

5

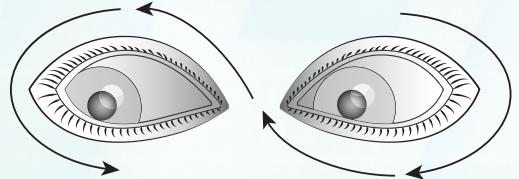
# ІНФОРМАТИКА



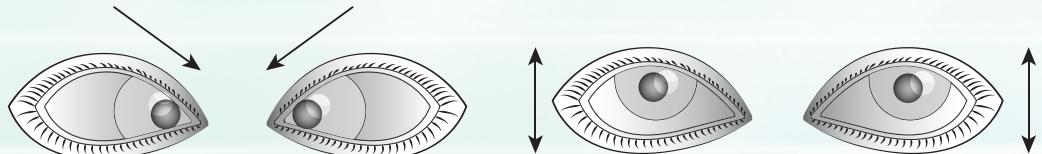
# Бережіть здоров'я!

## Комплекс вправ для очей

1. «Намалюйте» очима вісімку на протилежній стіні.

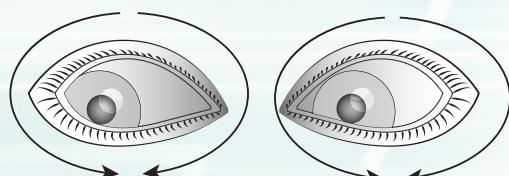


2. Подивітесь на кінчик носа, а потім на брови.

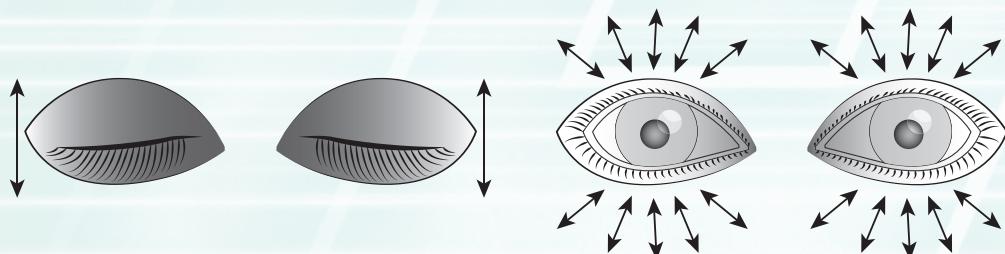


3. Сконцентруйте зір на віддаленому об'єкті (наприклад на лампочці на стелі) а потім — на близькому (наприклад на пальці перед обличчям).

4. Зробіть колові рухи очима вліво і вправо.



5. Міцно заплющте, а потім широко розплющте очі.



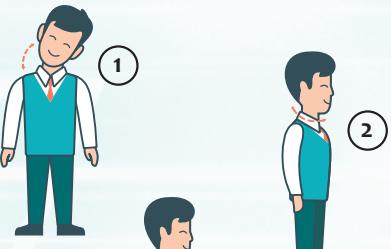
Виконуйте цей комплекс щонайменше двічі на день.  
Кожну вправу повторюйте 5 разів.

## Правильно сидіть за комп'ютером



## Комплекс вправ для збереження здоров'я

1. Повільно нахиліть голову до правого плеча, потім до лівого.



2. Повільно нахиліть голову вперед, потім назад.



3. Поверніть голову вправо, потім уліво.



4. Підніміть руки вверх, зчепіть пальці та потягніться.



5. Зчепивши руки в замок, зробіть руками кругові рухи.

Кожну вправу повторюйте 5 разів.

6. Зробіть плавні нахили тулуба назад.

**5**

# **ІНФОРМАТИКА**

**Підручник для 5 класу  
закладів загальної середньої освіти**

**Рекомендовано  
Міністерством освіти і науки України**

**Харків  
Видавництво «Ранок»  
2018**

УДК [004:37.016](075.3)  
I-74

**Авторський колектив:**  
О. О. Бондаренко, В. В. Ластовецький,  
О. П. Пилипчук, Є. А. Шестопалов

**Рекомендовано Міністерством освіти і науки України**  
(наказ Міністерства освіти і науки України від 11.10.2018 № 1085)

Видано за рахунок державних коштів. Продаж заборонено

I-74     **Інформатика** : підруч. для 5 кл. закл. загал. серед. освіти / [О. О. Бондаренко, В. В. Ластовецький, О. П. Пилипчук, Є. А. Шестопалов]. — Харків : Вид-во «Ранок», 2018. — 160 с. : іл.

ISBN 978-617-09-4781-9

УДК [004:37.016](075.3)



**Інтернет-підтримка**  
Електронні матеріали  
до підручника розміщено на сайті  
[interactive.ranok.com.ua](http://interactive.ranok.com.ua)

ISBN 978-617-09-4781-9

© Бондаренко О. О., Ластовецький В. В.,  
Пилипчук О. П., Шестопалов Є. А., 2018,  
© ТОВ Видавництво «Ранок», 2018

# ДОРОГІ П'ЯТИКЛАСНИКИ ТА П'ЯТИКЛАСНИЦІ!

Ви тримаєте в руках підручник з інформатики, призначений саме для вас, учнів і учениць 5 класу. У початковій школі ви дізналися про базові поняття інформатики, навчилися працювати з комп’ютером, створювати графічні й текстові документи, шукати в Інтернеті інформаційні матеріали, маєте поняття про алгоритми та їх виконавців.

У цьому навчальному році на уроках інформатики вас очікує чимало цікавого та корисного, а пропонований підручник буде вашим надійним помічником. Як же з ним працювати?

Підручник складається із чотирьох розділів. На початку кожного розділу ви знайдете рубрику «Повторюємо». Вона допоможе вам згадати відомості, які ви вивчали в попередніх класах і які будуть корисні для засвоєння нового матеріалу.

Розділ 1 «Інформаційні процеси та системи» присвячений значною мірою повторенню та закріпленню матеріалу, вивченого в початковій школі, щоб ви змогли успішно засвоїти подальші розділи, розділ 2 «Мережеві технології та Інтернет» присвячений пошуку інформації в Інтернеті, розділ 3 «Опрацювання текстових даних» — створенню текстових документів ізображеннями й таблицями, а розділ 4 «Алгоритми та програми» — ознайомленню з алгоритмами й основними поняттями мови програмування Python.

Розділ підручника складається з параграфів і практичних робіт за темами розділу. Кожен параграф містить теоретичні відомості за темою уроку, приклади використання отриманих знань, питання для самоперевірки, вправу, комп’ютерний тест.

«Питання для самоперевірки» допоможуть з’ясувати, чи зрозуміли ви вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт за комп’ютером. «Вправа» складається з теоретичних і практичних завдань. Виконуючи ці завдання, ви навчитеся краще працювати з комп’ютером. Оцінити свої



знання, уміння та навички вам допоможе «*Комп'ютерне тестування*» з автоматичною перевіркою відповідей. Його можна пройти на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua) або в онлайн-режимі, користуючись програмою, запропонованою вчителем.

У підручнику ви знайдете інструкції до практичних робіт. Щоб виконати кожну з них, потрібно повторити матеріал, вивчений на попередніх уроках, — тоді ви зможете успішно застосувати свої знання, виконуючи завдання за комп'ютером.

У тексті підручника використано такі позначки:

| Запам'ятайте

| Розгляньте приклад

| Зверніть увагу

| Знайдіть відповідь у підручнику

| Знайдіть відповідь в Інтернеті

| Виконайте практичне завдання за комп'ютером

| Виконайте тестові завдання за комп'ютером, використовуючи матеріали сайта «Інтерактивне навчання»

| Виконайте завдання підвищеної складності

| Виконайте завдання в парах

*Бажаємо натхнення та успіхів!*

# РОЗДІЛ 1

## ІНФОРМАЦІЙНІ ПРОЦЕСИ ТА СИСТЕМИ



- § 1. Безпека під час роботи з комп'ютером
- § 2. Інформаційні процеси та системи
- § 3. Апаратна та програмна складові інформаційної системи
- § 4. Комп'ютер як пристрій для опрацювання даних
- § 5. Операційна система та її інтерфейс
- § 6. Операції над файлами та папками

Практична робота 1. Операції над файлами та папками

## ПОВТОРЮЄМО



На уроках інформатики у 2–4 класах ви ознайомилися з поняттям інформації, знаєте, як інформацію класифікують: за способами сприйняття — візуальна (зорова), аудіальна (слухова), нюхова, смакова, тактильна (дотикова); за способом подання — текстова, графічна, числовая, звукова, відео тощо.

Ви маєте уявлення про основні інформаційні процеси, тобто про дії, які можна виконувати з інформацією: передавати, приймати, зберігати, захищати, опрацьовувати.

Для опрацювання інформації використовують комп’ютер, який має *апаратну* та *програмну складові*. Є найрізноманітніші комп’ютери: від кишенькових до суперком’ютерів.

Ви розумієте, що призначенням комп’ютера є опрацювання інформації, і знаєте, що для цього потрібне відповідне програмне забезпечення.

1. Які бувають види інформації за способом її сприйняття людиною?
2. У який спосіб можна подати інформацію?
3. Які ви знаєте інформаційні процеси?
4. Де використовують комп’ютери?
5. Для чого призначено програмне забезпечення?



У цьому розділі ви детальніше ознайомитеся з будовою та роботою комп’ютера; навчитеся безпечно та грамотно працювати з операційною системою — комплексом програм для керування комп’ютером.



## § 1. Безпека життєдіяльності під час роботи з комп'ютером

Ви вже вмієте користуватися комп'ютером. Але не зайвим буде повторити основні правила та норми, яких слід дотримуватися в комп'ютерному класі (рис. 1.1).

- !** Будь-які правила безпеки та норми поведінки в класі базуються на таких принципах:
- не нашкодь собі;
  - не нашкодь і не заважай іншим;
  - не зіпсуй обладнання і пристрої.



Рис. 1.1

Комп'ютер сконструйовано так, щоб не допустити випадкового ураження користувача електричним струмом. Однак лише грамотне використання та дисциплінованість гарантують безпечну роботу на комп'ютері.

Поміркуйте, чи можна гарантувати безпеку «умільцям», які самовільно «ремонтують» комп'ютер. Чи можна гарантувати безпеку дітям, які бігають і штовхаються в комп'ютерному класі?

Джерелом небезпеки можуть бути розетки з розбитими корпусами (рис. 1.2), дроти з пошкодженою ізоляцією, кабелі на підлозі, обладнання комп'ютерів зі знятими кришками.



Рис. 1.2

Кишеневий комп'ютер (мобільний телефон, смартфон та ін.) також може бути небезпечним. Через неправильне користування або низьку якість іноді вибухають акумулятори цих пристрій, спричиняючи опіки й навіть пожежі.

## Правила поведінки в комп'ютерному класі

Комп'ютерний клас, у якому встановлено багато комп'ютерів, є приміщенням підвищеної небезпеки. Тому практичні заняття можуть проходити лише в присутності й під керівництвом учителя або лаборанта.

У комп'ютерному класі потрібно дотримуватися таких **правил безпеки**:

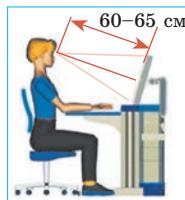
- заходити в клас спокійно, не штовхатися, не бігти, щоб зайняти «країці» комп'ютер;
- складати портфелі у відведеному місці; брати із собою на робоче місце лише необхідні речі;
- сідати тільки на вказане вчителем робоче місце;
- вмикати й вимикати комп'ютери лише з дозволу лаборанта або вчителя;
- працювати чистими й сухими руками;
- повідомляти вчителю про негаразди в роботі комп'ютерів та нестандартні ситуації;
- у разі небезпечної ситуації організовано залишити комп'ютерний клас.

**!** Учням категорично забороняється виконувати будь-який ремонт комп'ютерів та іншого електрообладнання.

## Правила роботи за комп'ютером

Чи дотримуєтесь ви правил безпечної роботи за комп'ютером у домашніх умовах?

Під час роботи за комп'ютером спина повинна мати опору, лінія погляду — перпендикулярна до екрана (рис. 1.3), відстань від очей до екрана монітора — не менш ніж 60–65 см (на довжину руки).



ПРАВИЛЬНО



НЕПРАВИЛЬНО

Рис. 1.3

Час безперервної роботи за комп’ютером не має перевищувати 15 хвилин. Потім необхідно відволіктися на інший вид діяльності або відпочити.

За висновками вчених, під час роботи за комп’ютером людина в 10–15 разів рідше моргає й рухає очима, ніж зазвичай. Це призводить до пересихання рогівки ока та перенапруження очних м’язів. Працюючи за комп’ютером, потрібно час від часу виконувати комплекс вправ для очей (рис. 1.4).

Вправи потрібно виконувати сидячи, відвернувшись від комп’ютера: спину тримати прямо, очі розплющені, погляд — прямо перед собою. Кожну вправу слід виконувати п’ять разів.

### Комплекс вправ для очей

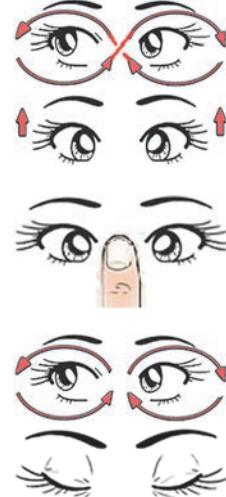
1. «Намалюйте» очима вісімку на протилежній стіні.
  2. Подивіться на кінчик носа, а потім на брови.
  3. Сконцентруйте зір на віддаленому об’єкті (наприклад на лампочці на стелі), а потім — на близькому (наприклад на пальці перед обличчям).
  4. Зробіть колові рухи очима вліво і вправо.
  5. Міцно заплющте, а потім широко розплющте очі.
- 

Рис. 1.4

### Питання для самоперевірки



1. Яких правил поведінки в комп’ютерному класі ви дотримуєтесь?
2. Чого не можна робити учням у комп’ютерному класі?
3. Що може бути джерелом небезпеки в комп’ютерному класі?
4. Як правильно сидіти за комп’ютером?



5. Назвіть безпечний, на вашу думку, час безперервної роботи за комп'ютером для школярів.
6. Яка має бути відстань від очей користувача до екрана монітора?
7. Чому потрібно виконувати комплекс вправ для очей?
8. Наведіть власні приклади вправ для очей.

**Вправа 1**

► Виконайте завдання за комп'ютером.

- 1) Завантажте клавіатурний тренажер і потренуйтесь, набираючи пропонований текст українською мовою. Повідомте результат учителю.
- 2) Завантажте графічний редактор. Намалюйте комп'ютер, схожий на той, який стоїть на вашому робочому місці, і позначте на ньому місця, що можуть бути небезпечними. Додайте підписи, що саме є джерелом небезпеки.

**Комп'ютерне тестування**

Виконайте тестове завдання 1 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 2. Інформаційні процеси та системи

Світ, що нас оточує, складається з предметів і явищ, які називають об'єктами. Кожен об'єкт має певні властивості. З об'єктами відбуваються певні дії.

### Інформація та повідомлення

Інформація та повідомлення є основними поняттями інформатики.

Найбільш узагальненим означенням поняття інформація можна вважати наступне.



**Інформація** — це відомості про об'єкти Всесвіту.

У різних сферах людської діяльності застосовують також інші більш вузькі означення цього терміна.

У побуті інформацією вважають лише ті відомості, які є для людини новими й корисними (рис. 2.1).

У Законі України «Про інформацію» вживается таке означення цього поняття: інформація — це будь-які відомості чи дані, які можуть бути збережені на матеріальних носіях або відображені в електронному вигляді.

Повідомлення — це форма існування інформації. Оскільки без повідомлень не можна сприйняти інформацію, то кажуть, що інформація передається за допомогою повідомлень.



Рис. 2.2

Людина отримує повідомлення органами чуття (зір, слух, нюх, смак, дотик тощо) (рис. 2.2), технічні пристрої — за допомогою різних датчиків і сенсорів. Деякі тварини мають органи чуття для сприйняття інформації, недоступної для людини.

- 1 Змії особливими органами сприймають теплове (інфрачервоне) випромінювання. Кити можуть створювати й сприймати наднизькі звуки (інфразвуки), а кажани — надвисокі (ультразвуки), яких людина не чує. Завдяки таким звукам кажани навіть «перемовляються» та орієнтуються в просторі.

Людина має й особливі органи чуття. Наприклад, завдяки вестибулярному апарату ви навіть із заплющеними очима легко визначаєте положення свого тіла в просторі (нахили, повороти, перевороти).



Рис. 2.1

## Класифікація інформації

Інформацію можна класифікувати за різними ознаками. Розглянемо деякі з них.

Ознаки	Види інформації	Приклади
За способом сприйняття	Візуальна, аудіальна, нюхова, смакова, тактильна	Інформація, яку ви сприймаєте, слухаючи вчителя, є аудіальною (звуковою)
За формою подання	Текстова, числовая, графічна, звукова, відео, комбінована	Інформація, яку ви сприймаєте, читаючи цей підручник, є текстовою
За призначенням	Масова, спеціальна, особиста, розважальна, наукова, медична тощо	Дивлячись новини по телебаченню, ви сприймаєте масову інформацію
За корисністю	Корисна, непотрібна, шкідлива	Коли ви вирішуєте зранку, чи брати із собою парасольку, інформація Українського гідрометеоцентру є корисною

## Інформаційні процеси

Як ви знаєте, з інформацією можна виконувати певні дії.



**Інформаційні процеси** — це дії над інформацією.

До основних інформаційних процесів належать *передавання, отримання, зберігання, захист та опрацювання інформації*.

- 2 Коли вчитель пояснює новий матеріал, то він передає інформацію, а ви її отримуєте. Коли ви записуєте щось у зошит — зберігаєте інформацію (рис. 2.3).



Рис. 2.3

Існує є багато інших інформаційних процесів.

*Передавання* інформації багатьом приймачам називають її поширенням.

*Захист* інформації полягає або в обмеженні її передавання, або в особливому кодуванні — шифруванні.

*Зберігання* великої кількості інформації називають накопиченням.

*Опрацювання* інформації полягає у створенні нової інформації шляхом перетворення початкової інформації. Наприклад, людина складає вірш під враженням від дивовижної картини світанку.

Інформацію можна використовувати. Наприклад, застосовуючи знання з інформатики, ви можете скласти комп’ютерну програму. Інформацію можна також оцінювати, ототожнювати, тлумачити, зіставляти, обмірювати тощо.

## Дані

Передавати, приймати, зберігати, опрацьовувати тощо можна лише інформацію, подану певним чином, наприклад, у вигляді звуків, зображенень, слів, ієрогліфів, електричних, світлових та інших сигналів.



**Дані** — це інформація, подана в зручній для виконання інформаційних процесів формі.

Скажімо, інформація у вигляді наскельного малюнка в печері може зберігатися тисячоліттями, але непридатна для передавання (рис. 2.4). Текстові дані на паперових носіях (книжки, журнали, газети тощо) значно легше передавати, хоча зберігання даних на них не таке тривале.



Рис. 2.4

Сучасні засоби опрацювання й передавання електронних повідомлень не можуть зберігати інформацію на своїх носіях навіть упродовж кількох десятків років. Лише завдяки надзвичайно легкому та швидкому копіюванню й розповсюдженю інформації сучасними засобами її можна вважати практично незнищеною.



- 3 Тривають розробки щодо збільшення довговічності й самих носіїв даних. У 2013 році вчені Інституту проблем реєстрації інформації Національної академії наук України розробили технологію зберігання даних на оптичних сапфірових дисках, які можуть зберігати дані впродовж десятків тисяч років (рис. 2.5).



Рис. 2.5

## Інформаційні системи

Слово «система» перекладається з грецької як ціле, що складається із частин.



**Інформаційна система** — це сукупність взаємопов'язаних засобів і методів здійснення інформаційних процесів.

**Засоби інформаційної системи** — це об'єкти живої та неживої природи. Засобами можуть бути біологічні клітини, люди, вода, механічні й електронні пристрої тощо (рис. 2.6).

**Методи інформаційної системи** — це способи дії (взаємодії) об'єктів. Методами можуть бути танці бджіл задля розповсюдження інформації серед бджолиного рою (рис. 2.7), усне спілкування людей тощо.



Рис. 2.6



Рис. 2.7



Рис. 2.8

На рис. 2.8 зображені інформаційну систему — учнівський клас на уроці малювання. Засобами такої системи є вчитель та учні, класне приміщення, дошка, крейда (маркери), олівці, проектор, репродукції картин, аркуші паперу та ін. Методами такої системи є усне пояснення вчителя, демонстрування та обговорення репродукцій картин, презентація, відтворення учнями малюнків олівцями на папері тощо.

## Роль інформаційних технологій у житті сучасної людини

Слово «технологія» з грецької перекладається як майстерність, уміння. На відміну від технології матеріального виробництва, інформаційна технологія зорієнтована на роботу з інформацією.



**Інформаційна технологія** — сукупність методів і процесів опрацювання інформації із застосуванням комплексу відповідних засобів.

Стрімкий розвиток сучасних інформаційних технологій став можливим завдяки поєднанню досягнень у таких галузях, як інформатика, математика, електроніка, психологія, філософія, економіка та ін. Розглянемо кілька пристрій для опрацювання даних, які набули найбільшого поширення.

Пристрій		Призначення
Комп'ютер		Здійснення майже всіх інформаційних процесів
Програвач (плеєр)		Відтворення аудіо- або відеозаписів
Калькулятор		Виконання операцій над числами
Фото- та відеокамера		Фіксування нерухомих або рухомих зображень
Навігатор GPS		Визначення поточного місця розташування пристрою на Землі, прокладання маршруту та адресний пошук

## ► РОЗДІЛ 1. Інформаційні процеси та системи

Завдяки розвитку сучасних інтернет-технологій світ став значно «прозорішим» і «тіснішим». Так, за кілька секунд ви можете зв'язатися телефоном з абонентом на іншій півкулі Землі (рис. 2.9), отримати електронного листа, поспілкуватися в режимі відео з будь-яким користувачем, пограти в комп’ютерну гру, «помандрувати» світом за допомогою геосервісів, дізнатися про новини, прогноз погоди, рецепт смачної страви, заплатити за комунальні послуги та багато-багато іншого.



Рис. 2.9

### Питання для самоперевірки



1. Що таке інформація, повідомлення, дані?
2. Наведіть приклади інформації.
3. Назвіть основні інформаційні процеси.
4. Назвіть носії для зберігання інформації.
5. Якими засобами передають інформацію?
6. Наведіть приклади опрацювання інформації.
7. Що таке інформаційна технологія?
8. Опишіть роль інформаційних технологій у житті людини.

### Вправа 2



► Виконайте завдання за комп’ютером.

- 1) Завантажте клавіатурний тренажер і потренуйтесь набирати пропонований текст українською та іноземною мовами. Повідомте результат учителю.
- 2) За допомогою графічного редактора підготуйте колаж із зображень комп’ютера і ще двох пристрій, згаданих у параграфі. Зображення відшукайте в Інтернеті або візьміть із папки, запропонованої вчителем.



### Комп’ютерне тестування



Виконайте тестове завдання 2 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 3. Апаратна та програмна складові інформаційної системи

Як ви знаєте, інформаційна система забезпечує приймання, перетворення, опрацювання, збереження інформації та передавання результатів опрацювання споживачу: людині, машині, іншій інформаційній системі. Комп'ютер є прикладом технічної інформаційної системи.

### Складові комп'ютера та їх призначення

Персональний комп'ютер складається з апаратної частини і програмного забезпечення.

**Апаратна частина** (від англ. *hardware* — тверда частина) складається зі з'єднаних між собою різноманітних пристрій, які можна побачити, доторкнутися.

**Програмне забезпечення** (від англ. *software* — м'яка частина) складається з програм, установлених на комп'ютері та призначених для забезпечення його роботи.

! Пристрой апаратної частини комп'ютера не можуть опрацьовувати дані за відсутності програмного забезпечення. Саме взаємодія апаратної та програмної складових лежить в основі роботи комп'ютера.

### Апаратна складова комп'ютера

Більшість сучасних комп'ютерів побудовано за принципами, які обґрунтував американський учений Джон фон Нейман (рис. 3.1) ще в 1948 році. Відтоді змінився зовнішній вигляд комп'ютерів, з'явилися нові пристрої. Але загальна схема роботи комп'ютера лишилась без змін.

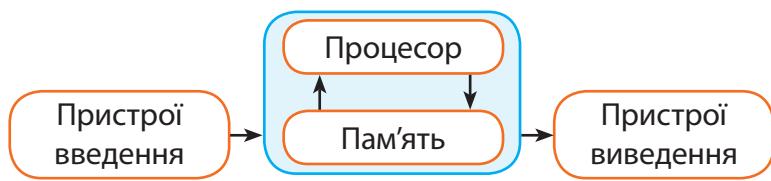


Рис. 3.1

## ► РОЗДІЛ 1. Інформаційні процеси та системи



Апаратну складову комп'ютера можна розподілити на пристрой введення, пристрой виведення, пристрой опрацювання, пристрой зберігання (рис. 3.2).



Рис. 3.2

**Пристрої введення** призначені для введення даних у комп'ютер. Основними пристроями введення інформації є *клавіатура* і *миша*.

- 1 Для введення текстової і графічної інформації також застосовується *електронно-оптичний сканер*, для введення звукової інформації — *мікрофон*, для введення команд під час роботи комп'ютерних ігор і тренажерів — *джойстик*.

**Пристрої виведення** призначені для виведення результатів опрацювання даних на комп'ютері у вигляді чисел, таблиць, текстів, графіків, малюнків, креслень, анімації, звуків, тобто у ви-

гляді, зручному для сприйняття людиною. Основним пристроєм виведення інформації є *монітор*.

- 2** Друкування на невеликих аркушах паперу текстів, малюнків і креслень виконується за допомогою *принтера*, а креслень на великих аркушах — за допомогою *плотера*. Для виведення звуку використовують *звукові колонки*.

Пристрої введення-виведення називають *зовнішніми пристроями* комп’ютера.

**Пристроем опрацювання** є *процесор*. Окрім опрацювання даних він виконує функцію керування пристроями комп’ютера. Процесор складається з арифметико-логічного пристрою, де виконуються всі арифметичні й логічні операції, і пристрою керування, який координує взаємодію всіх пристройів комп’ютера під час їх роботи.

**Пристрої зберігання** (пам’яті) призначені для тимчасового або постійного зберігання даних та програм, за допомогою яких виконується їх опрацювання.

Пристрої зберігання поділяють на внутрішні та зовнішні.

Розглянемо популярні зовнішні пристрої зберігання.

- *Накопичувач на магнітних дисках* (вінчестер) (рис. 3.3), призначений для зберігання великого обсягу даних, складається з одного або кількох дисків із магнітним покриттям. Механізм вінчестера й самі диски містяться в корпусі, що зазвичай установлюється в системний блок комп’ютера.
- *Оптичний диск* — це диск із прозорого пластику з непрозорим шаром, чутливим до лазерного променя. Диск обертається в дисководі й під дією лазерного променя вздовж спіральної доріжки утворюються мікроскопічні непрозорі ділянки — таким чином здійснюється запис. Потім за допомогою лазерного променя, що по-різному відбивається від робочого шару, проводиться зчитування інформації (рис. 3.4).



Рис. 3.3

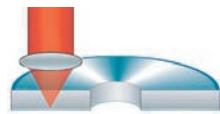


Рис. 3.4



- 3** На оптичні диски типів CD-R, DVD-R проводиться одноразовий запис, а на CD-RW, DVD-RW — багаторазовий запис.
- *Пристрій флеш-пам'яті*, на відміну від інших носіїв даних, не містить рухомих деталей, запам'ятування здійснюється на основі електричних зарядів. Пристрій має невелику масу й малі розміри.

## Програмне забезпечення комп’ютера



**Програмне забезпечення** — це набір різноманітних програм, які керують роботою комп’ютера, забезпечують ведення діалогу з користувачем, за їх допомогою обробляють різноманітні дані, створюють нові програми.

Програмне забезпечення комп’ютера складається із системного, службового та прикладного програмного забезпечення (рис. 3.5).

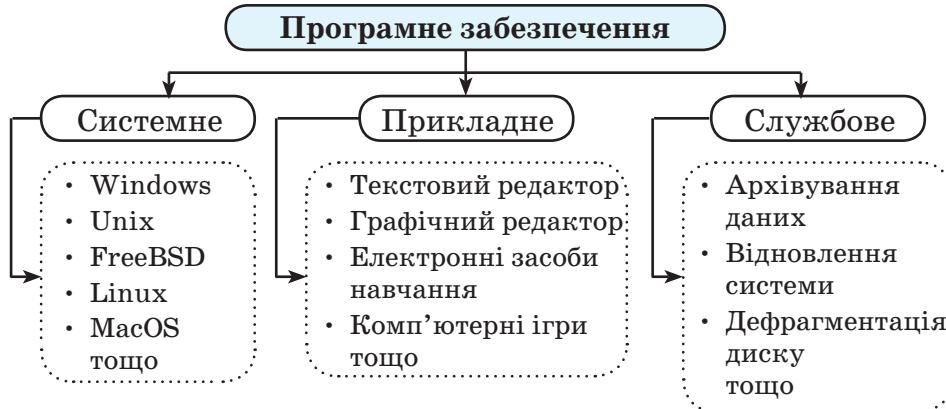


Рис. 3.5

До системного програмного забезпечення належать операційна система, системні програми, які виконують функції керування ресурсами комп’ютера.

**Операційна система** — це набір програм, що автоматично завантажуються під час вмикання комп’ютера й використовуються для керування обчислювальними процесами, програмними й апаратними ресурсами.

До **прикладного програмного забезпечення** належать програми, що безпосередньо забезпечують виконання необхідних користувачу робіт.

Прикладне програмне забезпечення розподіляють на програми загального призначення (текстові та графічні редактори, програми для роботи в мережі тощо); програми спеціального призначення (використовують у медицині, поліції, магазинах, банках тощо); інструментальні програмні засоби (програми для створення інших програм).

До **службового програмного забезпечення** належать так звані утиліти — службові програми для обслуговування дисків, створення та підтримки архівів, боротьби з вірусами тощо.

### Питання для самоперевірки



1. Що таке «інформаційна система»?
2. Назвіть складові комп'ютера.
3. Як взаємодіють апаратна та програмна частини комп'ютера?
4. Що входить до програмного забезпечення комп'ютера?
5. Яке призначення операційної системи?
6. Які пристрої становлять апаратну частину комп'ютера?
7. Назвіть пристрої введення інформації.
8. Які пристрої призначенні для виведення інформації?



### Вправа 3



- Виконайте завдання за комп'ютером.
- 1) Завантажте клавіатурний тренажер і потренуйтесь набирати пропонований текст українською та іноземною мовами. Повідомте результат учителю.
  - 2) За матеріалом параграфа побудуйте схему, яка показує класифікацію пристройів комп'ютера. Яку програму використано? Чи зручно вносити зміни в отриману схему?



### Комп'ютерне тестування

Виконайте тестове завдання 3 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 4. Комп'ютер як пристрій для опрацювання даних

Сучасний світ неможливо уявити без комп'ютерної техніки. У перекладі з англійської мови слово «комп'ютер» означає «обчислювач», проте зараз він застосовується не тільки для обчислень.

### Використання комп'ютерів

Комп'ютер — це універсальний пристрій для опрацювання інформації. Слово «універсальний» означає, що комп'ютер може застосовуватися для опрацювання, зберігання й передавання інформації, поданої в різних форматах — числовому, графічному, текстовому, звуковому.

Комп'ютери використовують у найрізноманітніших галузях людської діяльності.

За допомогою комп'ютерів можна, наприклад, здійснювати прогнозування погоди, планування та управління виробництвом, проведення медичних досліджень (комп'ютерний томограф (рис. 4.1)), продаж залізничних і авіаквитків, обслуговування різних заходів; підготовку до видання газет, журналів і книжок, цифровий аудіо- і відеозапис, створювати відео- й анімаційні фільми, розробляти ігрові та навчальні програми.

Завдяки комп'ютерам функціонують банківські, біржові, криміналістичні бази даних.



Рис. 4.1

## Види комп'ютерів

Сучасні комп'ютери можна класифікувати за різними ознаками. Якщо розглядати комп'ютери, призначенні для персонального користування (ПК), то їх можна класифікувати таким чином.

