(num[,n])</code>	Число <code>num</code> , округлене до n знаків після коми; якщо аргумент n відсутній, повертає найближче ціле	<code>x = 1.5278</code> <code>y = round(x)</code> <code>z = round(x, 2)</code>	<code>y = 2</code> <code>z = 1.53</code>
<code>eval(s)</code>	Значення арифметичного виразу, запис якого зберігається в рядку <code>s</code>	<code>a = eval("(3+21)/3+4")</code>	<code>a = 12.0</code>

Доступ до математичних функцій надає модуль `math`:

```
from math import*
```

Розглянемо деякі функції модуля `math`:

Функція	Що повертає	Приклад	Результат
<code>sqrt(x)</code>	Квадратний корінь (square root) з x	<code>a = sqrt(81)</code>	<code>a = 9.0</code>
<code>factorial(num)</code>	Факторіал цілого числа <code>num</code>	<code>a = factorial(5)</code>	<code>a = 120</code>
<code>pi</code>	Константа $\pi = 3.1415\dots$	<code>c = 2*pi*5</code>	<code>c = 31.415926\dots</code>

По мірі вивчення математики ви зможете відшукати в модулі `math` знайомі функції.

Випадкові числа

Ви вже застосовували випадкові числа для створення ігрових або тестових програм. Згадаємо: щоб отримати випадкове число, необхідно завантажити модуль `random`:

```
from random import*
```

Щоб отримати випадкове число, слід викликати одну з функцій: `randint(x1, x2)` — ціле з діапазону від $x1$ до $x2$ ($x1 \leq x2$); `random()` — дробове з проміжку від 0.0 до 1.0. Дробові числа з інших діапазонів отримують за допомогою математичних операцій:

```
number1 = 10+random()*50    # Значення від 10.0 до 60.0
print(number1)              # Може бути виведено 52.493371860628294
```

Звичайні дроби

Можливості Python не обмежуються цілими числами та десятковими дробами. Можна запрограмувати дії над звичайними дробами. Щоб почати використовувати ці дроби, слід імпортувати модуль `fractions`:

```
from fractions import*
```

Для створення об'єкта класу `Fraction`, слід передати конструктору класу два цілі числа: значення чисельника та знаменника дроби.

- Створимо об'єкт класу `Fraction`, значення якого дорівнює $\frac{2}{3}$:

```
>>> Fraction(2, 3)
```

! Python не дозволить створити дріб із нульовим знаменником:

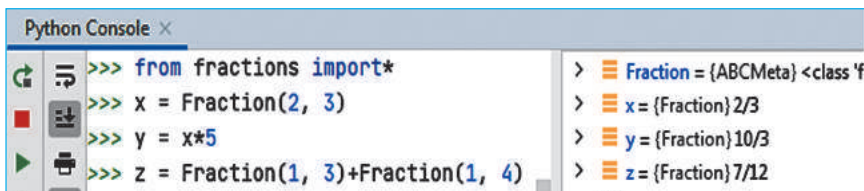
```
>>> Fraction(0, 0)
ZeroDivisionError: Fraction(0, 0)
```

Значення чисельника об'єкта `x` зберігається в атрибуті `x.numerator`, знаменника — в атрибуті `x.denominator`:

```
>>> x = Fraction(5, 7)
>>> x.denominator    # Буде виведено 7
```

Із дробами можна виконувати прості арифметичні дії, в результаті яких створюється об'єкт класу `Fraction`. Дроби в об'єктах класу `Fraction` скорочуються автоматично.

- Проаналізуйте команди для виконання дій над звичайними дробами й отримані значення змінних (рис. 32.1).



```
Python Console x
>>> from fractions import*
>>> x = Fraction(2, 3)
>>> y = x*5
>>> z = Fraction(1, 3)+Fraction(1, 4)
> Fraction = {ABCMeta} <class 'f
> x = {Fraction} 2/3
> y = {Fraction} 10/3
> z = {Fraction} 7/12
```

Рис. 32.1

Перетворити звичайний дріб на десятковий можна, скориставшись функцією перетворення типу `float()`:

```
>>> float(Fraction(1, 4))
0.25
```

Python надає можливість перетворити десятковий дріб на звичайний із використанням методу `from_float()`. Метод приймає число типу `float` і повертає звичайний дріб (об'єкт класу `Fraction`); відношення чисельника до знаменника цього дробу наближене до даного значення.

```
>>> from fractions import*
>>> Fraction.from_float(0.5)
Fraction(1, 2)
>>> Fraction.from_float(0.6)
Fraction(5404319552844595, 9007199254740992)
```

- ! Через особливості двійкового кодування дробових чисел
- `Fraction.from_float(0.6) ≠ Fraction(6, 10)`:

```
>>> Fraction.from_float(0.6) == Fraction(6, 10)
False
```

Таким чином, мова Python забезпечує доступ до найуживаніших математичних функцій, які ви можете використовувати для програмування як завгодно складних математичних обчислень.

Питання для самоперевірки



1. Обчисліть значення, яке повертає функція:

а) <code>abs(-6.9)</code> ;	в) <code>int(6.9)</code> ;	г) <code>round(1.5278, 2)</code> ;
б) <code>divmod(16, 2)</code> ;	г) <code>divmod(16, 3)</code> ;	д) <code>round(6.9)</code> .

 Результат перевірте в консолі Python.
2. Як отримати випадкове дробове число в діапазоні від 0 до 100?
3. Обчисліть значення, яке повертає функція модуля `math`:

а) <code>sqrt(25)</code> ;	б) <code>factorial(6)</code> ;	в) <code>10*pi</code> .
----------------------------	--------------------------------	-------------------------
4. Складіть програму для знаходження значення виразу $\sqrt{|x-1|}$ для введеного з клавіатури значення x .
5. Якого значення набуває змінна x у результаті виконання операторів:

а) <code>x = Fraction(3, 5)</code> ;	в) <code>x = Fraction(2, 5)+Fraction(3, 5)</code> ;
б) <code>x = 2+Fraction(3, 5)</code> ;	г) <code>x = float(Fraction(3, 4))</code> ;
6. Складіть і випробуйте програму для обчислення гіпотенузи та площі прямокутного трикутника з катетами a і b .

Вправа 32



- Складіть програму для розв'язування задачі.
Задача. На святкуванні дня народження Христини Матвій з'їв $\frac{a}{b}$ торта (рис. 32.2), а Остап — $\frac{c}{d}$ торта. Скільки торта з'їла Христина, якщо його маса становить маса кг?



Рис. 32.2

- 1) Завантажте середовище PyCharm. Створіть Python file із назвою Вправа 32. Завантажте модуль fractions:


```
from fractions import*
```
- 2) Запишіть оператори введення з клавіатури значень змінних a , b , c , d , $masa$. Створіть об'єкти x , y класу Fraction:


```
x = Fraction(a, b)
y = Fraction(c, d)
```
- 3) Присвойте змінній z значення частини торта, що залишилась для Христини. Змінній v присвойте значення маси решти торта.
- 4) Перетворіть змінну v на тип float. Виведіть отримане значення з трьома десятковими знаками.
- 5) Запустіть програму з такими даними: Матвій з'їв $\frac{1}{3}$ торта, Остап з'їв $\frac{1}{4}$ торта, а торт важив 1 кг. Скільки з'їла Христина?
- 6) Створіть точки зупинки в рядку коду з оператором print. Виберіть пункт Debug у меню Run. Уведіть значення для змінних a , b , c , d , $masa$. Проаналізуйте значення всіх змінних на вкладці Debugger в розділі Variables.



Комп'ютерне тестування



Виконайте тестове завдання 32 із автоматичною перевіркою результату.



§ 33. Рядковий тип даних

Для опрацювання текстових даних у 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'
```

! Початок і кінець рядка позначають лапками одного виду.

Рядок, який містить апостроф, 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 за допомогою функції `str()` (рис. 33.1).

```
>>> str(1.5e+5)
'150000.0'
>>> str(1/3)
'0.3333333333333333'
>>> str(15)
'15'
```

Рис. 33.1

Доступ до символів у рядку

Для опрацювання всіх символів рядка зручно скористатись циклом `for <змінна> in <рядок>`, під час виконання якого змінна по чергово набуває значень окремих символів рядка.

1 Виведемо символи рядка, введеного з клавіатури, поставивши після кожного з них двокрапку:

```
s = input()
```

```
for C in s: print(C, end = ',')
```

До певного символу в рядку можна звернутися за його індексом (номером позиції в рядку; нумерація починається з 0). Відлік символів можна також вести з кінця рядка, починаючи з `-1` (рис. 33.2).

Крім того, можна **добувати зріз (підрядок)**, тобто копіювати деяку послідовність символів із рядка або частину рядка за певним правилом.

