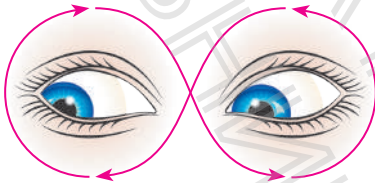


ІНФОРМАТИКА

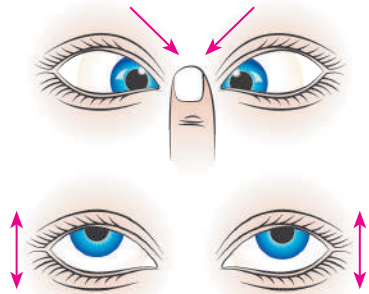


КОМПЛЕКС ВПРАВ ДЛЯ ОЧЕЙ

1. «Намалюйте» очима вісімку на протилежній стіні.

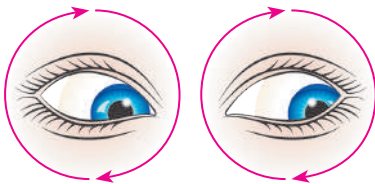


2. Подивіться на кінчик носа, а потім на брови.

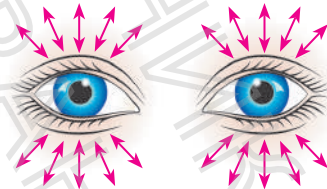


3. Сконцентруйте зір на віддаленому об'єкті (наприклад на лампочці на стелі), а потім — на близькому (наприклад на пальці перед обличчям).

4. Зробіть колові рухи очима вліво і вправо.



5. Міцно заплющте, а потім широко розплющте очі.



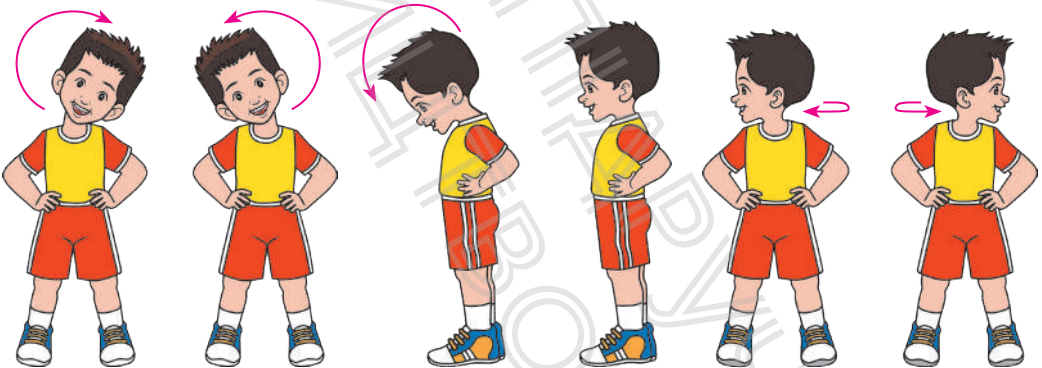
Виконуйте цей комплекс щонайменше двічі на день.

Кожну вправу повторюйте 5 разів.

ЗДОРОВ'Я!

КОМПЛЕКС ВПРАВ ДЛЯ ЗБЕРЕЖЕННЯ ЗДОРОВ'Я

1. Повільно нахиліть голову до правого плеча, потім до лівого.
2. Повільно нахиліть голову вперед, потім назад.
3. Повільно поверніть голову вправо, потім уліво.



4. Підніміть руки вгору, зчепіть пальці та потягніться.
5. Зчепивши руки в замок, зробіть руками кругові рухи.
6. Виконайте плавні нахили тулуба назад.

Виконуйте цей комплекс щонайменше двічі на день.

Кожну вправу повторюйте 5 разів.

ІНФОРМАТИКА

Підручник для 5 класу
закладів загальної середньої освіти

Рекомендовано
Міністерством освіти і науки України

Харків
Видавництво «Ранок»
2023



УДК 004:37.016(075.3)

I-74

Авторський колектив:

Олена Бондаренко, Василь Ластовецький,
Олександр Пилипчук, Євген Шестопалов

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 08.02.2022 № 140)

Підручник створено відповідно до модельної навчальної програми
«Інформатика. 5–6 класи» для закладів загальної середньої освіти
(авт. Ривкінд Й. Я., Лисенко Т. І., Чернікова Л. А., Шакоцько В. В.)

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», 2022. — 192 с. : іл.

ISBN 978-617-09-7917-9

УДК 004:37.016(075.3)



ISBN 978-617-09-7917-9

© Бондаренко О. О., Ластовецький В. В.,
Пилипчук О. П., Шестопалов Є. А., 2022
© ТОВ Видавництво «Ранок», 2022

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

Рубрика «Повторюємо» допоможе вам згадати відомості, які ви вивчали в попередніх класах і які будуть корисні для засвоєння нового матеріалу. «Питання для самоперевірки» допоможуть з'ясувати, чи зрозуміли ви вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт за комп'ютером.

«Вправи» складаються із завдань теоретичного і практичного спрямування. Під час їх виконання ви закріпите навички роботи за комп'ютером.

Оцінити свої знання та вміння вам допоможе «Комп'ютерне тестування» з автоматичною перевіркою відповідей. Тести за кожною темою курсу, а також підсумкові роботи до розділів розміщено на інтерактивній освітній платформі IZZI.

У підручнику ви знайдете покроковий опис виконання практичних робіт. Щоб виконати кожену роботу, потрібно повторити матеріал, вивчений на попередніх уроках, — тоді зможете успішно застосувати свої знання, виконати завдання за комп'ютером.

Бажаємо натхнення та успіхів!

Авторський колектив

У тексті підручника використано такі позначення:



Запам'ятайте



Розгляньте приклад



Ознайомтеся з цікавою інформацією



Зверніть увагу



Дайте відповідь на питання



Виконайте завдання за комп'ютером



Виконайте завдання підвищеної складності



Виконайте завдання в парах



Знайдіть відповідь в інтернеті



Виконайте тестові завдання за комп'ютером

РОЗДІЛ 1

ІНФОРМАЦІЙНІ ПРОЦЕСИ ТА СИСТЕМИ



- § 1. Інформація та повідомлення. Правила роботи за комп'ютером
 - § 2. Інформаційні процеси. Інформаційні технології
 - § 3. Використання комп'ютерів
 - § 4. Комп'ютер як інформаційна система. Складові комп'ютера
 - § 5. Операційна система та її інтерфейс
 - § 6. Операції над файлами та папками
- Практична робота 1. Виконання операцій над файлами та папками

ПОВТОРЮЄМО



Ви вже знаєте, що відомості про навколишній світ і процеси, які в ньому відбуваються, називають *інформацією*. Людина отримує інформацію у вигляді повідомлень, які сприймає *органами чуття*. За *способом сприйняття* повідомлення поділяють на візуальні, смакові, нюхові, аудіальні й тактильні, за *способом подання* — текстові, графічні, звукові, мультимедійні.