Стаціонарні комп'ютери		
Настільний комп'ютер (десктоп)		Складається із системного блоку, до якого під'єднано монітор, клавіатуру й мишу
Моноблок		Системний блок і монітор складають єдине ціле
Мікрокомп'ютер		Розміщується в мініатюрному корпусі, не має пристрою виведення, тому приєднується до монітора. Миша або клавіатура під'єднується через убудовані USB-порти або Bluetooth
Портативні комп'ютери		
Ноутбук та нетбук		У корпусі об'єднано дисплей, клавіатуру, тачпад і акумуляторні батареї
Планшетний ноутбук та планшет		Корпус складається з дисплея із сенсорним екраном, є клавіатура, яка або складається, або висувається з ніші під екраном. У планшетів засобом уведення — сенсорний екран
Кишеневковий персональний комп'ютер та смартфон		Мають малі розміри та великий запас часу автономної роботи акумулятора. Засіб уведення — сенсорний екран або висувна клавіатура

## ► РОЗДІЛ 1. Інформаційні процеси та системи

Із розвитком комп’ютерної індустрії пов’язане поняття суперкомп’ютера. Це обчислювальна машина, яка за своїми технічними характеристиками значно перевершує решту комп’ютерів.

Система Intel ASCI RED, побудована на замовлення Міністерства енергетики США, виконує 3200 млрд операцій на секунду. Для виконання таких розрахунків за допомогою калькулятора людині необхідно було б витратити 100 тис. років!

В Україні створено потужні центри суперкомп’ютерних обчислень у Національному технічному університеті «Київський політехнічний університет» та Інституті кібернетики ім. В. М. Глушкова НАН України (рис. 4.2).

Суперкомп’ютери застосовують для складних обчислень, проведення наукових досліджень, передбачення погоди та стихійних явищ і виконання багатьох інших завдань. Постійна модернізація суперкомп’ютерів є неодмінною умовою для успішного розвитку людства.



Рис. 4.2

### Питання для самоперевірки



1. Наведіть приклади використання комп’ютерів.
2. Яке призначення персонального комп’ютера?
3. Що таке моноблок?
4. Які характеристики має нетбук?
5. Які характеристики має смартфон?
6. Назвіть види комп’ютерів.
7. Де використовують мікрокомп’ютери?
8. Для чого створюють суперкомп’ютери?

### Вправа 4



- Виконайте завдання за комп’ютером.
- 1) Завантажте клавіатурний тренажер і потренуйтесь набирати пропонований текст із буквами різних алфавітів і спеціальними символами. Повідомте результат учителю.



- 2) Знайдіть в Інтернеті технічні характеристики моделі комп'ютера, який вас зацікавив. Який вид комп'ютерів, на вашу думку, є найзручнішим для учнів 5 класу?

### Комп'ютерне тестування



Виконайте тестове завдання 4 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 5. Операційна система та її інтерфейс

Найважливіші програми на комп'ютері — це програми операційної системи.



**Операційна система** — це набір програм, призначений для забезпечення взаємодії всіх пристройів комп'ютера і виконання різних команд користувача.

Операційна система забезпечує взаємодію пристройів і програм під час виконання заданої користувачем роботи. Вона дає людині змогу не занурюватися в тонкощі роботи програм та пристройів комп'ютера.

### Інтерфейс користувача



**Інтерфейс користувача** — це сукупність засобів операційної системи для взаємодії між користувачем і апаратно-програмними засобами комп'ютера.

Перші операційні системи мали текстовий інтерфейс. Команди керування слід було пам'ятати та вводити з клавіатури.

Графічний інтерфейс є наочнішим: команди вибираються зі списків (меню), додаткові параметри зазначаються в діалогових



вікнах, об'єкти відображаються у вигляді значків-піктограм із назвами під ними.

Під об'єктами розуміють усе, чим управляє операційна система: папки, файли, мережа, диски, пристрой.

- 1** Об'єкти операційної системи часто мають піктограми (значки) із зображенням, схожим на оригінал: система, телефон і модем, миша, принтери і факси, екран тощо (рис. 5.1).



Рис. 5.1

Значки можуть розташовуватися на робочому столі, у вікнах папок. Кожна Windows-програма або папка може мати власну, не схожу на інші піктограму. За виглядом піктограми майже завжди можна визначити тип об'єкта (рис. 5.2).

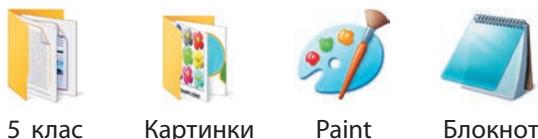


Рис. 5.2

Операції над об'єктами виконують переважно за допомогою миші.

## Файли

Усі опрацьовані за допомогою комп'ютера дані та програми зберігаються у файлах.

Вхідна інформація, результати її опрацювання, програми, тексти, малюнки, музика, фільми тощо у зовнішній пам'яті рівнозначні: для комп'ютера це — файли.



**Файл** — це іменована сукупність даних, розміщених на носіеві, які в процесі зберігання, передавання та опрацювання вважаються єдиним цілим.

Ім'я файла складається з назви і розширення, відокремлених крапкою (рис. 5.3).

Ім'я файла може містити літери (англійські, українські та ін.) і цифри, а також майже всі розділові і спеціальні знаки: \_ \$ # & @ ! % ( ) { } ‘ ~ ^.

Не можна використовувати службові символи: / \ : \* ? “ < > -. Іноді можуть виникнути проблеми в разі використання українських літер і, е, є, г, І, Е, ІІ, Г.

Здебільшого називу файла призначає користувач, а розширення — програма, якою цей файл створено.

Залежно від розширення операційна система надає файлу піктограму для його позначення й визначає, за допомогою яких програмних засобів цей файл може бути відкритим. Зміна розширення користувачем може привести до «нерозуміння» комп’ютером типу файла й неможливості його автоматичного відкривання.

Розглянемо деякі з розширень, які часто використовуються, та відповідні їм типи файлів.



Рис. 5.3

.txt — текстові файли	.doc, .docx — файли документів Word
.mp3 — музичні файли	.xls, .xlsx — файли документів Excel
.com, .exe — виконувані програми	.bmp, .png — файли зображень

## Папки

На дисках сучасного комп’ютера можуть зберігатися одночасно сотні тисяч файлів. Для їх упорядкування використовують папки (каталоги).

У звичайній картонній папці зберігають паперові документи. Папка ж на диску містить дані про розміщення вкладених у неї файлів та інших папок.

## ► РОЗДІЛ 1. Інформаційні процеси та системи

Кожна папка має назву, розширення для папки не використовується.

- 2 Розпочинаючи роботу над проектом, який складається з багатьох файлів, варто створити для нього окрему папку, щоб потім було легко відшукати потрібні для роботи файли.



Петренко



Мої фото

Рис. 5.4

## Ярлики

Користуватися значками об'єктів не завжди зручно. Значок об'єкта може бути схований у «глибині» папок, і його пошук займе тривалий час. Щоб уникнути цього, шлях записують у спеціальний файл, під час відкриття якого система знаходить і відкриває сам об'єкт. Такий файл називають ярликом об'єкта.



**Ярлик** — це засіб швидкого доступу, який містить лише шлях до відповідного об'єкта.

Ярлики для дисків, папок, програм і документів можна створити за допомогою команд контекстного меню.

Піктограма ярлика така сама, як у зв'язаного з ним об'єкта, тільки в лівому нижньому куті він має маленький квадрат, усередині якого зображено стрілку (рис. 5.5).

- 3 Ярлик відіграє роль «запобіжника» від необережних дій користувачів-початківців: видалення ярлика не призводить до знищення об'єкта.



5 клас



Картинки



Paint



Блокнот

Рис. 5.5



### Питання для самоперевірки

1. Що таке операційна система?
2. Які функції виконує операційна система?
3. Назвіть об'єкти операційної системи.
4. Що таке інтерфейс користувача?
5. Що таке файл?
6. Як записується ім'я файла?
7. Опишіть призначення папок.
8. Що таке значки, ярлики?



### Вправа 5

► Виконайте завдання за комп'ютером.



- 1) Завантажте тренажер редагування тексту та потренуйтесь, виправляючи текст із зайвими символами. Повідомте результат учителю.
- 2) З'ясуйте, користуючись Інтернетом, яке призначення клавіші Print Screen (або PrtSc). Зробіть знімок робочого столу, щоб було видно розташовані на ньому значки. Запустіть графічний редактор і вставте знімок на малюнок. Збережіть файл із назвою Знімок.



### Комп'ютерне тестування

Виконайте тестове завдання 5 з автоматичною перевіркою результу на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 6. Операції над файлами та папками

На робочому столі й у папках ви бачите багато різноманітних значків і ярликів об'єктів. Щоб у майбутньому уникнути втрати даних, необхідно навчитися правильно виконувати операції над файлами та папками.



## Вікно папки «Комп’ютер»

Папка Комп’ютер відображає диски та пристрой, які під’єднані до комп’ютера (рис. 6.1).

Щоб відкрити вікно папки **Комп’ютер**, достатньо двічі клацнути на її значку, розташованому на робочому столі.

Вигляд, зміст і навіть назва вікна Комп’ютер залежить від типу операційної системи, а також від теми оформлення. В операційних системах Windows XP, Windows 7 і Windows 10 вони різні.

У Windows 10 вікно називається Цей комп’ютер і має такий вигляд, як показано на рис 6.2.

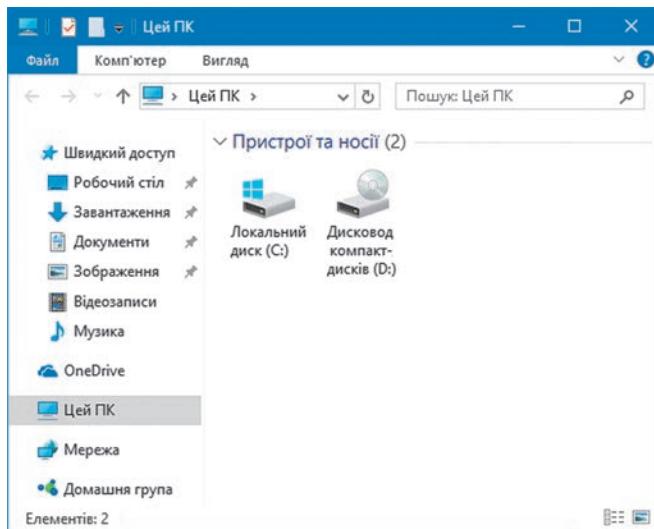


Рис. 6.2

Як бачимо, вікно має знайомі вам кнопки, якими його можна згорнути на панелі завдань —, розгорнути на весь екран □ і повернути до початкового розміру ☐, а також закрити ✕.

## Створення папки

Люди часто зберігають паперові документи в папках, які підписують відповідно до виду документів, що в них будуть зберігати-



Комп’ютер  
Рис. 6.1

ся (особові справи, накази тощо). Папки розміщують у канцелярській шафі або в іншій папці. Розташування документів в іменованих папках полегшує знаходження кожного окремого документа.

Папки на комп'ютері використовують так само: кожна папка має відповідну до її вмісту назву, зберігається на диску або в іншій папці.



**Папка** — це елемент файлової системи, контейнер для зберігання файлів та інших папок.

Папку можна створити на робочому столі, усередині іншої папки, на знімному носії даних, на вінчестері.

Щоб створити власну папку на диску D:, потрібно:

- 1) відкрити вікно Комп'ютер і на панелі навігації вибрати Диск D:;
- 2) на вільному місці клацнути правою кнопкою — відкриється контекстне меню;
- 3) установити вказівник на команду Створити — відкриється додаткове меню (рис. 6.3);

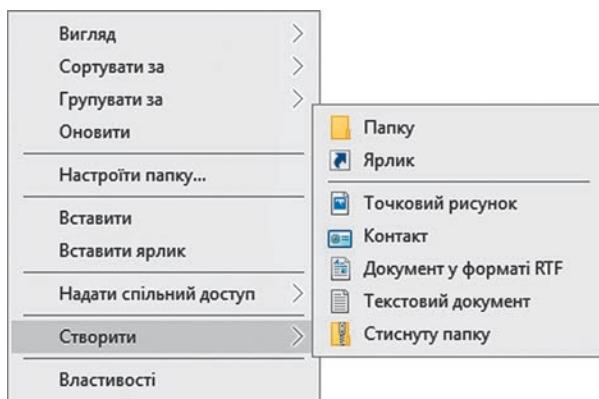


Рис. 6.3

- 4) установити вказівник миші на команді Папку і клацнути.

У вікні папки з'явиться піктограма нової папки, поряд — поле для введення назви (рис. 6.4);

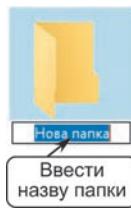


Рис. 6.4

## ► РОЗДІЛ 1. Інформаційні процеси та системи

- 5) увести з клавіатури назву папки й натиснути клавішу Enter — буде створено нову папку з уведеною назвою.

! Необхідно запам'ятати, де міститься потрібна папка та її назву, щоб потім використовувати її для подальшої роботи.

### Створення ярлика

Для швидкого доступу до своїх файлів можна створити ярлик власної папки на робочому столі комп'ютера.

Для створення ярлика папки потрібно:

- 1) установити вказівник миші на значок папки й викликати контекстне меню;
- 2) навести вказівник на команду Надіслати — з'явиться додаткове меню (рис. 6.5);

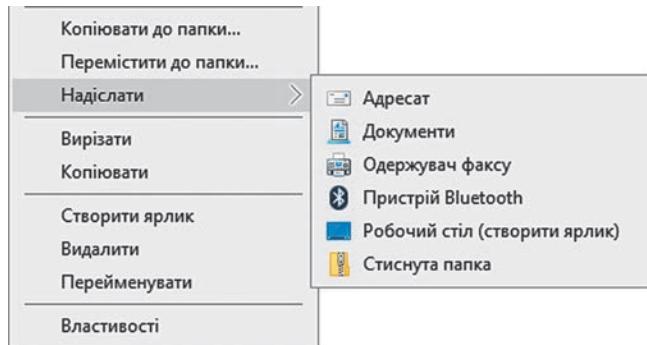


Рис. 6.5

- 3) вибрати у додатковому меню команду Робочий стіл (створити ярлик) і клацнути;
- 4) закрити вікно — на робочому столі з'явиться ярлик із назвою папки.

Тепер, щоб відкрити власну папку, досить на робочому столі двічі клацнути на її ярлику.

### Робота з папками, ярликами та файлами

Папки, ярлики та файли (далі — об'єкти) за необхідності можна перейменовувати, копіювати, переміщувати, видаляти й відновлювати.

Щоб **перейменувати об'єкт**, слід клацнути його ім'я і, коли зміниться колір фону тексту, клацнути ще раз — з'явиться курсор, і можна ввести нове ім'я.

Щоб **скопіювати об'єкт**, потрібно:

- 1) на значку об'єкта викликати контекстне меню, у якому вибрати команду Копіювати;
- 2) відкрити папку, де буде розміщена копія об'єкта;
- 3) викликати контекстне меню й вибрати команду Вставити — об'єкт з'явиться на новому місці й такий самий залишиться на старому.

Скопіювати об'єкт в іншу папку також можна, натиснувши комбінації клавіш Ctrl+C (копіювати) і Ctrl+V (вставити).

Щоб **перемістити об'єкт**, потрібно:

- 1) на значку об'єкта викликати контекстне меню, у якому вибрати команду Вирізати;
- 2) перевести вказівник миші в потрібне місце, викликати контекстне меню й вибрати команду Вставити — об'єкт з'явиться на новому місці та зникне на старому.

Для переміщення об'єкта можна також використати сполучення клавіш Ctrl+X (вирізати) та Ctrl+V (вставити).

Щоб **видалити об'єкт**, потрібно:

- 1) на його значку викликати контекстне меню;
- 2) вибрати команду Видалити — об'єкт зникне й переміститься в Кошик.

Для видалення об'єкта також можна використати клавішу Delete.

Щоб **відновити об'єкт**, необхідно відкрити Кошик, на значку потрібного об'єкта викликати контекстне меню, у якому вибрати команду Відновити.

**!** Переміщення, копіювання та видалення об'єктів (переміщення в Кошик) можна здійснити перетягуванням значка об'єкта правою або лівою кнопкою миші.

Після перетягування, наприклад, файла в папку правою кнопкою миші, у контекстному меню буде запропоновано вибрати один із варіантів дії:

- копіювати;
- перемістити;
- створити ярлик (рис. 6.6).

Копіювати сюди
<b>Перемістити сюди</b>
Створити ярлики
Скасувати

Рис. 6.6

### Питання для самоперевірки



1. Які кнопки керування має вікно папки Комп'ютер?
2. Як згорнути вікно на панель завдань? Як його розгорнути?
3. Що таке контекстне меню? Як ним користуватися?
4. Де можна створити папку?
5. Як створити власну папку на диску D?
6. Як створити й надіслати на робочий стіл ярлик власної папки?
7. Які дії виконуються над папками, файлами та ярликами?
8. Дослідіть і опишіть інші способи створення папки.

### Вправа 6



- Виконайте завдання за комп'ютером.
- 1) Завантажте тренажер редагування тексту й потренуйтесь, виправляючи текст із пропущеними символами. Повідомте результат учителю.
  - 2) Створіть на робочому столі папку з назвою 5 клас. Відкрийте її і створіть у ній папки з назвами трьох навчальних предметів. Закрийте вікно й видаліть папку 5 клас у кошик. Відновіть папку та переконайтесь, що відновилися також і внутрішні папки.

### Комп'ютерне тестування



Виконайте тестове завдання 6 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)





# Практична робота 1

## Операції над файлами та папками



**Завдання:** навчитися користуватися засобами операційної системи для операцій над файлами та папками.

**Обладнання:** комп’ютер із встановленою операційною системою.

### Хід роботи

*Під час роботи з комп’ютером дотримуйтесь правил безпеки.*

- 1. Підготуйте комп’ютер до роботи.
- 2. Створіть у вказаному вчителем місці власну папку, назвіть її власним прізвищем.
- 3. Створіть у власній папці чотири папки з назвами: Малюнки, Тексти, Музика, Різне.
- 4. Створіть на робочому столі ярлик власної папки. Відкрийте власну папку за допомогою ярлика.
- 5. Відкрийте папку Малюнки; згорніть її на панель завдань.
- 6. Розгорніть вікно на весь екран, закрийте папку.
- 7. У папку Малюнки з указаної вчителем папки скопіюйте графічний файл.
- 8. Відкрийте графічний файл, збережений у папці Малюнки; закрийте програму перегляду.
- 9. У папку Тексти з указаної вчителем папки скопіюйте текстовий файл.
- 10. У папку Музика з указаної вчителем папки скопіюйте музичний файл.

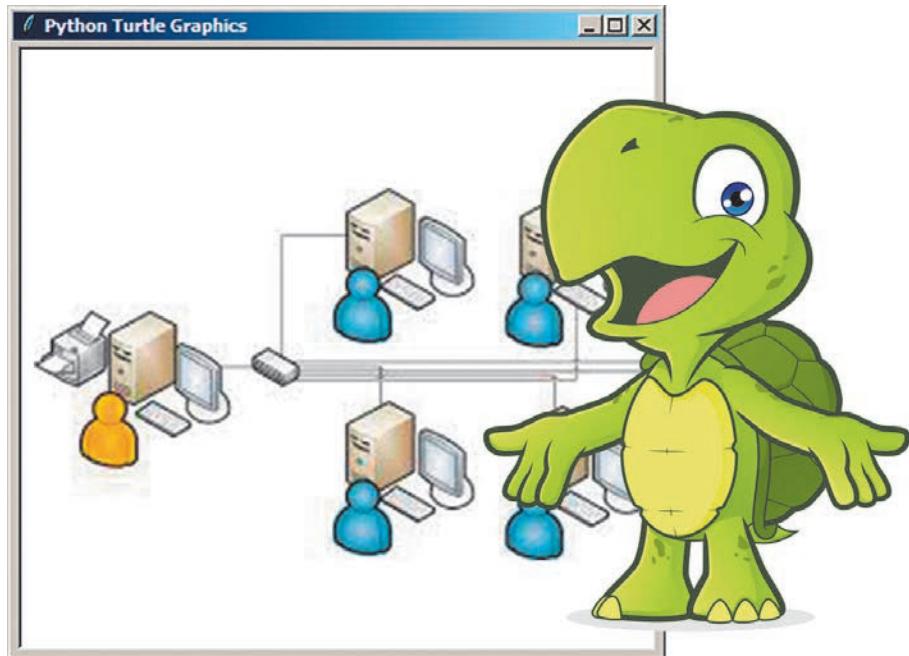
## ► РОЗДІЛ 1. Інформаційні процеси та системи

- **11.** Прослухайте музичний файл, збережений у папці Музика, закрійте програвач.
- **12.** Закрійте всі вікна, закінчіть роботу на комп'ютері.

**Зробіть висновок:** які засоби для здійснення операцій над файлами та папками надає операційна система комп'ютера?

# РОЗДІЛ 2

## МЕРЕЖЕВІ ТЕХНОЛОГІЇ ТА ІНТЕРНЕТ



§ 7. Локальна і глобальна комп'ютерні мережі

§ 8. Безпечне користування Інтернетом

§ 9. Пошук інформації в Інтернеті

Практична робота 2. Пошук інформації в Інтернеті

## ПОВТОРЮЄМО



Як ви знаєте, комп'ютерна мережа — це сукупність комп'ютерів, які можуть надавати доступ до своїх апаратних (дисководи, принтери, сканери тощо) і програмних (програми, дані тощо) ресурсів.

Комп'ютери в мережі відрізняються за функціями, які вони виконують. Комп'ютер, що надає ресурси в мережу, називають *сервером*, а той, який ці ресурси використовує, — *клієнтом* (або *робочою станцією*).

Найбільша глобальна комп'ютерна мережа — *Інтернет*. Вона надає користувачам усього світу можливості спілкування, пошуку й перегляду інформаційних матеріалів.

Вам відомо, як здійснювати простий пошук в Інтернеті за ключовими словами з використанням пошукових систем. Прийшов час вдосконалити навички зберігати та опрацьовувати знайдені матеріали.

1. Що таке комп'ютерна мережа?
2. Назвіть апаратні й програмні ресурси комп'ютерної мережі.
3. Який комп'ютер у мережі називається сервером?
4. Який комп'ютер у мережі називається клієнтом?
5. Що таке Інтернет?
6. Які можливості надає Інтернет користувачу?



Опрацювавши цей розділ, ви дізнаєтесь про особливості будови й роботи комп'ютерних мереж, навчитеся шукати потрібну інформацію, зберігати її та опрацьовувати.



## § 7. Локальна і глобальна комп'ютерні мережі

Комп'ютерні мережі, залежно від охоплюваної території, бувають двох основних видів: локальна й глобальна.

### Локальна комп'ютерна мережа

Локальна мережа (LAN) об'єднує комп'ютери, розташовані на невеликій відстані один від одного в межах школи, банку, підприємства тощо.

Для під'єднання до мережі комп'ютер повинен мати спеціальний пристрій — **мережевий адаптер**, який у сучасних комп'ютерах зазвичай уже вбудований. Для побудови локальної мережі потрібен інший пристрій — **мережевий комутатор** (від англ. *switch* — перемикач). До нього кабелями під'єднують комп'ютери й інші мережеві пристрої (принтери, сканери тощо). Можна побудувати також безпровідну локальну мережу.

#### ► Сервер та робоча станція

Комп'ютер у мережі виконує одну з функцій: або надає мережеві ресурси, або використовує їх.



**Комп'ютер**, який надає ресурси, називають **сервером**. Комп'ютер, який використовує ресурси, називають **клієнтом**, або **робочою станцією** (рис. 7.1).

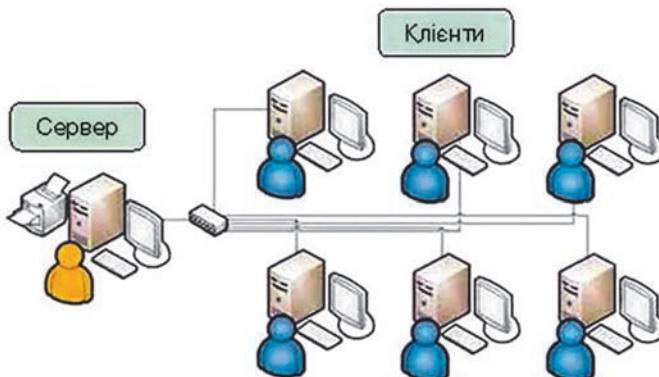


Рис. 7.1



## ► Адресація в мережі

Обмін даними між пристроями в мережі відбувається відповідно до певних правил, які називають **протоколами**. Згідно з ними кожен комп’ютер у мережі має спеціальний номер — IP-адресу (читається «ай-пі-адреса»).

IP-адреса складається із чотирьох чисел у межах від 0 до 255, відокремлених крапками.

- 1 Сервер локальної мережі шкільного комп’ютерного класу може мати IP-адресу 192.168.0.1. Жодні два комп’ютери в мережі не можуть мати одинакових IP-адрес.

## ► Вхід у локальну мережу

Для входу до локальної мережі призначена папка Мережеве оточення. У вікні цієї папки містяться ярлики мережевих ресурсів, які відкривалися раніше (рис. 7.2).

Щоб знайти певний ресурс, потрібно:

- 1) на додатковій панелі вибрати команду Показати комп’ютери робочої групи — у робочому полі вікна з’являться значки доступних комп’ютерів робочої групи;
- 2) двічі клацнувши значок потрібного комп’ютера, перейти до роботи з відкритими ресурсами цього комп’ютера — відкривати папки, файли, створювати ярлики тощо.

## ► Спільне використання ресурсів

Деякі папки можуть бути відкриті лише для читання. У таку папку неможливо ні записати новий файл, ні змінити та зберегти наявний. Мережеві ресурси іншого комп’ютера стають недоступними після його вимкнення. Тому буває корисно зробити локальну копію файла навіть тоді, коли немає потреби вносити до нього зміни.

Якщо до одного з комп’ютерів локальної мережі підключено принтер і до нього відкрито доступ через мережу, то користувачі інших комп’ютерів можуть друкувати на ньому свої документи (тексти, малюнки). Принтер та інші пристрої можуть також приєднуватися безпосередньо до мережі.



Мережеве  
оточення

**Рис. 7.2**

## Глобальна комп'ютерна мережа

Глобальна мережа (WAN) об'єднує комп'ютери та локальні мережі, розташовані в різних містах, державах і континентах.



**Інтернет** — глобальна комп'ютерна мережа, що складається з мільйонів комп'ютерів у всьому світі.

Інтернет об'єднує локальні мережі навчальних і наукових закладів, промислових підприємств, службових установ тощо, а також окремі комп'ютери. В Інтернеті циркулює різноманітна інформація, доступ до якої забезпечується майже з будь-якого куточка Землі.

2 Користувачі Інтернету можуть швидко знайти необхідну інформацію, що зберігається на серверах у Канаді чи Японії, надіслати повідомлення з одного комп'ютера на інший чи на мобільний телефон, поспілкуватися з друзями в режимі реального часу, знайти партнера для гри в шахи тощо.

Глобальна мережа створює можливість проводити наради, відеоконференції, «мандрувати» світом, стежити за курсами валют, робити покупки в інтернет-магазинах тощо.

### ► Провайдери



**Провайдер** (від англ. *provider* — постачальник) — це організація, що надає послуги, пов'язані з доступом до глобальної мережі.

Серед функцій провайдерів найважливішими є:

- забезпечення доступу до Інтернету;
- виділення дискового простору для сайтів (хостинг);
- підтримка роботи поштових скриньок;
- послуги зв'язку з передачі голосової інформації.

### ► Доменне ім'я

Разом із IP-адресою комп'ютерів для адресації ресурсів в Інтернеті використовують доменні імена.



**Доменне ім'я** — текстова адреса комп'ютера або іншого ресурсу в Інтернеті.

Доменне ім'я складається з назв кількох доменів, відокремлених крапками.

Останнім у доменному імені є домен першого (верхнього) рівня, який зазвичай вказує тип організації чи державу; назва домену другого рівня найчастіше є ім'ям сервера даних; найпершим зліва є власне ім'я ресурсу (рис. 7.3).



Рис. 7.3

Таким чином, до кожного домену 1-го рівня належать багато доменів 2-го рівня; до кожного домену 2-го рівня належать багато доменів 3-го рівня і т. д.

Кожному доменному імені відповідає певна IP-адреса, але один ІР-адресі може відповідати декілька доменних імен.

## Гіпертекст



**Гіпертекст** — це текстовий документ, який містить гіперпосилання (слова, фрази, графічні об'єкти), клацання на яких дає змогу перейти до іншого місця в документі або до іншого документа (рис. 7.4).

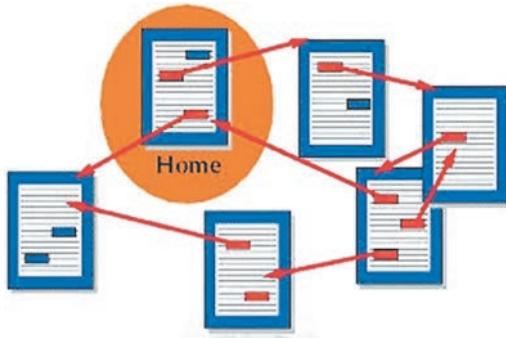


Рис. 7.4

Деякі гіперпосилання в разі клацання можуть викликати виконання певних дій: відкриття файла, завантаження поштового клієнта для написання електронного листа тощо. Гіперпосилання виділяють кольором символів, підkreслюванням, рамкою малюнка тощо. Під час наведення вказівника на гіперпосилання він набуває вигляду руки: .

Документ, пов'язаний із гіперпосиланням, може бути розміщений на будь-якому комп'ютері в мережі.

### Питання для самоперевірки



1. Які мережі називають локальними?
2. Поясніть різницю між сервером і робочою станцією.
3. Опишіть принцип адресування комп'ютерів у мережі.
4. У чому полягає спільне використання ресурсів?
5. Розкажіть, що ви знаєте про Інтернет.
6. Яку роботу виконує провайдер?
7. Опишіть найпопулярніші служби Інтернету.
8. Що таке гіпертекст? Наведіть приклади.

### Вправа 7



- Виконайте завдання за комп'ютером.
- 1) З'ясуйте назву робочої групи, до якої належить ваш комп'ютер, його ім'я в мережі та опис.
  - 2) У мережевій папці, указаній учителем, створіть папку, назвіть її власним прізвищем. Скопіюйте до папки вказаний файл, розміщений на вашому комп'ютері.

### Комп'ютерне тестування



Виконайте тестове завдання 7 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 8. Безпечне користування Інтернетом

Глобальна мережа Інтернет — це необмежений ресурс, який може бути використаний як для навчання, так і для відпочинку та спілкування з друзями. Маючи великий інформаційний, навчальний та розважальний потенціал, Інтернет також може бути небезпечним і становити певний ризик, особливо для дітей.

### Небезпеки, пов'язані з використанням Інтернету

Користувач глобальної мережі має знати про загрози, що існують в Інтернеті.

- **Віруси** — комп'ютерні програми, що можуть завдавати шкоди: знищувати, пошкоджувати дані або порушувати роботу комп'ютера.
- **Троянські коні** — шкідливі програми, що можуть викрачати дані чи використовувати комп'ютер зі зловмисною метою.
- **Сайти**, які пропагують насилля та поведінку, що несе загрозу життю та здоров'ю.
- **Хакерство** — отримання несанкціонованого доступу до комп'ютерних систем, як правило, із метою отримання сектреної інформації.
- **Спам** — примусова масова розсылка кореспонденції рекламиного чи іншого характеру.

Існують засоби, що запобігають цим загрозам або навіть унеможливлюють їх.

Програма	Призначення
Брандмауер	Програмне й аппаратне забезпечення, що захищає локальну мережу від небезпек, які є в Інтернеті
Антивіруси	Службові програми для знаходження комп'ютерних вірусів та відновлення заражених файлів, а також для запобігання зараженню комп'ютера
Спам-фільтр	Програма, що використовується для фільтрації електронної пошти та посилань, блокує листи від небажаних адресатів

## Правила безпечно користування Інтернетом

Інтернет є публічним місцем. Працюючи онлайн, слід дотримуватися основних правил безпеки так само, як ви дотримуєтесь правил дорожнього руху.

**Основні правила безпечної роботи** в глобальній мережі Інтернет для школярів такі:

- не надавайте незнайомим особам свої дані й інформацію про близьких і рідних;
- спілкуючись в Інтернеті, не вказуйте свої особисті дані, а користуйтесь псевдонімом (ніком);
- якщо в мережі потрібно пройти реєстрацію, то робіть її так, щоб у ній не зазначалося ніякої особистої інформації;
- не здійснюйте фінансові операції через мережу Інтернет без нагляду дорослих;
- не погоджуйтесь на зустріч із незнайомими людьми;
- повідомляйте батьків, старших друзів у разі отримання інформації, що має характер залякування;
- не заходьте на незнайомі сайти;
- пам'ятайте, що не можна довіряти всій інформації, яку розміщено в Інтернеті.

## World Wide Web



**World Wide Web (WWW)** — це сукупність пов'язаних гіперпосиланнями документів, розміщених на безлічі комп'ютерів, із наданням можливості доступу до них через Інтернет.