```
>>> s = 'Це рядок'
>>> s[3]
'р'
>>> s[-3]
'д'
```

Рис. 33.2

Оператор добування зрізу з рядка:

$$z = s[ij:step],$$

де z — змінна, в яку зберігається підрядок рядка s ; i — індекс початку зрізу; j — індекс межі кінця зрізу (не входить у зріз); $step$ — крок, з яким вибираються символи.

2 Розглянемо застосування операції добування зрізу на прикладі рядка $s = \text{'Сонечко яскраво сяє'}$.

Приклад	Опис	Результат
$z = s[8:15]$	Зріз від символу з номером 8 рядка s включно до символу з номером 14	$z = \text{'яскраво'}$
$z = s[:7]$	Зріз від початку рядка до символу з номером 6	$z = \text{'Сонечко'}$
$z = s[8:]$	Зріз від символу з номером 8 рядка s до кінця рядка	$z = \text{'яскраво сяє'}$
$z = s[::2]$	Копіювання символів з рядка s із кроком 2	$z = \text{'Снчоякаосє'}$
$z = s[::-1]$	Запис символів рядка s у зворотному порядку	$z = \text{'єяс оваркся окченоС'}$

Дії над рядковими величинами

Згадаємо прийоми опрацювання рядків. Розглянемо таблицю:

Операція	Опис	Приклад
$\text{len}(s)$	Кількість символів у рядку	$k = \text{len}(\text{'Сорока'})$ $\# k = 6$
$\text{subs in } s$	Визначення наявності підрядка subs у рядку s	$a = \text{'y' in 'Python'}$ $\# a = \text{True}$
$s1+s2$	Об'єднання рядків $s1$ і $s2$ в один рядок s (конкатенація)	$s = \text{'Сонце'} + \text{' ' + 'сяє'}$ $\# s = \text{'Сонце сяє'}$
$n*s$	Дублювання рядка s n разів	$s1 = \text{'+'*5}$ $\# s1 = \text{'+++++'}$

Мова Python надає й інші можливості для опрацювання текстів. Відповідні операції доступні у вигляді методів рядкових об'єктів.

У наведених далі прикладах методи викликаються для змінної `s`, яка містить опрацьований рядок. Розглянемо таблицю:

Назва методу	Опис дії
<code>s.find(s1)</code>	Шукає у рядку <code>s</code> задану групу символів і повертає номер першого символу шуканого підрядка в рядку або <code>-1</code> , якщо в <code>s</code> немає підрядка <code>s1</code>
<code>s.replace(sfind, sins)</code>	У рядку <code>s</code> замінює підрядок <code>sfind</code> підрядком <code>sins</code>
<code>s.count(subs)</code>	Повертає кількість входжень підрядка <code>subs</code> у рядок <code>s</code>

Рядки в Python належать до незмінюваних об'єктів. Тобто рядкові методи в Python не змінюють рядка, а утворюють новий. Якщо потрібно далі в програмі використовувати результат виконання методу, його слід зберегти в новій змінній:

```
m = 'шпала'.replace('па', 'ко')
print(m)           # Буде виведено: школа
```

Багаторядкове текстове поле

Багаторядкове текстове поле є об'єктом класу `Text`. За замовчуванням висота створюваного віджета дорівнює 24 знакомісцям, ширина становить 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 літерах.

3 Створимо об'єкт `text1` класу `Text`.

```
text1 = Text(width = 35, height = 6, wrap = WORD)
text1.pack()
```

Якщо в текстове поле вводити більше рядків тексту, ніж висота поля, то його вміст автоматично прокручуватиметься вниз. Під час перегляду текст можна прокручувати вгору й униз за допомогою колеса миші та клавіш керування курсором.

Методи `Text`

Основні методи в `Text` такі самі, як у `Entry`, — `get()`, `insert()`, `delete()`. Проте для багаторядкового поля необхідно зазначити два індекси — номер рядка і номер символу в ньому, відокремлені крапкою.

Нумерація рядків починається з одиниці, а стовпців — з нуля.

4

Виведемо в поле об'єкта `text1` вірш.

Щоб вірш виводився по рядках, слід додати в початок кожного рядка символ перенесення рядка `\n` (рис. 33.3):

```
text1.insert(1.0, 'Вірш про дружбу \nВ житті усім потрібна дружба: \n'  
'І для розваг, і для журби, \nй для служби.')
```

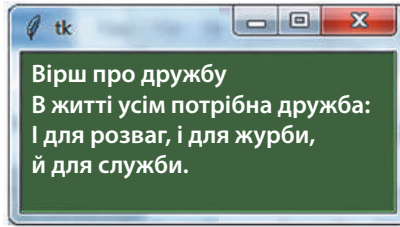


Рис. 33.3

Застосовуючи метод `get()`, слід мати на увазі, що в кінці кожного рядка та в кінці всього тексту багаторядкового поля завжди є символ перенесення рядка. Якщо цей символ в кінці тексту не потрібен, то метод слід викликати так: `s = text.get(1.0, 'end-1c')`. Якщо в константі `'end-1c'` замість числа 1 вказати інше ціле число, то буде відкинута відповідну кількість символів.

Питання для самоперевірки



- Що таке зріз рядка? Як його отримати?
- Визначте, якого значення набуває змінна `a` після виконання операторів, якщо `s = 'Сонечко'`:

а) <code>a = s[2:5];</code>	в) <code>a = s[5];</code>	г) <code>a = 5*s[2];</code>
б) <code>a = s[:3];</code>	г) <code>a = s[::2];</code>	д) <code>a = s[:: -1].</code>
- Запишіть оператор, що реалізує таку дію:
 - отримати підрядок рядка `s` з 2-го символу по 5-й включно;
 - отримати підрядок з останніх п'яти символів рядка `s`.
- Розгадайте фразу і запишіть оператор для її розшифрування:
 - `s = 'я клеюгбмлщюй кіеннфгошрщмззатхиткуур';`
 - `s = 'мофинвафин пайфинтонфин'.`
- Складіть програму для визначення, чи є паліндромом уведений з клавіатури рядок `s`.
- Поясніть порядок використання текстового поля.

Вправа 33



- ▶▶ Скласти програму для шифрування тексту.
- 1) Завантажте середовище PyCharm. Створіть Python file із назвою Вправа 33. Завантажте модуль tkinter. Створіть вікно програми із заголовком Шифрувальник. Завершіть програму оператором `root.mainloop()`.
 - 2) Створіть об'єкт `text` класу `Text`. Усі віджети розміщуйте за допомогою методу `grid()`. Уставте в поле віджета текст:
`text.insert(1.0,"Видаліть цей текст \nі напишіть свій.")`
 - 3) Створіть кнопки — об'єкти `b_shifr`, `b_delete` класу `Button` (рис. 33.4). Обробникам подій для кнопок надайте назви `shifrText`, `deleteText`.
 - 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+' '
    for i in range(0, len(s), 2):
        text.insert(END, s[i:i+2][::-1])
```
- Випробуйте програму.
 Уведіть текст і натисніть кнопку Зашифрувати.
 Проаналізуйте, який спосіб шифрування використано.
- 6) Придумайте свій спосіб шифрування тексту і змініть опис функції `shifrText()`, щоб застосувати цей спосіб.

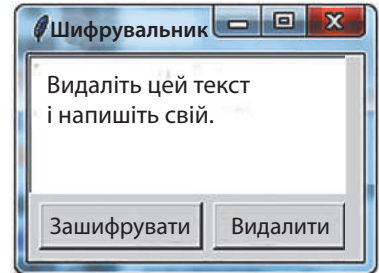


Рис. 33.4



Комп'ютерне тестування



Виконайте тестове завдання 33 із автоматичною перевіркою результату.



§ 34. Робота з текстовими файлами

Якщо для розв'язування задачі під час виконання програми потрібно вводити багато даних або результати роботи програми треба зберегти для подальшого використання, введення/виведення даних зручно здійснювати з використанням текстових файлів.

! Файл може бути джерелом інформації (під час введення) і приймачем інформації (під час виведення даних).

Розглянемо засоби Python для опрацювання наявних файлів і створення нових. У будь-якому випадку робота з файлом складається з трьох етапів (рис. 34.1):

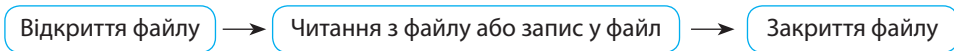


Рис. 34.1

Відкриття файлу

Для відкриття файлу користуються вбудованою функцією `open()`: `f = open(<шлях до файлу>, <режим відкриття>)`

де `f` — змінна для збереження файлового об'єкта, який використовується для читання та/або запису у файл.

Якщо шлях до файлу складається лише з імені файлу, Python буде шукати цей файл у папці, в якій міститься програма.

Параметр режим відкриття визначає, які дії можна буде виконувати з файлом. Основні режими відкриття такі:

- 'r' — тільки для читання (встановлюється, якщо не вказати);
- 'w' — тільки для запису. Знищує вміст файлу, якщо файл існує, і створює новий файл для запису, якщо не існує;
- 'a' — для дозапису. Інформація додається в кінець наявного файлу; якщо файл не існує, його буде створено.

Читання з файлу

Читання даних із файлу означає отримання доступу до цих даних.

Функція `read([n])` зчитує `n` символів, якщо задано ціле значення `n`. Якщо ж воно відсутнє, то зчитуються всі дані з файлу.

- 1 У файлі text.txt міститься рядок Привітання від Python!. У консоль виводиться вміст файлу (рис. 34.2).

```
f = open('text.txt')
print(f.read(1))      # Зчитати 1 символ із файлу
print(f.read())      # Зчитати всі інші символи
f.close()
```

```
===== R
П
ривітання від Python!
>>>
```

Рис. 34.2

Функція `readline()` зчитує із файлу один рядок. Після зчитування елемента (символу або рядка) з файлу поточним (доступним) стає наступний елемент. Якщо файл містить декілька рядків тексту, то можна прочитати файл порядково, скориставшись циклом `for`:

```
f = open('numbers.txt')
for line in f: print(line, end = " )      # Якщо не вказати атрибут end = ",
f.close()                                # виведуться зайві порожні рядки
```

Запис у файл

Запис у файл можна здійснювати за допомогою функції `print()`, зазначаючи у списку параметрів, що виведення спрямовується у файл:

```
print(<список виведення>, file = <файловий об'єкт>)
```

Для того щоб у файлі збереглися зміни, його потрібно обов'язково закрити, викликавши для файлового об'єкта метод `close()`.