Дії, які можна виконувати з інформацією: пошук, опрацювання, передавання, зберігання тощо, називають *інформаційними процесами*. Інформацію опрацьовують за допомогою *комп'ютера*. Існують різноманітні комп'ютери: від кишенькових до суперкомп'ютерів.

1. Що таке інформація?
2. Як людина сприймає інформацію?
3. Які бувають повідомлення за способом їх сприйняття людиною?
4. У який спосіб можна подати інформацію?
5. Де використовують комп'ютери?
6. Наведіть приклади передавання інформації.



У цьому розділі ви більше дізнаєтеся про будову та роботу комп'ютера; навчитеся безпечно та грамотно працювати з операційною системою — комплексом програм для керування комп'ютером.

§ 1. ІНФОРМАЦІЯ ТА ПОВІДОМЛЕННЯ. ПРАВИЛА РОБОТИ ЗА КОМП'ЮТЕРОМ

Упродовж життя людина сприймає безліч повідомлень, опрацьовує, зберігає або передає отриману інформацію. Як це відбувається, вивчає особлива наука — *інформатика*.

Інформатика — наука, що вивчає властивості інформації та методи її створення й опрацювання з використанням комп'ютерів, інших технічних пристроїв.

Що таке інформація

До основних понять інформатики належать *інформація* та *повідомлення*.

Всесвіт, що нас оточує і частиною якого є ми самі, складається з об'єктів (предметів, процесів і явищ). Та поряд із матеріальною складовою Всесвіту існує й нематеріальна складова — її називають інформацією (рис. 1.1).



Рис. 1.1

Інформація — відомості про об'єкти Всесвіту.

У різних сферах людської діяльності застосовують також інші означення цього терміна. У побуті інформацією часто вважають лише ті відомості, що для людини є новими.

У Законі України «Про інформацію» сказано, що інформація — це будь-які відомості чи дані, які можна зберегти на матеріальних носіях або відобразити в електронному вигляді.

Інформацією обмінюються за допомогою повідомлень.

Повідомлення — форма подання інформації у вигляді зображень, жестів, звуків, інших символів.

Без повідомлень не можна сприйняти інформацію. Ту саму інформацію можна передати різними повідомленнями, а те саме повідомлення може нести різну інформацію.

Людина сприймає повідомлення та отримує інформацію за допомогою органів чуття: зору, слуху, нюху, смаку, дотику (рис. 1.2). Розгляньмо, як розрізняють повідомлення за способом подання та способом сприйняття (рис. 1.3).



Рис. 1.2

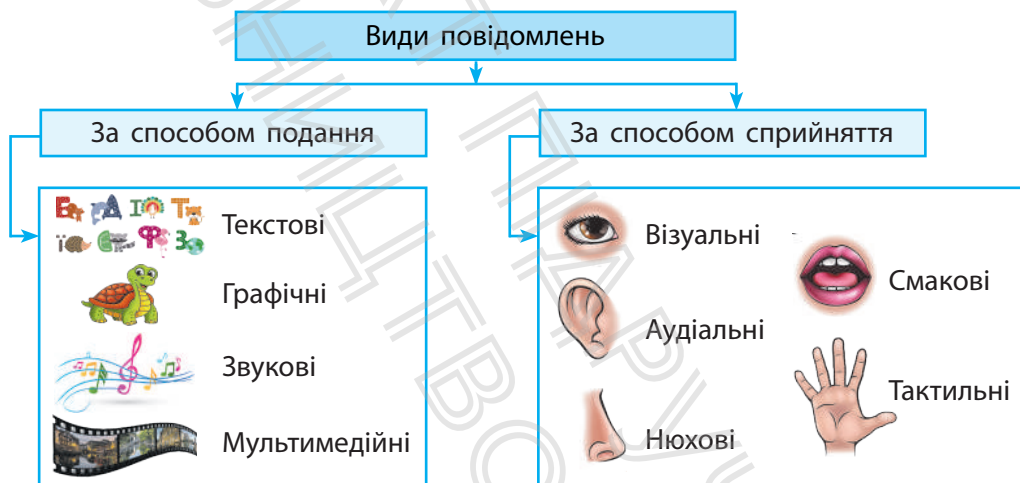


Рис. 1.3

- Людина має й інші органи чуття. Завдяки вестибулярному апарату ви навіть із заплющеними очима легко визначаєте положення свого тіла в просторі (нахили, повороти) (рис. 1.4). Подібними «органами» конструктори обладнують сучасних роботів, яких вони навчили ходити, стрибати і навіть робити сальто.



Рис. 1.4

Чи знаєте ви, що технічні пристрої отримують повідомлення за допомогою різних датчиків і сенсорів?



Світильник, що має *датчик руху* (рис. 1.5, а), допомагає економити електроенергію, бо засвічується тоді, коли перед ним з'являється рухомий об'єкт. *Датчик диму* (рис. 1.5, б), приєднаний до системи сигналізації, забезпечує приміщення від пожежі.



Рис. 1.5

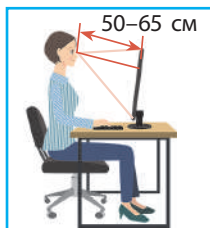
Правила поведінки в комп'ютерному класі

У комп'ютерному класі потрібно бути дуже уважними. У приміщенні з багатьма комп'ютерами джерелом небезпеки може бути, наприклад, розетка з розбитим корпусом, пошкоджений дріт тощо. У такому випадку слід повідомити вчителя / вчительку.

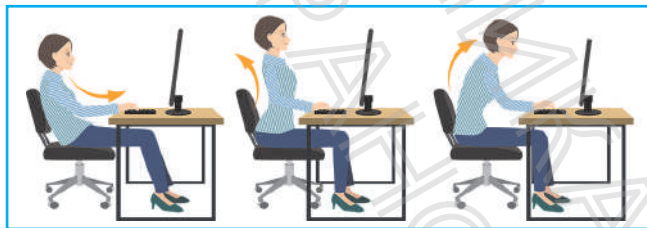
Із правилами роботи за комп'ютером ознайомтеся за QR-кодом або посиланням rnk.com.ua/105676. Під час занять необхідно робити перерви і виконувати вправи (див. форзац 1).



Час неперервної роботи становить 15–20 хв. Розгляньте рис. 1.6. Як неправильно сидіти за комп'ютером і чому?