Інформацію у World Wide Web подано у вигляді гіпертекстових документів — **веб-сторінок**. Кілька пов'язаних гіперпосиланнями веб-сторінок, які мають спільну тему та розміщення, називають **веб-сайтом** або просто **сайтом**.

Розміщені на серверах сайти створюють гіпертекстовий простір документів, доступ до якого можливий у будь-який час і з будь-якого під'єднаного до Інтернету комп'ютера.



Для перегляду веб-сторінок потрібна спеціальна програма — **браузер** (від англ. *browser* — оглядач, провідник). Браузер може відображати текстову або графічну інформацію, відтворювати звуки, анімацію та відео, розміщені на сайтах. Користувач за допомогою браузера може переглядати веб-сторінки та здійснювати наявності у веб-просторі за допомогою гіперпосилань.

Браузери дають змогу переглядати веб-сторінки з віддаленого комп’ютера (режим онлайн) або збережені на комп’ютері користувача (режим офлайн).

- 1** Популярним браузером довгий час була програма Internet Explorer, вбудована в операційну систему Windows. Існують й інші браузери: Opera, Mozilla Firefox, Google Chrome тощо (рис. 8.1).



Рис. 8.1

## Адреса веб-сторінки

Адресу веб-сторінки вводять одним із таких способів:

- 1) у полі адреси набрати з клавіатури адресу необхідної веб-сторінки;
- 2) у відкритій веб-сторінці клацнути на гіперпосиланні й перейти на нову веб-сторінку. Щоб не втратити зображення поточної сторінки, слід на гіперпосиланні клацнути правою кнопкою миші й у контекстному меню вибрати Відкрити в новій вкладці;
- 3) зі списку меню Уподобання або Журнал вибрати адресу необхідної веб-сторінки.

Після введення адреси веб-сторінки автоматично встановлюється зв’язок із сервером, веб-сторінка завантажується й відкривається.

## Робота із сайтами

Сьогодні організації, підприємства, навчальні заклади, торгово-вельні центри та приватні особи розміщують на веб-сайтах найрізноманітнішу інформацію.

- 2 Sinoptik.ua — прогноз погоди; novaposhta.ua — доставка товарів; megogo.net — розваги та ін.

Якщо в полі адреси браузера ввести адресу abetka.ukrlife.org, на екран буде виведено головну сторінку сайта Весела абетка (рис. 8.2). На ньому можна знайти багато цікавих матеріалів (казки, усмішки, загадки, скромовки тощо).

Клацанням на гіперпосиланнях Абетки, Казки, Читанка, Усмішки, Приказки, Скоромовки і т. д. здійснюється перехід на інші сторінки. На цих сторінках також є гіперпосилання для подальших переходів.



Рис. 8.2

**Питання для самоперевірки**

1. Які небезпеки пов'язані з використанням Інтернету?
2. Назвіть основні правила безпечної роботи в Інтернеті.
3. Що може відбутися після клацання на гіперпосиланні?
4. Для чого призначені програми-браузери?
5. Назвіть поширені веб-браузери.
6. Яким чином можна ввести адресу сайта?
7. Назвіть кілька поширених сайтів.
8. Як перейти з однієї веб-сторінки на іншу?



Рис. 8.3

**Вправа 8**

► Виконайте завдання за комп'ютером.

- 1) Запустіть програму-браузер. Наберіть в адресному рядку браузера адресу [abetka.ukrlife.org](http://abetka.ukrlife.org) і натисніть Enter. Відшукайте на сайті загадку про зиму. Скопіюйте її в текстовий документ та збережіть файл з іменем Загадка в папку Тексти власної папки.
- 2) Відкрийте сайт, указаний учителем. Відшукайте на ньому малюнок на тему «Моя країна» та збережіть його у власній папці. Закінчіть роботу в Інтернеті й на комп'ютері.

**Комп'ютерне тестування**

Виконайте тестове завдання 8 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



# § 9. Пошук інформації в Інтернеті

## Сервери пошуку

Пошук інформації здійснюють за допомогою серверів пошуку. Такі спеціалізовані сервери можуть переглядати сотні тисяч і навіть мільйони веб-сторінок у пошуках необхідної користувачам інформації (рис. 9.1).



Рис. 9.1

- 1 До спеціалізованих серверів належать пошукові системи Shukalka, Google, Yahoo, Lycos тощо (рис. 9.2).



Рис. 9.2

Отримавши пошуковий запит, програма знаходить сторінки, що містять задані ключові слова, упорядковує їх і передає користувачеві у вигляді списку веб-сторінок.

Для пошуку інформації вбудованими засобами браузера слід завантажити браузер, увести у відповідне поле ключові слова та натиснути клавішу Enter. Після цього почнеться пошук веб-документів, що містять ключові слова, і через деякий час у робочому полі вікна з'явиться список посилань на знайдені сайти.

### ► Сервер пошуку Google

Одним із найпопулярніших пошукових серверів є Google.

Щоб скористатися пошуковим сервером Google, потрібно в адресному рядку набрати його ім'я google.com.ua і натиснути Enter — після завантаження на екрані з'являється початкова сто-

## ► РОЗДІЛ 2. Мережеві технології та Інтернет

рінка (рис. 9.3) Google; далі потрібно ввести ключові слова в текстове поле, натиснути кнопку Пошук Google та дочекатися результатів пошуку.

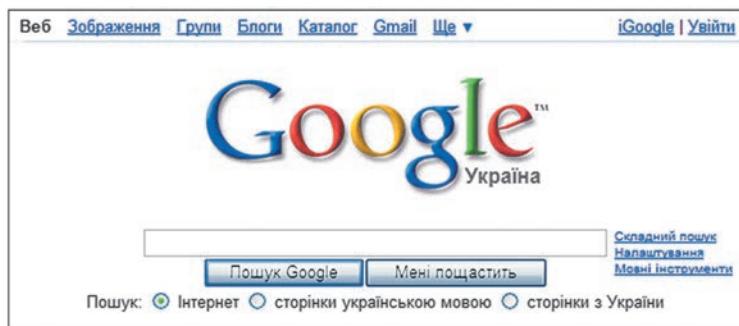


Рис. 9.3

### ► Стратегія пошуку інформації

У результаті пошуку можна отримати величезну кількість посилань. Тому вирішальне значення має правильний набір ключових слів:

- шукайте більше, ніж по одному слову: краще написати *птахи України*, ніж *птахи*;
- порядок самих ключових слів також впливає на результати;
- пишіть без помилок; використовуйте синоніми, якщо список знайдених сторінок занадто великий;
- не починайте слова з великої літери, крім власних назв;
- якщо хочете знайти точну фразу, візьміть її в лапки: «*Людина нібито не літає, а крила має*». Лапки кажуть Google шукати слова саме в такій послідовності та в такій формі.

### ► Завантаження файлів з Інтернету й авторське право

Деякі гіперпосилання вказують не на інші сторінки, а на файли, які можна завантажити й зберегти на своєму комп’ютері (англ. *download* — завантажити). У разі клачання на такому посиланні відкривається вікно Зберегти як..., у якому слід уточнити папку для збереження й ім’я файла і, класнувши Зберегти, роз-

почати завантаження файла. Клацнувши Скасувати, можна припинити завантаження файла.

Усі матеріали, розміщені в Інтернеті, кимось створені, тобто мають своїх власників. Лише власники мають право підписувати матеріали своїм іменем і розповсюджувати їх. Таке право називають авторським. Його охороняє Закон України про авторське право.

Копіювати матеріали з Інтернету для використання в особистих цілях дозволено, але передавати та подавати такий матеріал як власний не можна. Наприклад, якщо ви використовуєте матеріал для своєї презентації, то маєте посилатися на джерело.

#### ► Критичне оцінювання інформації, отриманої з Інтернету

Мережа Інтернет — це неояжне джерело ресурсів. Однак інформація, подана на багатьох сайтах Інтернету, не завжди перевірена, якісна й, отже, надійна.

Глобальна мережа може містити матеріали з орфографічними та граматичними помилками, фактичними неточностями, спотвореною інформацією тощо. Робота з ресурсами такої якості потребує від користувачів розвитку спеціальних умінь, а саме вміння спостерігати, розпізнавати та критично аналізувати знайдену інформацію (рис. 9.4).



Рис. 9.4

### Використання мережі Інтернет для навчання

Ви звички отримувати нові знання в школі. Але багато цікавого й корисного можна знайти в Інтернеті. Навчання в гло-



бальній мережі має певні переваги порівняно з іншими способами набуття знань:

- навчальні матеріали доступні в будь-який час, що дозволяє планувати час роботи на власний розсуд;
- засоби спілкування з іншими учнями дозволяють швидше зрозуміти складні місця;
- в Інтернеті можна шукати додаткові матеріали.

Проте є й недоліки:

- доводиться багато часу проводити перед екраном комп’ютера;
- не кожна людина може змусити себе працювати, якщо раптом трапиться складне завдання.

Не варто засмучуватись, якщо не вдається опанувати якийсь інтернет-курс. Можливо, він розрахований на старших учнів або невдало побудований. У такому разі слід порадитися зі старшиими й вибрати інший навчальний сайт.

**Основні варіанти використання інтернет-ресурсів** такі.

1. *Різноманітні види дистанційної освіти.* Багато курсів для учнів є на сайтах <http://osvita-online.com.ua/>, <https://www.ed-era.com/courses/> та інших.



**Рис. 9.5**

2. *Інтелектуальні змагання.* Популярними серед учнів є інтернет-олімпіади. Той, хто бажає взяти участь у такій олімпіаді, реєструється на сайті й виконує відповідні завдання.
3. *Проектна робота.* Можна включитися в роботу над реальним проектом і вчитися, спілкуючись з іншими учасниками.

4. *Самостійне навчання.* Наприклад, на сайті <https://code.org/> вже майже 2 мільйони учнів з усього світу вивчають основи програмування.



### Питання для самоперевірки

1. Якими засобами здійснюється пошук інформації в Інтернеті?
2. Опишіть процес пошуку за допомогою вбудованих засобів пошуку.
3. Опишіть призначення та дію серверів пошуку.
4. Як потрапити на веб-сторінку пошукового сервера?
5. У якому вигляді пошуковий сервер подає результати пошуку?
6. Назвіть найпопулярніші сервери пошуку.
7. Чи поширюється авторське право на матеріали в Інтернеті?
8. Що ви розумієте під авторським правом?



Рис. 9.6



### Вправа 9



- Виконайте завдання за комп’ютером.
- 1) За допомогою пошукової системи [shukalka.com.ua](http://shukalka.com.ua) знайдіть указану вчителем пісню.
  - 2) Завантажте вибрану пісню та збережіть, не змінюючи імені файла, у папці Музика власної папки.
  - 3) Прослухайте пісню із папки Музика. Закінчіть роботу в Інтернеті й на комп’ютері.



### Комп’ютерне тестування

Виконайте тестове завдання 9 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)





## Практична робота 2

### Пошук інформації в Інтернеті



**Завдання:** навчитися шукати інформацію в Інтернеті, засвоїти основні правила безпечної роботи в Інтернеті.

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

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтесь правил безпеки.*

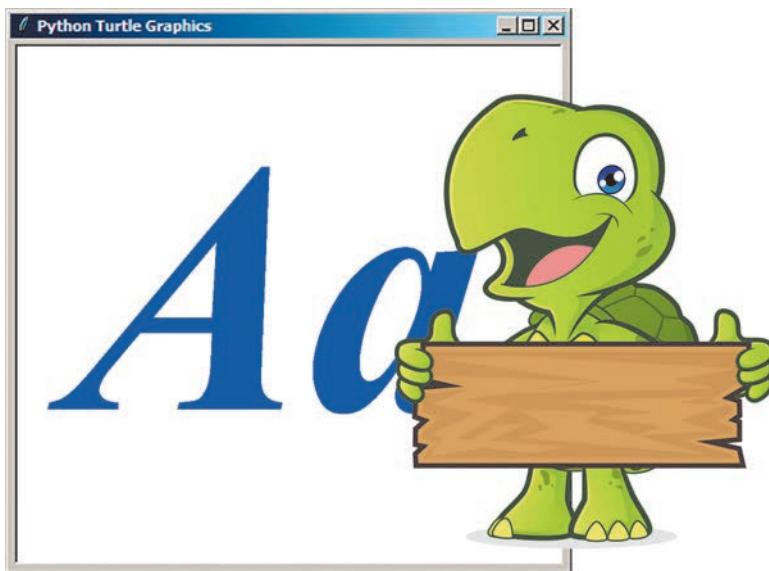
- 1. Підготуйте комп'ютер до роботи. Запустіть браузер.
- 2. Перейдіть на сайт пошукової системи.
- 3. Відшукайте в Інтернеті текст казки («Колобок», «Рукавичка», «Ріпка» тощо).
- 4. Запустіть текстовий редактор і скопіюйте текст казки до документа.
- 5. Збережіть текстовий документ у власну мережеву папку. Яке ім'я файла підійде найкраще?
- 6. На інших сайтах, указаних учителем, відшукайте дві ілюстрації до казки й збережіть у власну папку. Закінчіть роботу в Інтернеті й на комп'ютері.



**Зробіть висновок:** як шукати в Інтернеті потрібні інформаційні матеріали та зберігати їх на комп'ютері?

# РОЗДІЛ 3

## ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ



§ 10. Основні об'єкти текстового документа

§ 11. Фрагмент тексту. Форматування символів

§ 12. Форматування абзаців

Практична робота 3. Створення текстового документа

§ 13. Додавання зображень із файла та їх форматування

§ 14. Додавання, редагування та форматування таблиць

§ 15. Сторінки документа та їх форматування

Практична робота 4. Оформлення текстового документа

## ПОВТОРЮЄМО



Ви вже маєте навички роботи з документами в середовищі текстового редактора Блокнот і текстового процесора Word. Ви знаєте, що текстовий документ може містити різні *об'єкти*, основними з яких є *символ* і *абзац*. Окрім тексту, документ може містити малюнки, таблиці.

Кожен об'єкт має свої властивості. Властивостями символу є шрифт, накреслення, розмір і колір. Абзац має такі властивості, як вирівнювання, міжрядковий інтервал, відступи тощо. Значення властивостей можна змінювати.

Ви умієте виконувати операції над текстовими об'єктами: *редагувати* (видаляти, додавати, переміщувати, копіювати), *форматувати* (встановлювати шрифт, змінювати колір і накреслення символів, розташування абзаців тощо).

1. З яким текстовим процесором ви вже працювали?
2. Назвіть об'єкти текстового документа.
3. Які операції можна виконувати над текстовими документами?
4. Які властивості має символ?
5. Які властивості має абзац?
6. Що таке форматування текстового документа?



Опрацювавши цей розділ, ви навчитеся здійснювати за допомогою текстового процесора редагування та форматування текстового документа, а також додавати в текстовий документ таблиці та рисунки, готовувати документ до друку й друкувати його.



## § 10. Основні об'єкти текстового документа

### Властивості сторінки та об'єктів на ній

Сторінка як об'єкт текстового документа має такі властивості: розміри сторінки, розміри полів, орієнтація сторінки, наявність колонтитулів тощо (рис. 10.1).

**Розміри сторінки** — це висота й ширина сторінки документа. Найпоширеніші стандартні розміри сторінки: А4 —  $21 \times 29,7$  см, А5 —  $14,8 \times 21$  см.

**Поля** — це області сторінки вздовж її країв, у яких не міститься основний текст документа. На сторінці є верхнє, нижнє, ліве та праве поля. Рекомендовані розміри полів такі: ліве — 2,5 см, праве — 1,5 см, верхнє та нижнє поле — по 2 см. Ліве й праве поля частіше залишаються незаповненими, а на верхньому та нижньому полях можуть розміщуватися колонтитули.

**Колонтитули** — це службові повідомлення на полях сторінки документа.

Розрізняють *верхній* та *нижній* колонтитули. Вони можуть містити номери сторінок, називу документа або поточного розділу, прізвище автора, графічні зображення тощо. Інформація, розміщена в колонтитулах, відображається на всіх сторінках документа або частині їх.

**Орієнтація сторінки** — це спосіб розміщення тексту на сторінці. Розрізняють *книжкову* (вертикальну) й *альбомну* (горизонтальну) орієнтації.

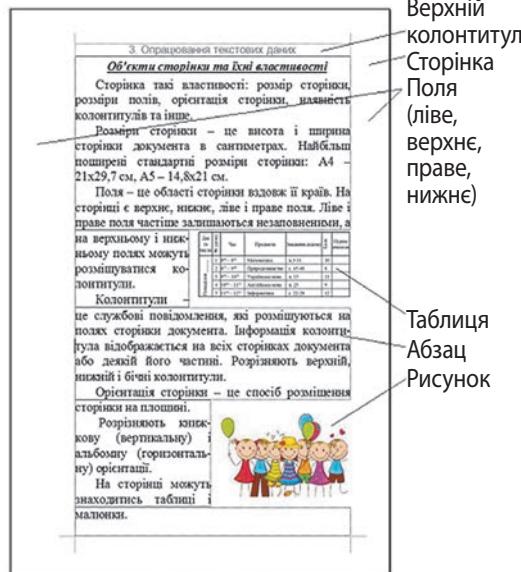


Рис. 10.1

## ► РОЗДІЛ 3. Опрацювання текстових даних

Окрім текстів сторінка може містити графічні об'єкти: рисунки, таблиці, діаграми, фотографії, формули тощо.

### Об'єкти текстового документа

Найменшою складовою текстового документа є символ. Групи символів утворюють слова, зі слів складаються речення, які, у свою чергу, групуються в абзаци (рис. 10.2).

Абзац (три рядки)	Символ	Слово	Речення (виділене)
	\	Був собі дід Андрушка, а в <b>нього</b> баба Марушка.	/ А в бабі дочечка Мінка, а в дочки собачка Хвінка. <b>А в собачки товаришка — киця Варварка.</b> А в киці вихованка — мишка Сіроманка.

Рис. 10.2

**Символ** позначає літеру, цифру, знак пунктуації, знак арифметичної операції чи спеціальний знак.

**Слово** — послідовність символів, відділена від інших пробілом.

**Речення** — розділена пробілами послідовність слів, яка починяється з великої літери та закінчується крапкою, знаком питання, знаком оклику або трьома крапками.

**Абзац** — це частина тексту, друк якого закінчується натисканням клавіши Enter. Абзац може займати один або кілька рядків.

### Дії над об'єктами текстового документа

Над текстовим документом та його об'єктами можна виконувати такі дії:

Назва дії	Опис дії
Уведення та зберігання	Уведення з клавіатури символів у документ та збереження у вигляді файла на диску
Редагування	Виправлення помилок, видалення, переміщення, копіювання, вставляння символів, слів, абзаців, рядків тощо
Форматування	Зміна вигляду документа: встановлення кольорів тексту, вирівнювання абзаців, зміна орієнтації сторінки тощо
Вставляння об'єктів	Додавання до документа рисунків, таблиць, формул, діаграм, схем тощо

Назва дії	Опис дії
Макетування	Підготовка документа до друку: оформлення заголовків, розбивка на сторінки, нумерація їх, опрацювання рисунків, створення змісту тощо
Друкування	Виведення на папір усіх або вибраних сторінок створеного документа, а також його фрагментів

## Програмне забезпечення для опрацювання текстів

Усі дії над об'єктами текстового документа забезпечує текстовий процесор, наприклад, Microsoft Word (далі — Word), що входить до складу програмного пакета Microsoft Office. Він є найпопулярнішим серед програм для створення та опрацювання листів, статей, рефератів, пояснівальних записок, курсових і дипломних робіт тощо.

### Способи запуску Word:

- за допомогою ярлика на робочому столі;
- за допомогою панелі швидкого запуску;
- відкривання будь-якого документа Word.

У перших двох випадках на екрані з'явиться вікно з порожньою робочою областю, в останньому випадку в робочій області з'явиться текст відкритого документа (рис. 10.3).

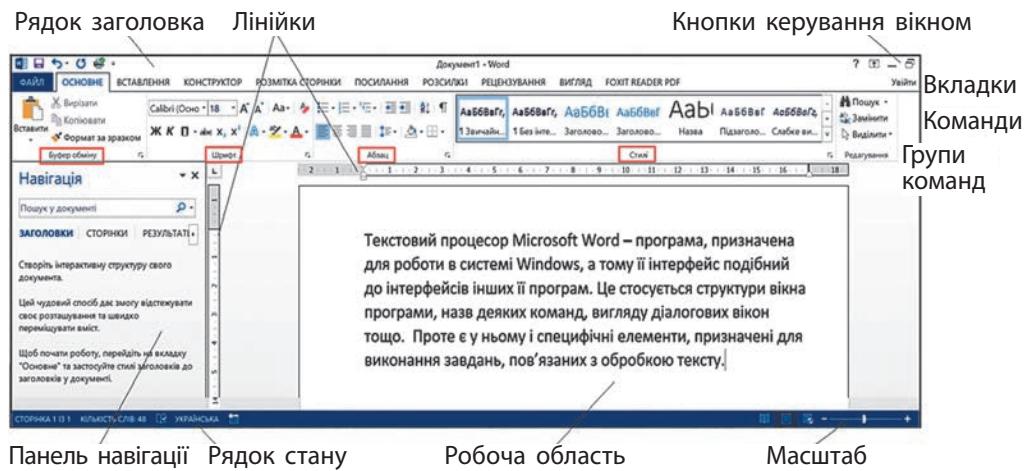


Рис. 10.3

## ► РОЗДІЛ 3. Опрацювання текстових даних

Вікно програми містить **рядок заголовка**, де зазначено ім'я активного документа (того, з яким у цей час працюють) і **назву програми**. Під рядком заголовка розташовані вкладки (ОСНОВНЕ, ВСТАВЛЕННЯ ...), нижче Команди, ще нижче Групи команд (Шрифт, Абзац ...). Більшу частину вікна займає **робоча область**, у якій міститься текстовий документ.

Ліворуч і зверху робочої області розміщено лінійки, за допомогою яких можна визначати та встановлювати розміри об'єктів документа, а праворуч і знизу — смуги прокручування для перевірки частин документа, яких не видно в робочій області.

Нижче від робочої області розташовано **рядок стану**, у якому подається корисна інформація та містяться кнопки для переключання режимів роботи з програмою.

### Параметри сторінки

Сторінка має певні розміри полів, орієнтацію, розмір та ін. (рис. 10.4).

Щоб установити розміри полів, орієнтацію та розмір сторінки, потрібно:

- відкрити вкладку Розмітка сторінки (на рис. 10.4 інструменти групи Параметри сторінки);
- відкрити потрібний список, вибрати один з варіантів або за допомогою лічильників установити потрібні розміри.

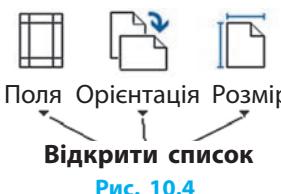


Рис. 10.4

### Установлення масштабу

Масштаб відображення документа на екрані монітора встановлюють за допомогою бігунка в правому нижньому куті вікна або за допомогою вкладки Вигляд: вибирають команду Масштаб — відкриється вікно Масштаб (рис. 10.5), де слід зазначити масштаб або на лічильнику Відсоток установити потрібне число.

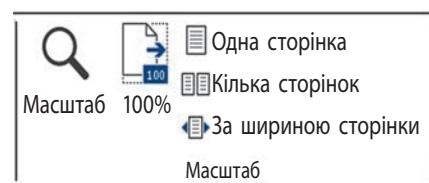


Рис. 10.5

## Збереження текстового документа

Після завершення роботи з документом його потрібно зберегти.  
**Для збереження створеного нового документа** потрібно:

- 1) виконати команду Файл → Зберегти → Огляд. Відкриється діалогове вікно Збереження документа (рис. 10.6);

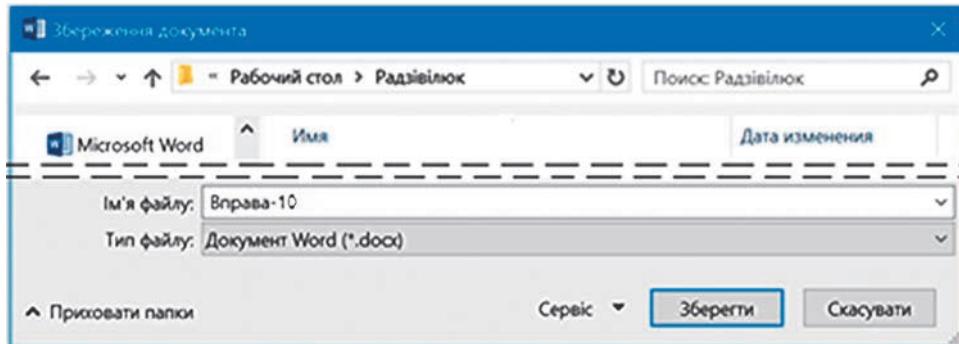


Рис. 10.6

- 2) відкрити папку, у якій має зберігатися новий документ (тут Робочий стіл → Радзівілок);
- 3) записати ім'я файла (тут — Вправа 10);
- 4) клікнути кнопку Зберегти.

### Питання для самоперевірки



1. Назвіть об'єкти сторінки текстового документа.
2. Які поля має сторінка?
3. Яке призначення колонтитулів?
4. Назвіть об'єкти тексту. Дайте визначення абзацу.
5. Опишіть вікно текстового процесора Word.
6. Як установити параметри сторінки?
7. Як установити потрібний масштаб?
8. Як зберегти новий документ?

### Вправа 10



- Створіть новий документ, установіть для нього вказані властивості, збережіть у власній папці.
- 1) Завантажте текстовий процесор. Створіть новий документ.

- 2) Задайте розмір паперу — А5, поля — по 2 см, орієнтацію — книжкову.
- 3) Установіть масштаб — За ширину сторінки.
- 4) Наберіть свої ім'я та прізвище.
- 5) Збережіть текст у файл з іменем Вправа 10 у папку Тексти власної папки.
- 6) Закрийте всі вікна. Завершіть роботу на комп'ютері.



### Комп'ютерне тестування

Виконайте тестове завдання 10 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 11. Фрагмент тексту. Форматування символів

### Операції над фрагментом тексту

Текстовий процесор створює можливість вирізати, копіювати або переносити на інше місце вибрану частину тексту (символ, слово, речення, кілька речень, абзац, кілька абзаців) — **фрагмент тексту**.

**!** Виконувати дії з фрагментом тексту можна лише після його виділення.

Для **виділення слова** достатньо двічі клацнути на ньому.

Для **виділення кількох слів** необхідно розмістити вказівник миші перед першим символом, притиснути ліву кнопку миші, перемістити вказівник після останнього символу й відпустити кнопку. Виділений фрагмент тексту буде позначено **темним фоном**. Є й інші способи виділення фрагмента тексту.

Для **скасування виділення** достатньо клацнути поза виділеним фрагментом.

Над виділеним фрагментом тексту можна виконувати такі дії: вирізати (або копіювати) в буфер обміну; вставляти з буфера обміну в потрібне місце (навіть в іншому документі).

### ► Редагування тексту

У тексті можуть траплятися помилки трьох основних видів:

- 1) зайвий символ — його потрібно видалити;
- 2) пропущений символ — його необхідно вставити;
- 3) неправильний символ — його слід замінити.

Розглянемо способи виправлення помилок (рис. 11.1):



**Рис. 11.1**

Ознайомимося зі способами редагування символів докладніше.

- 1) Для **видалення зайвого символу**, розміщеного ПРАВОРУЧ від курсора, слід натиснути клавішу Delete, а розміщеного ЛІВОРУЧ від курсора — клавішу Backspace.
- 2) Для **вставляння пропущеного символу** слід установити курсор на місце, де він має бути, і ввести з клавіатури потрібний (на рис. 11.1 — буква В). Символи, розміщені праворуч від вставленого, зсуваються на одну позицію вправо.
- 3) Для **замінення символу** слід видалити неправильний символ клавішою Delete (на рис. 11.1 — буква У) і ввести потрібний символ (на рис. 11.1 — буква О). На місці неправильного символу з'явиться введений.

### Властивості символів

Символ як об'єкт текстового документа має такі властивості, як шрифт, накреслення, розмір, колір. Властивості можна встановити для одного або кількох символів, слів, речень, рядків, а також для всього тексту.

! Щоб установити потрібні властивості, фрагмент тексту необхідно виділити.

## ► Добір і зміна шрифту

Основні властивості шрифту можна встановлювати за допомогою відповідних інструментів на вкладці Основне (рис. 11.2) або за допомогою діалогового вікна Шрифт — для його виклику необхідно клацнути спеціальний значок. У вікні Шрифт вибирають вид, накреслення, розмір, колір, підкреслення, спеціальні ефекти тощо.



Рис. 11.2

- 1** Це приклад шрифту Times New Roman розміру 12 пунктів
- Це приклад шрифту Arial розміру 12 пунктів**
- Це приклад шрифту Lucida Console розміру 10 пунктів**
- Це приклад шрифту Courier New розміру 11 пунктів**
- Це приклад шрифту Garamond розміру 14 пунктів**
- Це приклад шрифту Monotype Corsiva розміру 14 пунктів**

## ► Розмір шрифту

Розмір шрифту та інтервали між символами й рядками вимірюють у типографських пунктах (пт). Один пункт дорівнює 0,376 мм.

## ► Накреслення символів

У вікні Шрифт можна установити накреслення: звичайний, курсив, напівжирний, напівжирний курсив. Кнопки для вибору накреслення шрифту зазвичай містяться на стрічці інструментів (Ж — жирний, К — курсив, П або Ч — підкреслений).

## ► Спеціальні ефекти

До виділеного тексту можна додати спеціальні ефекти.

- 2** Закреслений;

надрядковий знак  $A^2 + B^2 = C^2$ ;

підрядковий знак  $C_2H_5OH$ ;  
подвійне підкresлення;  
контур тощо.

Для перетворення малих літер на великі та виконання інших додаткових команд форматування на стрічці інструментів можуть бути додаткові кнопки.

#### ► Міжсимвольний інтервал

У разі потреби можна збільшити або зменшити інтервал між символами. Для цього необхідно виділити потрібний фрагмент тексту та у вікні Шрифт відкрити вкладку Додатково (рис. 11.3):

- для збільшення інтервалу між символами слід збільшити значення лічильника З кроком: (тут — 1,3 pt). У полі Інтервал автоматично встановиться режим Розріджений;
- для зменшення інтервалу між символами необхідно зменшити значення лічильника З кроком:. У полі Інтервал установиться режим Ущільнений.

Після виконання певної дії слід натиснути кнопку ОК.

Міжсимвольний інтервал

Масштаб:	100%	<input type="button" value="▼"/>
Інтервал:	Розріджений	<input type="button" value="▼"/>
Розташування:	Немає	<input type="button" value="▼"/>
<input type="checkbox"/> Кернінг для знаків розміром:		<input type="button" value="▲"/> <input type="button" value="▼"/> пунктик і більше

Рис. 11.3

#### Питання для самоперевірки



1. Наведіть приклади фрагментів тексту.
2. Які є способи виділення фрагмента тексту?
3. Які операції можна виконувати над фрагментом тексту?
4. Які властивості має символ як об'єкт текстового документа?
5. Опишіть способи редагування символів.
6. Як вибрati шрифт, накреслення, розмір і коліr символів?
7. Як зменшити та збільшити інтервал між символами?

**Вправа 11**

► Створіть новий документ, установіть для нього певні властивості шрифтів, збережіть документ у власній папці.

**1)** Завантажте текстовий процесор. Задайте розмір паперу — А5, поля — по 2 см, орієнтацію — книжкову, шрифт — Arial, розмір — 10, накреслення — звичайний, масштаб — За ширину сторінки, вирівнювання — За лівим краєм.

**2)** Наберіть такий текст.

Україна — найбільша за площею країна Європи.

Територію України омивають Чорне та Азовське моря.

Державними символами України є гімн, прапор і герб.



**3)** Для першого рядка встановіть такі властивості: шрифт — Arial, розмір — 12, накреслення — курсив, колір — червоний.

**4)** Для другого рядка встановіть такі властивості: шрифт — Times New Roman, розмір — 14, накреслення — підкреслений.

**5)** Для третього рядка встановіть такі властивості: розмір — 14, накреслення — напівжирний, інтервал між символами — ущільнений 0,8 pt.

**6)** Збережіть текст у файл з іменем Вправа 11 у папку Тексти власної папки. Закінчіть роботу з текстовим процесором.

**Комп'ютерне тестування**

Виконайте тестове завдання 11 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 12. Форматування абзаців. Створення однорівневих списків

Текст зазвичай поділяється на абзаци. **Абзац** — це частина тексту, що завершується недрукованим символом, який уводиться натисканням клавіші Enter. Щоб його побачити або приховати, необхідно на вкладці Основне в групі Абзац натиснути кнопку ¶.