- 2 Запишемо у файл один рядок:

```
f = open('f1.txt', 'w')
print('111111', file = f)
f.close()
```

- 3 Файл f1.txt містить декілька чисел, записаних в окремих рядках. Розглянемо, як зчитати всі числа, знайти їх суму і дозаписати значення суми в кінець файлу (рис. 34.3).

```
f = open('f1.txt', 'r')
suma = 0
for line in f:
    suma = suma+int(line)
f.close()
f = open('f1.txt', 'a')
print('сума =', suma, file = f)
f.close()
```

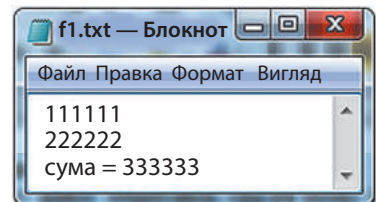


Рис. 34.3

Діалогові вікна відкриття та збереження файлів

Модуль `filedialog` надає доступ до діалогових вікон для вибору або задання імені файлу. Для завантаження модуля слід записати команду (для скорочення подальших записів позначимо його `fd`):

```
from tkinter import filedialog as fd
```

Розглянемо функції модуля `filedialog`.

Функція `fd.askopenfilename()` надає діалогове вікно для відкриття файлу, функція `fd.asksaveasfilename()` — вікно для збереження файлу. Обидві функції повертають шлях до файлу, але самі його не відкривають і не зберігають. Це потрібно робити описаними вище засобами.

4 Створимо вікно з кнопками для відкриття і збереження файлу (рис. 34.4).

```
from tkinter import*
from tkinter import filedialog as fd
def insertText():
    file_name = fd.askopenfilename()
    f = open(file_name)
    text.insert(1.0, f.read())
    f.close()
def extractText():
    file_name = fd.asksaveasfilename()
    f = open(file_name, 'w')
    f.write(text.get(1.0, END))
    f.close()
root = Tk()
text = Text(width = 25, height = 8)
text.pack()
Button(root, text = "Відкрити", command = insertText).pack()
Button(root, text = "Зберегти", command = extractText).pack()
```

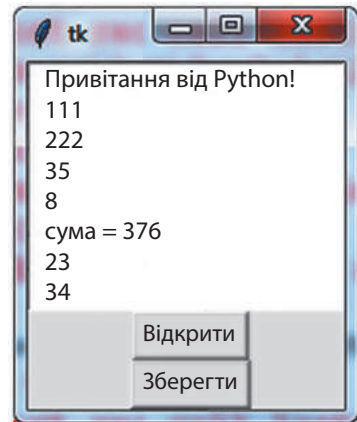


Рис. 34.4

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 до текстового файлу?
4. Обговоріть переваги використання діалогових вікон для відкриття і збереження файлів.

5. Що означає `as fd` у команді `from tkinter import filedialog as fd`?
6. Файл `t.txt` містить декілька рядків символів (і цифр, і літер). Складіть програму, що переписує цифри з файлу `t.txt` у файл `cif.txt`, всі інші символи — у файл `lit.txt`.

Вправа 34



▶▶ Написати програму, що підраховує кількість заданих символів у фрагменті тексту.

- 1) Створіть текстовий файл `f_text.txt` і запишіть у нього фрагмент статті з вікіпедії про вашу область.
- 2) Завантажте модулі `tkinter`, `filedialog` (як `fd`), `messagebox`.
- 3) Створіть вікно із заголовком `Аналіз тексту`. Розмістіть у вікні віджет `entry` класу `Entry` для задання шуканого символу, віджет `text` класу `Text` — для показу вмісту файлу `f_text.txt`.
- 4) Додайте до вікна два віджети `Button`: для відкриття файлу і для підрахунку числа входжень шуканого символу (рис. 34.5).

- 5) Опишіть функцію `loadText()` — обробник події для кнопки `b_open`:

```
def loadText():
    file_name = fd.askopenfilename()
    f = open(file_name)
    text.insert(1.0, f.read())
    f.close()
```



- 6) Опишіть функцію `count_C()` — обробник події для кнопки `b_count`:

```
def count_C():
    c = entry.get()
    k = text.get(1.0, END).count(c)
    messagebox.showinfo('Відповідь', 'k = '+str(k))
```

Випробуйте роботу програми для різних значень символу `c`.

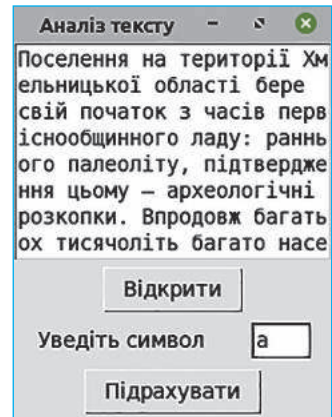


Рис. 34.5

Комп'ютерне тестування



Виконайте тестове завдання 34 із автоматичною перевіркою результату.



§ 35. Логічний тип даних. Умовні оператори

До простих типів даних у Python належить логічний тип `bool`. Змінна типу `bool` може набувати значення `True` (істина) або `False` (хибність).

Прості і складені умови

Значень типу `bool` набувають вирази, побудовані за допомогою операцій порівняння (**прості умови**):

<code>==</code>	дорівнює	<code>></code>	більше	<code>>=</code>	більше або дорівнює
<code>!=</code>	не дорівнює	<code><</code>	менше	<code><=</code>	менше або дорівнює

1 Проаналізуємо результати обчислення логічних виразів:

<pre>>>> print(2*2==4) True</pre>	Оскільки $2 \cdot 2 = 4$ — істинна рівність, то значення відповідного логічного виразу дорівнює <code>True</code>
<pre>>>> NUM = 10 >>> NUM%3 == 0 False</pre>	Отримано відповідь на запитання «Чи ділиться значення змінної <code>NUM</code> на 3?»

Логічні вирази для перевірки виконання кількох умов (**складені умови**) будують із використанням наведених логічних операцій:

Операція	Назва	Приклад	Значення
<code>or</code>	або логічний добуток	$2*2 == 7$ or $4 <= 5$ $3 > 8$ or $12 < 5$ or $9 == 1$	<code>True</code> <code>False</code>
<code>and</code>	і логічна сума	$2*2 == 7$ and $4 <= 5$ $3 <= 8$ and $12 > 5$ and $9 != 1$	<code>False</code> <code>True</code>
<code>not</code>	не логічне заперечення	<code>not 2*2 == 4</code> <code>not 8 < 3</code>	<code>False</code> <code>True</code>

2 Перевіримо, чи належить `NUM` проміжку `[7; 20]`:

```
>>> NUM = 12
>>> NUM >= 7 and NUM <= 20
True
```

Якщо змінній присвоїти значення логічного виразу, вона належатиме до типу `bool`:

```
>>> a = 3 > 7
>>> a
False
```

► Вбудовані 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
```

Функція `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 послідовностей дій. Пригадаємо синтаксис та семантику умовних операторів мови Python, призначених для програмування розгалужень.

► Умовний оператор `if`

Виконання оператора `if` (англ. *якщо*) можна описати так: «якщо умова істинна, то виконати ці оператори». Оператор `if` реалізує алгоритмічну конструкцію неповне розгалуження (рис. 35.1).

Синтаксис оператора:

```
if <умова>:
    <оператори>
```

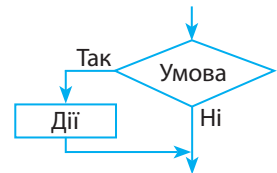


Рис. 35.1

Оператори записуються з обов'язковим однаковим відступом від лівого краю. Якщо оператор лише один, його можна записати після двокрапки в одному рядку з умовою.

5 Складемо програму для упорядкування введених з клавіатури 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)
```

► Умовний оператор `if...else`

Алгоритмічній конструкції повне розгалуження (рис. 35.2) відповідає умовний оператор `if...else` (англ. *якщо...інакше*).

Синтаксис оператора:

```
if <умова>:
    __<оператори 1>
else:
    __<оператори 2>
```

Якщо умова виконується, тобто має значення `True`, то виконується блок дій `<оператори 1>`. В іншому випадку виконується блок дій `<оператори 2>`, записаний після службового слова `else`. Команди, вкладені в гілки `if` і `else`, записуються з однаковим відступом.