ПРАВИЛЬНО



НЕПРАВИЛЬНО

Рис. 1.6



Питання для самоперевірки

1. Поясніть, що таке інформація; повідомлення.
2. Як розрізняють інформацію за різними ознаками?
3. Чого не можна робити в комп'ютерному класі?
4. Чому положення, наведені на рис. 1.6, є неправильними?
5. Який безпечний час роботи за комп'ютером для дітей?
6. Обговоріть правила безпеки та можливі наслідки їх недотримання з батьками чи старшими людьми.



Вправа 1

- Ознайомитися з робочим місцем, пригадати основи роботи за комп'ютером із дотриманням правил безпеки.



1. Запустіть клавіатурний тренажер і потренуйтеся набирати пропонований текст українською мовою. Повідомте результат учителю/вчительці.

2. Запустіть графічний редактор (наприклад, Paint).

3. Намалюйте комп'ютер, який стоїть на вашому робочому місці (рис. 1.7).



4. На малюнку позначте місця, що можуть бути джерелом небезпеки.



5. Намалюйте на екрані «монітора» комп'ютера пейзаж на власний розсуд.

6. Покажіть малюнок учителю/вчительці.

Збережіть файл із назвою Вправа 1. Завершіть роботу з графічним редактором.



Рис. 1.7



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102343).



§ 2. ІНФОРМАЦІЙНІ ПРОЦЕСИ. ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ

Об'єкти довкілля надзвичайно різноманітні. Кожен об'єкт має певні властивості й може виконувати певні дії (або дії можуть виконуватися над ним).

Інформаційні процеси

Із точки зору людини інформація є об'єктом, а дії над нею називають інформаційними процесами.



Інформаційні процеси — це дії над інформацією.

До основних інформаційних процесів належить *передавання* повідомлень, *пошук* потрібної інформації, *опрацювання* та *зберігання* інформації.



Коли вчитель/вчителька пояснює матеріал, то *передає* інформацію, а ви *отримуєте*. Коли ви записуєте щось у зошит — *зберігаєте* інформацію, а коли розв'язуєте задачу — *опрацьовуєте* (рис. 2.1).



Рис. 2.1



13 січня 2022 року виведено на навколосезну орбіту український супутник «Січ-2-30». Він *збирає* інформацію про нашу планету, фотографуючи її у видимих і інфрачервоних променях, і *передає* вченим, які її *опрацьовують*.

Опрацювання інформації полягає у створенні нової для людини інформації шляхом перетворення наявної.



Наприклад, дівчинка складає вірш, перебуваючи під враженням від дивовижної природи карпатської полонини (рис. 2.2).



Рис. 2.2

Окрім основних інформаційних процесів, існує багато інших: поширення інформації, її захист, накопичення тощо.

Дані

Щоб повідомлення було придатним для здійснення окремого інформаційного процесу, його необхідно подати певним чином (у вигляді звуків, зображень, слів, сигналів тощо).



Дані — повідомлення, подані у формі, зручній для опрацювання, зазвичай за допомогою технічних засобів.

Скажімо, повідомлення, подане у формі, з якою може працювати пристрій, є для нього даними. Дані зберігають, записуючи їх на носії даних (папір, карта пам'яті тощо) (рис. 2.3).



Рис. 2.3



Дані у вигляді наскельного малюнка в печері можуть зберігатися тисячоліттями. Так, в Україні поблизу Мелітополя є унікальне місце — Кам'яна Могила. На її кам'яних — понад 3 тис. зображень людей, диких і свійських тварин, різноманітних знаків, петрогліфів (рис. 2.4, а).

Текстові дані на паперових носіях — книжках, журналах, газетах тощо (рис. 2.4, б) значно легше передавати, хоча зберігання даних на них не таке тривале.



а



б

Рис. 2.4

Завдяки надзвичайно швидкому копіюванню та розповсюдженню інформації сучасними засобами її можна вважати практично незнищеною.



Тривають розробки зі збільшення довговічності носіїв даних. 2013 року вчені Інституту проблем реєстрації інформації Національної академії наук України розробили оптичний сапфіровий диск, який може зберігати дані впродовж десятків тисяч років (рис. 2.5).



Рис. 2.5

Інформаційні системи

Слово «система» походить із грецької мови і означає «ціле, що складається з частин».



Інформаційна система — сукупність взаємопов'язаних об'єктів, у якій відбуваються інформаційні процеси.

Об'єкти живої та неживої природи, які становлять будь-яку інформаційну систему, називають *засобами інформаційної системи*. Способи дії (або взаємодії) цих об'єктів називають *методами інформаційної системи*.



На рис. 2.6 зображено інформаційну систему — учнівський клас на уроці малювання. *Засоби системи:* учитель, діти, приміщення, дошка, олівці, аркуші паперу й ін. *Методами системи* є усне пояснення вчителя/вчительки, демонстрування та обговорення репродукцій картин, відтворення дітьми малюнків олівцями на папері тощо.



Рис. 2.6

Існують технічні (наприклад, робот), біологічні (людина), соціальні (група людей) та інші інформаційні системи.

Інформаційні технології навколо нас

Інформатика є теоретичною базою для розвитку інформаційних технологій. Слово «технологія» з грецької перекладається як «майстерність», «уміння».

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

Як відомо, ще перші комп'ютери повністю виправдали свою назву (від англ. *computer* — обчислювач). Адже саме для автоматичного виконання великих обсягів дуже складних обчислень їх і створювали.

Нині комп'ютери постійно вдосконалюються, розширюється сфера їх застосування. Набувають поширення нові види комп'ютерів (рис. 2.7): роботи-помічники в господарстві, роботизовані іграшки тощо. Комп'ютери дали змогу створити такі цікаві технології, як «розумний дім» (англ. *smart home*). Його домашня автоматика здатна спростити побутові проблеми людям похилого віку, людям з інвалідністю (рис. 2.8).



Рис. 2.7



Рис. 2.8

Віднедавна зростає популярність дистанційного навчання. Його основою є мережеві *інформаційно-комунікаційні технології*, які створюють можливість учнівській і вчительській спільнотам взаємодіяти на відстані.



Завдяки інтернету безпілотні автомобілі пересуваються маршрутом, реагуючи на дорожню ситуацію; камери відеоспостереження (рис. 2.9) сповіщають через мобільні пристрої про порушників; наручні пристрої (рис. 2.10) виконують функції крокоміра, тонометра для вимірювання кров'яного тиску; пульта дистанційного керування телевізором (рис. 2.11) тощо.



Рис. 2.9



Рис. 2.10



Рис. 2.11

Ми навели лише окремі приклади сучасних інформаційних технологій, розвиток яких не зупиняється ні на мить.