### ► Дії з абзацами

Під час набирання й редагування тексту можна виконувати такі дії з абзацами:

Назва дії	Опис дії
Розбити абзац на два абзаци	В абзаці, який необхідно розділити на два, потрібно встановити курсор у місці поділу й натиснути клавішу Enter
Об'єднати два абзаци в один	Установити курсор після останнього символу в абзаці, до якого потрібно приєднати наступний абзац, і натиснути клавішу Delete
Вставити порожній абзац	Установити курсор наприкінці абзацу, після якого слід уставити порожній абзац, і натиснути клавішу Enter
Видалити порожній абзац	Установити курсор на порожньому абзаці й натиснути клавішу Delete

### Властивості абзаца

Вам уже відомо, що абзац як об'єкт текстового документа має певні властивості: вирівнювання, відступи, міжрядковий інтервал та ін.

Абзац можна вирівнювати за лівим краєм, по центру, за правим краєм, за ширину. Для вирівнювання абзацу достатньо встановити на нього курсор і на вкладці Основне кладці відповідну кнопку на стрічці інструментів (рис. 12.1).

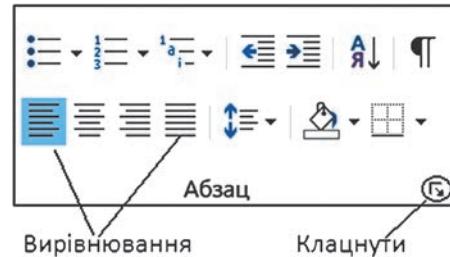


Рис. 12.1

### ► Відступи та виступи

Абзац може мати відступи і виступи (рис. 12.2).

**Відступ ліворуч** — установлення величини відступу абзацу від лівої межі робочої області сторінки.

**Відступ праворуч** — установлення величини відступу абзацу від правої межі робочої області сторінки.

### ► РОЗДІЛ 3. Опрацювання текстових даних

Спеціальний: **Перший рядок На:** — встановлення зсуву пра-воруч від першого рядка абзацу щодо лівої межі абзацу.

Спеціальний: **Навислий На:** — встановлення зсуву ліворуч від першого рядка абзацу щодо лівої межі абзацу.

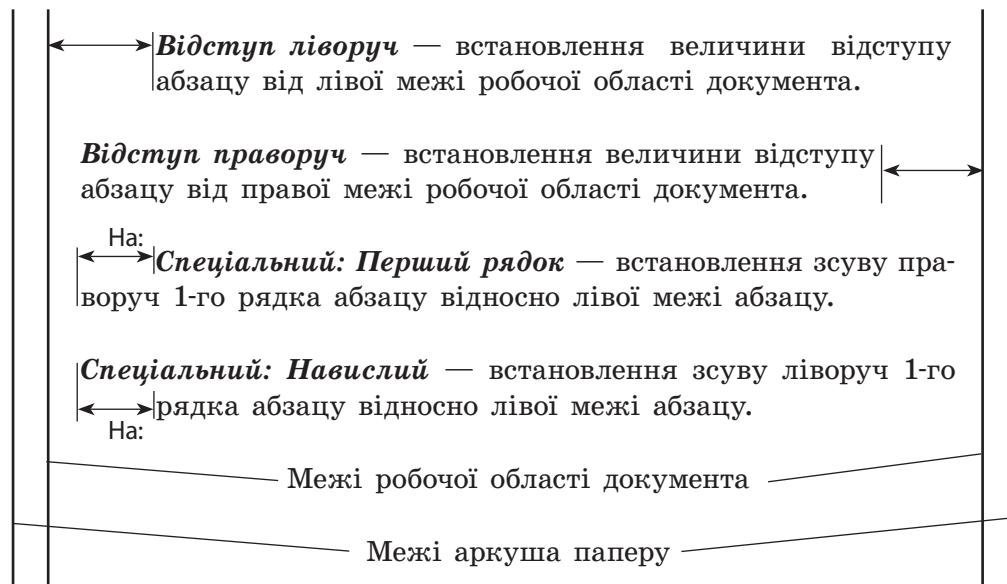


Рис. 12.2

Для встановлення відступів абзацу можна також скористати-ся горизонтальною лінійкою (рис. 12.3). Щоб змінити величини відступів і виступів, достатньо перетягти на нове місце відповід-ний маркер.

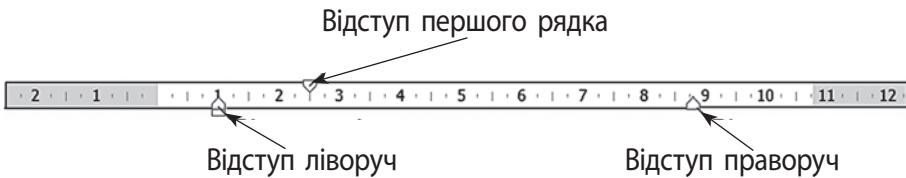


Рис. 12.3

Установити відступи й виступи можна за допомогою відповід-них інструментів вкладки Основне або діалогового вікна Абзац, для виклику якого слід клацнути спеціальний значок (рис. 12.4).

Діалогове вікно Абзац можна викликати командою Абзац... із контекстного меню, класнувши правою кнопкою миші на потрібному абзаці.



Рис. 12.4

Для встановлення відступів і виступів необхідно:

- 1) установити курсор на абзац і викликати діалогове вікно Абзац;
- 2) у діалоговому вікні Абзац (на рис. 12.5 подано його фрагмент) за допомогою лічильників Ліворуч (тут 0,5 см) і Праворуч (тут 0,7 см) установити потрібні лівий і правий відступи.

Відступ або виступ першого рядка встановлюється в списку Спеціальний за допомогою лічильника На::



Рис. 12.5

### ► Міжабзацний і міжрядковий інтервали

У текстовому документі можна встановлювати й змінювати інтервали між абзацами та рядками.

**Міжабзацний інтервал** встановлюють за допомогою лічильників Перед: і Після: (на рис. 12.5 Перед: — 12 pt, Після: — 6 pt).

Для встановлення міжрядкового інтервалу необхідно:

- 1) на вибраному абзаці або групі виділених абзаців установити вказівник;
- 2) відкрити список Міжрядковий інтервал у діалоговому вікні Абзац;

## ► РОЗДІЛ 3. Опрацювання текстових даних

3) скористатися одним зі способів:

- задати один зі стандартних інтервалів: одинарний, 1,5 рядка, подвійний або інше (рис. 12.6);
- вибравши Точно, установити на лічильнику На: число-ве значення інтервалу в пунктах менше 12 (стиснутий) або більше 12 (розтягнутий);
- вибравши Множинний, установити на лічильнику На: значення менше 1 — міжрядковий інтервал стиснутий або більше 1 — розтягнутий (на рис. 12.6 установлено 1,08);

4) натиснути кнопку ОК.

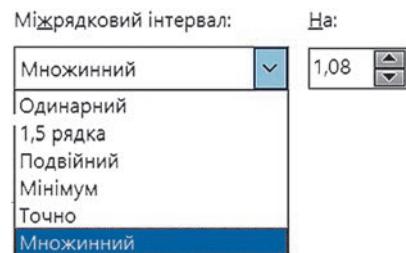


Рис. 12.6

## Створення однорівневих списків

Текстовий процесор дає змогу подавати текст у вигляді списків, маркувати списки символами, нумерувати цифрами або літерами.



**Маркований список** — це послідовність абзаців, на початку кожного з яких стоїть маркер (спеціальний символ).



**Нумерований список** — це послідовність абзаців, на початку кожного з яких стоїть порядковий номер або літера.

Створити однорівневий список можна за допомогою інструментів групи Абзац на вкладці Основне. Для цього необхідно виділити потрібні абзаци, для *маркованого списку* відкрити бібліотеку маркерів і вибрати потрібний маркер (рис. 12.7), для *нумерованого списку* відкрити бібліотеку нумерованих списків і вибрати потрібну нумерацію; натиснути кнопку ОК.

У подальшому під час поповнення списку нові елементи будуть відповідно маркуватися або нумеруватися. У нумерованих списках у разі вставляння або вилучення елементів нумерація відновлюється автоматично.

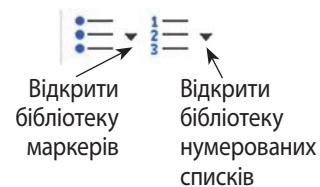


Рис. 12.7

Маркований список можна перетворювати на нумерований і навпаки. Для цього необхідно виділити весь список і вибрати для нього новий вигляд.

### Питання для самоперевірки



1. Які операції виконують над абзацами?
2. Опишіть послідовність дій для розбиття одного абзацу на два.
3. Наведіть послідовність дій для об'єднання двох абзаців.
4. Як можна вирівняти абзац?
5. Які відступи може мати абзац?
6. Які виступи може мати абзац?
7. Як установити потрібні інтервали перед абзацем і після нього?
8. Назвіть засоби створення однорівневого списку.

### Вправа 12



- Виконайте вказані дії над абзацами, створіть однорівневий список.
- 1) Завантажте текстовий процесор. Задайте: розмір паперу — А5, поля — по 2 см, орієнтацію — книжкову, шрифт — Arial, розмір — 12, накреслення — звичайний, масштаб — За ширину сторінки, вирівнювання — За лівим краєм. Інтервал Перед: і Після: — 0 pt, Міжрядковий інтервал: — Одинарний.
  - 2) Наберіть такий текст:
    - 1 абзац. Робити «пропуск» між словами. Після розділового знака ставити «пропуск».
    - 2 абзац. До і після дефісу «пропуск» не ставити.
    - 3 абзац. До і після тире ставити «пропуск».
  - 3) Перший абзац розбийте на два абзаци. Об'єднайте перші два абзаци до попереднього вигляду. Після другого абзацу вставте порожній абзац. Після нового четвертого абзацу вставте два порожні абацаці.
  - 4) Установіть відступи для первого абзацу: ліворуч — 1 см, праворуч — 2 см, для первого рядка — 1 см. Установіть відступи для другого абзацу: ліворуч — 0 см,



праворуч — 3 см, виступ навислий для першого рядка — 1 см.

- 5)** Установіть для третього абзацу: відступ ліворуч — 1 см, відступ праворуч — 0 см, виступ навислий для першого рядка — 3 см.
- 6)** Після останнього рядка наберіть список із 5 своїх друзів, промаркуйте його. Змініть нумерацію списку на арабські числа. Для всього тексту встановіть міжрядковий інтервал із множником 1,1.
- 7)** Збережіть текст у файл з іменем Вправа 12 у папку Тексти власної папки. Закінчіть роботу з текстовим процесором.



#### Комп’ютерне тестування

Виконайте тестове завдання 12 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## Практична робота 3

### Створення текстового документа



**Завдання:** навчитися створювати й форматувати простий текстовий документ.

**Обладнання:** комп’ютер зі встановленою програмою Microsoft Word.

#### Хід роботи

*Під час роботи з комп’ютером дотримуйтесь правил безпеки.*

- 1. Підготуйте комп’ютер до роботи. Завантажте текстовий процесор. Задайте розмір паперу — А5, поля — по 2 см, орієнтацію — книжкова, шрифт — Arial, розмір — 12, накреслення — звичайний, масштаб — Одна сторінка, вирівнювання — За лівим краєм, інтервал Перед: і Після: — 0 pt, міжрядковий інтервал — 1,5 рядка.

- 2. Наберіть текст заяви довільного змісту (не менше чотирьох рядків; зразок не повторювати!) на ім'я класного керівника або директора школи від власного імені. Потім відформатуйте текст за зразком (рис. 12.8).

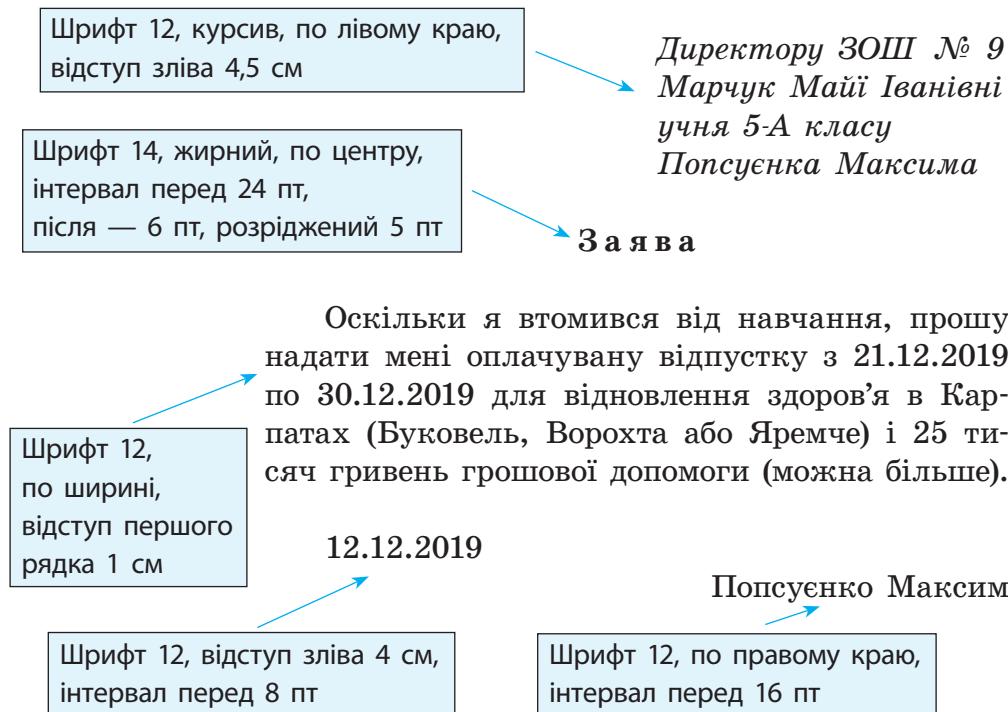


Рис. 12.8

- 3. Для перших 4 рядків установіть розмір — 12, накреслення — курсив, відступ зліва — 4,5 см.
- 4. Для заголовка встановіть розмір — 14, накреслення — напівжирний, вирівнювання — по центру, інтервал: перед — 24 pt, після — 6 pt, розріджений на 5 pt.
- 5. Для змістової частини заяви встановіть шрифт 12, вирівнювання за ширину, відступ першого рядка — 1 см.
- 6. Для прізвища заявника встановіть розмір — 12, за правим краєм, інтервал перед — 18 pt. Для дати встановіть розмір 12, відступ зліва — 4 см, інтервал перед — 6 pt.

## ► РОЗДІЛ 3. Опрацювання текстових даних

- 7. Збережіть документ у файлі з іменем Робота 3 у папці Тексти власної папки. Закрійте всі вікна. Закінчіть роботу на комп'ютері.

**Зробіть висновок:** які засоби для створення й форматування простого текстового документа надає текстовий процесор Microsoft Word?

## § 13. Додавання зображень із файлів та їх форматування

У документ можна додавати зображення з папок на поточному комп'ютері або шукати зображення в Інтернеті, щоб вставляти їх у документи.

### Вставляння зображень з комп'ютера

Щоб вставити зображення в документ із файла, який зберігається на комп'ютері, необхідно:

- 1) установити курсор на те місце документа, де має бути зображення;
- 2) у вкладці Вставлення в групі Ілюстрації натиснути кнопку Зображення (рис. 13.1) — з'явиться вікно Цей комп'ютер;

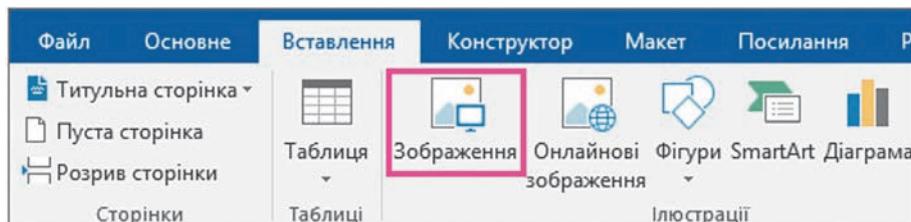


Рис. 13.1

- 3) у вікні Цей комп'ютер знайти й відкрити папку із зображеннями;

## § 13. Додавання зображень із файлів та їх форматування

- 4) перейти в папці до зображення, яке потрібно вставити, вибрати його й натиснути кнопку Вставити — через деякий час зображення з'явиться в документі.

**Положення зображення в тексті** може бути різним. Для налаштування «взаємодії» зображення й тексту необхідно:

- 1) клацнути потрібне зображення, щоб активізувати його;
- 2) викликати на зображенні контекстне меню й вибрати в ньому команду Формат зображення;
- 3) у діалоговому вікні Формат зображення, що з'явиться (рис. 13.2), вибрати вкладку Розташування;
- 4) у групі Обтікання вибрати потрібний варіант взаємодії тексту й зображення (тут — навколо рамки);
- 5) у групі Горизонтальне вирівнювання ввімкнути перемикач для розташування зображення на сторінці (тут — справа);

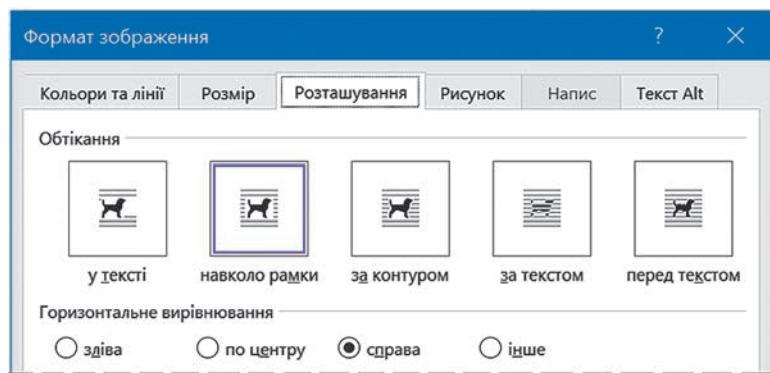


Рис. 13.2

- 6) підтвердити вибір, натиснувши кнопку OK.

## Операції над зображенням

Як вам відомо, зображення можна копіювати, переміщувати, змінювати його розмір тощо.

### ► Зміна розмірів зображення

Щоб змінити розміри зображення, уже вставленого в текстовий документ, необхідно виконати такі дії (рис. 13.3):

- 1) активувати зображення, клацнувши його, — з'являться кутові маркери;

## ► РОЗДІЛ 3. Опрацювання текстових даних

- 2) навести вказівник на один із маркерів навколо зображення так, щоб з'явилася двостороння стрілка;
- 3) щоб пропорційно збільшити (або зменшити) розміри зображення, перетягти кутовий маркер від центру (або до центру) й у разі досягнення необхідних розмірів відпустити;
- 4) щоб розтягнути (або стиснути) зображення в одному з напрямків, навести вказівник на один із маркерів на середині сторони й перетягти його від центру (або до центру) й у разі досягнення потрібних розмірів відпустити.



Рис. 13.3

### ► Переміщення зображення

Для переміщення зображення в інше місце документа необхідно:

- 1) навести на зображення вказівник — він набуде вигляду чотиристоронньої стрілки (рис. 13.4);
- 2) притиснути ліву кнопку миші й, не відпускаючи її, перетягти зображення на потрібне місце в документі.



Рис. 13.4

### ► Копіювання зображення

Копіювання зображення здійснюється так само, як і копіювання текстового фрагменту: на зображені викликати контекстне меню, у якому вибрati команду Копіювати; на місці вставлення в цьому або іншому документі викликати контекстне меню та вибрati в ньому команду Вставити.

**!** Команду Вставити можна виконувати багато разів, щоразу отримуючи нову копію зображення.

У разі потреби копію зображення можна перемістити в інше місце. Копія зображення з'явиться на новому місці, а його оригінал залишиться на старому.



### Питання для самоперевірки

1. Звідки можна додавати зображення в документ?
2. Як вставити зображення в текстовий документ?
3. Яким засобом установлюється положення зображення в тексті?
4. Що таке обтікання зображення текстом?
5. Як налаштовувати обтікання зображення текстом?
6. Які маркери з'являються у виділеному зображенні?
7. Як змінюють розміри зображення?
8. Як перемістити зображення?

### Вправа 13



► У текстовий документ вставити зображення, зробити його копії, виконати над ними вказані дії.

- 1) Завантажте текстовий процесор. Задайте розмір паперу — А5, поля — по 2 см, орієнтацію — книжкову, шрифт — Times New Roman, розмір — 10, звичайний, масштаб — За ширину сторінки, вирівнювання — За лівим краєм, інтервал Перед: і Після: — 0 pt, міжрядковий інтервал — Одинарний.
- 2) Наберіть абзац із трьох рядків довільного тексту (наприклад із підручника).
- 3) Виділіть набраний абзац і командами Копіювати → Вставити створіть документ із 15 рядків. Уставте в документ указане вчителем зображення.
- 4) Виберіть для зображення обтікання текстом — Навколо рамки, розташування — за лівим краєм; висота зображення — 3 см.
- 5) Зробіть чотири копії зображення, розмістіть їх в один рядок.
- 6) Одне із зображень розтягніть, інше — стисніть. Поверніть два зображення за зразком (рис. 13.5). Згрупуйте зображення.
- 7) Збережіть документ у файл з іменем Вправа 13 у папку Тексти власної папки. Закінчіть роботу з текстовим процесором і на комп’ютері.





Рис. 13.5

**Комп’ютерне тестування**

Виконайте тестове завдання 13 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 14. Додавання, редагування та форматування таблиць

Для більшого уточнення даних у текстовий документ можна вставити таблицю.

### Додавання таблиці

Щоб вставити таблицю, необхідно:

- 1) установити курсор у текстовому документі на місце, де має бути таблиця;
- 2) у вкладці Вставлення в групі Таблиці натиснути кнопку Вставить таблицю — з’явиться вікно Вставлення таблиці (рис. 14.1);
- 3) на лічильниках установити потрібну кількість стовпців і рядків;

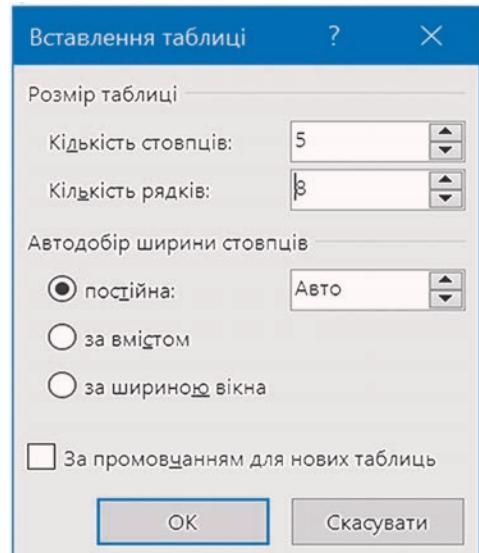


Рис. 14.1

- 4) підтвердити вибір, натиснувши кнопку ОК — у вказаному місці з'явиться таблиця із зазначеною кількістю стовпців і рядків.

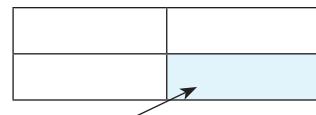
## Виділення в таблиці

Пригадаємо, що таблиця складається з рядків, стовпців, клітинок. Ці об'єкти перед початком роботи з таблицею необхідно виділити.

Щоб **виділити одну клітинку**, достатньо біля її лівої межі (рис. 14.2) установити вказівник і, коли з'явиться чорна стрілка, клацнути — клітинка зафарбується в сірий колір.

Щоб **виділити кілька клітинок**, достатньо на першій із них установити вказівник і, коли з'явиться чорна стрілка, натиснути ліву кнопку миші і, не відпускаючи її, перетягти стрілку до останньої клітинки й відпустити кнопку — клітинки зафарбуються в чорний колір.

Щоб **зняти виділення**, достатньо клацнути за межами таблиці.



Виділити клітинку

Рис. 14.2

## Редагування та форматування таблиці

### ► Зміна розмірів стовпців і рядків

Можна, не виділяючи клітинок таблиці, перевести вказівник на внутрішню вертикальну (горизонтальну) межу стовпчика (рядка) — з'явиться двостороння стрілка (рис. 14.3).

Натиснути ліву кнопку миші і, не відпускаючи її, перетягти стрілку для збільшення або зменшення ширини стовпчика (рядка).

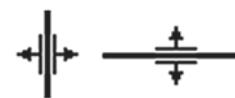


Рис. 14.3

Поки клітинки порожні, із клітинки в клітинку можна пересуватися клавішами керування курсором. Якщо клітинки мають текст, то клавішами «ліворуч» або «праворуч» пересуваються від символу до символу до кінця або до початку тексту і лише тоді — у сусідню клітинку.

## ► РОЗДІЛ 3. Опрацювання текстових даних

### ► Форматування клітинок

Таблиці можна надати бажаного вигляду, розділивши або об'єднавши клітинки, а також додавши або видаливши стовпці, рядки чи межі. Розглянемо операції над рядками та стовпцями.

**Об'єднати клітинки** — виділити дві й більше суміжних горизонтальних і (або) вертикальних клітинок, викликати на них контекстне меню й вибрати команду **Об'єднати клітинки...**

**Розділити клітинки** — на виділеній клітинці слід викликати контекстне меню, вибрати команду **Розділити клітинки...** й у контекстному меню **Поділ клітинок** зазначити потрібну кількість стовпців і рядків (рис. 14.4).

**Вставити стовпець** — виділити стовпець, викликати на ньому контекстне меню, вибрати команду **Вставити** (рис. 14.5), вибрати команду **Вставити стовпці зліва** або **Вставити стовпці справа**. Щоб вставити кілька стовпців, потрібно стільки ж стовпців виділити й виконати попередні дії.

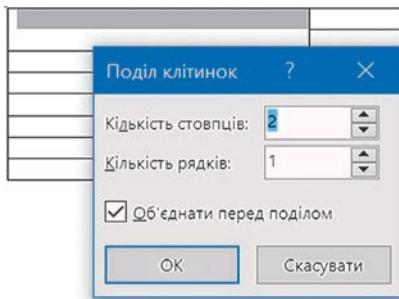


Рис. 14.4

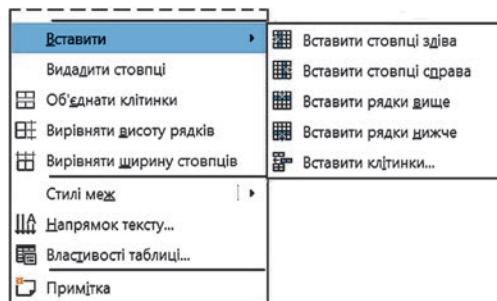


Рис. 14.5

**Вставити рядок** — виділити рядок, вибрати команду **Вставити рядки вище** або **Вставити рядки нижче**. Щоб **вставити кілька рядків**, потрібно стільки ж рядків виділити й виконати попередні дії.

! Усі перелічені операції над рядками та стовпцями можна виконати за допомогою інструментів групи **Рядки та стовпці**, що містяться на вкладці **Макет**.

**Змінити напрямок тексту** — у виділеній клітинці можна повернути горизонтальний текст вертикально і навпаки, для чого

на ньому викликати контекстне меню, вибрати команду Напрямок тексту — з'явиться діалогове вікно Напрямок тексту, у якому клацнути потрібну орієнтацію (рис. 14.6).



Рис. 14.6

#### Вертикальне вирівнювання

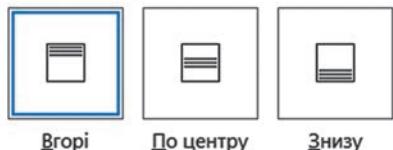


Рис. 14.7

**Вирівняти текст у клітинках** — виділити потрібні клітинки з текстом, викликати на них контекстне меню, вибрати команду Властивості таблиці... — з'явиться нове контекстне меню, у якому відкрити вкладку Клітника й вибрати Вертикальне вирівнювання (рис. 14.7). Горизонтальне вирівнювання виконується тими самими засобами, що й для абзацу.

### ► Межі таблиці

Для встановлення меж таблиці необхідно:

- 1) виділити таблицю, клацнувши значок «+» у лівому верхньому куті;
- 2) викликати контекстне меню; вибрати команду Властивості таблиці, вкладку Таблиця;
- 3) вибрати команду Межі й заливка;
- 4) відкрити вкладку Межі, вибрати потрібний тип меж у списку Тип: і потрібний стиль меж у списку Стиль:;
- 5) вибрати колір і ширину меж у відповідних вікнах — передній вигляд можна спостерігати праворуч від вікна;
- 6) клацнути кнопку ОК.

### Питання для самоперевірки



1. Як вставити в документ таблицю?
2. Як змінити розміри стовпця або рядка?
3. Як виділити для роботи кілька клітинок, стовпців або рядків?
4. Як об'єднати кілька клітинок таблиці в одну клітинку?

## ► РОЗДІЛ 3. Опрацювання текстових даних

5. Як розбити одну клітинку таблиці на кілька клітинок?
6. Як вирівняти ширину стовпців або рядків?
7. Як змінити напрямок тексту в таблиці?
8. Яким чином установити потрібні межі таблиці?

### Вправа 14



► У текстовому документі створіть таблицю, заповніть і відформатуйте її до вказаного вигляду.

- 1) Завантажте текстовий процесор. Задайте розмір паперу — А5, поля — по 2 см, орієнтацію — альбомна, шрифт — Arial, розмір — 10, звичайний, масштаб — За ширину сторінки, вирівнювання — За лівим краєм, інтервал Перед: і Після: — 0 pt, міжрядковий інтервал — Одинарний.
- 2) Створіть таблицю із 7 стовпців і 6 рядків.
- 3) У клітинки заголовка таблиці введіть потрібний текст: «Дні та числа», «№ уроку», «Час» і т. д.
- 4) Вирівняйте текст у клітинках, установіть розміри стовпців згідно із щоденником (рис. 14.8).



Дні та числа	№ уроку	Час	Предмети	Завдання додому	Бали	Підпис учителя
Понеділок	1	8.30–9.15	Математика	п. 3–31	10	
	2	9.25–10.10	Природознавство	с. 45–48	8	
	3	10.20–11.05	Українська мова	п. 15	11	
	4	11.25–12.10	Англійська мова	п. 25	9	
	5	12.20–13.05	Інформатика	с. 22–28	12	

Рис. 14.8

- 
- 5) У відповідні клітинки введіть Понеділок \_\_\_, порядкові номери та час проведення уроків, назви предметів, завдання додому. Оформіть межі таблиці.
  - 6) Скопіюйте таблицю і вставте під початковою. Відредактуйте її, унісши дані для наступного дня тижня, унесіть зміни в клітинки Предмети, Завдання додому, Бали.
  - 7) Збережіть текст у файл з іменем Вправа 14 у папку Тексти власної папки.



### Комп'ютерне тестування

Виконайте тестове завдання 14 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 15. Сторінки документа та їх форматування



**Колонтитул** — це текст, розміщений на зарезервованому місці вгорі і (або) внизу аркуша для друкування номера сторінки, назви книги, розділу тощо.

Щоб установити колонтитули, необхідно:

- 1) у вкладці Вставлення на панелі Колонтитули вибрати Верхній колонтитул або Нижній колонтитул — з'явиться вікно Убудовані, у якому для першого випадку знаходиться такий фрагмент (рис. 15.1);
- 2) вибрати Змінити верхній колонтитул (Змінити нижній колонтитул) — документ перемкнеться в режим роботи з колонтитулами, а основний текст документа зробиться блідим і стане недоступним для редагування;
- 3) в обведену лінією область колонтитула ввести текст, який можна форматувати як звичайний текст;
- 4) натиснути кнопку Закрити колонтитули або двічі клацнути на основному тексті — на всіх сторінках з'являться встановлені колонтитули, і основний текст стане доступним для редагування.



Змінити верхній колонтитул



Видалити верхній колонтитул

Рис. 15.1

### Нумерація сторінок

Коли документ має більше двох сторінок, то може виникнути потреба у їх нумерації. Робиться це так:

- 1) у вкладці Вставлення на панелі Колонтитули відкрити список Номер сторінки;

- 2) вибрати місце розташування номера сторінки;
- 3) виконати команду Формат номерів сторінки... — з'явиться вікно Формат номера сторінки (рис. 15.2);
- 4) у списку Формат номерів: вибрати нумерацію цифрами, буквами тощо;
- 5) у перемикачі Нумерація сторінок вибрати продовжити або почати з:;
- 6) для останнього випадку на лічильнику встановити число, з якого має починатися нумерація;
- 7) натиснути кнопку OK.

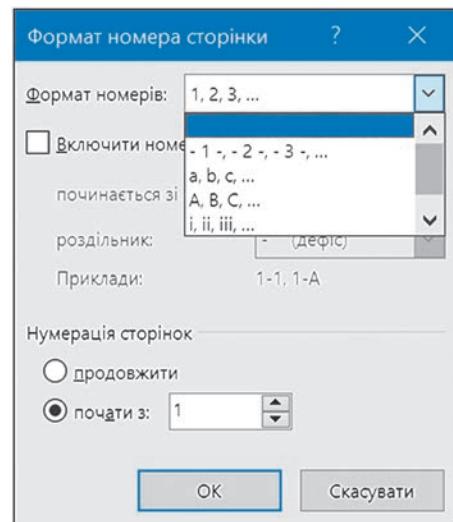


Рис. 15.2

## Стильове оформлення тексту



**Стиль** — це іменований набір значень параметрів оформлення тексту (шрифти й вирівнювання заголовків, відступи й інтервали абзаців, марковані та нумеровані списки тощо).