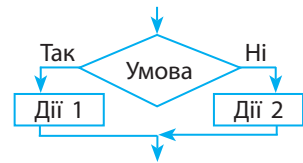


Рис. 35.2

6 Складемо програму для перевірки, чи введені 3 числа є довжинами сторін прямокутного трикутника. Для перевірки скористаємось теоремою Піфагора:

```
a, b, c = map(int, input().split())
if a**2 == b**2 + c**2 or b**2 == a**2 + c**2 or 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 умовного оператора знову використовують оператор `if`.

7 Складемо програму, яка б за уведеним цілим числом виводила слово «рік» у правильному відмінку (рис. 35.3). Проаналізувавши форми слова «рік», помічаємо, що форма слова залежить від останньої цифри числа K . Проте значення від 11 до 14 слід опрацювати окремо (алгоритм на рис. 35.4).

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. 35.3

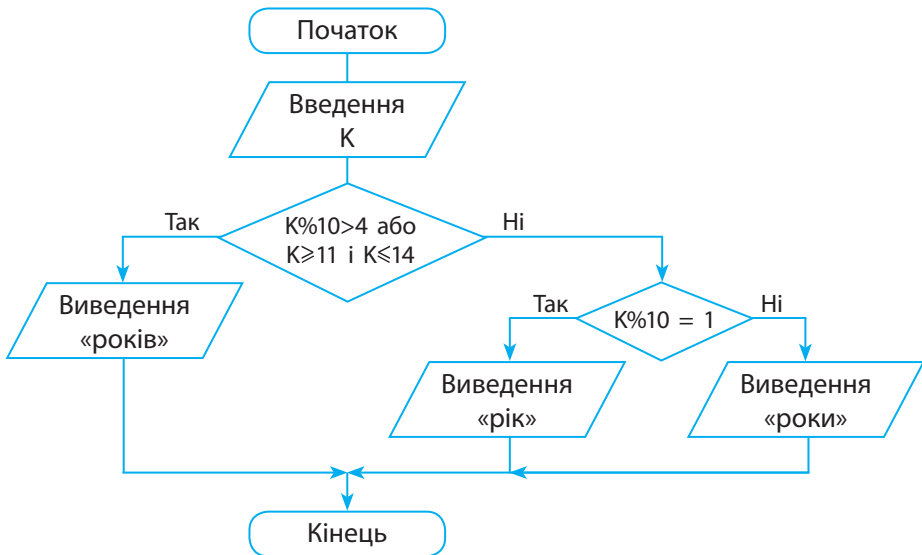


Рис. 35.4

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')
  
```

► Багатоваріантне розгалуження

Перевірку умови в гілці `else` можна записати скорочено, скориставшись ключовим словом `elif` (англ. *else if* — інакше, якщо).

Це дозволяє нам реалізувати багатоваріантне розгалуження (рис. 35.5).

Синтаксис оператора:

```
if <умова1>:
    <оператори 1>
elif <умова2>:
    <оператори 2>
...
elif <умова N>:
    <оператори N>
else:
    <оператори>
```

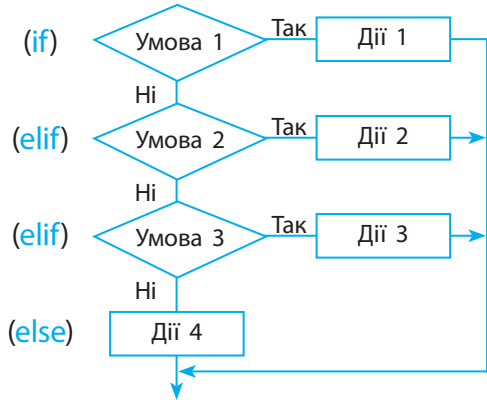


Рис. 35.5

8

Розглянемо задачу. Визначити, в якій координатній чверті лежить точка з координатами x , y ($x \neq 0$, $y \neq 0$) (рис. 35.6).

```
x = int(input('x = ?'))
y = int(input('y = ?'))
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')
```

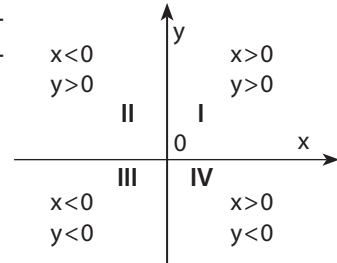


Рис. 35.6

У цьому прикладі складені умови, що перевіряються в гілках `if...elif`, охоплюють усі можливі варіанти розташування точки, що відповідають умові задачі. Тому якась із гілок буде виконана, і гілка `else` у цьому коді була б зайвою. Але якщо є ймовірність того, що змінна x або y набуде значення 0 , то слід додати гілку `else` і записати в ній оператори для його опрацювання. Наприклад:

```
else: print('Точка лежить на осі')
```

Розглянуті умовні оператори 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 операторів:

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 операторів:
 - 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 умовний оператор такого змісту: менше з двох значень x і y замінити нулем, а у випадку їх рівності — замінити нулями обидва.
4. Запишіть умовний оператор для обчислення значення y :
 - а) $y = \begin{cases} 1 - \sin a, & \text{якщо } -1 < a < 1; \\ 0, & \text{для всіх інших;} \end{cases}$
 - б) $y = \begin{cases} x, & \text{якщо } x < 0; \\ x^2, & \text{якщо } 0 \leq x \leq 5; \\ 2x, & \text{якщо } x > 5. \end{cases}$
5. Наведіть приклад використання оператора `if...elif...else`.
6. Складіть програму, яка за введеним значенням температури t виводить повідомлення про стан, у якому перебуває вода за такої температури (твердий, рідкий, газоподібний).

Вправа 35



- ▶▶ Скласти програму для знаходження коренів квадратного рівняння вигляду $ax^2 + bx + c = 0$.
 - 1) Створіть Python file із назвою Вправа 36.
 - 2) Завантажте модулі `tkinter`, `messagebox` (як `mb`), `math`.
 - 3) Створіть вікно програми із заголовком Квадратне рівняння. Завершіть програму оператором `root.mainloop()`.
 - 4) Розробіть інтерфейс програми за зразком (рис. 35.7). Віджетам `Entry` дайте назви `enA`, `enB`, `enC`.
 - 5) опишіть функцію `btn_click()` — обробник події для кнопки Знайти корені. Значення a , b , c можна зчитати з полів `enA`, `enB`, `enC` у такий спосіб:

```
a, b, c = map(float, [enA.get(), enB.get(), enC.get()])
```

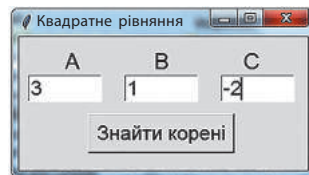


Рис. 35.7

Результати обчислення виводьте до діалогового вікна, попередньо сформувавши рядок. Наприклад, якщо рівняння має два корені, значення коренів можна вивести так:

```
s = 'x1 =' + str(round(x1, 2)) + 'x2 =' + str(round(x2, 2))
mb.showinfo('Два корені', s)
```

- б) Підберіть такі набори значень a , b , c , щоб квадратне рівняння мало один корінь; два корені; не мало коренів.

Комп'ютерне тестування



Виконайте тестове завдання 35 із автоматичною перевіркою результату.

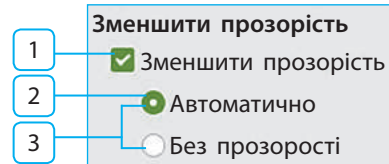


§ 36. Елементи вибору

Вам, напевно, доводилось робити вибір певних значень із числа запропонованих у діалоговому вікні за допомогою таких елементів керування, як перемикачі та прапорці (рис. 36.1).

У Tkinter радіокнопки, з яких складаються перемикачі, є об'єктами класу `Radiobutton`, прапорці — об'єктами класу `Checkbox`.

Розглянемо особливості створення та програмування елементів вибору.



- 1 — прапорець
- 2 — радіокнопка
- 3 — перемикач

Рис. 36.1

Елемент керування Радіокнопка (`Radiobutton`)

Перемикачі, що складаються з двох або більше радіокнопок (віджетів класу `Radiobutton`), використовують, коли потрібно вибрати один із декількох варіантів. Радіокнопки в перемикачі взаємопов'язані, тобто у разі вибору іншої радіокнопки попередня вимикається.

Радіокнопки перемикача зв'язує спільна змінна. Назву цієї змінної зазначають в атрибуті `variable` кожної радіокнопки. Ця змінна має бути екземпляром одного з описаних у Tkinter класів: `BooleanVar`, `IntVar`, `DoubleVar`, `StringVar` і може набувати, відповідно, логічних, цілих, дробових або рядкових значень. Щоб **отримати значення змінної**, слід викликати метод `get()`, а щоб його змінити, — метод `set()`.

Якщо радіокнопку вибрано, спільна змінна отримує значення її атрибуту `value`, і навпаки, надавши спільній змінній певне значення, можна увімкнути радіокнопку з відповідним значенням `value`.

Отже, для **створення перемикача** потрібно:

- 1) створити змінну `var` класу `IntVar` для керування радіокнопками;
- 2) створити потрібну кількість об'єктів класу `Radiobutton` із різними значеннями властивості `value`;
- 3) занести в атрибут `variable` усіх об'єктів назву змінної `var`.

1 Створимо три радіокнопки із заголовками "red", "yellow", "green".

```
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)
```

Під час запуску програми виявиться ввімкненою радіокнопка `rbR`, оскільки 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, слід радіокнопки пов'язати з обробниками подій.

2 Пов'яжемо всі кнопки з функцією-обробником `change()`, у якій, залежно від того, яка з радіокнопок активна, змінюватимемо колір тла форми (рис. 36.2):

```
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')
```

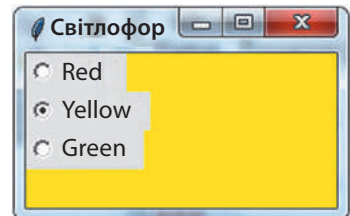


Рис. 36.2

3 Розробимо програму Конструктор піц, яка під час формування замовлення дає змогу користувачеві вибирати розмір піци (рис. 36.3).

```
from tkinter import*
def btn_click():
    lab2.config(text = 'Вартість: ' +str(var.get()*60)+ ' грн')
```

```
root = Tk()
root.title('Конструктор піц')
var = IntVar()
var.set(2)
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)
lab2 = Label(root, text = 'Вартість: ', width = 20, anchor = W)
lab2.pack(padx = 5, pady = 10)
root.mainloop()
```

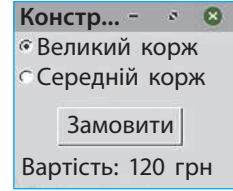


Рис. 36.3

Елемент Прапорець (Checkbutton)

Прапорець — об'єкт класу `Checkbutton` — використовують у програмі там, де потрібно вмикати або вимикати певний режим, параметр тощо.