Питання для самоперевірки

1. Що таке дані?
2. Назвіть основні інформаційні процеси.
3. Наведіть приклади носіїв для зберігання даних.
4. Якими засобами передають інформацію?
5. Що таке інформаційна технологія?
6. Наведіть приклади інформаційних технологій у житті.



Вправа 2

■ Ознайомитися з пристроями опрацювання даних.



1. Запустіть клавіатурний тренажер і потренуйтеся набирати пропонований текст українською та іноземною мовами. Повідомте результат учителю/вчительці.
- 2. Запустіть браузер і знайдіть в інтернеті зображення комп'ютера та двох пристроїв опрацювання даних.
3. Запустіть графічний редактор (наприклад, Paint).
4. Скопіюйте і вставте знайдені зображення за зразком на рис. 2.12. Запишіть назви пристроїв.



Рис. 2.12



5. Зробіть колаж із цих зображень або їх фрагментів.
6. Покажіть роботу вчителю/вчительці. Збережіть файл із назвою Вправа 2. Завершіть роботу за комп'ютером.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102345).



§ 3. ВИКОРИСТАННЯ КОМП'ЮТЕРІВ

Як відомо, комп'ютер — універсальний пристрій для опрацювання даних. І хоч дослівно його назва означає «обчислювач», нині він став придатним для опрацювання, зберігання та передавання різних даних: числових, текстових та ін.


Комп'ютер як помічник

Комп'ютери використовують у різних галузях людської діяльності. Так, за їх допомогою можна прогнозувати погоду, керувати автомобілем (автопілот), проводити медичні дослідження (комп'ютерний томограф), навчатися дистанційно, продавати залізничні й авіаквитки, створювати відео- й анімаційні фільми, розробляти ігрові та навчальні програми.

Види комп'ютерів

Комп'ютери поділяють за різними ознаками. Якщо розглядати комп'ютери, призначені для персонального користування, то їх можна класифікувати, наприклад, так:

Стаціонарні комп'ютери	
<p>Настільний комп'ютер (десктоп)</p> 	<p>Складається із системного блоку, до якого приєднано монітор, клавіатуру й мишу</p>
<p>Моноблок</p> 	<p>Системний блок і монітор становлять єдине ціле</p>

Портативні комп'ютери	
Ноутбук та нетбук	 <p>У корпусі об'єднано дисплей, клавіатуру, тачпад і акумуляторну батарею</p>
Планшетний ноутбук та планшет	 <p>Корпус містить сенсорний екран (пристрій уведення), може бути складана або висувна клавіатура</p>
Кишеньковий ПК та смартфон	 <p>Мають малі розміри та значний час автономної роботи від акумулятора. Пристрій уведення — сенсорний екран або висувна клавіатура</p>
Мікро-комп'ютер	 <p>Має мініатюрний корпус, до якого приєднують решту пристроїв (монітор, мишу, клавіатуру тощо)</p>

- Звичайний персональний комп'ютер виконує мільярди операцій за секунду. Для виконання таких розрахунків людині з калькулятором знадобиться понад 100 років.

Розвиток комп'ютерної індустрії дав змогу створити *суперкомп'ютер* — обчислювальну машину, можливості якої значно перевершують решту. Його застосовують для наукових досліджень, передбачення стихійних явищ тощо.

- 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. 3.1).



Рис. 3.1



Питання для самоперевірки

1. Наведіть приклади використання комп'ютерів.
2. Яке призначення персонального комп'ютера?
3. Що таке моноблок?
4. Опишіть особливості смартфона.
5. Назвіть види комп'ютерів.
6. Для чого створюють суперкомп'ютери?



Вправа 3

- Розглянути основні елементи комп'ютера.
1. Запустіть клавіатурний тренажер і потренуйтеся набирати текст, який складається з букв і спеціальних символів. Повідомте результат учителю/вчительці.
 2. Запустіть графічний редактор. Запустіть браузер.
 3. Відкрийте створений раніше файл Вправа 1. Підпишіть назви складових комп'ютера на зображенні (рис. 3.2).
 4. Знайдіть в інтернеті зображення ноутбука, планшета, смартфона та додайте їх нижче на малюнок.
 5. Знайдіть відомості про квантовий комп'ютер та додайте до свого малюнка коротке текстове повідомлення.
 6. Збережіть файл із назвою Вправа 3. Завершіть роботу з браузером і графічним редактором.



Рис. 3.2



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102346).



§ 4. КОМП'ЮТЕР ЯК ІНФОРМАЦІЙНА СИСТЕМА. СКЛАДОВІ КОМП'ЮТЕРА

Інформаційна система забезпечує приймання, перетворення, опрацювання, збереження інформації та передавання результатів людині, машині або іншій інформаційній системі. Комп'ютер є технічною інформаційною системою.

Складові комп'ютера та їх призначення

Персональний комп'ютер має дві складові: **апаратну**, або апаратне забезпечення (англ. *hardware* — тверді вироби), — з'єднані між собою різноманітні пристрої, та **програмну**, або програмне забезпечення (англ. *software*), — програми, що запускаються та виконуються на комп'ютері.

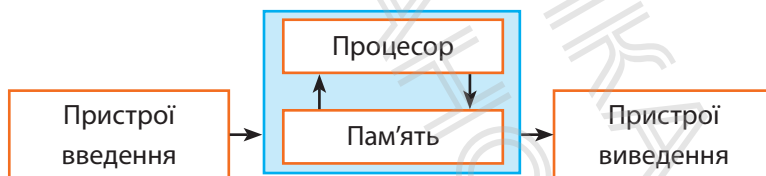
! Пристрої комп'ютера не можуть опрацьовувати дані без програм. Саме завдяки взаємодії апаратної та програмної складових працює комп'ютер.

Апаратна складова

Сучасні комп'ютери здебільшого базуються на принципах, які обґрунтував американський учений Джон фон Нейман (рис. 4.1, а) ще 1948 року. Відтоді змінився зовнішній вигляд комп'ютерів, з'явилися нові пристрої, проте загальна схема роботи лишилася без змін (рис. 4.1, б).



а



б

Рис. 4.1

Апаратну складову комп'ютера можна поділити на пристрої введення, виведення, опрацювання та зберігання.