Стиль установлюється для окремого слова, кількох слів, для всього абзацу, для кількох виділених абзаців або всього документа.

Для нового документа у вкладці Основне пропонується набір стилів, повний перелік яких відкривається після натискання кнопки в правому нижньому куті (рис. 15.3).

Щоб застосувати стиль до окремих символів, їх слід попередньо виділити.

Щоб застосувати стиль для абзацу, необхідно встановити курсор на нього, відкрити список Стиль і вибрати потрібний. Текст абзацу набуде вигляду згідно з вибраним стилем.

! Щоб застосувати єдиний стиль для кількох абзаців або всього документа, необхідно їх попередньо виділити.

<u>АаБбВвГг</u>	<u>АаБбВвГг</u>	<b>Розділ</b>	<b>1.</b>	<b>АаБб</b>	<u>АаБбВвГг</u>	<b>1.1.1.1.</b>	<b>1.1.1.1.1.</b>	<b>1.1.1.1.1.1.</b>	<b>1.1.1.1.1.1.1.</b>	<b>АаБбВвГг</b>	<u>АаБбВвГг</u>	<b>АаБбВвГг</b>
1 Абзац с...	Виділення	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...
<b>1.1.1.1.1.1.</b>	<b>1.1.1.1.1.1.1.</b>	<b>1.1.1.1.1.1.1.1.</b>	<b>1.1.1.1.1.1.1.1.1.</b>	<b>1.1.1.1.1.1.1.1.1.1.</b>	<b>АаБбВвГг</b>	<u>АаБбВвГг</u>	<b>АаБбВвГг</b>	<u>АаБбВвГг</u>	<b>АаБбВвГг</b>	<u>АаБбВвГг</u>	<b>АаБбВвГг</b>	<u>АаБбВвГг</u>
1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Заголов...	1 Назва	1 Назва	1 Назва о...
АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт	АаБбВвГт
Підзаголо...	Строгої	1 Без інте...	Слабке ви...	Сильне в...	Цитата	Насичена...						
ААББВВГТ,	ААББВВГ	АаБбВвГт	АаБбВвГт	АаБбВвГт								
Слабке п...	Сильне п...	Назва кн...	1 Абзац с...									

 Створити стиль  
 Очистити форматування  
 Застосувати стилі

Рис. 15.3

## Стилі заголовків

Заголовки розділів, підрозділів тощо мають бути оформлені з використанням однакових стилів по всьому документу для забезпечення створення змісту документа. Стилі заголовків можна встановлювати за допомогою списку Стиль на панелі інструментів.

Встановлення для заголовків розділів, підрозділів тощо стилів Заголовок 1, Заголовок 2 дасть змогу автоматично сформувати зміст документа.

Виділіть текст, який необхідно включити в зміст, а потім на меню Основне виберіть стиль заголовка, наприклад Заголовок 1.

Повторіть цю дію з усіма фрагментами, які мають відображатися в змісті. Наприклад, якщо книга, яку ви пишете, містить кілька розділів, можна застосувати стиль Заголовок 1 до назв розділів. А до назв підрозділів можна застосувати стиль Заголовок 2.

## Створення змісту

Для створення змісту необхідно виконати такі дії:

- 1) установити курсор клавіатури в кінці або на початку документа;
- 2) перейти на вкладку Посилання, натиснути кнопку Зміст, а потім зі списку вибрати стиль Автоматичний зміст.

Під час установлення на змісті вказівник миші набуде вигляду руки. Достатньо притиснути клавішу Ctrl і клацнути на потрібній назві розділу, як на екрані з'явиться його текст.

## Друк і попередній перегляд документа

Друк і попередній перегляд виконується в одному вікні. Щоб отримати доступ до них, клацніть елементи Файл → Друк. Документ відобразиться праворуч.

Щоб переглянути кожну сторінку, натискайте стрілку внизу вікна попереднього перегляду (тут поточна сторінка 15, усього сторінок 18).

Якщо ж текст замалий для читання, настроюйте масштаб за допомогою повзунка 52% .

Якщо до комп'ютера під'єднано кілька принтерів, відкрийте список Принтер і виберіть потрібний пристрій. Виберіть потрібну кількість копій (на рис. 15.5 встановлено 1).

Можна друкувати всі сторінки документа або вибрати потрібні (на рис. 15.5 встановлено 2 і 4). Для друкування вибраних сторінок натисніть кнопку Друк.

## Друк

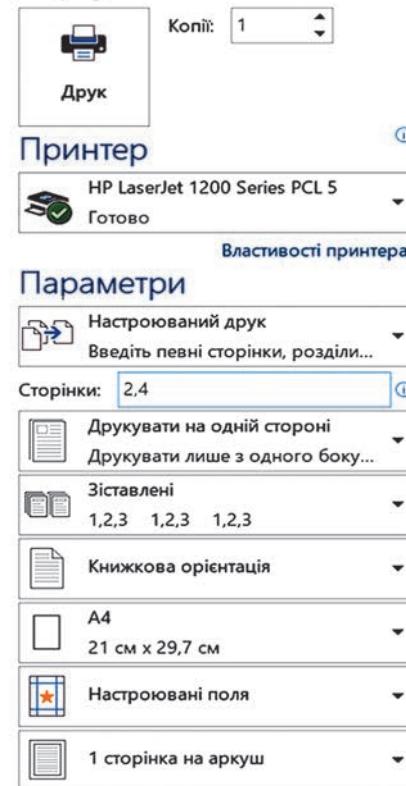


Рис. 15.5

### Питання для самоперевірки



1. Що таке колонтитул?
2. Як вставити колонтитул?
3. Як видалити колонтитул?
4. Як пронумерувати сторінки?
5. Як видалити нумерацію сторінок?
6. Як установити потрібний стиль?
7. Як побудувати зміст документа?
8. Як надрукувати текстовий документ?

**Вправа 15**

- Створіть багатосторінковий документ, виконайте вказані дії над ним, створіть зміст.
- 1)** Підготуйте комп’ютер до роботи. Скопіюйте до власної папки запропонований учителем багатосторінковий документ, відкрийте його.
  - 2)** Установіть: розмір паперу — А5, поля верхнє, нижнє, праве — 20 мм, ліве — 25 мм, орієнтація — книжкова, шрифт — Times New Roman, розмір шрифту — 14, інтервал — 1,5 рядка, вирівнювання тексту — за ширину, відступ першого рядка — 1,25 см.
  - 3)** Пронумеруйте сторінки документа знизу по центру, починаючи з другої. На першій сторінці номер не ставте.
  - 4)** Створіть верхній колонтитул, у який внесіть свої прізвище, ім’я та клас. Відформатуйте текст колонтитула: вирівнювання — по центру, шрифт — Arial, розмір шрифту — 12, червоний.
  - 5)** У документі зробіть кілька заголовків стилями Заголовок 1 та Заголовок 2 відповідно до змісту тексту.
  - 6)** На останній сторінці документа створіть дворівневий зміст.
  - 7)** Збережіть документ у файл з іменем Вправа 15 у папку Тексти власної папки.

**Комп’ютерне тестування**

Виконайте тестове завдання 15 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## Практична робота 4

### Оформлення текстового документа



**Завдання:** навчитися створювати складний текстовий документ.  
**Обладнання:** комп’ютер зі встановленою програмою Microsoft Word.

**Хід роботи**

*Під час роботи з комп’ютером дотримуйтесь правил безпеки.*

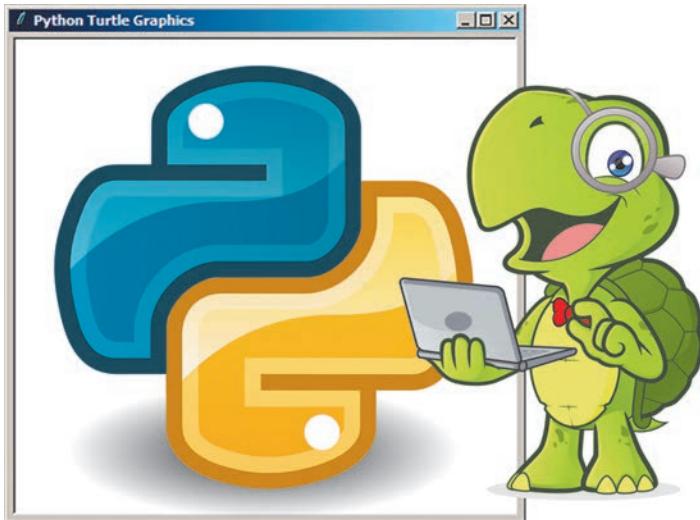
### ► РОЗДІЛ 3. Опрацювання текстових даних

- 1. Підготуйте комп’ютер до роботи. Завантажте текстовий процесор.
- 2. Знайдіть в Інтернеті та скопіюйте в текстовий документ короткі описи трьох областей України (тієї, де живете, та двох сусідніх).
- 3. Для всього документа виберіть команду Очистити формат. Видаліть усі порожні рядки. Установіть розмір паперу — А4, поля верхнє, нижнє, праве — 20 мм, ліве — 25 мм, орієнтація — книжкова, шрифт — Times New Roman, розмір шрифту — 14, інтервал Перед — 0 pt, інтервал Після — 0 pt; міжрядковий інтервал — 1,5 рядка, вирівнювання тексту — за ширину, відступ першого рядка — 1,25 см.
- 4. Установіть стилі Заголовок 1 для назв областей, Заголовок 2 — для розділів Географія, Промисловість, Населення тощо.
- 5. У кінці документа на окремій сторінці створіть автоматизований дворівневий зміст.
- 6. Для стилю Заголовок 1 виберіть шрифт — Arial, розмір — 14 pt, вирівнювання — по центру, із нової сторінки, інтервал після — 12 pt.
- 7. Для стилю Заголовок 2 виберіть шрифт — Arial, розмір — 13 pt, вирівнювання — по центру, не відривати від наступного, інтервал Перед — 12 pt, інтервал Після — 6 pt.
- 8. У верхній колонтитул запишіть власні ім’я, прізвище та клас.
- 9. Відформатуйте текст колонтитула: розмір шрифту — 16 pt, вирівнювання — по центру, колір — червоний.
- 10. Для всього документа встановіть міжрядковий інтервал — 1,5 рядка. Пронумеруйте сторінки знизу від центра, починаючи з другої. На першій сторінці номер не ставте.
- 11. На першій сторінці у верхньому лівому куті розмістіть власне фото. Поновіть зміст.
- 12. Збережіть документ у файл з іменем Робота 4 у папку Тексти власної папки. Надрукуйте документ. Закінчіть роботу з текстовим процесором.

**Зробіть висновок:** які засоби для створення складного текстового документа надає текстовий процесор Microsoft Word?

# РОЗДІЛ 4

## АЛГОРИТМИ ТА ПРОГРАМИ



- § 16. Алгоритм та його властивості
  - § 17. Виконавець алгоритмів і система його команд
  - § 18. Способи опису алгоритму. Алгоритмічні структури
  - § 19. Середовище опису й виконання алгоритмів
  - § 20. Основні поняття мови програмування Python
  - § 21. Лінійні алгоритми
  - § 22. «Черепашача» графіка
- Практична робота 5. Складання та виконання лінійних алгоритмів
- § 23. Алгоритми з розгалуженнями
  - § 24. Вкладені розгалуження
- Практична робота 6. Складання та виконання алгоритмів із розгалуженнями
- § 25. Алгоритми з повтореннями. Цикл із параметром
  - § 26. Алгоритми з повтореннями. Цикл із передумовою
- Практична робота 7. Складання та виконання алгоритмів із повтореннями

## ПОВТОРЮЄМО



На уроках інформатики в молодшій школі ви ознайомилися з поняттям алгоритму й навчилися складати алгоритми для розв'язування різних задач. Ви вже знаєте, що алгоритм можна подавати словесним і графічним способами, а також у вигляді програм, записаних певною мовою програмування.

Мова програмування — це штучна мова, яка є системою позначень і правил для запису алгоритмів у формі, придатній для їх виконання на комп'ютері. Для зручної розробки програм існують спеціальні засоби їх створення — системи програмування. Усе, що ми робимо за допомогою комп'ютера — від пошуку інформації в Інтернеті до комп'ютерної гри — можливо зауважи програмному коду, написаному програмістами. Ви вже ознайомилися з основними поняттями програмування, працюючи в програмному середовищі Scratch.

1. Що таке алгоритм?
2. Що таке програмування?
3. Що таке мова програмування?
4. Які способи опису алгоритму ви знаєте?
5. З якими мовами програмування ви вже знайомі?



▶ *Опрацювавши цей розділ, ви ознайомитеся з популярною сучасною мовою програмування Python, якою можна створювати програми для розв'язування різних задач, таких як розробка ігор, створення веб-сайтів, виконання обчислень тощо.* ▶

## § 16. Алгоритм та його властивості

### Поняття алгоритму

Згадаймо поняття алгоритму, знайоме вам із курсу інформатики 2–4 класів.



**Алгоритм** — це скінчена послідовність указівок на виконання дій, спрямованих на розв'язування задачі.

Слово «алгоритм» походить від імені арабського математика Аль-Хорезмі (800–847 pp.). *Абу Абдулла Абу Джраф Мухаммад ібн Муса аль-Хорезмі* (рис. 16.1) сформулював правила чотирьох арифметичних дій над багатозначними числами. Латиною ім'я автора європейці писали як «Algorithmi», і спочатку алгоритмами називали саме ці чотири правила виконання арифметичних дій. Із часом алгоритмами почали називати способи розв'язування найрізноманітніших задач.

У повсякденному житті людина стикається з алгоритмами, що визначають послідовність дій різної природи. Із курсу математики вам відомі алгоритми виконання арифметичних операцій над багатоцифровими числами, алгоритми знаходження коренів лінійних рівнянь та ін.

1 Рецепти приготування страв, порядок підготовки автоматичного пристрою до використання та інші послідовності розпоряджень, виконання яких дає змогу досягти поставленої мети, є алгоритмами.

Для багатьох ігор існують алгоритми виграшу, якщо результат гри залежить не від випадкового збігу обставин, а від кмітливості гравця й попередніх розрахунків.

2 Гра Баше. Є 11 предметів. За один хід гравець може взяти 1, 2, 3 предмети. Програє той, кому дістанеться останній предмет. Алгоритм виграшу для першого гравця:  
1-й хід: узяти два предмети.



Рис. 16.1

*2-й хід і далі:* брати стільки предметів, щоб кількість предметів, узятих разом із суперником за черговий хід, у сумі складала 4.

Уміння складати алгоритми є дуже важливим для людини будь-якої професії.

## Властивості алгоритму

Алгоритм складається з окремих кроків, які потрібно виконати в певному порядку. Якщо порушити порядок виконання кроків або пропустити якийсь крок, то алгоритм не буде виконаний до кінця або призведе до неправильного результату.

**3** Розглянемо алгоритм відкривання дверей.

1. Дістань ключ.
2. Встав ключ у замкову шпарину.
3. Двічі поверни ключ за годинниковою стрілкою.
4. Вийми ключ.

Чи відкриємо ми двері, якщо поміняти місцями вказівки 2 і 3?

Алгоритм має певні властивості. Щоб алгоритм виконав своє призначення, його необхідно будувати за певними правилами (рис. 16.2).

**Дискретність** означає, що алгоритм повинен складатися з окремих кроків, кожний з яких має завершуватися.

**Визначеність** означає однозначність тлумачення правил виконання кроків і порядку їх виконання. Алгоритм не повинен містити команди, які можуть сприйматися виконавцем неоднозначно.

**Виконуваність** означає, що алгоритм, призначений для певного виконавця, може містити лише ті команди, які виконавець здатний виконати. Так, алгоритм для виконавця Першокласник не може містити команду «Побудуй бісектрису даного кута», хоча

### Властивості алгоритму

- дискретність
- визначеність
- виконуваність
- скінченність
- масовість
- результативність
- формальність

Рис. 16.2

така команда може бути в алгоритмі, призначеному для виконавця Восьмикласник.

**Скінченність** — обов'язкова виконуваність алгоритму. Алгоритм має складатися зі скінченної кількості кроків, кожен з яких потребує для свого виконання скінченного проміжку часу.

4 Наведена послідовність команд є нескінченною.

1. Візьміть число 2.
2. Помножте задане число на 10.
3. Додайте до результату 5.
4. Якщо одержано додатне число, то перейдіть до команди 3, якщо ні, то припиніть виконання алгоритму.

**Масовість** означає можливість виконання алгоритму для різних входних даних. Наприклад, ви вивчали алгоритм знаходження коренів рівняння виду  $ax + b = c$  для розв'язування рівнянь  $5x + 3 = 8$ ,  $17x + 6 = 40$  та інших.

**Результативність** означає, що після виконання послідовності вказівок алгоритму має бути отримано конкретний результат. Наприклад, послідовність вказівок «налий води; увімкни плитку; вимкни плитку» не є результативною, якщо потрібно було нагріти воду.

**Формальність** означає, що будь-який виконавець, здатний сприймати й виконувати вказівки алгоритму (навіть не розуміючи його змісту), може виконати завдання за заданим алгоритмом. Як відомо, автоматизовані пристрої правильно розв'язують багато задач за заданими їм алгоритмами, хоча змісту задач вони, безумовно, розуміти не можуть.

### Питання для самоперевірки



1. Назвіть основні властивості алгоритмів і поясніть сутьожної з них.
2. Розгляньте заданий алгоритм.
  1. Прочитайте число а1.
  2. Прочитайте число а2.

3. Поділіть число  $a_1$  на число  $a_2$ .
  4. Запишіть результат.
- Чи має цей алгоритм властивості масовості та визначеності?
3. **Задача:** задане число, більше за 1, зменшити до 1 шляхом ділення на 2. Алгоритм розв'язування задачі:
    1. Поділіть задане число на 2.
    2. Якщо результат не дорівнює 1, то виконайте команду 1, інакше припиніть виконання алгоритму.

Чи має цей алгоритм властивість скінченності?

  4. Чи можна скласти алгоритми розв'язування таких задач:
    - a) знайти корінь рівняння  $ax + b = c$ ;
    - b) відвідати театр;
    - c) вивести новий сорт пшениці;
    - d) сконструювати машину для виконання домашніх завдань?
-  5. Ознайомтеся з наведеним алгоритмом отримання окропу.
1. Налийте воду в чайник.
  2. Відкрийте кран газової конфорки.
  3. Поставте чайник на плиту.
  4. Почекайте, поки вода закипить.
  5. Піднесіть запалений сірник до конфорки.
  6. Вимкніть газ.
- Визначте правильну послідовність дій, яка дозволить запобігти нещасному випадку.
- 6\*. **Задача:** перевізнику потрібно човном переправити через річку вовка, козу й капусту по одному. Опишіть алгоритм дій перевізника, виходячи з того, що небезпечно залишати разом без нагляду козу й капусту, вовка й козу.

**Вправа 16**

Скласти алгоритми розв'язування задач.

Ще із часів Середньовіччя відомі задачі на переливання рідини. У таких задачах за допомогою посудин, ємності яких заздалегідь відомі (посудини не мають поділок),

потрібно відміряти певну кількість рідини. Задачі на переливання бувають двох типів:

 <b>1</b>	 <b>2</b>
<p>Задачі, у яких необхідно отримати задану кількість рідини за допомогою кількох порожніх посудин із нескінченного джерела рідини</p>	<p>Задачі, у яких необхідно розділити рідину на задані частини з більшої посудини за допомогою кількох менших за ємністю посудин</p>

Для розв'язування задач можна застосувати такі алгоритми.

Алгоритм I	Алгоритм II
<ol style="list-style-type: none"> <li>1. Наповніть велику посудину</li> <li>2. Налийте рідину з більшої посудини в меншу</li> <li>3. Вилийте рідину з меншої посудини</li> <li>4. Повторіть дії 1–3, поки не буде отримано зазначену в умові завдання кількість рідини</li> </ol>	<ol style="list-style-type: none"> <li>1. Із більшої посудини наповніть посудину проміжної ємності</li> <li>2. Налийте рідину з проміжної посудини в найменшу</li> <li>3. Перелийте рідину з найменшої посудини в більшу</li> <li>4. Повторіть дії 2–3 доти, доки посудина проміжної ємності не стане порожньою</li> </ol>

Під час розв'язування таких задач необхідно дотримуватися певних правил:

- наливати в посудину рівно стільки рідини, якою є ємність (об'єм) посудини;
- виливати з меншої посудини в більшу всю рідину;
- виливати з посудини можна або всю рідину, або до заповнення рідиною посудини меншої ємності.



Складіть алгоритми розв'язування таких задач.

- 1)** Як, маючи два відра ємністю 2 л і 5 л, набрати з водопровідного крана 1 л води?
- 2)** Яким чином із річки можна принести рівно 6 л води, якщо є тільки два відра: ємністю 4 л і 9 л?
- 3)** Хлопців потрібно пофарбувати паркан. Він має 12 л фарби й хоче відлити половину, але в нього немає посудини ємністю 6 л. У нього є дві посудини ємністю 8 л і 5 л. Яким чином налити 6 л фарби в посудину ємністю 8 л?



### Комп'ютерне тестування



Виконайте тестове завдання 16 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 17. Виконавець алгоритмів і система його команд

Кожен алгоритм орієнтовано на певного виконавця. Виконавцем алгоритму може бути людина, тварина, комп'ютер, система людина — машина, система верстат — автомат тощо, яких «навчено» виконувати вказівки алгоритму. Якщо виконавцем є деякий пристрій, то вираз «виконавця навчено виконувати вказівку» означає, що пристрій може виконати задану вказівку автоматично, без зовнішнього втручання.



**Виконавець** — людина, тварина чи пристрій, здатні діяти за заданим алгоритмом.

Щоб скласти алгоритм, орієнтований на конкретного виконавця, необхідно знати характеристики цього виконавця.

### Характеристики виконавця

Кожен виконавець існує в певних умовах — середовищі — і може виконувати певний набір дій (рис. 17.1). Перш ніж скла-

дати алгоритм розв'язування задачі, необхідно дізнатися, у якому середовищі які дії може виконувати виконавець і за допомогою яких команд.

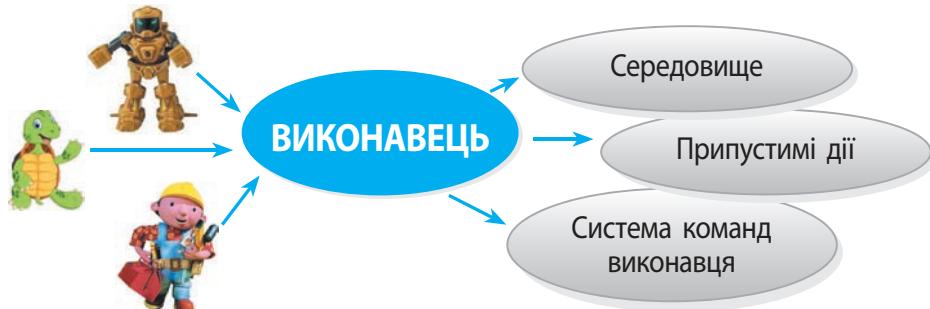


Рис. 17.1

**Середовище** — «місце існування» виконавця.

**Припустимі дії** — обмежений набір дій, що вміє виконувати певний виконавець.

Описати виконавця означає вказати його припустимі дії. Досяжні цілі — результати, які виконавець може отримати при виконанні припустимих дій.

**Система команд виконавця** — повний перелік команд, за якими виконавець може виконати одну або серію припустимих дій. Виконавця можна уявити як пристрій із кнопковим керуванням. Настикання кнопки означає виклик однієї команди.

**Відмова** — подія, що виникає в разі виклику команди в неприпустимому для цієї команди стані середовища.

Розглянемо виконавця Кресляра, який призначений для побудови малюнків на полі розміром  $5 \times 5$  клітинок (рис. 17.2) і вміє виконувати три команди: підніми перо, опусти перо, перейди до точки  $(x,y)$ .

На початку роботи Кресляр розташовується в точці  $(0,0)$  і тримає перо піднятим. Якщо перо опущене, під час пересування Кресляра за ним залишається слід.

Припустимим діям виконавця відповідає система команд:

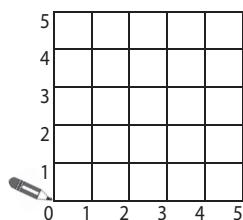


Рис. 17.2

Система команд виконавця	Дії за командами
Підніми перо	Підготуватися до переміщення без сліду
Опусти перо	Підготуватися до переміщення зі слідом
Перейди до точки $(x,y)$	Перейти до точки з координатами $(x,y)$

Відмова Кресляра виникає, якщо він отримує команду перейти до точки, яка розміщується за межами поля.

- 1 Складемо для виконавця Кресляр алгоритм побудови даху будиночка (рис. 17.3).

1. Перейди до точки  $(0,3)$ .
2. Опусти перо.
3. Перейди до точки  $(2,5)$ .
4. Перейди до точки  $(3,5)$ .
5. Перейди до точки  $(5,3)$ .
6. Перейди до точки  $(0,3)$ .
7. Підніми перо.

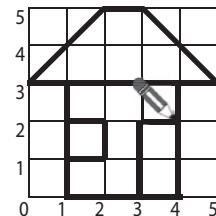


Рис. 17.3

Виконуючи алгоритм, виконавець може не розуміти смислу того, що він робить, і тим не менше отримувати потрібний результат, тобто виконавець діє формально (рис. 17.4).

Комп'ютер —  
формальний виконавець

Помилки робить не комп'ютер,  
а розробник алгоритму

Рис. 17.4

### Питання для самоперевірки



1. Поясніть поняття «виконавець алгоритму». Перелічіть характеристики виконавця.
2. Опишіть систему команд виконавця, щоб реалізувати алгоритм обчислення за формулою  $p = (1 + x) : (1 - x)$ .
3. Якими припустимими діями ви оснастили б автомат, що заміняє:
  - a) касира в магазині;
  - b) двірника;
  - b) вахтера?

- 4.** Виконавець уміє виконувати команди: помнож число на 2; знайди суму двох чисел. Складіть для виконавця алгоритм обчислення виразу  $y = 2a + 3b + 4c$  для заданих  $a$ ,  $b$  і  $c$ .
- 5\*.** Виконавець уміє виконувати команди: помнож число на 2; збільш число на 1. Складіть для виконавця алгоритм отримання:
- числа 4 з 1;
  - числа 5 з 1;
  - числа 100 з 1.
- 6\*.** Фірма «Електронні прилади» випустила автоматизовану ванну, пульт керування якої має дві кнопки: Долити 5 л і Злити 3 л. Складіть алгоритм доливання у ванну 4 л води за якомога меншу кількість команд.

**Вправа 17**

Склади алгоритми малювання фігур виконавцем Кресляр. Виконавець Кресляр розміщується на нижній частині аркуша і може виконувати такі команди:

Система команд виконавця	Дії за командами
Зроби крок	Пройти 1 см, залишаючи слід
Повернися	Повернутися на 90° ліворуч

Складіть для Кресляра алгоритм малювання таких фігур:

1)

2)

3)

**Комп'ютерне тестування**

Виконайте тестове завдання 17 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 18. Способи опису алгоритму. Алгоритмічні структури

### Форми подання алгоритмів

Існують різні форми подання алгоритмів — словесна, словесно-формульна, графічна, у вигляді програмного коду та інші — залежно від того, на якого виконавця орієнтований алгоритм.

У повсякденному житті найчастіше застосовується словесна форма. Алгоритм подається як послідовність окремих пронумерованих пунктів, кожен з яких містить команду на виконання певної дії. Команди виконують одну за одною в порядку зростання їх номерів, якщо немає спеціальної вказівки на перехід до виконання команди за іншим номером в алгоритмі.

Словесна форма подання алгоритму є найприйнятнішою для опису інструкцій побутового характеру, поведінки в разі виникнення певної надзвичайної ситуації, кулінарних рецептів тощо.

- 1** Подамо алгоритм приготування «швидкої» піци (рис. 18.1) у словесній формі.
  1. Змішайте сметану з томатною пастою, додайте спеції.
  2. Отриманий соус намастіть на основу для піци.
  3. Зверху викладіть шматочки помідорів і болгарського перцю, половинки маслин.
  4. Посипте натертим сиром.
  5. Поставте піцу в духову шафу на 5 хвилин для запікання.



Рис. 18.1

Алгоритми обчислень зручно подавати у вигляді формул. Так, алгоритм обчислення площин прямокутного трикутника для виконавця Учень 5 класу можна подати у вигляді  $S = a \cdot b : 2$ , де  $a, b$  — катети трикутника.

Записуючи алгоритми, часто комбінують словесне та формульне подання запису команд. Для графічного подання алгоритмів використовують блок-схеми.



**Блок-схема алгоритму** — графічне подання алгоритму у вигляді блоків, сполучених лініями.

Кожна команда міститься в блоці певного вигляду:

Назва блоку	Опис дії
Початок/ завершення	Позначає початок або кінець алгоритму
Процес	Позначає дію, яку потрібно виконати. Може бути позначена як указівка виконати окрему дію (додати два числа, накреслити лінію), так і послідовність логічно об'єднаних у блок дій (виконати розрахунки за заданими формулами, зробити малюнок), тобто певний процес
Введення/ виведення	Позначає введення вхідних даних або виведення вихідних даних
Умова	Позначає перевірку деякої умови. Умова може набувати одного з двох значень — TRUE (істина) або FALSE (хибність)

Розглянемо подання алгоритму в графічному вигляді.

- 2 На рис. 18.2 зображено блок-схему алгоритму знаходження середнього арифметичного трьох чисел, де:

$a, b, c$  — вхідні дані;

$S$  — проміжний результат;

$S_r$  — вихідні дані (результат виконання алгоритму).

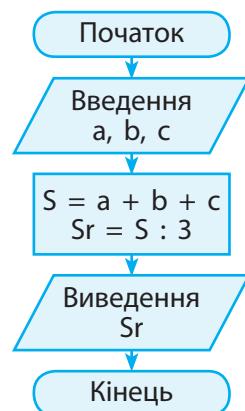


Рис. 18.2

## Базові алгоритмічні структури

Під час конструювання алгоритмів використовують три базові алгоритмічні структури: *слідування, розгалуження, повторення*. Згадаймо означення цих алгоритмічних структур.



**Слідування** — це така форма організації вказівок в алгоритмі, за якої дії виконуються послідовно одна за одною, без пропусків або повторень (рис. 18.3).



Рис. 18.3

### Алгоритм створення комп’ютерної програми.

1. Складіть алгоритм.
2. Напишіть програму.
3. Відлагодьте програму на комп’ютері.
4. Отримайте результат розв’язування задачі.



**Розгалуження** — це така форма організації дій, коли залежно від виконання або невиконання певної умови виконується одна з двох вказівок (рис. 18.4).

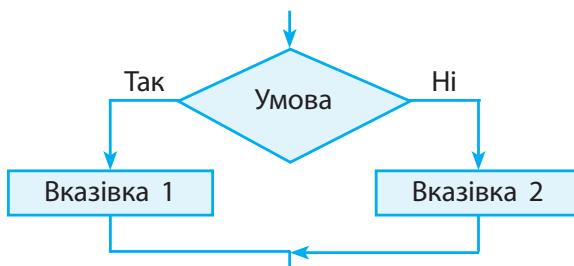


Рис. 18.4

Умову подають у вигляді запитання, сформульованого так, щоб допускалась лише одна з двох відповідей: «так» або «ні». Перевірка умови має бути допустимою дією виконавця.

Якщо умова істинна, то виконується Вказівка 1 (гілка Так); якщо умова хибна, то Вказівка 2 (гілка Ні).

Після виконання однієї з вказівок виконавець переходить до наступної після розгалуження команди.

- 3** Складемо блок-схему алгоритму обирання розваги: якщо в касі є квитки, то придбати квиток і переглянути фільм, інакше — піти на прогулку до парку (рис. 18.5).

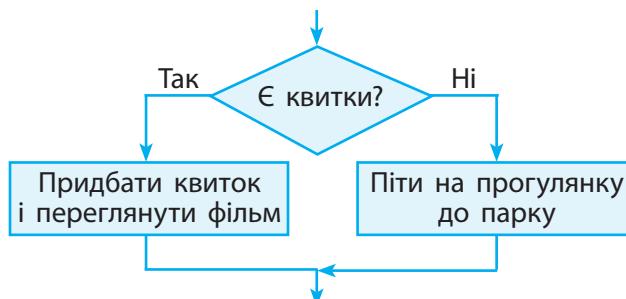


Рис. 18.5



**Повторення (цикл)** — це така форма організації дій, за якої одна і та сама послідовність дій виконується кілька разів залежно від певної умови (рис. 18.6).