Об'єкти класу `Checkbutton` можна також розташувати групою, але при цьому кожна позначка встановлюється або знімається незалежно від стану інших прапорців. Тобто в групі `Checkbutton` можна встановити одночасно декілька прапорців, а також зняти всі.

Кожний об'єкт класу `Checkbutton` повинен мати свою змінну — екземпляр одного з перелічених вище класів:

```
cb_var = IntVar()
cb_var.set(0)
```

Встановлюючи або знімаючи прапорець, користувач задає значення цієї змінної, таким чином впливаючи на подальшу роботу програми.

Об'єкт `cb` класу `Checkbutton` створюється такою командою:

```
cb = Checkbutton(text = "<підпис>", variable = cb_var,
                 onvalue = 1, offvalue = 0)
```

Якщо встановити прапорець, то пов'язана з ним змінна набуває значення `onvalue`, а якщо зняти — значення `offvalue`.

4 Для перевірки стану прапорця та активізації пов'язаних із ним параметрів слід отримати значення змінної:
`if cb_var.get() != 0: <активізація параметрів, пов'язаних із прапорцем>`

5 Додамо до програми Конструктор піц (див. приклад 3) можливість вибирати складники піци (рис. 36.4).

Перед рядком із викликом конструктора `Button()` додамо команди для створення і налаштування прапорців:

```
Label(root, text = 'Виберіть складники:',
      width = 20, anchor = W).pack(pady = 10)
```

```
cvar1 = BooleanVar()
```

```
cvar1.set(0)
```

```
c1 = Checkbutton(text = "Цибуля",
                variable = cvar1, onvalue = 1, offvalue = 0)
```

```
c1.pack(anchor = W)
```

```
cvar2 = BooleanVar()
```

```
cvar2.set(0)
```

```
c2 = Checkbutton(text = "Моцарела", variable = cvar2, onvalue = 1, offvalue = 0)
```

```
c2.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)
```

```
    lab2.config(text = 'Вартість: ' +str(suma)+ ' грн')
```

Віджети `Radiobutton` і `Checkbutton` підтримують більшість атрибутів зовнішнього вигляду, які є в інших елементів графічного інтерфейсу.

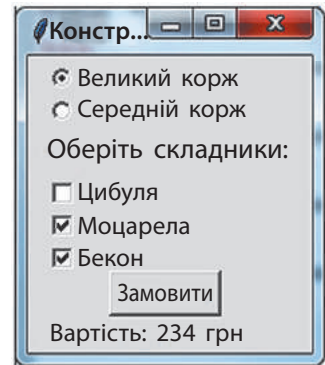


Рис. 36.4

Питання для самоперевірки



1. У яких випадках використовують компонент `Radiobutton`?
2. Що забезпечує використання компонента `Checkbutton`?
3. Як забезпечити 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 36



▶▶ Створити довідник «Міста України». Залежно від вибору перемикача виводити до текстового поля інформацію про місто.

- 1) Підготуйте декілька текстових файлів з інформацією про міста України. Збережіть файли в папку Міста України.
- 2) Створіть вікно програми із заголовком Міста України. Збережіть файл у папку Міста України. Додайте віджет `text1` класу `Text`:
`text1 = Text(width = 24,`

`height = 10, wrap = WORD)`

- 3) Створіть перемикач із 5 радіокнопок із підписами, що відповідають назвам міст (рис. 36.5):

```
var = IntVar()
```

```
var.set(0)
```

```
kyiv = Radiobutton(text = 'Київ',
```

```
variable = var, value = 0, command = change)
```

```
lviv = Radiobutton(text = 'Львів', variable = var, value = 1,
```

```
command = change)
```

```
<...>
```

```
kyiv.grid(row = 0, column = 0, sticky = W)
```

```
lviv.grid(row = 1, column = 0, sticky = W)
```

```
<...>
```

- 4) Опишіть спільний обробник події для перемикачів — функцію `change()`. Залежно від значення `var.get()` зі списку вибирається ім'я файлу з інформацією про вибране місто:

```
def change():
```

```
files = ['Київ.txt', 'Львів.txt', 'Дніпро.txt', 'Одеса.txt', 'Харків.txt']
```

```
view(files[var.get()])
```

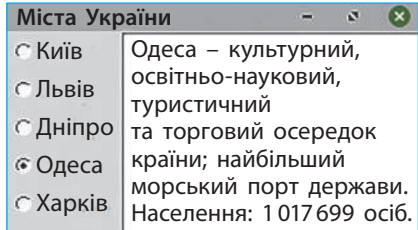


Рис. 36.5

- 5) Опишіть функцію `view()`, у якій до поля віджета `text1` виводиться вміст файлу з назвою з аргумента `filename`.

- 6) Додайте кілька перемикачів із назвами ваших улюблених міст, підготуйте повідомлення про них. Випробуйте програму.



Комп'ютерне тестування

Виконайте тестове завдання 36 із автоматичною перевіркою результату.



§ 37. Алгоритми з повтореннями. Цикли

Як відомо, однією з базових алгоритмічних структур є цикл.



Повторення (цикл) — це алгоритмічна структура, за допомогою якої певна послідовність дій виконується декілька разів.

Ви маєте досвід запису алгоритмів із повторенням (циклів) мовою Python і вмієте використовувати два види операторів циклу: із параметром (`for`) та з умовою (`while`).

Згадаємо: 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 визначає певна послідовність значень (символи рядка, елементи списку тощо). Блок-схему циклу подано на рис. 37.1.

Синтаксис циклу `for`:

```
for x in <послідовність>:
    <тіло циклу>
```

Змінна `x` — параметр циклу — послідовно набуває значень кожного члена послідовності.

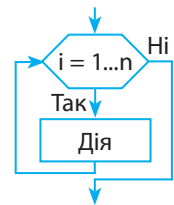


Рис. 37.1

1 Застосуємо цикл `for` для посимвольного опрацювання рядка. Виведемо всі символи рядка, що міститься в змінній `s`, окрім «зірочок»:

```
s = 'AB*C*D*EF*G'
```

```
for C in s:
    # Перебір у змінній C символів рядка s
    if not C == '*': print(C, end = " ")
```

2 Дано список днів тижня. З'ясуємо, скільки з них припадають на вихідні (сб, нд).

```
D = ['пн', 'нд', 'чт', 'сб', 'пт', 'сб', 'вт', 'ср', 'нд', 'чт', 'чт', 'нд', 'вт', 'пн', 'ср', 'нд', 'сб']
```

```
K = 0
```

```
for day in D:
    # Перебір у змінній day елементів списку D
    if day in ['сб', 'нд']: K += 1
print(K)
```

Числові послідовності можна задавати за допомогою вбудованої функції `range()`:

```
range([<початок>], <кінець>[, <крок>])
```

Вона створює об'єкт, який генерує послідовність цілих чисел від значення початок до кінець із заданим кроком, при цьому значення кінець у послідовність не входить. Чергове число генерується при кожному звертанні до такого об'єкта, тому навіть для дуже довгих послідовностей він не займає багато пам'яті.

- 3 Надрукувати числа від 20 до 24. Якщо крок дорівнює 1, цей параметр можна не зазначати:

```
for i in range(20, 25): print(i)
```

Надрукувати числа від 0 до 3. Якщо початок дорівнює 0, цей параметр можна не зазначати:

```
for i in range(4): print(i)
```

Надрукувати числа від 5 до 1. Якщо потрібно вести відлік у зворотному порядку, крок має бути від'ємним:

```
A = range(5, 0, -1)
for i in A: print(i)
```

Цикл із передумовою

Цикл із передумовою (або цикл `while`) буде повторюватися доти, поки це потрібно. Блок-схему оператора `while` наведено на рис. 37.2.

Синтаксис оператора:

```
while <умова>:
    <тіло циклу>
```

Виконання оператора циклу `while` починається з обчислення значення логічного виразу — умови повторення циклу.

Якщо *умова істинна*, то виконуються оператори тіла циклу і керування повертається на перевірку умови.

Якщо *умова хибна*, то виконання циклу припиняється і виконується оператор, який є наступним після оператора `while`.

- ! Якщо при першій перевірці умова виявиться хибною, то тіло циклу `while` не виконається жодного разу.