Пристрої введення інформації (рис. 4.2) призначені для введення даних у комп'ютер. Основні пристрої введення — *сенсорний екран, клавіатура і миша.*



Рис. 4.2



Для введення текстової і графічної інформації також застосовується *електронно-оптичний сканер*, для введення звукової інформації — *мікрофон*, для керування в комп'ютерних іграх і тренажерах — *джойстик (жермо).*

Пристрої виведення інформації (рис. 4.3) призначені для виведення результатів опрацювання даних на комп'ютері у вигляді тексту, таблиць, малюнків, звуків, електричних та інших сигналів, тобто у вигляді, зручному для сприйняття людиною або керувальним пристроєм. Основний пристрій виведення — *монітор.*



Рис. 4.3



Пам'ятаймо, що пристрої введення і виведення разом називають *зовнішніми пристроями* комп'ютера.

Пристроєм опрацювання інформації є процесор (рис. 4.4). Він також керує іншими пристроями комп'ютера. Процесор по чергово виконує команди програми.

Пристрої зберігання інформації, або пам'ять (рис. 4.5), призначені для тимчасового чи постійного зберігання даних і програм.



Рис. 4.4

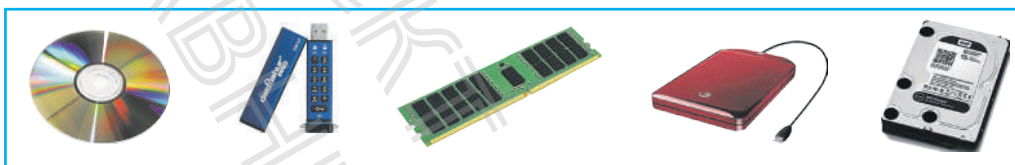


Рис. 4.5

Розрізняють внутрішні та зовнішні пристрої пам'яті.

Зовнішні пристрої пам'яті — це накопичувачі на жорстких магнітних дисках, накопичувачі на оптичних дисках, флешпам'ять тощо.

Програмна складова

Отже, програмне забезпечення — це набір програм, які керують роботою комп'ютера, забезпечують ведення діалогу з людиною, за їх допомогою опрацьовують дані, створюють нові програми. Програмне забезпечення комп'ютера поділяють на системне, прикладне та службове (рис. 4.6, с. 23).

До *системного програмного забезпечення* належать операційна система, драйвери пристроїв та інші програми, які виконують функції керування ресурсами комп'ютера.

До *прикладного програмного забезпечення* належать програми, що забезпечують виконання різноманітних робіт. Це можуть бути текстові та графічні редактори, браузері; програми, що використовують у лікарнях, поліції, банках; програми для створення інших програм тощо.

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

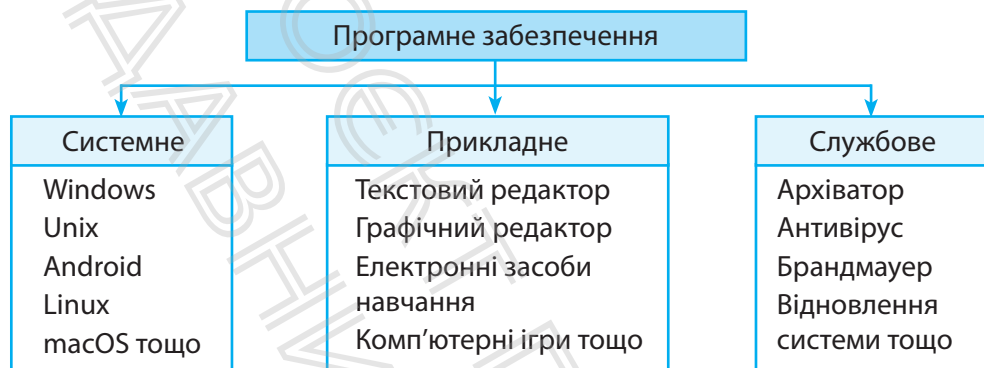


Рис. 4.6



Питання для самоперевірки

1. Що таке інформаційна система?
2. Які пристрої становлять апаратну частину комп'ютера?
3. Назвіть пристрої введення інформації.
4. Які пристрої призначені для виведення інформації?
5. Що входить до програмного забезпечення комп'ютера?
6. Яке призначення операційної системи?



Вправа 4

- Розглянути елементи апаратної складової комп'ютера.
- 👤 1. Запустіть клавіатурний тренажер і потренуйтеся набирати пропонований текст українською та іноземною мовами. Повідомте результат учителю/вчительці.
- 2. Запустіть графічний редактор. Намалуйте схему за зразком (рис. 4.7, с. 24).
- 3. У верхньому прямокутнику на схемі додайте напис Апаратна складова комп'ютера.

4. У *другому* рядку в прямокутниках допишіть назви видів пристроїв комп'ютера.
5. У *третьому* рядку й далі в прямокутниках запишіть по декілька назв пристроїв кожного виду.
6. Збережіть файл із назвою Вправа 4. Завершіть роботу з графічним редактором.

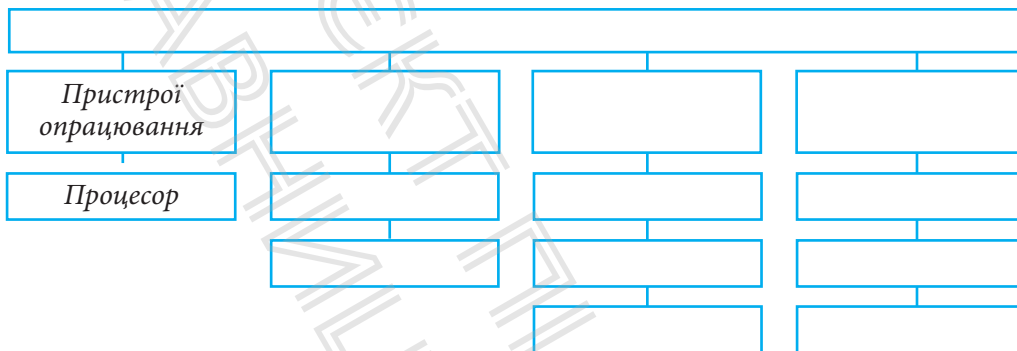


Рис. 4.7



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102347).



§ 5. ОПЕРАЦІЙНА СИСТЕМА ТА ЇЇ ІНТЕРФЕЙС

Після ввімкнення комп'ютера розпочинається завантаження та запуск *операційної системи* (ОС) — набору програм, які забезпечують взаємодію всіх пристроїв комп'ютера і виконання інших програм.

Інтерфейс користувача

Як відбувається діалог комп'ютера і людини?

Інтерфейс користувача — сукупність засобів для взаємодії між людиною та комп'ютером.

Перші ОС мали *текстовий інтерфейс*: команди вводилися з клавіатури. *Графічний інтерфейс* наочніший: команди вибираються з меню, параметри зазначаються в діалогових вікнах, об'єкти зображуються значками — піктограмами з підписом. Під об'єктами розуміють усе, чим керує ОС: папки, файли, мережу, диски, пристрої.