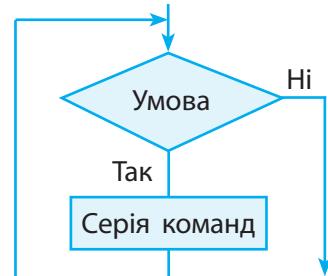


Рис. 18.6

Сукупність вказівок, що повторюється під час кожного проходження (ітерації) циклу, називається **тілом циклу**.

Наприклад, на рис. 18.6 подано циклічний алгоритм, у якому певні вказівки повторюються доти, поки задана умова є істинною.

Під час виконання такого алгоритму спочатку перевіряється умова. Якщо умова *істинна*, то тіло циклу виконується й відбувається повернення на перевірку умови.

Якщо умова є *хібною*, то виконання циклу припиняється.

**!** Якщо умова в команді повторення виявиться хібною за першої перевірки, то тіло циклу не виконається жодного разу.

Якщо під час повторення циклу умова незмінно залишається істинною, то цикл може повторюватися нескінченно (кажуть, що програма «зациклена»).

- 4 Блок-схему алгоритму збирання яблук подано на рис. 18.7.

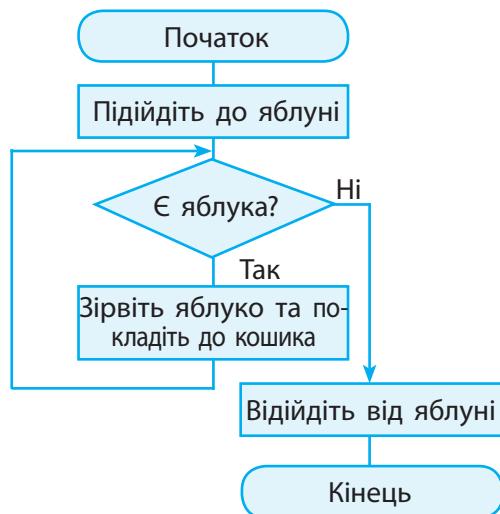


Рис. 18.7

Під час складання алгоритму розв'язування задачі необхідно отримуватися чіткого плану дій.

1. Уважно прочитайте умову задачі.
2. З'ясуйте:
  - а) що дано — *вхідні дані* (аргументи);
  - б) що потрібно знайти — *вихідні дані* (результати).
3. Визначте спосіб розв'язування задачі та виявіть необхідні проміжні величини.
4. Складіть блок-схему алгоритму.
5. Перевірте правильність складання алгоритму для різних значень аргументів.

## Програма та мова програмування

Програма — це алгоритм розв'язування певної задачі, записаний мовою програмування.



**Мова програмування** — це система позначень, яка використовується для запису алгоритмів для їх виконання за допомогою комп’ютера.

Існує багато мов програмування. Ви вже знайомі з візуальною мовою програмування Scratch і основними поняттями, які будуть корисними під час вивчення інших мов програмування.

### Питання для самоперевірки



- Назвіть основні способи опису алгоритмів.
- Як подаються базові алгоритмічні структури на блок-схемі?
- Наведіть приклади умов, які можуть бути використані для організації розгалуження.
- Поясніть переваги подання алгоритму у вигляді блок-схеми.
- Наведіть приклад алгоритму з розгалуженням, поданого у словесному вигляді.
- Наведіть приклад алгоритму з повторенням, поданого у словесному вигляді.

### Вправа 18



► Виконати завдання.

- Складіть блок-схему алгоритму обчислення площі прямокутного трикутника.
- Визначте істинність умови  $a > 5$ , якщо значення  $a$  дорівнює:
  - 8;
  - 5;
  - 2.
- Знайдіть значення величини  $c$  після виконання команди розгалуження (рис. 18.8), якщо початкове значення  $b$  дорівнює:
  - 4;
  - 6;
  - 10.

4. Знайдіть значення величини  $x$  після виконання команди повторення (рис. 18.9), якщо початкове значення  $x$  дорівнює:  
а) 4; б) 9; в) 10.

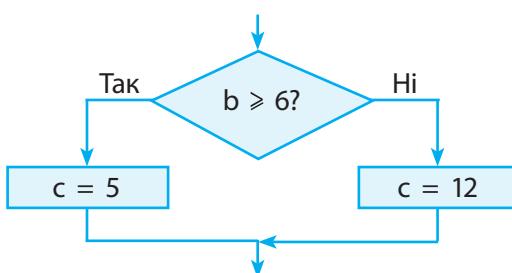


Рис. 18.8

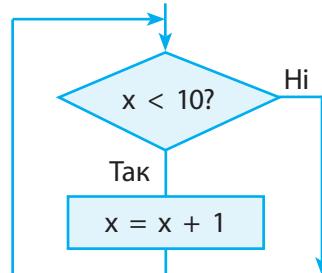


Рис. 18.9

5. Складіть блок-схему алгоритму для розв'язування задачі: якщо ціна книжки не перевищує  $n$  гривень, купити цю книжку й сувенір, в іншому випадку купити тільки книжку.  
6. Складіть блок-схему виграшного алгоритму у грі Баше для першого гравця (с. 91).



### Комп'ютерне тестування

Виконайте тестове завдання 18 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 19. Середовище опису та виконання алгоритмів

Починаємо знайомство з популярною сучасною мовою програмування Python, яка застосовується для розв'язування різних задач: написання прикладних програм, створення ігор, розробки веб-сайтів.

Мова програмування Python була створена в 1991 році нідерландським програмістом *Гвідо ван Россумом* (рис. 19.1) і названа ним на честь скетч-серіалу «Літаючий цирк Монті Пайтона» (англ. *Monty Python's Flying Circus*).

Мова Python підтримується всіма операційними системами. Існують версії для Linux, Windows. Наразі використовують дві версії Python: 2.x і 3.x. У нашому підручнику розглянемо версію Python 3.



Рис. 19.1

## Установлення програмного середовища Python

Перш ніж почати програмувати на Python, середовище програмування потрібно встановити на комп’ютер. Завантажити файл для інсталяції можна із сайта [python.org](http://www.python.org) (рис. 19.2).



Рис. 19.2

Разом із Python 3 на комп’ютер буде встановлено програму IDLE — орієнтоване на початківців середовище програмування, у якому є засоби для написання та налагодження програм мовою Python.

### Алгоритм установлення програмного середовища Python:

1. Зайдіть на сайт <http://www.python.org>
2. Клацніть Downloads, щоб відкрити сторінку завантаження (рис. 19.3).



Рис. 19.3

3. Клацніть кнопку з версією Python 3.6.1.
4. Після завантаження інсталяційного файла двічі клацніть на ньому, щоб установити Python на своєму комп’ютері.

5. Запустіть IDLE. Для цього виберіть команди Пуск → → Всі програми → Python → IDLE. Відкриється вікно IDLE (рис. 19.4).

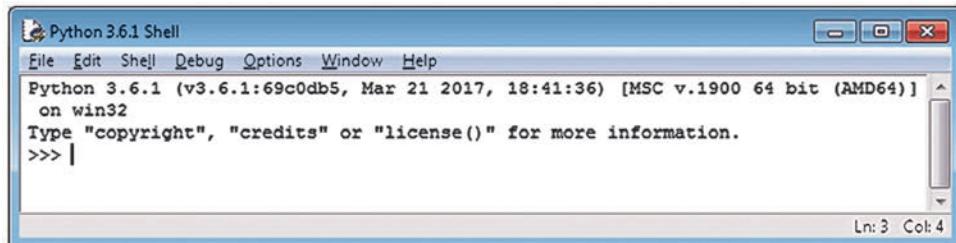


Рис. 19.4

Ми можемо починати програмувати, записуючи команди після позначки >>>.

Рядок >>> називається **запрошенням**, і його наявність означає, що комп’ютер готовий прийняти вашу першу команду.

У Python існують два види вікон: *вікно програми* і *вікно консолі* (IDLE). У вікні програми можна вводити й зберігати програмний код для подальшого виконання, а у вікні консолі — уводити команди й одразу отримувати результат виконання.

## Знайомство з IDLE

Вікно IDLE є вікном консолі. У ньому відображаються результати виконання програми й повідомлення про помилки в програмному коді.

На рис. 19.5 проілюстровано виконання команд у вікні IDLE.

 A screenshot of the Python 3.6.1 Shell window showing the following session:
 

```
>>> a = 5
>>> b = 3
>>> a + b
8
>>> a * b
15
>>> |
```

 Three callout boxes explain the results:
 

- Box 1: "Команди присвоєння значень змінним a і b" points to the first two lines of code.
- Box 2: "Команда знаходження суми двох чисел" points to the third line of code ("a + b") and its result (8).
- Box 3: "Результат виконання команди a + b" points to the result of the multiplication ("a \* b") and its result (15).

Рис. 19.5

Команди, які введено у вікні консолі, виконуються після натискання клавіші Enter, і результат одразу виводиться у вікні IDLE.

- 1 Щоб зрозуміти, як працює IDLE, після позначки >>> запишемо команду print ('Hello, World!') і натиснемо клавішу Enter. У вікні IDLE буде виведено привітання (рис. 19.6).

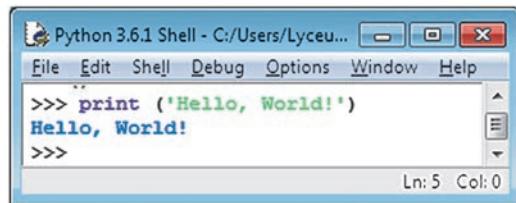


Рис. 19.6

Вікно IDLE зручно використовувати для того, щоб зрозуміти, що виконує та чи інша команда. Але для створення великих програм, які необхідно зберігати й редактувати, використовують вікно програми.

## Вікно програми

Вікно програми призначене для введення та редактування тексту програми.

Щоб відкрити вікно програми, в IDLE виберіть команду File → New File. Відкриється окреме вікно програми, яка до першого збереження має ім'я Untitled (рис. 19.7).

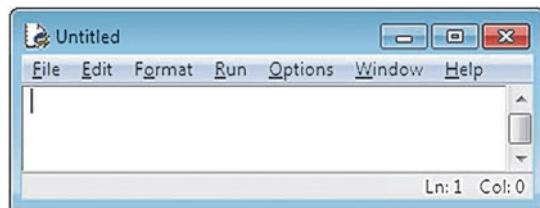


Рис. 19.7

На рис. 19.8 наведено алгоритм роботи з програмою в середовищі Python у вікні програми.



Рис. 19.8

- 2 Створимо програму розв'язання задачі за поданим вище алгоритмом роботи з програмою.

*Задача:* знайти суму і добуток двох чисел.

1. Уведемо код у вікні програми (рис. 19.9).

2. Виберемо команду File → Save As. Уведемо ім'я файла Перша програма і натиснемо Save. Файл збережено з розширенням .py.

```

print ('Наша перша програма')
a = 5
b = 3
print (a+b)
print (a*b)

```

Лн: 5 Col: 11

Рис. 19.9

3. Виберемо команду Run → Run Module (або натиснемо F5).  
4. Переглянемо результат роботи програми у вікні консолі IDLE (рис. 19.10).

```

Python 3.6.1 Shell
File Edit Shell Debug Options Window Help
Наша перша програма
8
15
>>>

```

Лн: 18 Col: 4

Рис. 19.10

Програму не можна запустити на виконання, поки вона не збережена. У разі спроби виконати програму без збереження з'явиться попередження. Зверніть увагу, що попередження виділено червоним кольором.

У Python різні складові програмного коду виділяються певними кольорами:

Команди Python, наприклад, print, — фіолетового кольору
Рядки в лапках виділені зеленим кольором. Якщо дужки навколо текстового рядка теж виділені зеленим, десь не вистачає лапок
Службові слова мови Python (if, while та інші) — помаранчевого кольору
Помилки у вікні програми та повідомлення про помилки у вікні консолі виділяються червоним

Кольорові підказки допомагають уникати помилок під час уведення тексту програми.



### Питання для самоперевірки

1. Які види вікон є у програмному середовищі Python?
2. Які повідомлення відображаються у вікні консолі?
3. Як відкрити вікно програми?
4. Для чого призначено вікно програми?
5. Опишіть алгоритм роботи з програмою в середовищі Python.
6. Поясніть, чому складові тексту програми виділяються різними кольорами.



### Вправа 19

- Скласти програму для додавання двох цілих чисел.
- 1) Завантажте програму Python → IDLE, виберіть команду File → New File (Файл → Новий файл), щоб відкрити вікно програми, уведіть текст програми:

```
print ('Моя перша програма')
a = 5
b = 10
c = a + b
print ('c='c)
```

  - 2) Для збереження файла програми виберіть команду File → → Save As (Файл → Зберегти як). У вікні збереження файла виберіть власну папку, уведіть ім'я файла Вправа19 і натисніть Save (Зберегти). Файл збережено з розширенням .py.
  - 3) Для запуску програми на виконання виберіть команду Run → Run Module (Виконати → Виконати модуль) або натисніть клавішу F5. Програма почне виконуватися. Якщо з'явиться повідомлення про помилку, перевірте, чи правильно набрано текст програми.
  - 4) Перегляньте результат виконання програми у вікні консолі. Закрийте вікно IDLE.
  - 5) Повторно виконайте програму. Відкрийте її: увійдіть у власну папку, установіть курсор на позначку програми та відкрийте контекстне меню, у якому виберіть команду Edit with IDLE (Змінити в IDLE) зі с. 112. Повторно запустіть програму Вправа19.
  - 6) Закінчіть роботу, закривши вікно IDLE.



### Комп’ютерне тестування

Виконайте тестове завдання 19 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 20. Основні поняття мови програмування Python

Для написання програми певною мовою програмування необхідно знати алфавіт мови, правила запису команд та правила їх використання в програмі.

### Алфавіт мови Python

Як і будь-яка мова, Python має алфавіт, синтаксис, семантику.

**Алфавіт мови** — набір символів, що може використовуватись у програмному коді.

Під час створення програм мовою Python можна використовувати такі символи:

- літери латинського алфавіту: A...Z, a...z;
- цифри: 0...9;
- знаки арифметичних операцій, спеціальні символи:  
+ — \* / \ ^ = < > ( ) . , : ; ' # \_.

У мові Python використовують також комбінації символів і службові слова, що мають фіксований зміст:

- комбінації символів: <=, >=, <>, = =; !=, \*\*;
- службові слова: and, elif, if, as, else, import тощо.

**Синтаксис мови** — сукупність правил побудови команд мовою програмування.

**Семантика мови** — сукупність правил виконання комп’ютером команд, записаних мовою програмування.

## Величини в мові Python

Окремий інформаційний об'єкт (число, символ, рядок тощо) називають **величиною**. Основними характеристиками величини є *назва, вид і значення*.

**Вид величини** визначає спосіб використання величини в програмі. Величина може бути *константою* (тобто *постійною*) або *змінною*.

**Константи** — це величини, значення яких не можуть змінюватися в ході виконання програми. Прикладом константи може бути число (5, 1.23) або рядок: «Це рядок!».

**Змінні** — це величини, значення яких можуть змінюватися в ході виконання програми.

З'ясуємо правила, яких необхідно дотримуватися під час надання *імен змінним*.

- Першим символом імені має бути літера чи знак нижнього підкреслювання \_.
- Решта імені може складатися з літер, чисел або знаків нижнього підкреслювання.
- Не можна використовувати спеціальні символи, такі як /, # або @. Не можна використовувати пробіли.
- Імена змінних чутливі до регістру. Наприклад, myName і myName — це різні змінні.
- Не можна надавати змінним імена команд, наприклад print.

1

Правильні імена: i, \_\_my\_name, name\_23, a1, b2.

Неправильні імена: 2things, this is spaced out, my-name.

## Типи величин у мові Python

**Тип величини** визначається обсягом пам'яті, необхідним для її збереження, множиною припустимих значень величини та операціями, які можна над нею виконувати.

Основними типами величин є *числа* й *рядки*.

У Python є два типи числових даних: цілі числа (int) і дійсні (float) — дробові числа.

Цілі числа використовують для лічби (перший, другий, третій...). Кількість учнів у класі, кількість предметів ми зазвичай указуємо за допомогою цілих чисел. Дійсні числа потрібні, коли ми хочемо вказати частину чого-небудь, наприклад, 3.5 м, 1.25 грн.

Як роздільник між цілою та дробовою частиною дійсного числа використовується крапка.

**Рядок** — це взята в одинарні або подвійні лапки послідовність будь-яких символів: цифр, літер, розділових знаків та ін.

У змінних рядкового типу зберігають фрагменти тексту.

## Математичні оператори Python

Обчислення здійснюються за допомогою арифметичних виразів. Арифметичні вирази будується з імен змінних, констант, знаків арифметичних операцій і круглих дужок так, як в математиці.

Розглянемо математичні операції та відповідні їм символи операторів:

Операція	Символ операції	Приклад	Результат
Додавання	+	Res = 15+3	Res = 18
Віднімання	-	A = Res-10	A = 8
Множення	*	A = A*2	A = 16
Ділення	/	Res = 5/2	Res = 2.5
Обчислення неповної частки від ділення	//	Res = 5//2	Res = 2
Обчислення остачі	%	Res = 5%2	Res = 1
Піднесення до степеня	**	A = 4**2	A = 16

## Випадкові числа

При створенні ігрових або тестових програм іноді потрібно вибрati число з певного проміжку випадковим чином. У програмуванні випадкові числа отримують за допомогою спеціальних пристройів — генераторів випадкових чисел. Щоб отримати випад-

кове число, необхідно за допомогою команди `import` завантажити в Python функцію `randint`. Функція `randint(x1,x2)` вибирає ціле випадкове число в діапазоні від  $x_1$  до  $x_2$ .

- 2** Отримаємо випадкове число в діапазоні від 1 до 10.

```
>>> from random import*
>>> randint (1,10)
8
>>> randint (1,10)
6
```

### Синтаксичні помилки

Якщо Python не може зрозуміти введену вами команду, то він може вивести у відповідь повідомлення про синтаксичну помилку `SyntaxError`. Місце помилки у вікні консолі помічається червоним кольором. Після команди, що містить помилку, виводиться повідомлення про помилку.

**!** Уважно читайте такі повідомлення — це допоможе зрозуміти, у чому помилка, і вправити її.

- 3** Розглянемо програмний код:

```
>>> (x+5))
SyntaxError: invalid syntax
```

У виразі допущено помилку: надруковано зайву дужку — у консолі з'являється повідомлення про синтаксичну помилку.

### Питання для самоперевірки



1. Назвіть основні характеристики величини.
2. Назвіть та охарактеризуйте види величин.
3. Які послідовності символів неприпустимо використовувати як імена (ідентифікатори) й чому?
 

а) suma;	д) a-4;
б) w1;	е) if;
в) primer 1;	ж) 8a.
г) (sum);	
4. Чим визначається тип величини?

5. Обчисліть значення виразів:
  - a)  $7/2$ ; г)  $123//100$ ;
  - б)  $7//2$ ; д)  $123\%10$ ;
  - в)  $7\%2$ ; е)  $(123//10)\%10$ .
6. Як отримати випадкове число в діапазоні від 1 до 100?

**Вправа 20**

► Розв'яжіть задачу в середовищі Python.

**Задача:** обчислити суму цифр двоцифрового числа  $a$ .

- 1) Завантажте програму Python → IDLE. В IDLE виберіть команду File → New File, щоб відкрити вікно програми. У вікні програми наберіть текст програми:

```
a = 25
a1 = a//10
a2 = a%10
sum = a1+a2
print (sum)
```

- 2) Для збереження файлу виберіть команду File → Save As. У вікні збереження файлу виберіть власну папку, уведіть ім'я файла Вправа20 і натисніть Save.
- 3) Для запуску програми на виконання виберіть команду Run → Run Module або натисніть F5.
- 4) Перегляньте результат виконання програми. Запустіть програму кілька разів, змінюючи значення  $a$ . Закрийте вікно IDLE.
- 5) Для повторного виконання програми відкрийте контекстне меню, у якому виберіть команду Edit with IDLE.
- 6)\* Змініть програму таким чином, щоб обчислювалася сума цифр трицифрового числа. Запустіть програму Вправа20. Збережіть програму, закривши вікно IDLE.

**Комп'ютерне тестування**

Виконайте тестове завдання 20 з автоматичною перевіркою результу на сайті  
[interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 21. Лінійні алгоритми



Алгоритм, у якому використовується тільки структура слідування, називається **лінійним**.

Будь-яка програма має прийняти вхідні дані (введення), опрацювати їх і повернути результат (виведення). У програмах, що реалізують лінійні алгоритми, використовуються команди введення даних, присвоєння та виведення. На рис. 21.1 наведено логічну структуру програми, у якій реалізований лінійний алгоритм.

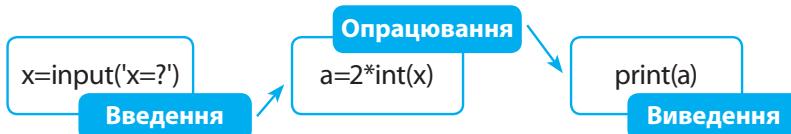


Рис. 21.1

### Введення даних

Команда `input()` призначена для введення даних із клавіатури. Коли програма зустрічає команду `input`, вона припиняє роботу й очікує, поки користувач уведе дані й натисне `Enter`. У дужках може бути записана підказка користувачеві, що саме потрібно ввести. Ця підказка виводиться на екран.

1 Випробуємо у вікні консолі, як працює `input()`:

```
>>> name = input('Як тебе звати?')
```

Як тебе звати? Петро

Значення, отримане від команди `input`, Python сприймає як рядок (послідовність літер), навіть якщо ми ввели число.

2 У разі спроби додати до значення змінної `a` число 3 виникне помилка, оскільки Python не знає, як додати число до рядка.

```
>>> a = input('a=?')
```

`a=?5`

```
>>> a+3
```

Traceback (most recent call last):

```
File "<pyshell#20>", line 1, in <module>
```

`a+3`

`TypeError: must be str, not int`

Необхідно виконати перетворення введеного значення на число за допомогою функції `int()`.

Функція `int(s)` перетворює рядок *s* на ціле число.

- 3** Тепер помилки немає:

```
>>> a = input('a=?')
a = ?3
>>> int(a)+5
8
```

Під час запису дійсного числа у формі десяткового дробу для розділення цілої та дробової частин як десятковий роздільник використовується крапка «.».

Функція `float(s)` перетворює рядок *s* на дробове число.

- 4** Введення з клавіатури дійсного значення:

```
>>> x = input('x=?')
x = ?5.25
>>> float(x)+3
8.25
```

## Указівка присвоєння

У мові Python, як і в більшості сучасних мов програмування, ми присвоюємо значення змінної за допомогою знаку `=`. Вираз присвоювання, такий як `x = 7`, наказує комп’ютеру запам’ятати число 7 у змінній *x*.

**Загальний вигляд команди присвоєння:**

$$A = B,$$

де *A* — ім’я змінної, *B* — константа, змінна або вираз.

Схема виконання вказівки присвоєння: спочатку обчислюється значення виразу в правій частині вказівки присвоєння, потім це значення надається змінній, ім’я якої записане в лівій частині (рис. 21.2).

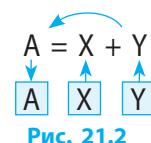


Рис. 21.2

Під час виконання команди присвоєння ділянка пам’яті, відведена для змінної, заповнюється новим значенням, одночасно знищуючи старе. Тому в правій частині виразу може бути вказане тільки ім’я змінної.

- 5 Наступні команди присвоюють змінній `rabbits` значення 5, потім те саме значення присвоюють змінній `hats`:

```
>>> rabbits = 5
>>> hats = rabbits
```

У виразах можна використовувати змінні. Якщо в правій частині оператора присвоєння записати вираз, то змінна в лівій частині набуває значення виразу.

Нехай  $a = 10$ . Тоді після виконання вказівки присвоєння  $a = a + 5$  змінна  $a$  отримає значення 15.

- 6 При спробі виконати оператор  $a + 2 = b$  буде виведено повідомлення про синтаксичну помилку (рис. 21.3).

```
>>> a=5
>>> a+2=b
SyntaxError: can't assign to operator
\ \ |
```

Рис. 21.3

## Виведення значень змінних

У попередніх прикладах ми вже стикалися з командою `print`, яка потрібна, щоб вивести текст у вікно консолі.

- 7 За допомогою команди `print` можна дізнатися значення змінної.

```
>>> print (rabbits)
5
```

- 8 Якщо потрібно вивести значення кількох змінних або виразів, їх необхідно перелічiti через кому:

```
>>> x = 4
>>> print (x, 2*x, 3*x)
4 8 12
```

- 9 За допомогою команд `input` і `print` можна організувати діалог користувача з програмою:

```
>>> name = input('Як тебе звати?')
Як тебе звати? Петро
>>> print ('Привіт, 'name)
Привіт, Петро
```

## Коментарі в програмі



**Коментар** — це текст, призначений для читання людиною, а не комп’ютером. Коментар — це підказка, яку програмісти записують у своїй програмі.

Щоб комп’ютер відрізнив команди від коментарів, у мові Python перед текстом коментаря ставиться знак #. Редактор IDLE виділяє коментарі червоним кольором, нагадуючи про те, що ці фрагменти коду будуть проігноровані.

10 Кomentar пояснює призначення наступної команди:

```
# Запит імені користувача
s = input('Як тебе звати?')
```

### Питання для самоперевірки



- Для чого призначена команда `input()`?
- Для чого призначена команда `print()`?
- Назвіть константи та змінні у списку виведення:  
`print ('a=', a, 5, '3 * b', 3*b)`
- Чому дорівнює значення  $x$  після виконання послідовності присвоювань:
 

a) $y = 2; x = y;$	b) $x = 5; x = -x;$
б) $x = 8; x = x + 2;$	г) $x = 10; x = x + 3?$
- Після виконання оператора  $x = y + x$  нові значення змінних  $x = 10, y = 3$ . Чому дорівнювали  $x$  і  $y$  до виконання оператора присвоєння?
- Після виконання оператора  $x = y + x$  нові значення змінних  $x = 3, y = 10$ . Чому дорівнювали  $x$  і  $y$  до виконання оператора присвоєння?



### Вправа 21



▶ Скласти програму-гру «Нісенітниця». Алгоритм гри такий. Програма просить увести відповіді на запитання «Хто?», «Де?», «Коли?» і зберігає введені дані в трьох різних змін-

них. Зі створених змінних із додаванням жартівливої кінцівки формується речення та виводиться на екран.

- 1) Відкрийте вікно IDLE і створіть нове вікно програми.
- 2) Уведіть текст програми:

```
c1 = input('Хто?')
c2 = input('Де?')
c3 = input('Коли?')
phrase = c1+ ' +c2+ ' +c3+ ' співає пісню'
print(phrase)
```

- 3) Збережіть файл у власній папці з іменем Вправа21. Запустіть програму на виконання, дайте відповіді на питання програми, проаналізуйте результат виконання програми.
- 4) Додайте до програмного коду оператор для введення відповіді на питання «Що робив?». Збережіть відповідь у змінній *c4*.
- 5) Додайте значення змінної *c4* до рядка *phrase*.
- 6) Збережіть програму й запустіть її на виконання.

#### Комп'ютерне тестування

Виконайте тестове завдання 21 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 22. «Черепашача» графіка

У середовищі Python Черепашкою звуться уявний робот — пристрій, який переміщається по екрану й повертається в заданих напрямках, водночас залишаючи (або, за вибором, не залишаючи) за собою намальований слід заданого кольору й ширини.

Положення та напрямок руху Черепашки відображає невелика чорна стрілочка, яка повільно пересувається по екрану. Це дає змогу відстежити рух Черепашки та зрозуміти, яким чином кожен рядок коду впливає на траєкторію її руху.

Черепашка допоможе нам вивчити основи комп'ютерної графіки, і ми будемо малювати за її допомогою цікаві рисунки.

## Система координат

Результат виконання Черепашкою команд відображається у графічному вікні Python Turtle Grafics (рис. 22.1).

Для визначення місцезнаходження Черепашки використовують координати. Ви вже знаєте, що таке координатна пряма, і вмієте визначати положення точки на прямій.

Але для малювання нам доведеться користуватися орієнтирами не тільки вздовж прямої, а й на площині. Будь-яка точка у вікні Python Turtle Grafics може бути задана парою чисел ( $X$ ,  $Y$ ).

**Координатні осі** — це дві координатні прямі, які перетинаються під прямим кутом. Центр вікна Python Turtle Grafics — точка перетину невидимих координатних осей — точка з координатами (0, 0). Вертикальна координата  $Y$  зростає знизу догори, а горизонтальна  $X$  — зліва направо.

На уроках математики ви працювали із числами, розташованими на координатній прямій праворуч від 0. Але горизонтальну числову пряму можна продовжити вліво, а вертикальну — униз від 0, а на променях ліворуч і знизу від 0 розташовані числа зі знаком мінус (-).

Повернемось до рис. 22.1. На ньому зображені 5 черепашок і вказано координати їх розташування.

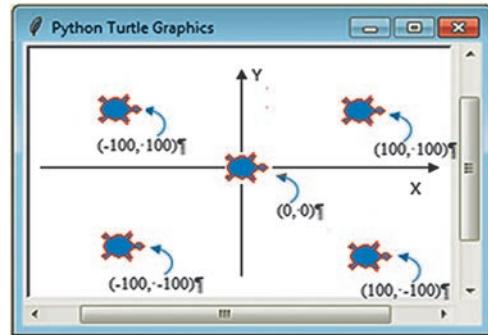


Рис. 22.1

## Команди Черепашки

Для завантаження команд для роботи з Черепашкою потрібна команда:

```
from turtle import *
```

Після введення цієї команди ви можете подавати Черепашці команди малювання. Якщо у вашій програмі є і команди введення-виведення, і команди малювання, перед початком малювання запишіть команду:

`pendown()` — почати малювати;

а завершивши створення малюнку — команду:

`penup()` — закінчити малювати.

Розглянемо команди малювання та їх призначення.

Команда	Призначення
<code>forward (n)</code>	Проповзти вперед n кроків (пікселів)
<code>left (angle)</code>	Повернути ліворуч на angle градусів
<code>right (angle)</code>	Повернути праворуч на angle градусів
<code>circle (r)</code>	Намалювати коло радіуса r
<code>circle (r,angle)</code>	Намалювати дугу радіуса   r   з градусною мірою angle
<code>goto (x,y)</code>	Перемістити Черепашку в точку з координатами (x, y)
<code>down ()</code>	Опустити перо
<code>up ()</code>	Підняти перо
<code>width (n)</code>	Установити ширину сліду Черепашки в n пікселів
<code>write (s)</code>	Вивести текстовий рядок s у точці перебування Черепашки
<code>clear ()</code>	Очищення області малювання

Застосуємо команди на прикладі.

1

Намалюємо трикутник, використовуючи команди малювання.

Уведіть ці команди у вікні консолі:

```
>>> from turtle import *
>>> forward(100)
>>> right(120)
>>> forward(100)
>>> right(120)
>>> forward(100)
```

Черепашка рухається, залишаючи за собою слід. Результат виконання команд малювання наведено на рис. 22.2.

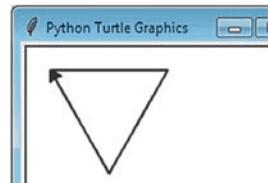


Рис. 22.2

## Створення кольорового малюнка

Черепашка може залишати не лише чорний, а й кольоровий слід. Ви можете використовувати велику кількість різноманітних кольорів.

Назва кольору (англійською мовою) береться в одинарні лапки, наприклад 'red', 'yellow', 'green' тощо. Відвідайте сайт [www.tcl.tk/man/tcl8.4/TkCmd/colors.htm](http://www.tcl.tk/man/tcl8.4/TkCmd/colors.htm), щоб переглянути повний список кольорів.

Команди для встановлення кольору сліду:

Команда	Призначення
color(s)	Установити колір сліду Черепашки в s
color(s1,s2)	Установити колір сліду Черепашки в s1, а колір заливки замкненої фігури — у s2

Команди для зафарбовування замкнених фігур:

Команда	Призначення
begin_fill()	Почати стежити за Черепашкою для заповнення області
end_fill()	Заповнити кольором s2 область, починаючи з begin_fill()

- 2 У вікні консолі введемо команди для малювання червоного кола, зафарбованого синім кольором.

```
>>> color('red', 'blue')
>>> begin_fill()
>>> circle(50)
>>> end_fill()
```

Результат виконання команд малювання кола наведено на рис. 22.3.