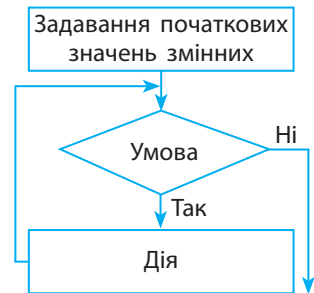


Рис. 37.2

- 4 Числами Фібоначчі називають послідовність чисел, які обчислюють за таким правилом: перші два числа дорівнюють 1, а кожне наступне дорівнює сумі двох попередніх. Математично це можна записати так:

$$\text{Fib}[1] = \text{Fib}[2] = 1; \text{Fib}[n] = \text{Fib}[n-1] + \text{Fib}[n-2].$$

Для визначення номера першого числа Фібоначчі, яке більше за 1000, потрібен цикл із умовою:

```
a = b = 1          # Перші два числа Фібоначчі
c = 2             # Наступне число
n = 3            # Змінна — лічильник чисел
while c < 1000:  # Поки останнє число менше від 1000:
    c = a + b    # — обчислюємо наступне число;
    a, b = b, c  # — зберігаємо два останніх у змінних a і b;
    n += 1       # — збільшуємо лічильник
print('число', c, 'має номер', n)
```

- 5 Запрограмуємо рух об'єкта полотном із відбиттям від країв полотна (рис. 37.3).

```
from tkinter import*
from random import*
import time
tk = Tk()
canvas = Canvas(tk, width = 400, height = 400)
canvas.pack()
x = randint(10, 350)      # Задання випадкових початкових координат
y = randint(10, 350)
id = canvas.create_oval(x, y, x+50, y+50, fill = 'red')  # Створення об'єкта
dx = dy = 5              # Задання кроку руху
while True:              # Організація нескінченного циклу
    if x > 350 or x < 10: dx = -dx  # Якщо об'єкт досягає меж полотна,
    if y > 350 or y < 10: dy = -dy  # напрямок руху змінюється
                                    # на протилежний
    x = x + dx                # Обчислення нових координат об'єкта
    y = y + dy
    canvas.move(id, dx, dy)    # Переміщення об'єкта в точку
                                # з новими координатами
    tk.update()               # Оновлення полотна
    time.sleep(0.05)          # Затримка виконання програми на 0,05 с
```

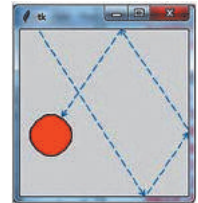


Рис. 37.3

Вкладені цикли



Вкладений цикл — цикл, який міститься в тілі іншого циклу.

Такий цикл також називають **внутрішнім**, а цикл, у якому він міститься, — **зовнішнім**.

Усередину вкладеного циклу може бути вкладено наступний цикл і так далі. У Python підтримується до 20 рівнів вкладеності циклів.

Принцип роботи вкладених циклів такий: 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. І так доти, поки не завершиться зовнішній цикл.

6 Обчислимо значення змінної $y = 2 \cdot k + n$ при всіх значеннях змінних $n=1, 2, 3$ і $k=2, 4, 7, 9$.

Якщо перебирати всі значення n і k , отримаємо 12 значень змінної y .

Скласти програму можна в такий спосіб: для кожного значення n перебрати всі значення k — від 2 до 8, тобто n використати як параметр зовнішнього циклу, k — як параметр внутрішнього циклу (рис. 37.4):

```
print('n k y')
for n in range(1, 4):
    for k in [2, 4, 7, 9]:
        y = 2*k+n
        print(n, k, y)
```

n	k	y
1	2	5
1	4	9
1	7	15
1	9	19
2	2	6
2	4	10
2	7	16
2	9	20
3	2	7
3	4	11
3	7	17
3	9	21

Рис. 37.4

7 Виведемо в консоль таблицю множення:

```
for i in range(1, 10):
    for j in range(1, 10):
        print(i*j, end = '\t')
    print('\n')
```

Зовнішній цикл `for i in range(1, 10)` виконується 9 разів.

Внутрішній цикл `for j in range(1, 10)` виконується 9 разів для однієї ітерації зовнішнього циклу i , відповідно, 81 раз для всіх ітерацій зовнішнього циклу.

У кожній ітерації внутрішнього циклу на консоль буде виводитися добуток чисел i та j (рис. 37.5).

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. 37.5

Питання для самоперевірки



- У вікні IDLE уведіть команду:

```
for i in range(65, 90): print(chr(i))
```

 Обговоріть результати виконання оператора `for`. Поекспериментуйте зі зміною параметрів функції `range()`.
- З'ясуйте, що буде надруковано в результаті виконання фрагмента програми:

а) <code>for i in range(5):</code>	б) <code>for x in 'рядок':</code>	в) <code>for x in range(5, 1, -1):</code>
<code>print(i*i)</code>	<code>print(2*x)</code>	<code>print(x)</code>
- З клавіатури вводиться 10 довільних цілих чисел. Запишіть оператори для виконання таких дій:

а) знайдіть суму та середнє арифметичне;	б) знайдіть суму тільки від'ємних доданків;	в) знайдіть найменше число.
--	---	-----------------------------
- Визначте значення змінної s після виконання операторів:

а) <code>s = 0</code> <code>i = 0</code> <code>while i < 5:</code> <code> i = i + 1</code> <code> s = s + i</code>	б) <code>s = 0</code> <code>i = 5</code> <code>while i > 1:</code> <code> s = s + i</code> <code> i = i - 1</code>	в) <code>s = 0</code> <code>i = 0</code> <code>while i < 5:</code> <code> i = i + 1</code> <code> s = s + 10</code>
---	---	--
- Складіть програму 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 виведе програма?

Вправа 37



- ▶▶ Вивести на екран усі прості числа на інтервалі $[a; b]$.
- 1) Як відомо, простими називають числа (крім 1), що діляться без остачі тільки на 1 і самі на себе.
Розглянемо алгоритм перевірки, чи є число c простим.
 - а) Уводиться змінна `prap`, яка відіграє роль прапорця (за її значенням можна визначити, чи відбулася деяка подія). Змінній `prap` присвоюється початкове значення `False`.
 - б) Перебираються всі числа D від 2 до $c//2$. Якщо число D є дільником числа c (тобто c ділиться на D без остачі), то змінній `prap` присвоюється значення `True`.
 - в) Після завершення перебору можливих дільників перевіряється значення змінної `prap`. Якщо воно в процесі повторень циклу змінилось на `True`, це означає, що c має принаймні три дільники, а отже, не є простим.
 - 2) Запишіть програму для визначення, чи є число c простим.
Увага: задання значення змінної c навмисно пропущено.
<задання значення змінної c >

```
prap = False
for D in range(2, c//2):
    if c%D == 0: prap = True
if prap == False: print(c, ' – просте число')
else: print(c, ' – складене число')
```
 - 3) За умовою задачі переберіть усі числа в інтервалі $[a; b]$, тобто для кожного c від a до b потрібно виконати алгоритм перевірки, чи є число c простим.
 - 4) Змініть програму так, щоб a і b вводилися з клавіатури.
 - 5) Додайте керуючий рядок циклу `for` для перебору значень c у діапазоні $[a; b]$.
Тілом циклу є код перевірки, чи є число c простим.
 - 6) Збережіть файл Python. Випробуйте програму з різними значеннями a і b . Поміркуйте, чи можна прискорити підрахунок, зменшивши кількість ітерацій циклу `for D...`



Комп'ютерне тестування

Виконайте тестове завдання 37 із автоматичною перевіркою результату.



§ 38. Словники. Віджет ComboBox

Списки дозволяють пов'язувати сукупність даних у єдину структуру. Якщо ж потрібно мати доступ до даних за певним ключем, застосовують складнішу структуру — словник (рис. 38.1).

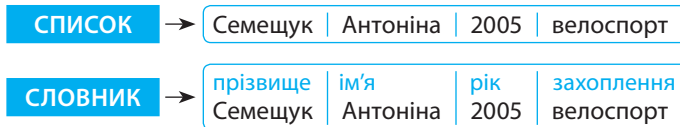


Рис. 38.1

На рисунку видно, що в словнику кожен елемент даних позначений ключовим словом, тобто словник поєднує два набори даних, в одному з яких всі значення різні: назви товарів та їх ціни, назви країн та їх столиць, характеристики автомобіля та їх значення тощо.

- Можна створити словник, який описує об'єкт людина, і зберегти в ньому назви властивостей об'єкта (ім'я, вік, стать, професія тощо) та їх значення. Якщо об'єднати словники в список, то можна ефективно опрацьовувати великі обсяги персональних даних.

Створення словників

Словник (об'єкт типу `dict`) — невпорядкована сукупність пар ключ: значення. Кожне значення в словнику має унікальний ключ, який є об'єктом одного з незмінюваних типів: рядком, логічним значенням, цілим або дійсним числом.

Щоб **створити словник**, потрібно помістити у фігурні дужки `{}` пари ключ: значення, розділені комами:

```
dictionary = {ключ1: значення1, ключ2: значення2, ...}
```

- Створимо порожній словник `zakuru` і словник `magazyn` для збереження пар товар: ціна.

```
zakuru = {}
```

```
magazyn = {'яблука': 15, 'груші': 25, 'огірки': 9, 'помідори': 15, 'морква': 9}
```

- Доступ до елементів словника здійснюється за ключем, а не за індексом, тому ключ не може повторюватися.

3 Отримаємо зі словника `magazyn` елемент із ключем "яблука":

```
>>> magazyn['яблука']
```

```
15
```

Якщо ключа в словнику немає, буде згенеровано помилку:

```
>>> magazyn['цибуля']
```

```
KeyError: 'цибуля'
```

Щоб попередити таку помилку, слід перевірити наявність ключа в словнику за допомогою виразу `<ключ> in <словник>`:

```
if 'цибуля' in magazyn:
```

```
    print(magazyn['цибуля'])
```

```
else: print("Товар відсутній")      # Буде надруковано "Товар відсутній"
```

Для додавання елемента у словник слід звернутися до елемента за його ключем і присвоїти йому значення. Якщо ключ у словнику вже існує, поточне значення буде замінено новим.