Значки об'єктів операційної системи часто мають піктограми із зображенням самих об'єктів: панель керування, миша, принтери і факси, екран тощо (рис. 5.1).

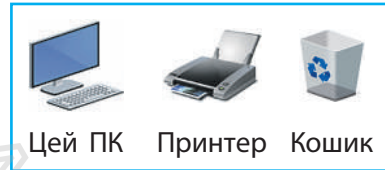


Рис. 5.1

Значки можуть міститися на робочому столі, у вікнах папок.

Кожна програма або папка може мати власну піктограму. За її виглядом часто можна визначити призначення об'єкта (рис. 5.2).

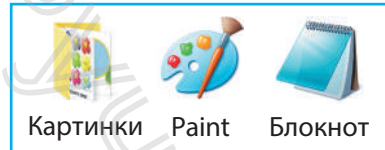


Рис. 5.2

Операції над об'єктами в ОС із графічним інтерфейсом зручно виконувати за допомогою миші чи сенсорного екрана.

► Файли

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

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

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

Ім'я файлу може містити літери (англійські, українські й ін.) і цифри, а також майже всі розділові знаки та спеціальні символи:

`_ $ # & @ ! % () { } ' ~ ^ .`

Здебільшого назву файлу призначає людина, 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 можна відкрити.

Розгляньмо деякі розширення та типи файлів:

.txt — текстові файли	.com, .exe — програми
.mp3, .wav — звукові файли	.xlsx — файли документів Excel
.docx — файли документів Word	.png, .jpg — файли зображень

► Папки

На дисках одночасно можуть зберігатися сотні тисяч файлів. Для їх групування використовують папки.

 **Папка** (або **тека**) — елемент файлової системи, контейнер для зберігання файлів та інших папок.

Кожна папка має назву; розширень не використовують.



Розпочинаючи роботу над проектом, який складається з багатьох файлів, варто створити для нього окрему папку. Це потім допоможе легко відшукати потрібні для роботи файли (рис. 5.4).

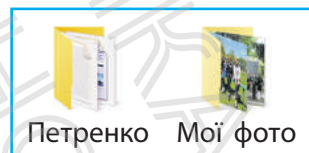


Рис. 5.4

Швидко відкривати потрібні об'єкти допомагає *ярлик* — спеціальний файл, який містить шлях до відповідного об'єкта й створюється командою контекстного меню.

Піктограма ярлика така, як у зв'язаного об'єкта, лише в лівому нижньому куті має квадрат зі стрілкою всередині (рис. 5.5). Ярлик захищає об'єкт від необережних дій людини: його видалення не призводить до знищення об'єкта.

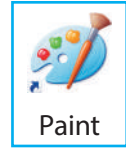


Рис. 5.5



Питання для самоперевірки

1. Які функції виконує операційна система?
2. Назвіть об'єкти операційної системи.
3. Що таке інтерфейс користувача?
4. Що таке файл?
5. Як записується ім'я файлу?
6. Поясніть різницю між значками об'єктів і ярликами.



Вправа 5

■ Ознайомитися з операційною системою Windows.

1. Запустіть браузер. Знайдіть в інтернеті відомості про призначення клавіші Print Screen та програми Ножиці.
2. Запустіть текстовий процесор, створіть документ, запишіть (скопійуйте) призначення клавіші Print Screen.
3. Скопіюйте в документ призначення програми Ножиці.
4. За допомогою клавіші Print Screen (програми Ножиці) зробіть скриншот робочого столу зі значками.
5. Уставте в документ скриншот. Збережіть файл із назвою Вправа 5. Завершіть роботу за комп'ютером.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102348).







§ 6. ОПЕРАЦІЇ НАД ФАЙЛАМИ ТА ПАПКАМИ

На робочому столі комп'ютера ми бачимо багато різних значків і ярликів об'єктів. Для успішної роботи з ними слід навчитися виконувати операції над файлами та папками.

Вікно папки «Комп'ютер»

Папка Комп'ютер (у Windows 10 називається Цей ПК) відображає диски та пристрої, які при'єднані до комп'ютера.

Вікно Цей ПК має знайомі кнопки (рис. 6.1), якими його можна: згорнути на панель завдань ; розгорнути на весь екран ; повернути до початкового розміру ; закрити .

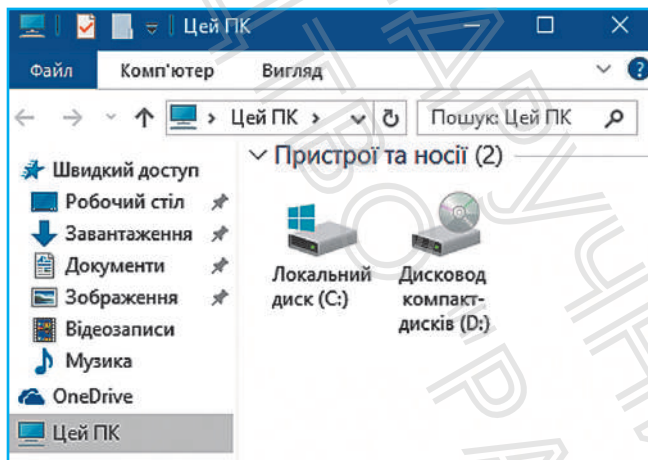


Рис. 6.1

Створення папки

Папки використовують для впорядкування файлів: кожна має відповідну до її вмісту назву і зберігається на диску.

- Щоб створити папку на диску, потрібно:
- 1) відкрити вікно Цей ПК і вибрати, наприклад, Диск D;
 - 2) вибрати команду меню Файл → Створити → Папку;
 - 3) увести (рис. 6.2) назву папки й натиснути клавішу Enter — буде створено нову папку з уведеною назвою.

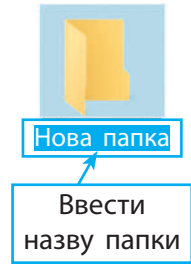


Рис. 6.2

! Щоб надалі користуватися папкою (текою), слід запам'ятати, де вона міститься, та її назву.

Створення ярлика

На робочому столі варто створити ярлик власної папки. Для створення ярлика папки або файлу потрібно:

- 1) викликати контекстне меню потрібного об'єкта;
- 2) навести вказівник на команду Надіслати (рис. 6.3);
- 3) клацнути в додатковому меню команду Робочий стіл (створити ярлик) — на робочому столі з'явиться ярлик із назвою папки, й тепер, щоб відкрити власну папку, досить двічі клацнути на її ярлику на робочому столі.