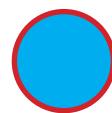


Рис. 22.3

### Питання для самоперевірки



- Яку команду потрібно застосувати для завантаження команд роботи із Черепашкою?
- Запишіть команди для Черепашки:
  - проповзти вперед 20 кроків;
  - намалювати коло радіуса 30 пікселів;

- 3) підняти перо;
- 4) установити зелений колір сліду Черепашки;
- 5) установити червоний колір сліду Черепашки, синій колір заливки замкненої фігури.

### Вправа 22



► Скласти програму малювання олімпійського символу.

- 1) Відкрийте вікно IDLE і створіть нове вікно програми.
- 2) Для малювання кожного кільця виконайте такий алгоритм: задати колір, яким будемо малювати; підняти перо, щоб під час переходу не малювати лінії; перейти на точку початку малювання кільця; опустити перо; малювати коло радіусом 45.

Уведіть текст програми:

```
color('blue') # Встановлюємо колір синій
up() # Піднімаємо перо
goto(-110, -25) # Переходимо в потрібну точку
down() # Опускаємо перо
circle(45) # Малюємо коло радіусом 45
from turtle import *
width(3) # Встановлюємо ширину сліду Черепашки
```

- 3) Текст, позначений фігурною скобкою, скопіюйте та послідовно вставте 4 рази наприкінці програми.
- 4) Відредактуйте програму для малювання всіх кілець за такими даними: друге — black (0, -25); третє — red (110, -25); четверте — yellow (-55, -75); п'яте — green (55, -75).

- 5) Над олімпійським символом червоним кольором запишіть власне прізвище та ім'я:

```
color('red')
up()
goto(0, 80)
down()
write('Лахай Оля')
```

На рис. 22.4 наведено результат малювання олімпійського символу.

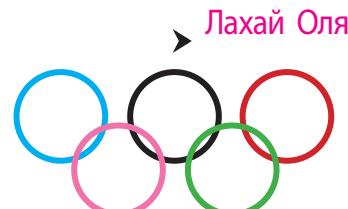


Рис. 22.4

- 6) Збережіть файл у власній папці з іменем Вправа22. Запустіть програму на виконання, проаналізуйте результат виконання програми у вікні Python Turtle Grafics. Закінчіть роботу, закривши вікно IDLE.

### Комп'ютерне тестування



Виконайте тестове завдання 22 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## Практична робота 5

### Складання та виконання лінійних алгоритмів



**Завдання:** створити програму «Калькулятор піци».

**Обладнання:** комп'ютер зі встановленим середовищем програмування Python.

Програма має виконати такі дії: запитати, скільки піц бажає замовити покупець; запитати ціну піци, зазначену в меню; обчислити підсумкову вартість покупки, ураховуючи знижку на 10 % на честь ювілею піцерії; вивести ціну покупки.

### Хід роботи

*Під час роботи з комп'ютером дотримуйтесь правил безпеки.*

- 1. Відкрийте вікно IDLE і створіть нове вікно програми.
- 2. Запишіть оператор для введення кількості піц:  
`number = int(input('Скільки піц замовляєте?'))`
- 3. Запишіть оператор для введення ціни однієї піци. Вартість однієї піци може бути дробовим числом, тому введене значення слід перетворити на дійсний тип (float):  
`cost = float(input('Скільки коштує одна піца?'))`

- 4. Запишіть оператори для обчислення ціни покупки без урахування знижки:

```
total = number * cost
print ('Ціна без знижки', total)
```

- 5. Запишіть оператори для обчислення величини знижки та ціни покупки з урахуванням знижки:

```
discount = total*0.1
total = total-discount
```

- 6. Виведіть значення змінних discount і total.

- 7. Збережіть файл у власній папці з іменем Pizza.py. Запустіть програму на виконання, перевірте її для різних початкових даних (рис. 22.5).

- 8. Закінчіть роботу, закривши вікно IDLE.

```
Python Shell
File Edit Shell Debug Options Window Help
Python34/pizza.py =====
=====
Скільки піц замовляєте? 2
Скільки коштує одна піца? 10.50
Ціна без знижки 21.0
Знижка 2.1
Ціна зі знижкою 18.9
>>>
Ln: 18 Col: 4
```

Рис. 22.5

**Зробіть висновок:** як складати та виконувати лінійні алгоритми для розв'язування задач?

## § 23. Алгоритми з розгалуженнями

### Логічні операції

Окрім уже відомих нам числового та рядкового типів даних, у мові Python є логічний тип bool. Змінна типу bool може набувати одного з двох значень — True (Істина) або False (Хибність).

- 1 Якщо змінній надати значення True, це буде змінна типу bool:

```
>>> type(True)
<class 'bool'>
>>> a = True
>>> print(a)
True
```

Умови в програмі записуються у вигляді логічних виразів. **Логічними** є вирази, результатом яких є True або False.

Простий логічний вираз (проста умова) утворюється за допомогою логічних операцій:

Знак операції	Значення	Логічний вираз	Результат
<code>==</code>	Дорівнює	<code>8==9</code>	<code>False</code>
<code>&gt;</code>	Більше	<code>8&gt;9</code>	<code>False</code>
<code>&lt;</code>	Менше	<code>8&lt;9</code>	<code>True</code>
<code>&gt;=</code>	Більше або дорівнює	<code>5&gt;=5</code>	<code>True</code>
<code>&lt;=</code>	Менше або дорівнює	<code>5&lt;=2</code>	<code>False</code>
<code>!=</code>	Не дорівнює	<code>2!=5</code>	<code>True</code>

Бувають ситуації, коли одночасно необхідно перевірити виконання кількох умов.



**Складена умова** — це кілька простих умов, з'єднаних логічними операціями AND (логічне I, інакше — логічний добуток), OR (логічне АБО, інакше — логічна сума), NOT (логічне заперечення).

Розглянемо приклади складених умов.

- 2 not `a <= 3` — рівнозначне виразу `a > 3`;  
`age >= 10 and age <= 18` — істинне тоді й тільки тоді, коли значення `age` розташовується в проміжку від 10 до 18 включно;  
`age < 10 or age > 18` — істинне для всіх значень `age`, які не належать проміжку від 10 до 18.

- 3 Логічні операції працюють і у вікні консолі. Виконаємо цей приклад, щоб ознайомитися з результатами обчислення логічних виразів:

```
>>> books = 10
>>> books == 5 # Перевірка, чи дорівнює books 5
False
>>> books > 1 # Перевірка, чи є books більшим за 1
True
>>> books >= 5
True
```

```
>>> books != 10 # Перевірка, чи не дорівнює books 10
False
>>> books == 10 or books == 5 # перевірка, чи дорівнює books 10
або 5
True
>>> books == 10 and books == 5 # перевірка, чи дорівнює books
одночасно 10 і 5
False
```

## Умовний оператор if

Оператори — це команди програми.

Оператор **if** призначено для виконання деякої послідовності дій у тому випадку, якщо істинною є зазначена умова. Цей умовний оператор відповідає алгоритмічній конструкції «неповне розгалуження» (рис. 23.1).

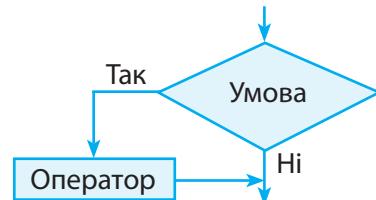


Рис. 23.1

**Синтаксис умовного оператора в неповній формі:**

**if <Умова>:**

    \_\_\_\_ <Оператор>

де \_\_\_\_ — обов'язковий відступ від лівого краю.

Оператор **if** перевіряє істинність зазначеної умови.

Якщо умова істинна (**True**), програма виконає блок дій, зазначених у частині <Оператор>.

Якщо умова хибна (**False**), то керування передається оператору, що міститься після оператора **if**.

- Після виконання цієї програми у вікні консолі отримаємо результат **Правильно**.

**books = 10**

**if books == 10:**

**print('Правильно!')**

- Вітання **Вітаю!** виводиться, якщо користувач уводить літеру **у**.  
**answer = input('Сьогодні твій день народження? (y/n)')**  
**if answer == 'y':**  
**print('Вітаю!')**

Приклад виконання команди наведено на рис. 23.2.

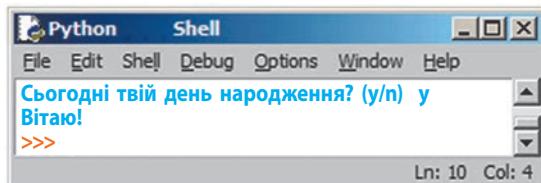


Рис. 23.2

### Умовний оператор if ... else

Алгоритмічній конструкції «повне розгалуження» відповідає умовний оператор if ... else (рис. 23.3).

**Синтаксис умовного оператора в повній формі:**

```

if <Умова> :
    <Оператор 1>
else:
    <Оператор 2>
  
```

Якщо умова *істинна* (True), програма виконає блок дій <Оператор 1>. Якщо умова *хібна* (False), виконується блок дій <Оператор 2>, який міститься після службового слова else.

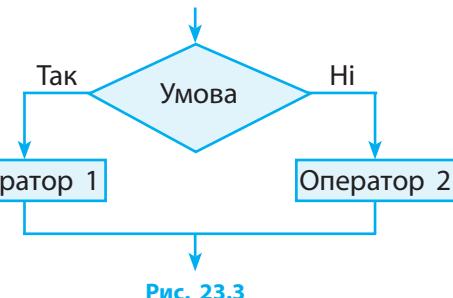


Рис. 23.3

6 Після виконання цієї програми у вікні консолі отримаємо результат Неправильно.

```

a = 13
if a == 12:
    print('Правильно')
else: # Інакше
    print('Неправильно')
  
```

7 Якщо користувач уводить літеру y, виводиться вітання Вітаю!, інакше текст Щастя тобі!.

```

answer = input('Сьогодні твій день народження? (y/n)')
if answer == 'y':
    print('Вітаю!')
else:
    print('Щастя тобі!')
  
```

- 8 Оператор if реалізує таку дію: якщо  $a > b$ , то змінні міняються значеннями так, щоб виявилося  $a < b$ .

```
a = 5
b = 2
if a > b :
    c = a
    a = b
    b = c
```

```
print ('a і b помінялися значеннями')
else :
    print ('обміну значеннями не потрібно')
print ('a = ', a, 'b = ', b)
```

Результат виконання програми наведено на рис. 23.4.

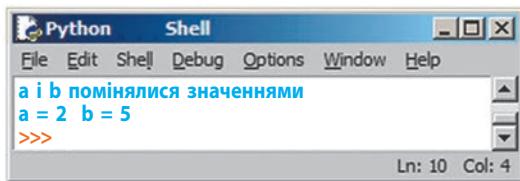


Рис. 23.4

Зверніть увагу на відступи від лівого краю в прикладі 8: команди, вкладені в гілки оператора if, записані на одній вертикалі.

Отже, оператор if є важливим інструментом програмування, який стане вам у пригоді під час створення тестових та ігрових програм.

### Питання для самоперевірки



- Як записується й виконується умовний оператор у неповній формі?
- Як записується й виконується умовний оператор у повній формі?
- Як виконуються логічні операції and, or, not?
- Яких значень набудуть змінні  $a$  і  $b$  після виконання умовних операторів, наведених нижче, для початкових значень  $a = 3$ ;  $b = 5$ ?

- |                 |                    |                       |                 |                 |
|-----------------|--------------------|-----------------------|-----------------|-----------------|
| 1) if $a > b$ : | 2) if $a \neq b$ : | 3) if $a \% 3 == 0$ : | 4) if $a > b$ : | 5) if $a < b$ : |
| $a = 0$         | $a = b$            | $a = a // 3$          | $a = a + 10$    | $a = 2 * a$     |
| else :          |                    |                       | else :          | else :          |
| $b = 0$         |                    |                       | $b = b + 10$    | $b = b * a$     |

5. Запишіть мовою програмування прості умови:
  - a)  $x$  більше 10;
  - б)  $x$  не більше  $y$ ;
  - в)  $x$  — парне число.
6. Запишіть мовою програмування складені умови:
  - а)  $2 < x < 10$ ;
  - б)  $x$  не належить проміжку (2, 10).

### Вправа 23



► Скласти програму, яка визначає, чи достатньо користувачеві років, щоб керувати автомобілем.

**Задача.** Потрібно ввести вік користувача та зберегти значення в змінній age; якщо  $age \geq 18$ , то вивести повідомлення «Ваш вік дозволяє керувати авто», інакше: змінній r присвоїти значення виразу  $(18 - age)$ , вивести повідомлення, че-рез скільки років користувач зможе отримати посвідчення.

- 1) Відкрийте вікно IDLE і створіть нове вікно програми.
- 2) Уведіть текст програми, наприклад такий:  

```
age = int(input('Скільки вам років?'))
if age>=18 :
    print ('Ваш вік дозволяє керувати авто')
else:
    r = 18-age
    print ('Доведеться почекати', r, 'років')
```
- 3) Збережіть файл у власній папці з іменем Вправа23.
- 4) Запустіть програму на виконання, проаналізуйте результат виконання програми у вікні консолі. Випробуйте програму для різних значень змінної age.
- 5)\* Змініть програмний код так, щоб у випадку  $age \geq 18$  виводилося повідомлення, скільки років людина вже могла б керувати автомобілем.
- 6) Закінчіть роботу, закривши вікно IDLE.

### Комп'ютерне тестування



Виконайте тестове завдання 23 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 24. Вкладені розгалуження

Під час розв'язування багатьох задач виникає необхідність у разі виконання (або невиконання) деякої умови знову робити вибір в залежності від іншої умови. У такій ситуації застосовують так звані вкладені розгалуження: в умовному операторі по гілці Так або Ні знову використовують умовний оператор if.

### Вкладені розгалуження

- Розглянемо задачу: знайти найбільше значення серед трьох чисел  $a$ ,  $b$ ,  $c$  (результат присвоїти змінній max).

Алгоритмічна структура розгалуження має тільки дві гілки, а в нас три варіанти (найбільшим може бути будь-яке із чисел  $a$ ,  $b$ ,  $c$ ), тому використаємо вкладені розгалуження.

Поміркуймо: якщо  $a > b$ , то найбільшим є більше із чисел  $a$  і  $c$ ; інакше найбільшим є більше із чисел  $b$  і  $c$ .

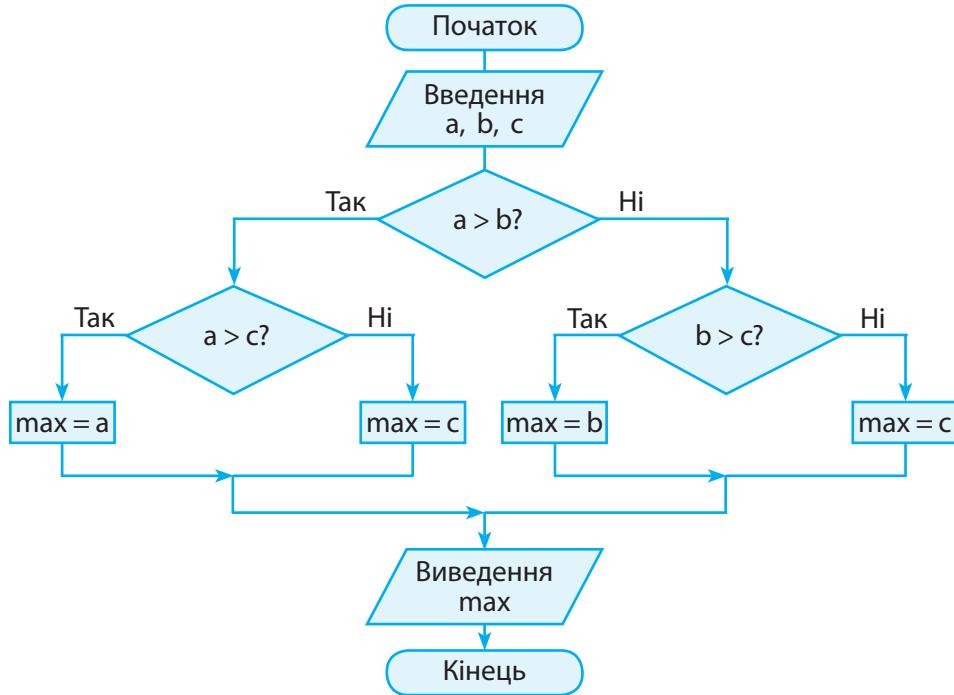


Рис. 24.1

Отже, після вибору більшого із чисел  $a$  і  $b$  знову необхідно вибрати більше із чисел  $a$  і  $c$ , а по гілці `False` — вибрати більше із чисел  $b$  і  $c$ .

Блок-схему алгоритму пошуку найбільшого серед трьох чисел наведено на рис. 24.1.

Запишемо програмний код цього алгоритму:

```
a=int(input('a=? '))
b=int(input('b=? '))
c=int(input('c=? '))
if a>b:
    if a>c:
        max=a
    else:
        max=c
else:
    if b>c:
        max=b
    else:
        max=c
print ('max= ', max)
```

Якщо  $a = 5$ ,  $b = 2$ , вираз  $a > b$  набуває значення `True`. По гілці `True` перевіряється умова  $a > c$ , яка в разі  $c = 9$  набуває значення `False`, тому змінна `max` набуває значення 9. Результат виконання програми наведено на рис. 24.2.

```
File Edit Shell Debug
==== RESTART: C:/h
a=? 5
b=? 2
c=? 9
max= 9
```

Рис. 24.2

## Відступи

У мові Python велике значення мають відступи команди від лівого краю вікна програми. Команди, вкладені в гілки умовного оператора, об'єднуються в блоки за величиною відступів.

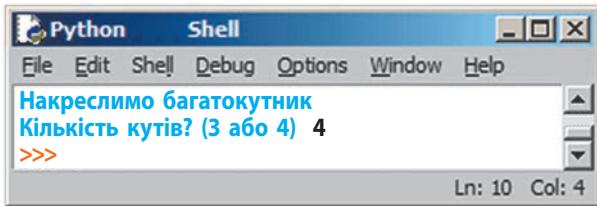
Відступ може бути будь-яким, головне, щоб у межах одного вкладеного блока відступ був однаковим.

- 2** Намалювати трикутник або квадрат за заданою кількістю кутів. Програма запитує кількість кутів у багатокутнику. Якщо кількість кутів не дорівнює 3 або 4, видається повідомлення

Помилка в кількості кутів, якщо кількість кутів дорівнює 3, то малюється трикутник, якщо 4 — малюється квадрат.

```
from turtle import *
print('Накреслімо багатокутник')
a_str = input('Кількість кутів? (3 або 4) ')
a = int(a_str)
if not (a == 3 or a == 4):
    print('Помилка в кількості кутів')
else:
    if a == 3:
        forward(100)
        right(120)
        forward(100)
        right(120)
        forward(100)
    else:
        forward(100)
        right(90)
        forward(100)
        right(90)
        forward(100)
        right(90)
        forward(100)
```

У вікні консолі:



У вікні Python Turtle Grafies:

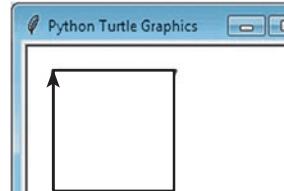


Рис. 24.3

Результат виконання програми наведено на рис. 24.3.

Отже, вкладені умовні оператори використовуються в тих випадках, коли необхідно перевірити послідовно дві або більше умови. Під час запису програмного коду уважно слідкуйте за відступами, щоб команди, вкладені в гілки умовного оператора if, були записані на одній вертикалі. По-перше, цього потребує синтаксис мови Python, по-друге, це робить текст програми наочнішим і зрозумілішим.

### Питання для самоперевірки



- Поясніть порядок виконання умовного оператора, у якому застосовані вкладені умовні оператори.
- Яке значення мають відступи команди від лівого краю вікна програми?

- 3.** З'ясуйте, яких значень набудуть змінні *a* і *b* після виконання умовного оператора if для початкових значень:

Фрагмент програмного коду	Початкові значення <i>a</i> , <i>b</i>	<i>a</i>	<i>b</i>
if <i>a</i> <3: if <i>a</i> > <i>b</i> : <i>a</i> = <i>b</i> else: <i>b</i> = <i>a</i>	<i>a</i> = 1, <i>b</i> = 5		
else: if <i>a</i> > <i>b</i> : <i>b</i> = <i>a</i> else: <i>a</i> = <i>b</i>	<i>a</i> = 3, <i>b</i> = 5		
	<i>a</i> = 8, <i>b</i> = 5		
	<i>a</i> = 1, <i>b</i> = 2		

- 4.** Напишіть програмний код, який визначає найменше з трьох чисел *a*, *b*, *c*.
- 5.** Напишіть програму, яка для двох чисел *a* і *b* виводить відповідь «*a*>*b*», «*b*>*a*» або «*a*=*b*».
- 6\*.** Напишіть програму, яка запитує вік школяра і визначає, у якій ланці він навчається: початкова школа (1–4 класи), основна (5–9 класи) або старша (10–11 класи).

#### Вправа 24



- Написати програму для визначення переможця з бігу на 100 м за результатами забігу трьох спортсменів, які посіли призові місця: Івана, Василя й Петра.
- 1) Відкрийте вікно IDLE і створіть нове вікно програми.
  - 2) Напишіть оператор для введення значення часу першого спортсмена:
- ```
t1 = float(input('Уведіть результат Івана'))
```
- 3) Напишіть оператори для введення значення часу *t2* і *t3* другого і третього спортсменів.
  - 4) Напишіть умовний оператор, який визначає, яке зі значень *t1*, *t2* і *t3* є найменшим:
- ```
if t1<t2 :  
    if t1<t3:  
        print ('Переміг Іван')
```

```

else:
    print ('Переміг Петро')
else:
    if t2<t3:
        print ('Переміг Василь')
    else:
        print ('Переміг Петро')

```

- 5)** Збережіть файл у власну папку з іменем Вправа24. Пере-вірте роботу програми для різних значень:
- |      |      |      |                |
|------|------|------|----------------|
| 11   | 11.2 | 11.1 | Переміг Іван   |
| 11.6 | 11.3 | 11.5 | Переміг Василь |
| 11.4 | 11.7 | 11.2 | Переміг Петро  |
- 6)** Закінчіть роботу, закривши вікно IDLE.



### Комп'ютерне тестування

Виконайте тестове завдання 24 з автоматичною перевіркою результату на сайті  
[interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## Практична робота 6

### Складання та виконання алгоритмів із розгалуженнями

**Завдання:** скласти програму для обчислення коренів рівняння  $ax + b = c$ .

**Обладнання:** комп'ютер зі встановленим середовищем програмування Python.

#### Хід роботи

*Під час роботи з комп'ютером дотримуйтесь правил безпеки.*

- **1.** Проаналізуйте словесний алгоритм розв'язування рівняння:  
якщо  $a = 0$ ,  $b = c$ , то коренем рівняння є будь-яке число;  
якщо  $a = 0$ ,  $b \neq c$ , то коренів немає;  
якщо  $a \neq 0$ , то  $x = (c - b) / a$ .
- **2.** Відкрийте вікно IDLE і створіть нове вікно програми.

- 3. Складіть програму на основі блок-схеми (рис. 24.4).

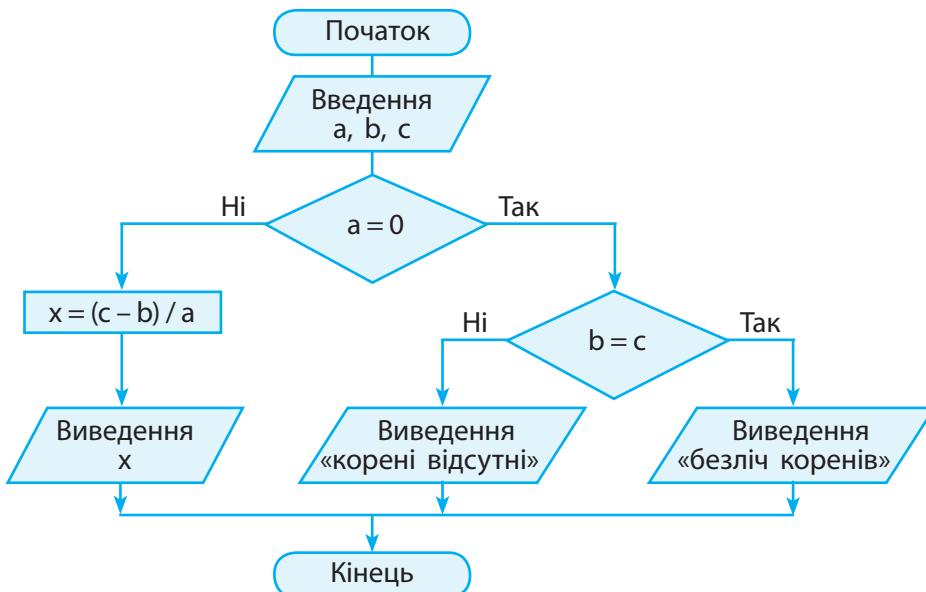


Рис. 24.4

```

# Програма розв'язування рівняння
a = int(input ('a=?'))
b = int(input ('b=?'))
c = int(input ('c=?'))
if a == 0:
    if b == c:
        print ('безліч коренів')
    else :
        print ('корені відсутні')
else :
    x = (c-b)/a
    print ('x = ', x)
    
```

- 4. Збережіть файл у власній папці з іменем LinRiv.py. Запустіть програму на виконання. Перевірте роботу програми для тестового набору значень.

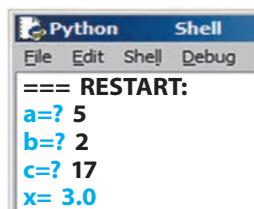


Рис. 24.5

- 5. Випробуйте програму для таких наборів коефіцієнтів:
  - а)  $a = 5; b = 3; c = 3;$
  - б)  $a = 0; b = 2; c = 17;$
  - в)  $a = 0; b = 4; c = 4.$
- 6. Закінчіть роботу, закривши вікно IDLE.

**Зробіть висновок:** як складати та виконувати алгоритми з розгалуженням для розв'язування задач?

## § 25. Алгоритми з повтореннями. Цикл із параметром



**Повторення (цикл)** — це алгоритмічна структура, за допомогою якої та сама послідовність дій виконується кілька разів.

Для запису алгоритмів із повторенням (циклів) мовою Python використовують два види операторів циклу: із параметром та з умовою. Серію команд, що повторюється під час виконання циклу, називають **тілом циклу**. Кожне виконання тіла циклу називають **ітерацією**.

### Повторення команд

Цикл `for` (цикл із параметром) повторює блок команд (тіло циклу) задану кількість разів, позбавляючи необхідності кількаразово писати одні й ті самі команди.

**Синтаксис оператора циклу `for`:**

```
for x in range(n) :
    <тіло циклу>
```

Насамперед ми вказуємо ключове слово `for`, потім — змінну  $x$ , яка буде лічильником циклу.

Ключове слово `in` наказує почергово надати змінній  $x$  усі цілі значення в діапазоні від 0 до  $n - 1$ . Не забувайте, що лічильник починає рахувати з 0, а не з 1.

Вбудована функція `range` повертає зростаючу послідовність цілих чисел, які можна використовувати як індекси всередині циклу.

Щоб дати зрозуміти комп'ютеру, які команди слід повторити, використовуються відступи. Можна зробити відступ дляожної повторюваної в тілі циклу команди, натискаючи для цього клавішу `Tab`.

- 1 Проаналізуйте зміну значень лічильника  $x$  у циклі `for`. Функція `range(4)` задає список значень [0, 1, 2, 3]:  

```
for x in range(4):
    print (x)
```

- 2 У прикладі 1 (с. 123) для малювання трикутника ми тричі, дляожної сторони трикутника, давали Черепашці команди `forward(100)` і `right(120)`. Замість цього можна повторити ці команди в циклі:

```
from turtle import *
for x in range(3):
    forward(100)
    right(120)
```

Результат виконання циклу наведено на рис. 25.1.

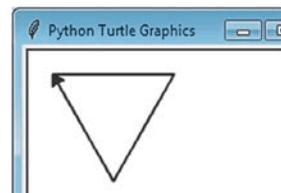


Рис. 25.1

## Лічильник циклу `for`

Лічильник циклу підраховує, скільки разів повторився цикл, почергово набуваючи всіх значень з указаного діапазону.

- 3 Діапазон значень може бути заданий у вигляді рядка.

Лічильник почергово набуває значення літер, з яких складається зазначений рядок (рис. 25.2).

```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
>>> a = 'Рядок'
>>> for x in a:
    print ('->', x)

-> Р
-> я
-> д
-> о
-> к
>>>
Ln: 13 Col: 4
```

Рис. 25.2

**Списки** — це набір значень, розділених комами, узятий у квадратні дужки. У списках можна зберігати значення будь-якого типу — числа або рядки.

- 4 Діапазон значень може бути заданий у вигляді списку. Лічильник почергово набуває заданих значень зі списку (рис. 25.3).

```

Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
>>> b = [2, 4, 6, 8, 10]
>>> for x in b :
    print ('->', x)
-> 2
-> 4
-> 6
-> 8
-> 10
>>>
Ln: 13 Col: 4
  
```

Рис. 25.3

Якщо список значень лічильника достатньо великий і числа в списку змінюються з певним кроком, можна застосувати функцію `range` з такими параметрами:

`range (від, до, крок)`

Відлік починається зі значення `від` і завершується на 1 раніше за значення `до`.

- 5 Функція `range (2, 11, 3)` надає змінній `x` значення від 2 до  $(11 - 1)$  із кроком 3, тому, хоча  $8 + 3 = 11$ , значення 11 уже до діапазону значень не входить (рис. 25.4).

```

Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
>>> for x in range(2, 11, 3) :
    print ('->', x)
-> 2
-> 5
-> 8
>>>
Ln: 30 Col: 4
  
```

Рис. 25.4

- 6 Якщо потрібно вести зворотний відлік (у бік зменшення значень), укажіть значення `від` більшим, ніж значення `до`, а крок має бути від'ємним (рис. 25.5).

```

Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
>>> for x in range(5, 1, -1) :
    print ('->', x)
-> 5
-> 4
-> 3
-> 2
>>>
Ln: 38 Col: 4
  
```

Рис. 25.5

## Застосування алгоритму повторення

Проаналізуємо виконання програми:

```
from turtle import *
for x in range(1,100,2):
    forward(x)
    left(90)
```

Під час кожного проходу циклу Черепашка малює лінію довжиною  $x$  пікселів і повертається вліво на  $90^\circ$  (рис. 25.6, а). Як ви знаєте, повне обертання навколо точки — це  $360^\circ$ . Отже, за 4 ітерації циклу Черепашка повертається до початкового напрямку  $\rightarrow$  ( $4 \cdot 90 = 360$ ). Програма малює спіраль, оскільки кожен відрізок на 2 пікселі довший за попередній: перший відрізок має довжину 1 піксель, другий — 3, останній (п'ятдесятій) — 99.

Поворот на  $90^\circ$  створює квадратну спіраль. Змінемо кут повороту на  $2^\circ$ : `left(92)`. Зміна лише одного значення викликає зміни у вигляді спіралі (рис. 25.6, б).

Змінимо команду `forward(x)` на `circle(x)`, яка малює коло радіусу  $x$  із поточної позиції:

```
for x in range(1,100,2):
    circle(x)
    left(90)
```

Отримаємо 4 набори кіл, оскільки після малювання кожного кола Черепашка повертається вліво на  $90^\circ$  (рис. 25.6, в). Щоб намалювати  $n$  кіл, потрібно повернати Черепашку на  $360/n$  градусів. Унесемо такі зміни до програми, щоб вона малювала 6 наборів кіл ( $360/6 = 60$ ), (рис. 25.6, г):

```
for x in range(1,100,2):
    circle(x)
    left(60)
```

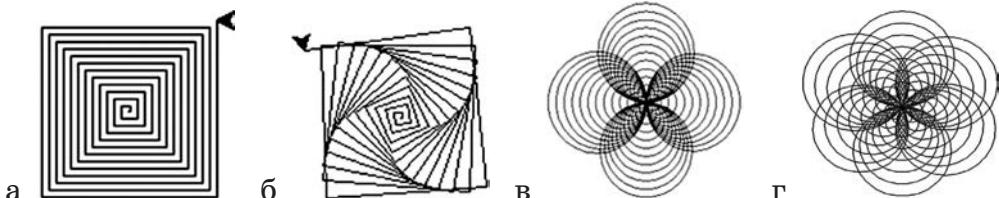


Рис. 25.6

Як зробити спіралі різнокольоровими?

Для цього нам потрібен список кольорів, а не один колір. Створимо список із чотирьох кольорів з іменем colors і помістимо у змінну з іменем colors:

colors = ['red', 'yellow', 'green', 'blue']

! Зверніть увагу, що звернутися до значення зі списку з номером  $x$  можна як до colors[x].

Наприклад, colors[0] = 'red', colors[3] = 'blue'.

Складне питання: як змусити Черепашку під час кожної ітерації циклу, коли  $x$  змінюється від 1 до 99, обирати одне з 4 значень списку?