4 Установимо для словника `magazyn` значення елемента з ключем 'морква':

```
>>> magazyn['морква'] = 5
```

Для видалення елемента за ключем застосовується оператор `del`:

```
del magazyn['яблука']
```

Методи словників

Розглянемо деякі методи об'єктів типу `dict` на прикладі:

```
books = {'І. Котляревський': 'Енеїда', 'М. Гоголь': 'Тарас Бульба',
         'Т. Шевченко': 'Гайдамаки', 'О. Гончар': 'Собор'}
```

Метод `keys()` повертає `dict_keys()` — ітеративне подання ключів словника:

```
>>> books.keys()
```

```
dict_keys(['І. Котляревський', 'М. Гоголь', 'Т. Шевченко', 'О. Гончар'])
```

Якщо потрібен саме список, то для перетворення викликають функцію `list()`:

```
>>> list(books.keys())
```

```
['І. Котляревський', 'М. Гоголь', 'Т. Шевченко', 'О. Гончар']
```

Метод `values()` повертає `dict_values()` — ітеративне подання значень словника. Щоб отримати список значень, викликають функцію `list()`:

```
>>> list(books.values())
```

```
['Енеїда', 'Тарас Бульба', 'Гайдамаки', 'Собор']
```

Перебір елементів словника

Послідовно перебрати всі елементи в словнику можна в циклі `for`. При переборі елементів ми отримуємо ключ поточного елемента і за ним можемо отримати значення елемента.

- 5 Переберемо всі ключі у словнику `books` і виведемо на екран ключі з відповідними значеннями (рис. 38.2):

```
for key in books:
    print(key, ' - ', books[key])
```

```
I. Котляревський – Енеїда
M. Гоголь – Тарас Бульба
T. Шевченко – Гайдамаки
O. Гончар – Собор
>>>
```

Рис. 38.2

Для перебору лише значень можна викликати метод `values()`.

- 6 Виведемо на екран всі значення зі словника `books` (рис. 38.3):

```
for value in books.values():
    print(value)
```

```
Енеїда
Тарас Бульба
Гайдамаки
Собор
>>>
```

Рис. 38.3

Елемент керування Поле зі списком (Combobox)

Віджет Tkinter Combobox — це розкритий список для вибору значень. Якщо натиснути на кнопку, можна побачити список варіантів, а при виборі одного з варіантів він стане поточним.

Клас Combobox описано в модулі Tkinter `ttk`, тому для використання віджета слід імпортувати цей модуль.

- 7 Створимо екземпляр віджета Combobox і присвоїмо значення, які міститиме список (рис. 38.4):

```
from tkinter import ttk
combo1 = ttk.Combobox(root,
    values = ['Енеїда', 'Тарас Бульба',
             'Гайдамаки', 'Собор'])
combo1.pack()
```

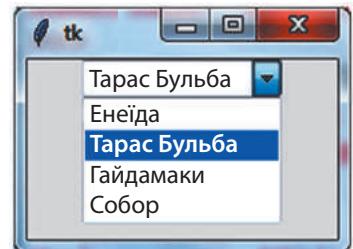


Рис. 38.4

Елемент, що виводиться в полі, можна програмно замінити на будь-який елемент зі списку, використовуючи метод `current(index)`.

- 8 Зробимо поточним елемент списку з індексом 3 (рис. 38.5):

```
...
combo1.current(3)
```

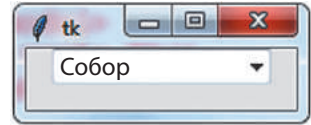


Рис. 38.5

Метод `current()` можна також використовувати для отримання індексу поточного (вибраного) елемента. Щоб отримати значення поточного елемента, слід використати метод `get()`.

- 9 Збережемо у змінних `index` і `item` індекс і значення вибраного елемента списку:

```
index = combo1.current()
item = combo1.get()
print("індекс вибраного елемента –", index, ", значення –", item)
```

У результаті виконання коду буде виведено рядок:
індекс вибраного елемента – 3, значення – Собор

Питання для самоперевірки



- Дано словник: `magazyn = {'яблука': 15, 'груші': 25, 'огірки': 9}`
Яким буде результат виконання операторів:
а) `magazyn['груші'];` б) `magazyn['морква']?`
- Як додати до словника `magazyn` товар капуста вартістю 5 грн?
- Опишіть способи послідовного перебору значень у словнику.
- Як задати вміст списку для віджета Combobox?
- Як отримати значення вибраного елемента зі списку віджета Combobox?
- Напишіть програму, яка моделює діалог в овочевому магазині: покупець вводить назву товару і кількість кілограмів, яку хоче придбати; програма обчислює загальну вартість покупки. Для збереження набору даних використайте словник із п. 1.

Вправа 38



- ▶▶ Створити програму Банкноти України.
Напишіть програму, яка виводить імена видатних особистостей України, яких зображено на грошових знаках.
- Завантажте середовище PyCharm. Створіть Python file із назвою Вправа 38.

Завантажте модулі tkinter, ttk, messagebox.

Створіть вікно розміром 400×250 із заголовком Банкноти.

- 2) Додайте до вікна віджети lab_nom класу Label, ent_nom класу Entry:

```
lab_nom = Label(root, text = 'Уведіть номінал банкноти:', font = 'Arial 12')
```

```
lab_nom.place(x = 10, y = 10)
```

```
ent_nom = Entry(root, width = 8, font = 'Arial 12')
```

```
ent_nom.place(x = 220, y = 10)
```

Додайте до вікна віджети lab_name класу Label, ent_name класу Entry для виведення імені особи, що зображена на банкноті.

- 3) Створіть об'єкт btn класу Button:

```
btn = Button(text = "Визначити ім'я", command = btn_click)
```

```
btn.place(x = 100, y = 40)
```

- 4) Створіть словник bank для збереження пар <номінал>: <ім'я>:

```
bank = {1000: 'Володимир Вернадський', 500: 'Григорій Сковорода',
        200: 'Леся Українка'}
```

Доповніть словник елементами, що відповідають іншим банкнотам. Відомості про банкноти можна отримати на сайті Національного банку України bank.gov.ua.

- 5) Опишіть функцію btn_click:

```
def btn_click():
```

```
    nom = int(ent_nom.get())
```

```
    if nom in bank:
```

```
        name = bank[nom]
```

```
        ent_name.delete(0, END)
```

```
        ent_name.insert(0, name)
```

```
    else: messagebox.showerror('Помилка', 'Такої банкноти в обігу немає')
```



- 6) Додайте до вікна полотно canvas для виведення зображення банкноти:

```
canvas = Canvas(height = 300, width = 500)
```

```
canvas.place(x = 100, y = 120)
```

Помістіть у папку з файлом програми графічні файли із зображеннями банкнот (наприклад, з назвами файлів 1000.png, 500.png тощо).

Створіть словник bank_image для збереження пар <номінал>: <ім'я файлу>:

```
bank_image = {1000: '1000.png', 500: '500.png', 200: '200.png'}
```

У тілі функції `btn_click()`: до операторів, що виконуються у випадку, коли умова `nom in bank` істинна, додайте оператори для зміни зображення в `canvas`:

```
im = bank_image[nom]
```

```
my_image = PhotoImage(file = im)
```

```
canvas.create_image(0, 0, anchor = NW, image = my_image)
```

Перевірте роботу програми.



Комп'ютерне тестування

Виконайте тестове завдання 38 із автоматичною перевіркою результату.



Практична робота 13

Створення програми з використанням словників і списків

Завдання: скласти програму, яка складає речення з трьох наборів слів, комбінуючи з них слова.

Обладнання: комп'ютер зі встановленою системою програмування PyCharm Community Edition.

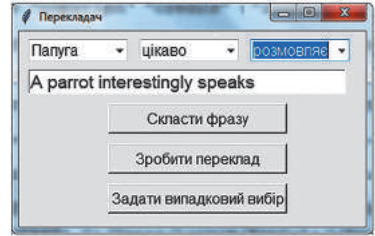
Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Створіть новий Python file Практична робота 13. Підключіть модулі: `tkinter`, `ttk`, `random`.
2. Створіть вікно програми із заголовком Перекладач.
3. Підготуйте словники, в яких будуть зберігатися слова для списків віджетів `Combobox`.


```
c1 = {'Собака': 'A dog', 'Кінь': 'A horse', 'Пташка': 'A bird', 'Сонце': 'The Sun'}
c2 = {'голосно': 'loudly', 'швидко': 'fast', 'високо': 'high', 'яскраво': 'brightly'}
c3 = {'гавкає': 'barks', 'бігає': 'runs', 'літає': 'flies', 'с'яє': 'shines'}
```

4. На основі словника `c1` сформууйте список ключів: `key1 = list(c1.keys())`
5. Так само створіть списки `key2`, `key3`.
6. Слова для складання речення будуть вибиратися з трьох розкривних списків (див. рисунок).
Створіть три віджети за зразком (розміщуйте методом `grid()`):
`combo1 = ttk.Combobox(root, width = 10, values = key1)`
`combo1.current(0)`
7. Створіть об'єкт класу `Entry` для виведення сформованого речення:
`fraza = Entry(text = "", width = 35)`
8. Створіть командну кнопку `Скласти фразу`:
`btn1 = Button(text = 'Скласти фразу', width = 20, font = 'Arial 12',
command = btn1_click)`
9. Створіть кнопки `Перекласти`, `Вибрати випадково`.
10. Запрограмуйте процедуру обробки події для командної кнопки `btn1`. Рядок `s` створюється як результат конкатенації рядків, вибраних в елементах `combo1...combo3`, між якими додаються пропуски. Отримане значення виводиться в текстове поле `fraza`.
`def btn1_click():`
 `s = combo1.get()+' '+combo2.get()+' '+combo3.get()`
 `fraza.delete(0, END)`
 `fraza.insert(0, s)`
11. Запрограмуйте процедуру обробки події для командної кнопки `btn2`. Рядок `s` створюється як результат конкатенації значень, що відповідають ключам, отриманим із віджетів `combo1...combo3`. Отримане значення виводиться в текстове поле `fraza`.
 `s = c1[combo1.get()+' '+c3[combo3.get()+' '+c2[combo2.get()]`
12. Запрограмуйте можливість випадкового формування речення. Створіть процедуру обробки події для командної кнопки `btn3`. Задайте індекс вибраного елемента випадковим чином:
 `combo1.current(randint(0, 4))`
Випробуйте програму. Перевірте роботу кнопок.