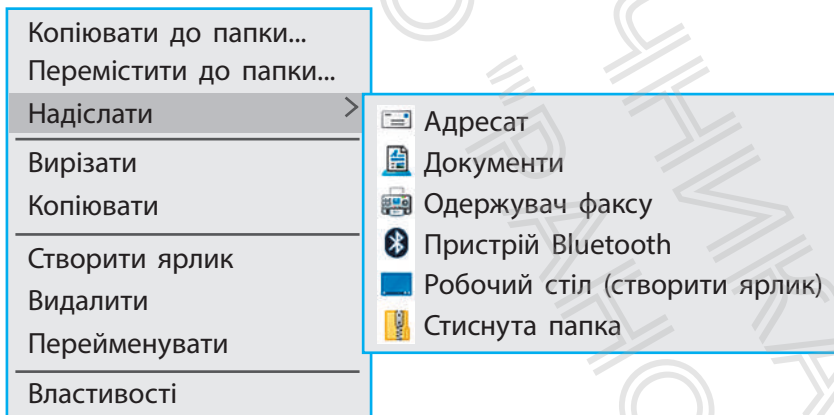


Рис. 6.3

Робота з папками, ярликами, файлами

Папки, ярлики та файли (далі — об'єкти) можна перейменувати, переміщувати, видаляти, відновлювати тощо.

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

Щоб **скопіювати об'єкт**, потрібно:

- 1) на значку об'єкта викликати контекстне меню, в якому вибрати команду Копіювати;
- 2) відкрити папку, де буде розміщена копія об'єкта;
- 3) викликати на вільному місці контекстне меню й вибрати команду Вставити — з'явиться копія об'єкта.

Щоб **перемістити об'єкт в іншу папку**, потрібно:

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

Щоб **видалити об'єкт**, слід на його значку викликати контекстне меню, вибрати команду Видалити — об'єкт переміститься в Кошик.

Щоб **відновити об'єкт**, слід відкрити Кошик, на значку об'єкта викликати контекстне меню і вибрати Відновити.

! Перемістити або скопіювати вибраний об'єкт можна також за допомогою клавіш: Ctrl+C (копіювати), Ctrl+X (вирізати) та Ctrl+V (вставити).

Після перетягування, наприклад, значка файлу в потрібну папку, правою кнопкою миші у контекстному меню можна вибрати один із варіантів дії (рис. 6.4): копіювати; перемістити; створити ярлики.

Копіювати сюди
Перемістити сюди
 Створити ярлики

 Скасувати

Рис. 6.4



Питання для самоперевірки

1. Які кнопки керування має вікно папки Комп'ютер?
2. Як згорнути вікно на панель завдань?
3. Що таке контекстне меню? Як ним користуватися?
4. Як створити власну папку на диску D:?
5. Як створити на робочому столі ярлик папки?
6. Які дії виконуються над папками, файлами, ярликами?



Вправа 6

■ Виконати розглянуті операції над файлами та папками.

1. Запустіть тренажер редагування тексту й потренуйтеся у виправлянні тексту з пропущеними символами. Повідомте результат учителю/вчительці.
2. Створіть на робочому столі папку з назвою 5 клас.
3. Відкрийте папку 5 клас і створіть у ній три папки з назвами навчальних предметів — на свій розсуд (рис. 6.5).

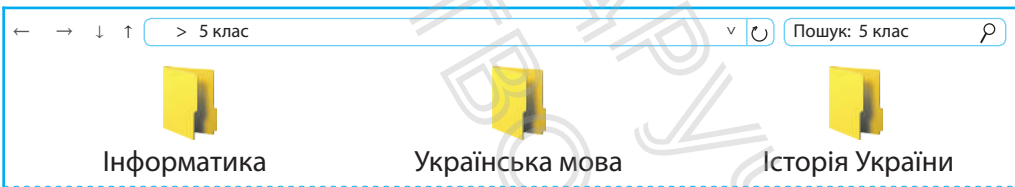


Рис. 6.5

4. Закрийте вікно папки. Видаліть папку 5 клас у Кошик.
5. Відкрийте папку Кошик. Відновіть папку 5 клас.
6. Відкрийте на робочому столі папку 5 клас. Чи відновилися внутрішні папки? Яке призначення папки Кошик?



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102349).



ПРАКТИЧНА РОБОТА 1

ВИКОНАННЯ ОПЕРАЦІЙ НАД ФАЙЛАМИ ТА ПАПКАМИ

Завдання: виконати операції над файлами та папками за-
собами операційної системи (ОС).

Обладнання: комп'ютер із ОС, навушники.

Хід роботи

Під час роботи за комп'ютером дотримуйтеся правил безпеки.

1. Підготуйте комп'ютер до роботи.
2. Створіть у запропонованому вчителем/учительською місці папку, назвіть її своїм прізвищем.
3. У власній папці створіть папки з назвами Малюнки, Тексти, Музика, Різне.
4. Створіть на робочому столі ярлик власної папки. Відкрийте її за допомогою ярлика.
5. Відкрийте папку Малюнки, згорніть її на панель завдань.
6. Розгорніть вікно папки Малюнки на весь екран і закрийте.
7. У папку Малюнки з папки, указаної вчителем/учительською, скопіюйте графічний файл.
8. У папку Тексти скопіюйте текстовий файл.
9. У папку Музика скопіюйте музичний файл.
10. Відкрийте графічний файл, що в папці Малюнки.
11. Запустіть музичний файл із папки Музика й закрийте.
12. Покажіть роботу вчителю/вчительці. Закрийте всі вікна.

Зробіть висновок: які засоби для виконання операцій над файлами та папками надає операційна система.

РОЗДІЛ 2

МЕРЕЖЕВІ ТЕХНОЛОГІЇ ТА ІНТЕРНЕТ



§ 7. Комп'ютерні мережі. Локальна мережа

§ 8. Глобальна комп'ютерна мережа

§ 9. Безпечне використання інтернету. Спілкування в інтернеті

§ 10. Пошук відомостей в інтернеті. Авторське право

Практична робота 2. Пошук інформації в інтернеті

ПОВТОРЮЄМО



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

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

Найбільша глобальна комп'ютерна мережа — «*Інтернет*». Вона надає користувачській спільноті можливості спілкування, пошуку й перегляду інформації. Ви вмієте здійснювати пошук за ключовими словами.

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



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

§ 7. КОМП'ЮТЕРНІ МЕРЕЖІ. ЛОКАЛЬНА МЕРЕЖА

Потреба в обміні даними наштовхнула дослідників на ідею з'єднати між собою два комп'ютери. Так з'явилася перша комп'ютерна мережа. З історією її виникнення пов'язані імена Дж. Ліклайдера, Пола Берена, Дональда Девіса.