Використаємо операцію % (остача від ділення). Під час обчислення виразу  $x \% 4$  ми можемо отримати 4 значення остачі (0, 1, 2, 3), що відповідає номерам кольорів у списку colors.

Повернемося до програми малювання квадратної спіралі, але крок циклу зробимо дорівнюючим 1 (рис. 25.7):

```
from turtle import *
colors = ['red','yellow','green','blue']
for x in range(100) :
    color(colors[x% 4])
    forward(x)
    left(90)
```

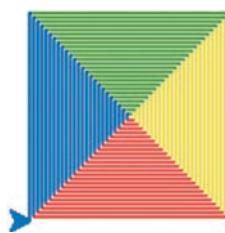


Рис. 25.7

Отже, уносячи нескладні зміни до операторів малювання в тілі циклу for, ми можемо отримувати складні цікаві зображення.

### Питання для самоперевірки



1. Поясніть правила виконання циклу for.
2. Яких значень набуває змінна  $x$  у ході виконання циклу: for x in range(5)?
3. Яких значень набуває змінна  $x$  у ході виконання циклу: for x in range(0, 10, 2)?

4. Що буде надруковано в ході виконання циклу:

a)  $s = \text{'цикл'}$   
for x in s:  
    print (x)

б)  $b = [1,3,5,7,9]$   
for x in b:  
    print (x)

5. Чому дорівнює  $s$  після виконання циклу:

a)  $s = 0$   
for a in range(5,  
    7, 1)  
    s = s + 1

б)  $s = 0$   
for a in range(5,  
    7, 1)  
    s = s + a

в)  $s = 0$   
for a in range(10,  
    5, 1)  
    s = s + 1

### Вправа 25



► Створити рисунки з використанням циклу for.

1) Відкрийте вікно IDLE і створіть нове вікно програми. Запишіть програму для побудови поданого малюнка (рис. 25.8).

```
from turtle import *
color('blue')
for i in range(1,100,2):
    up()
    goto(i*2,0)
    down()
    circle(i)
```



Рис. 25.8

2) Збережіть файл у власній папці з іменем Вправа25. Запустіть програму на виконання.

3) Поекспериментуйте з наведеним програмним кодом і дайте відповіді на такі запитання.

- Скільки кіл побудовано?
- Як зміниться малюнок, якщо змінити параметри функції range таким чином: range (1,100,6)?
- Як зміниться малюнок, якщо на початку рядка goto(i\*2,0) поставити символ #?
- Як змінити команду goto(i\*2,0), щоб малюнок мав такий вигляд, як на рис. 25.9?

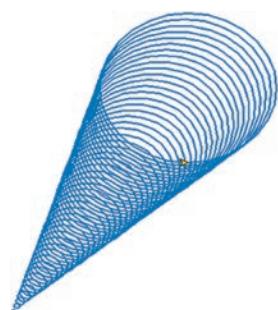


Рис. 25.9

- 4) Змініть програмний код так, щоб були намальовані 100 кіл, малювання яких починається з точок із випадковими координатами, а радіуси є випадковими числами в діапазоні від 5 до 50 в діапазоні від -200 до 200 по обох осіях. Кольори кіл почергово вибираються зі списку colors.

```
from turtle import *
from random import *
colors=['red', 'blue', 'yellow', 'green', 'violet', 'orange', 'purple']
width(3)
for i in range(1,100):
    color(colors[i % 7])
    up()
    goto(randint(-200,200),randint(-200,200))
    down()
    circle(randint(5,50))
```

- 5) Збережіть файл і запустіть програму. Отримуєте приблизно такий малюнок, як на рис. 25.10.



Рис. 25.10

- 6)\* Додайте до програми команди, потрібні для того, щоб кола були зафарбовані. Збережіть програму. Закінчіть роботу, закривши вікно IDLE.

#### Комп'ютерне тестування



Виконайте тестове завдання 25 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## § 26. Алгоритми з повтореннями. Цикл із передумовою

Цикл `for` зручно використовувати, якщо відома кількість повторень. Але часто цикл необхідно повторювати за певної умови.

### Як працює цикл `while`

Цикл `while` (поки) буде повторюватися, поки виконується задана умова. Ця умова називається умовою циклу. Результатом перевірки умови може бути `True` або `False`.

**Синтаксис оператора циклу з передумовою:**

`while <Умова>:`

`<Оператор>`

Тут `<Умова>` — логічний вираз, що є умовою; `<Оператор>` — тіло циклу, яке виконується під часожної ітерації. Виконання оператора циклу `while` починається з обчислення значення логічного виразу — умови циклу (рис. 26.1).

Якщо умова істинна, то виконуються оператори тіла циклу й керування повертається на перевірку умови. Якщо ж умова хибна, то виконується оператор, який є наступним після оператора `while`. Якщо під час першої перевірки умова виявиться хибною, тіло циклу не виконається жодного разу.

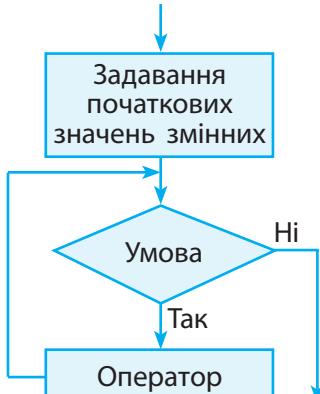


Рис. 26.1

- Для початкового значення  $x = 7$  цикл `while x < 10` виконується тричі (рис. 26.2, а). Результат виконання циклу наведено на рис. 26.2, б.

```

File Edit Shell Debug 
x=7
while x<10:
    x=x+1
    print (x)
  
```

а

```

File Edit Shell Debug 
==== RESTART:
8
9
10
  
```

б

Рис. 26.2

- 2 У циклі продовжується побудова відрізків і поворот Черепашки на  $170^\circ$ , поки Черепашка не повернеться в точку  $(0,0)$  ( $\text{abs}(\text{pos}()) < 1$ ). На рис. 26.3 зображене побудову відрізків за допомогою циклу while () .

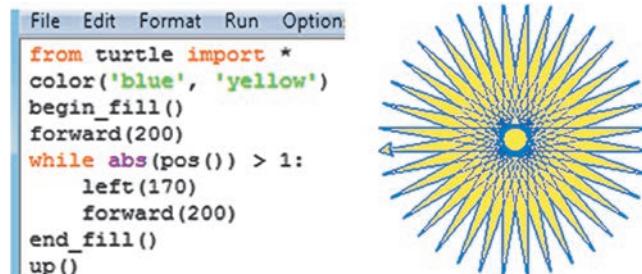


Рис. 26.3

## Нескінчений цикл

- 3 Наведений фрагмент програми ілюструє нескінчений цикл:
- ```
num = 0
while num < 20:
    print (num)
```
- У тілі циклу значення num не змінюється, тому умова  $\text{num}<20$  завжди істинна й поданий цикл є нескінченим.
- Якщо як умову циклу while задати True, умова ніколи не стане хибною й цикл не завершиться.
- 4 Програма випробовує витримку користувача — у нескінченному циклі пропонує користувачеві розв'язати задачу: знайти суму двох чисел (рис. 26.4).

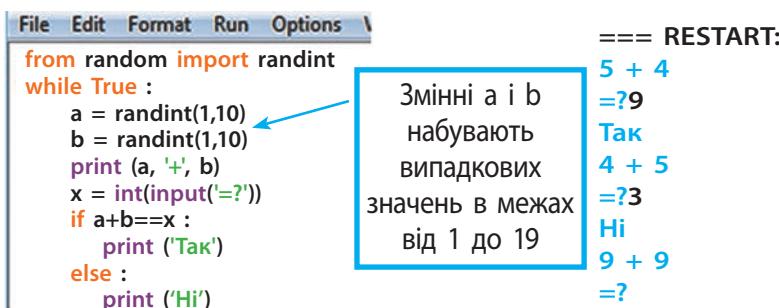


Рис. 26.4

Можна зупинити виконання нескінчленого циклу тільки примусово. Для цього у вікні консолі натисніть сполучення клавіш Ctrl+C.

### Питання для самоперевірки



1. Поясніть структуру та правила виконання циклу з умовою.
2. У чому відмінність у використанні циклу з параметром і циклу з умовою?
3. У якому випадку цикл while не виконається жодного разу?
4. У якому випадку виникає «нескінчений цикл»?

### Вправа 26



- Створити гру «Камінь-ножиці-папір».

**Задача:** камінь, ножиці, папір — популярна гра руками, відома в багатьох країнах. Часто використовується для жеребкування. Напишіть програму, яка визначає переможця у грі комп’ютера та людини. Схему визначення переможця наведено на рис. 26.6.

- 1) Відкрийте вікно IDLE і створіть нове вікно програми.

На початку програми напишіть коментарі — правила гри.

# Камінь ламає ножиці

# Ножиці ріжуть папір

# Папір накриває камінь

- 2) Завантажте функцію randint, необхідну для отримання випадкових чисел:

```
from random import randint
```

Значення для вибору зберігаємо у списку:

```
choice = ['камінь', 'ножиці', 'папір']
```

Програма запрошує до гри за допомогою оператора:



Рис. 26.6

```
print ('Давайте пограємо!')
```

Виводимо на екран підказку для гравця:

```
for i in range(3):
    print (i, choice[i])
```

**3)** Проаналізуйте алгоритм гри:

- Комп'ютер «загадує» ціле випадкове число в діапазоні від 0 до 2.
- Гравець робить свій вибір, уводячи число 0, 1 або 2.
- Виводиться повідомлення про вибір комп'ютера та вибір гравця.
- За правилами гри визначається переможець.
- Виводиться повідомлення про перемогу або нічию.

**4)** Алгоритм гри повторюється в циклі while, поки гравець не відмовиться грати далі (введе 'n'). Якщо гравець уведе 'y', то цикл повториться (рис. 26.7):

```
game='y'
while game!='n':
    comp=randint(0,2)
    x = int(input('Що обираєте (0-1-2)->'))
    your_choice = choice[x]
    comp_choice = choice[comp]
    print ('Комп\'ютер вибрал ',
          comp_choice)
    print ('Ваш вибір ', your_choice)
    # Якщо комп\'ютер вибрал КАМІНЬ
    if comp==0:
        if x==0:
            print ('НІЧИЯ')
        if x==1:
            print ('Виграв комп\'ютер')
        if x==2:
            print ('Ви виграли!')
    game = input('Бажаєте продовжити? (y/n)')
    print ('Дякую за гру!')
```

```
=====
RESTART:
Давайте пограємо!
0 камінь
1 ножиці
2 папір
Що обираєте (0-1-2)-> 0
Комп'ютер вибрав ножиці
Ваш вибір камінь
Ви виграли!
Бажаєте пограти? (y/n) y
Що обираєте (0-1-2)-> 0
Комп'ютер вибрав папір
Ваш вибір камінь
Виграв комп'ютер
Бажаєте пограти? (y/n) n
Дякую за гру!
>>> |
```

Рис. 26.7

**5)\*** У програмному коді визначається переможець тільки в тому випадку, коли комп'ютер вибрал камінь. Самостій-



но напишіть оператори розгалуження для випадків «якщо комп’ютер вибрав ножиці» і «якщо комп’ютер вибрав папір».

- 6) Запустіть програму на виконання, проаналізуйте роботу програми.
- 7) Закінчіть роботу, закривши вікно IDLE.



### Комп’ютерне тестування

Виконайте тестове завдання 26 з автоматичною перевіркою результату на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)



## Практична робота 7

### Складання та виконання алгоритмів із повтореннями



**Завдання:** скласти ігрову програму «Вгадай-но!» за поданим алгоритмом.

**Обладнання:** комп’ютер зі встановленим середовищем програмування Python.

- Алгоритм гри:**
- 1) комп’ютер генерує («задумує») випадкове число від 1 до 100;
  - 2) користувач робить спробу відгадати число;
  - 3) комп’ютер порівнює введене число з «задуманим»;
  - 4) комп’ютер надає відповідь: «Мое число більше» або «Мое число менше».

Якщо користувач відгадав число, виводиться повідомлення: «Вітаю! Число вгадане!». Інакше — надається наступна спроба.

Хід роботи

Під час роботи з комп’ютером дотримуйтесь правил безпеки.

- 1. Проаналізуйте блок-схему алгоритму (рис. 1): спробуйте пройти по гілках алгоритму з уявним числом, «задуманим» ніби комп’ютером (наприклад, 26). Нібто не знаючи цього, «вводьте» числа користувача, поки не одержите відповідь «Число вгадане».

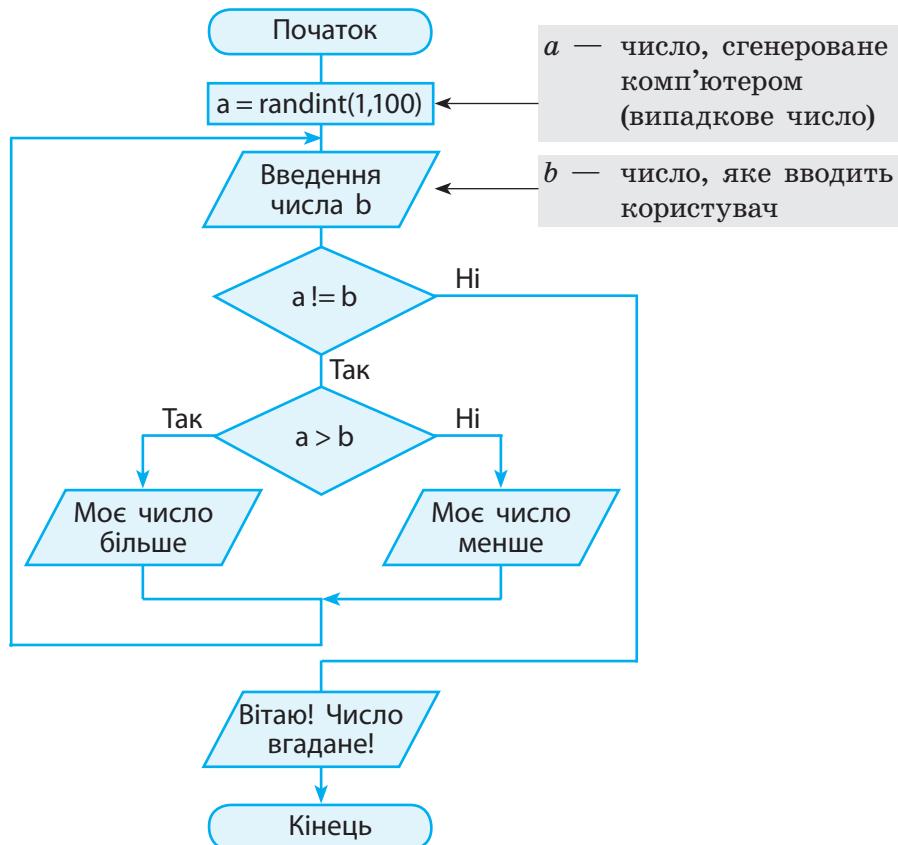


Рис. 1

- 2. Відкрийте вікно IDLE і створіть нове вікно програми.  
 ► 3. Почніть писати програмний код. Для того щоб при кожному запуску програми комп’ютер «загадував» випадкове

число, необхідно за допомогою команди `import` завантажити функцію `randint`.

Комп’ютер «задумує» ціле випадкове число в діапазоні від 1 до 100 і зберігає його в змінній `a`.

```
from random import randint  
a = randint(1,100)
```

Користувач робить спробу вгадати число:

```
b = int(input('Вгадай число :'))
```

- 4. Перевіряємо в циклі `while`, чи вгадане число. Цикл працює, поки  $a \neq b$ . Якщо число користувача менше за те, що задумав комп’ютер, видається підказка «Мое число більше», інакше — підказка «Мое число менше».

Після цього користувач знову робить спробу вгадати число.

```
while a != b:
```

```
    if a > b:
```

```
        print ('Мое число більше')
```

```
    else:
```

```
        print ('Мое число менше')
```

```
    b = int(input('Вгадай число: '))
```

- 5. Якщо при черговій перевірці умова  $a \neq b$  не виконується, це означає, що користувач угадав число на попередньому кроці циклу, і цикл припиняє роботу. Виводиться вітання з тим, що число вгадане (рис. 2).

```
print ('Вітаю! Число вгадане!')
```

- 6. Збережіть файл у власній папці з іменем `game.py`. Запустіть програму на виконання. Намагайтеся відгадати число за мінімальну кількість спроб. Закінчіть роботу, закривши вікно IDLE.

```
===== RESTART: C:/Users/.../Desktop/game.py =====
Вгадай число :50
Мое число більше
Вгадай число :75
Мое число менше
Вгадай число :62
Мое число менше
Вгадай число :56
Мое число менше
Вгадай число :53
Мое число більше
Вгадай число :55
Вітаю! Число вгадане!
```

Рис. 2

**Зробіть висновок:** як складати та виконувати циклічні алгоритми для розв’язування задач?

## Комп'ютерний словник

**Алгоритм** — скінчenna послідовність команд для виконавця, яка чітко визначає, які дії та у якому порядку потрібно виконати для розв'язування певної задачі.

**Алгоритм із повторенням** — алгоритм, у якому ті чи інші команди виконуються залежно від заданої умови.

**Алгоритм із розгалуженням** — алгоритм, у якому передбачено повторення деяких команд.

**Алфавіт мови програмування** — набір символів, який може використовуватись у програмному коді.

**Антивірус** — спеціалізована програма для знаходження комп'ютерних вірусів та відновлення заражених файлів, а також для запобігання зараженню комп'ютера.

**Блок-схема** — графічне подання алгоритму у вигляді організованої послідовності блоків.

**Браузер** — спеціальна програма для перегляду веб-сторінок.

**Величина** — окремий інформаційний об'єкт (число, символ, рядок тощо), основними характеристикими якого є назва, вид, тип і значення.

**Виконавець** — людина, тварина чи пристрій, здатні діяти за заданим алгоритмом.

**Відмова** — подія, що виникає в разі виклику команди в неприпустимому для цієї команди стані середовища.

**Вірус** — комп'ютерна програма, яка здатна до самокопіювання й убудовування в код інших програм, системні ділянки пам'яті, завантажувальні сектори і яка може завдавати шкоди: знищувати, пошкоджувати й викрадати дані.

**Гіпертекст** — текстовий документ, який містить гіперпосилання для переходу на інші об'єкти цього документа або іншого.

**Дані** — повідомлення, зафіковані на певному носії і подані у зручній для здійснення інформаційних процесів формі.

**Доменне ім'я** — текстова адреса комп'ютера або іншого ресурсу в Інтернеті.

**Змінна** — величина, значення якої може змінюватися в ході виконання програми.

**Інтернет** — глобальна комп'ютерна мережа, що складається з комп'ютерів у всьому світі.

**Інтерфейс користувача** — сукупність засобів операційної системи для взаємодії між користувачем і апаратно-програмними засобами комп'ютера.

**Інформаційна система** — сукупність взаємопов'язаних засобів і методів здійснення інформаційних процесів.

**Інформаційна технологія** — сукупність методів і процесів опрацювання інформації із затосуванням комплексу відповідних засобів.

**Інформаційний процес** — дії над інформацією.

**Інформація** — відомості про об'єкти Всесвіту.

**Клієнт (робоча станція)** — комп'ютер, який використовує ресурси мережі.

**Колонтитул** — текст, розміщений на зарезервованому місці вгорі і (або) внизу аркуша для друкування номера сторінки, назви книги, розділу тощо.

**Команда** — чіткий короткий наказ виконати певну дію.

**Коментар** — текст, призначений для читання людиною; підказка, яку програмісти записують у програмі.

**Комп'ютерно-орієнтовані засоби навчання** — апаратні та програмні засоби з елементами інтерактивності, призначені для застосування у процесі навчання для виконання розрахункових, лабораторних, дослідницьких, практичних робіт.

**Константа** — величина, значення якої не може змінюватися в ході виконання програми.

**Лінійний алгоритм** — алгоритм, у якому всі команди виконуються послідовно в порядку їх подання.

**Маркер (у тексті)** — спеціальний символ.

**Маркований список** — послідовність абзаців, на початку кожного з яких стоїть маркер.

**Методи інформаційної системи** — способи дії (взаємодії) об'єктів.

**Мова програмування** — штучна мова, що являє собою систему позначень і правил для запису алгоритмів у формі, придатній для подальшого їх виконання за допомогою комп'ютера.

**Нумерований список** — послідовність абзаців, на початку кожного з яких стоїть порядковий номер або літера.

**Операційна система** — сукупність програм, які автоматично завантажуються під час вмикання комп’ютера й використовуються для керування обчислювальними процесами, програмними й апаратними ресурсами.

**Папка** — елемент файлової системи, контейнер для зберігання файлів.

**Повідомлення** — сукупність відомостей, які зафіксовані на деякому носії для зберігання, передавання та опрацювання.

**Повторення (цикл)** — алгоритмічна структура, за допомогою якої та сама послідовність дій виконується кілька разів.

**Припустимі дії** — певний набір елементарних дій, які вміє виконувати певний виконавець.

**Пристрої введення** — пристрої, призначені для введення даних у комп’ютер.

**Пристрої виведення** — пристрої, призначені для виведення результатів опрацювання даних на комп’ютері у вигляді чисел, таблиць, текстів, графіків, малюнків, креслень, анімації, звуків.

**Провайдер** — організація, що надає послуги, пов’язані з доступом до глобальної мережі.

**Програма** — впорядкована послідовність команд для комп’ютера, виконання якої реалізує алгоритм розв’язування певної задачі.

**Програмне забезпечення** — сукупність програм, установлених на комп’ютері та призначених для забезпечення його роботи й розв’язування задач користувача.

**Програмування** — процес створення комп’ютерних програм.

**Протокол** — сукупність певних правил, за якими відбувається обмін даними між пристроями в мережі.

**Редактування документа** — виправлення помилок, унесення змін до документа.

**Робоча станція (клієнт)** — комп’ютер, який використовує ресурси.

**Сайт (веб-сайт)** — кілька пов’язаних гіперпосиланнями вебсторінок, які мають спільну тему та розміщення.

**Сервер** — комп’ютер, який надає ресурси.

**Середовище** — програма, що має засоби автоматизації підготовки та виконання програм користувача; «місце існування» виконавця.

**Синтаксис мови** — сукупність правил побудови команд мови програмування.

**Система команд виконавця** — повний перелік команд, які «розуміє» і може виконати виконавець.

**Слідування** — така форма організації вказівок в алгоритмі, за якої дії виконуються послідовно одна за одною, без пропусків або повторень.

**Службове програмне забезпечення (утиліти)** — службові програми, призначені для обслуговування дисків, створення і підтримки архівів, боротьби з вірусами тощо.

**Спам** — небажана масова розсилка листів рекламиного чи іншого характеру.

**Стиль** — іменований набір значень параметрів оформлення тексту (шрифти, вирівнювання заголовків, віdstупи, інтервали, списки тощо).

**Тіло циклу** — серія команд, які повторюються під час виконання циклу.

**Умова** — це запитання, сформульоване так, що допускає лише одну з двох відповідей: «так» або «ні».

**Файл** — розміщена на носієві іменована сукупність даних, що в процесі зберігання, передавання та опрацювання вважається єдиним цілим.

**Форматування документа** — зміна зовнішнього вигляду документа.

**Ярлик** — засіб швидкого доступу, який містить лише шлях до відповідного об'єкта.

# Алфавітний покажчик

## А

- Абзац 67  
Авторське право 51  
Адреса веб-сторінки 46  
Алгоритм 91  
— лінійний 117  
— з повтореням 139  
— з розгалуженням 129  
Антивірус 44

## Б

- Блок-схема 101  
Браузер 45

## В

- Величина 118  
Виконавець 96  
Вірус 44

## Г

- Гіпертекст 42

## Д

- Дані 13  
Доменне ім'я 42

## З

- Змінна 110

## І

- Інтернет 41  
Інформаційна система 14  
Інформаційна технологія 15  
Інформаційний процес 12  
Інформація 11

## К

- Клієнт (робоча станція) 39  
Колонтитул 57, 83  
Коментар 119  
Комп'ютер 22  
Константа 118

## М

- Міжсимвольний інтервал 69  
Мова програмування 105

## О

- Операційна система 21, 25

## П

- Папка 30  
Повідомлення  
Пошуковий сервер 50  
Програмне  
забезпечення 17, 20  
Припустимі дії 99  
Провайдер 41

## Р

- Редагування документа 58  
Розгалуження 102

## С

- Сайт (веб-сайт) 45  
Сервер 39  
Середовище 97  
Символ 62  
Слідування 102  
Спам 44  
Спам-фільтр 44  
Список 70  
Стиль 84

## У

- Умова 104  
Умовний оператор 131

## Ф

- Файл 26  
Форматування  
документа 58

## Ц

- Цикл 103

# **Зміст**

## **Розділ 1. Інформаційні процеси та системи**

|                                                                    |    |
|--------------------------------------------------------------------|----|
| § 1. Безпека під час роботи з комп'ютером.....                     | 7  |
| § 2. Інформаційні процеси та системи.....                          | 10 |
| § 3. Апаратна та програмна складові<br>інформаційної системи ..... | 17 |
| § 4. Комп'ютер як пристрій для опрацювання даних.....              | 22 |
| § 5. Операційна система та її інтерфейс .....                      | 25 |
| § 6. Операції над файлами та папками .....                         | 29 |
| Практична робота 1.                                                |    |
| Операції над файлами та папками .....                              | 35 |

## **Розділ 2. Мережеві технології та Інтернет**

|                                                    |    |
|----------------------------------------------------|----|
| § 7. Локальна і глобальна комп'ютерні мережі ..... | 39 |
| § 8. Безпечне користування Інтернетом .....        | 44 |
| § 9. Пошук інформації в Інтернеті .....            | 49 |
| Практична робота 2.                                |    |
| Пошук інформації в Інтернеті .....                 | 54 |

## **Розділ 3. Опрацювання текстових даних**

|                                                                |    |
|----------------------------------------------------------------|----|
| § 10. Основні об'єкти текстового документа .....               | 57 |
| § 11. Фрагмент тексту. Форматування символів.....              | 62 |
| § 12. Форматування абзаців .....                               | 66 |
| Практична робота 3.                                            |    |
| Створення текстового документа.....                            | 72 |
| § 13. Додавання зображень із файла<br>та їх форматування ..... | 74 |
| § 14. Додавання, редагування та форматування таблиць .....     | 78 |
| § 15. Сторінки документа та їх форматування .....              | 83 |
| Практична робота 4.                                            |    |
| Оформлення текстового документа.....                           | 87 |

|                                                              |     |
|--------------------------------------------------------------|-----|
| <b>Розділ 4. Алгоритми та програми</b>                       |     |
| § 16. Алгоритм та його властивості .....                     | 91  |
| § 17. Виконавець алгоритмів і система його команд.....       | 96  |
| § 18. Способи опису алгоритму. Алгоритмічні структури .....  | 100 |
| § 19. Середовище опису й виконання алгоритмів .....          | 106 |
| § 20. Основні поняття мови програмування Python .....        | 112 |
| § 21. Лінійні алгоритми.....                                 | 117 |
| § 22. «Черепашача» графіка .....                             | 121 |
| <b>Практична робота 5.</b>                                   |     |
| Складання та виконання лінійних алгоритмів.....              | 126 |
| § 23. Алгоритми з розгалуженнями.....                        | 127 |
| § 24. Вкладені розгалуження.....                             | 133 |
| <b>Практична робота 6.</b>                                   |     |
| Складання та виконання алгоритмів<br>із розгалуженнями ..... | 137 |
| § 25. Алгоритми з повтореннями. Цикл із параметром.....      | 139 |
| § 26. Алгоритми з повтореннями. Цикл із передумовою .....    | 146 |
| <b>Практична робота 7.</b>                                   |     |
| Складання та виконання алгоритмів<br>із повтореннями .....   | 150 |
| Комп'ютерний словник .....                                   | 153 |
| Алфавітний покажчик.....                                     | 157 |

## Відомості про користування підручником

| №<br>з/п | Прізвище та ім'я<br>учня / учениці | Навчаль-<br>ний рік | Стан підручника    |                 |
|----------|------------------------------------|---------------------|--------------------|-----------------|
|          |                                    |                     | на початку<br>року | в кінці<br>року |
| 1        |                                    |                     |                    |                 |
| 2        |                                    |                     |                    |                 |
| 3        |                                    |                     |                    |                 |
| 4        |                                    |                     |                    |                 |
| 5        |                                    |                     |                    |                 |

### Навчальне видання

*БОНДАРЕНКО Олена Олександровна  
ЛАСТОВЕЦЬКИЙ Василь Васильович  
ПИЛИПЧУК Олександр Павлович  
ШЕСТОПАЛОВ Євген Анатолійович*

### «ІНФОРМАТИКА»

підручник для 5 класу закладів загальної середньої освіти

Рекомендовано Міністерством освіти і науки України

Видано за рахунок державних коштів. Продаж заборонено

Провідний редактор І. Л. Морева. Редактори Л. А. Каюда, Ю. М. Миронова.  
Технічний редактор А. В. Пліско. Художнє оформлення В. І. Труфен.  
Комп'ютерна верстка О. В. Сміян, С. В. Яшиш. Коректор Н. В. Красна.

В оформленні підручника використано зображення,  
розміщені в мережі Інтернет для вільного використання

Підписано до друку 04.12.2018. Формат 70×90/16.

Папір офсетний. Гарнітура Шкільна. Друк офсетний.

Ум. друк. арк. 11,70. Обл.-вид. арк. 11,5. Тираж 13 113 прим. Зам. 6511-2018.

ТОВ Видавництво «Ранок»,  
вул. Кібальчича, 27, к. 135, Харків, 61071.

Свідоцтво суб'єкта видавничої справи ДК № 5215 від 22.09.2016.

Адреса редакції: вул. Космічна, 21а, Харків, 61145.

E-mail: office@ranok.com.ua. Тел. (057) 719-48-65, тел./факс (057) 719-58-67.

Надруковано у друкарні ТОВ «ТРИАДА-ПАК»,  
prov. Сімферопольський, 6, Харків, 61052.

Свідоцтво суб'єкта видавничої справи ДК № 5340 від 15.05.2017.

Тел. +38 (057) 703-12-21. E-mail: sale@triada.kharkov.ua

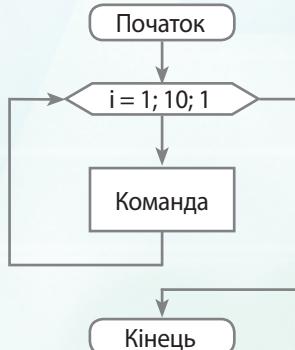
# Базові алгоритмічні структури



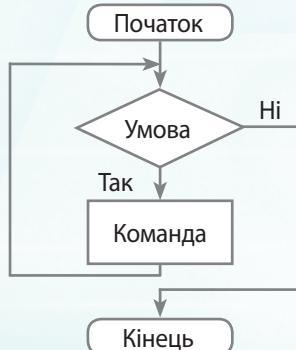
Лінійний алгоритм



Алгоритм із повторенням  
(цикл із параметром)



Алгоритм із повторенням  
(цикл із передумовою)



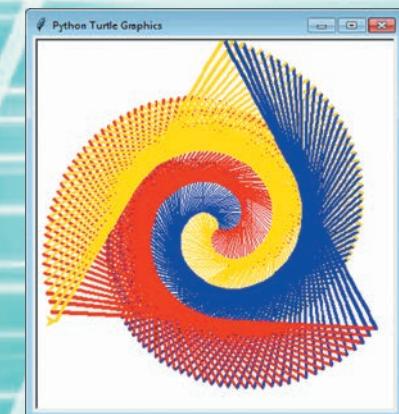
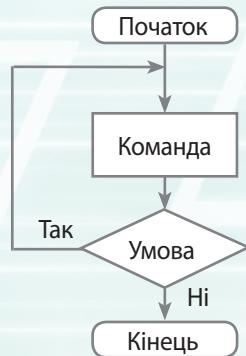
Алгоритм із розгалуженням  
(повне розгалуження)



Алгоритм із розгалуженням  
(неповне розгалуження)



Алгоритм із повторенням  
(цикл із післяумовою)



# ІНФОРМАТИКА

## 5 клас

### Особливості підручника:

- ▶ Мотивація навчальної діяльності
- ▶ Приклади практичного застосування набутих знань
- ▶ Система вправ для формування практичних навичок роботи за комп'ютером
- ▶ Питання для самоперевірки за кожною темою
- ▶ Покрокові описи практичних робіт

### Інтернет-підтримка дозволить:

- ▶ здійснити онлайн-тестування заожною темою
- ▶ ознайомитися з додатковими матеріалами до уроків

ВИДАВНИЦТВО  
**РАНОК**



ISBN 978-617-09-4781-9



9 786170 947819



Інтернет-підтримка  
[interactive.ranok.com.ua](http://interactive.ranok.com.ua)