Зробіть висновок: які переваги надає використання словників і розкривних списків.

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.

Виняток — помилка, яка не є синтаксичною і спричиняє аварійне припинення виконання програми.

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

Глибина кольору — кількість бітів, які використовуються для кодування певного кольору пікселя растрового зображення.

Графічний компонент — об'єкт певного класу, що має властивості та методи.

Декодування — перетворення закодованого повідомлення у форму, прийнятну для приймача.

Дискретизація сигналу — процес кодування неперервного сигналу послідовністю окремих значень його рівня.

Домен — певна зона в системі доменних імен, виділена власникові домену з метою забезпечення доступу через інтернет до інформації, що належить власникові домену.

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

Захоплення відео — процес перетворення відеосигналу із зовнішнього пристрою в цифровий формат і запису у відеофайл.

Захоплення звуку — процес перетворення звукового сигналу із зовнішніх джерел у цифровий формат і запису в аудіофайл.

Значення — характеристика величини, яка може багаторазово змінюватися в процесі опрацювання інформації.

Клас — опис об'єктів певного типу.

Кліп — відео- або аудіофрагмент, розташований на доріжці монтажного столу.

Код — набір символів або сигналів і правил їх використання для кодування повідомлень.

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 в потрібний формат.

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.

Методи класів — функції, що призначені для змінювання властивостей об'єктів — екземплярів цих класів.

Мультимедіа — комбінування в одному інформаційному продукті різних форм подання інформації: текстової, графічної, звукової, відео, з можливістю її інтерактивного використання.

Наслідування — це можливість створення нового класу (нащадка) на базі вже наявного (базового).

Подія — зовнішній вплив на елемент керування, на який цей елемент може реагувати.

Провайдер — організація, яка надає послуги, пов'язані з доступом до глобальної мережі інтернет.

Проект — це набір файлів, необхідних для створення програми в ІСР.

Розділ — структурна складова документа зі сталими налаштуваннями Word, яка завершується недрукованим символом розриву розділу.

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

Структура документа — ієрархічна схема розміщення складових документа.

Тег — записана в кутових дужках <> позначка мови HTML, яку опрацьовує браузер.

Титри — написи, які з'являються на початку або наприкінці відеофільму, на певному відеофрагменті або зображенні тощо.

Транслятор — програма, яка перекладає програму з мови програмування високого рівня на мову машинних кодів.

Функція — поймає блок програмного коду, до якого можна звернутися з будь-якого місця програми необхідну кількість разів.

Хостинг — можливість використання дискового простору сервера, постійно під'єданого до мережі, для розміщення, функціонування та обслуговування сайтів, файлових сховищ, електронної пошти тощо.

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

Алфавітний покажчик

А

Адаптер 30
Аудіоредактор 97

Б

Байт 12
Батьківський віджет 174

В

Величина 193
Виняток 163
Вкладений цикл 225
Властивості 182

Г

Гіпертекст 135
Глибина кольору 15

Д

Декодування 8
Дискретизація сигналу 86
Доменне ім'я 127

З

Захоплення відео 102
Захоплення звуку 96
Зміст 70

К

Клас 168
Код 8
Кодування 8
Колонтитул 66
Конвертер 104

М

Мережевий протокол 122
Мова програмування 155
Мультимедіа 32, 85

О

Область видимості змінної 191

П

Пам'ять комп'ютера 29
Парадигма програмування 156
Повторення 222
Подія 177
Показчик 72
Посилання 68
Провайдер 122
Проект 158

Р

Розділ 65
Рядок 215

С

Сайт 130
Складена умова 211
Словник 228
Стиль 63
Структура документа 61

Т

Тег 139
Титри 111

У

Успадкування 170

Ф

Функція 177

Х

Хмарний диск 76
Хостинг 130

Ч

Частота дискретизації 87

Ц

Цифровий звукозапис 85

Ш

Шаблон 64

Зміст

Передмова	3
Розділ 1. Кодування даних та апаратне забезпечення	
§ 1. Кодування та декодування повідомлень	7
§ 2. Двійкове кодування	10
§ 3. Кодування графічних даних	14
<i>Практична робота 1. Визначення довжини двійкового коду текстових даних</i>	20
§ 4. Історія опрацювання інформаційних об'єктів	21
§ 5. Архітектура комп'ютера. Технічні характеристики складових комп'ютера	27
<i>Практична робота 2. Визначення конфігурації комп'ютера під потребу</i>	34
Розділ 2. Опрацювання текстових даних	
§ 6. Середовище текстового процесора	37
§ 7. Робота з фрагментами тексту	44
§ 8. Форматування об'єктів текстового документа	50
<i>Практична робота 3. Створення текстового документа, що містить об'єкти різних типів</i>	57
§ 9. Структура складного текстового документа	59
§ 10. Зміст документа	68
<i>Практична робота 4. Автоматизоване створення змісту документа</i>	72
§ 11. Опрацювання документів на хмарному диску	74
<i>Практична робота 5. Створення документа на хмарному диску для спільного опрацювання</i>	82
Розділ 3. Опрацювання мультимедійних об'єктів	
§ 12. Поняття мультимедіа. Кодування аудіоданих	85
§ 13. Кодування відеоданих	89
<i>Практична робота 6. Використання аудіо- й відеоданих у програмах мовою Python</i>	94
§ 14. Програмне забезпечення для опрацювання звуку	96
§ 15. Програмне забезпечення для опрацювання відео	102
§ 16. Побудова відеоряду	107
§ 17. Опрацювання, зберігання та розміщення відеофільму в інтернеті	113

<i>Практична робота 7.</i> Створення відеофільму	117
<i>Практична робота 8.</i> Розміщення відеоматеріалів в інтернеті ...	118
Розділ 4. Створення та публікація вебресурсів	
§ 18. Обмін даними в інтернеті.....	121
§ 19. Адресація в інтернеті	126
§ 20. Створення сайтів	130
<i>Практична робота 9.</i> Конструювання сайтів із використанням системи керування змістом	135
<i>Практична робота 10.</i> Наповнення сайту вмістом	138
§ 21. Розмічання тексту засобами HTML.....	139
§ 22. Графічні зображення та гіперпосилання на вебсторінці....	143
§ 23. Розмічання таблиць засобами HTML.....	146
<i>Практична робота 11.</i> Створення головної сторінки сайту засобами HTML	150
Розділ 5. Алгоритми та програми	
§ 24. Сучасні мови програмування. Система програмування PyCharm	155
§ 25. Налаштування програмного коду.....	163
§ 26. Об'єкти, їх властивості та методи	168
§ 27. Графічний інтерфейс користувача	173
§ 28. Обробники подій	177
§ 29. Організація діалогу користувача з програмою.....	182
<i>Практична робота 12.</i> Введення і виведення даних.....	187
§ 30. Функції користувача.....	189
§ 31. Величини. Числові типи даних	193
§ 32. Математичні функції.....	198
§ 33. Рядковий тип даних	202
§ 34. Робота з текстовими файлами	207
§ 35. Логічний тип даних. Умовні оператори	211
§ 36. Елементи вибору	217
§ 37. Алгоритми з повтореннями. Цикли	222
§ 38. Словники. Віджет Combobox.....	228
<i>Практична робота 13.</i> Створення програми з використанням словників і списків	233
Комп'ютерний словник	235
Алфавітний покажчик.....	237

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»

Редактор *Л. А. Каюда*. Художнє оформлення *В. І. Труфена*.
Технічний редактор *А. В. Пліско*. Коректор *Н. В. Красна*.

Підписано до друку 28.05.2021. Формат 70×90/16.

Папір офсетний. Гарнітура Шкільна. Друк офсетний.

Ум. друк. арк. 17,55. Обл.-вид. арк. 14,2.

Тираж 58 689 прим. Зам. № 2906-2021.

ТОВ Видавництво «Ранок»,

вул. Кібальчича, 27, к. 135, Харків, 61071.

Свідоцтво суб'єкта видавничої справи ДК № 5215 від 22.09.2016.

Адреса редакції: вул. Космічна, 21а, Харків, 61165.

E-mail: office@ranok.com.ua. Тел. (057) 719-48-65, тел./факс (057) 719-58-67.

Підручник надруковано на папері українського виробництва

Надруковано у друкарні ТОВ «ТРИАДА-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: sale@triada.kharkov.ua