Комп'ютерна мережа складається з двох або більше комп'ютерів, з'єднаних каналами зв'язку, якими передаються дані. Залежно від охоплюваної території комп'ютерні мережі бувають двох типів: *локальні* та *глобальні*.

Організація роботи в локальній мережі

Локальна мережа (*LAN*, від англ. *Local Area Network* — локальна мережа) — мережа, що об'єднує комп'ютери та інші пристрої (принтери, сканери й ін.), розташовані близько один від одного — у межах школи, банку тощо.

Для приєднання до мережі комп'ютер повинен мати спеціальний пристрій — **мережевий адаптер** (рис. 7.1, а), а також потрібен інший пристрій — **мережевий комутатор** (англ. *switch* — перемикач) (рис. 7.1, б). До нього кабелями приєднують комп'ютери (рис. 7.1, в) й решту пристроїв.

Можна побудувати також безпроводну локальну мережу.



Рис. 7.1

► Сервер і клієнт

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, надані іншими, називають **клієнтом**, або **робочою станцією** (рис. 7.2).



Рис. 7.2

Щоб увійти в локальну мережу, в ОС Windows слід вибрати папку Мережеве оточення (рис. 7.3), у вікні якої містяться значки інших комп'ютерів і мережевих пристроїв, ярлики мережевих ресурсів, які відкривалися раніше.



Мережеве оточення

Рис. 7.3

Команда Показати комп'ютери робочої групи на додатковій панелі показує значки доступних комп'ютерів. Якщо ми двічі клацнемо на одному з них, то побачимо значки ресурсів цього комп'ютера, до яких є доступ через мережу.

► Спільне використання ресурсів

Під час колективної роботи в мережі слід пам'ятати, що деякі папки можуть бути доступними лише для читання. Це означає, що в таку папку неможливо ані записати новий файл, ані змінити та зберегти наявний.

Окрім того, мережеві ресурси іншого комп'ютера стають недоступними після його вимкнення. Таким чином доцільно зробити копію файлу на власному комп'ютері навіть тоді, коли немає потреби його змінювати.

Якщо до одного з комп'ютерів мережі приєднано принтер і відкрито доступ через мережу, то користувачі та користувачки інших комп'ютерів можуть друкувати документи.



Питання для самоперевірки

1. Які мережі називають локальними?
2. Поясніть різницю між сервером і робочою станцією.
3. У чому полягає спільне використання ресурсів?
4. Як знайти певний ресурс у мережі? Опишіть алгоритм.
5. Чому може бути недоступною мережева папка, з якою ви вже працювали раніше?



Вправа 7

- Навчитися працювати з файлами за допомогою локальної комп'ютерної мережі.
1. Запишіть, як називається робоча група, до якої належить ваш комп'ютер. Яке його ім'я в мережі?
 2. У зазначеній мережевій папці створіть папку.
 3. Назвіть створену папку своїм прізвищем та ім'ям.
 4. Скопіюйте до створеної папки вказаний учителем/учителькою файл, розміщений на вашому комп'ютері.
 5. Скопіюйте до створеної папки вказаний учителем/учителькою файл, розміщений на іншому комп'ютері.
 6. Зробіть скриншот вікна папки зі значками. Додайте в документ і збережіть у цій папці з назвою Вправа 7.
Підказка: в ОС Windows натискання сполучення клавіш Alt + PrtScr копіює в буфер обміну скриншот активного вікна. Після цього його можна вставити, наприклад, у документ.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102350).



§ 8. ГЛОБАЛЬНА КОМП'ЮТЕРНА МЕРЕЖА

Глобальна мережа (WAN) — це мережа, що об'єднує комп'ютери та локальні мережі, розташовані далеко одне від одного — у різних містах, державах і на континентах.

Організація роботи в глобальній мережі

Найбільшою глобальною мережею є «Інтернет».

«Інтернет» — глобальна комп'ютерна мережа, що складається з мільярдів окремих комп'ютерів у всьому світі.

Інтернет об'єднує локальні мережі навчальних і наукових закладів, промислових підприємств, державних установ тощо, а також окремі комп'ютери. В інтернеті циркулює різноманітна інформація, доступна майже з будь-якого куточка планети (рис. 8.1).



Рис. 8.1

У повсякденному житті нам часто доводиться чути слово «провайдер» (від англ. *provider* — постачальник).

Провайдер — це організація, що надає послуги, пов'язані з доступом до глобальної мережі.

1 У квітні 2021 року необмежений (безлімітний) доступ до інтернету з'явився на українській антарктичній станції «Академік Вернадський», розташованій на острові Галіндез архіпелагу Аргентинські острови (рис. 8.2).



Рис. 8.2

Для адресації ресурсів використовують *доменні імена*.

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

Доменне ім'я складається з назв кількох доменів (імен доменів), відокремлених крапками. Рівень доменів зростає справа наліво (рис. 8.3). Домен *1-го (верхнього) рівня* зазвичай вказує тип організації, державу; домен *2-го рівня* часто є іменем сервера даних; першим зліва є власне ім'я ресурсу.

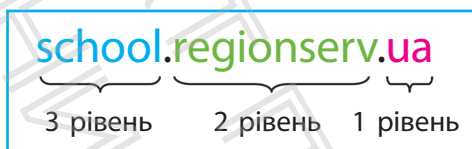


Рис. 8.3

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

Гіпертекст

Документи, розміщені в інтернеті, можуть містити зв'язки для швидкого доступу до відомостей в інших документах.

Гіпертекст — це текстовий документ, який містить гіперпосилання, клацання на яких дає змогу перейти до іншого місця в документі або до іншого документа.


Гіперпосилання виділяються кольором, підкресленням, рамкою малюнка тощо. Під час наведення на нього вказівник миші набуває вигляду руки: . Документ, пов'язаний із гіперпосиланням, може міститися на будь-якому комп'ютері в мережі (рис. 8.4).



Рис. 8.4

World Wide Web

Пригадаємо, що **World Wide Web (WWW)** — це сукупність гіпертекстових документів (вебсторінок), пов'язаних гіперпосиланнями та розміщених в інтернеті.

Вебсторінки, що мають спільну тему та розміщення, називають **сайтом**. Для перегляду вебсторінок потрібна спеціальна програма — **вебоглядач**, або **браузер** (від англ. *browser* — оглядач, провідник).

До найпопулярніших браузерів належать такі (рис. 8.5):



Opera



Mozilla Firefox



Google Chrome



Microsoft Edge

Рис. 8.5

У вікні браузера Google Chrome (фрагмент на рис. 8.6) є елементи керування, які можна бачити в інших браузерах.

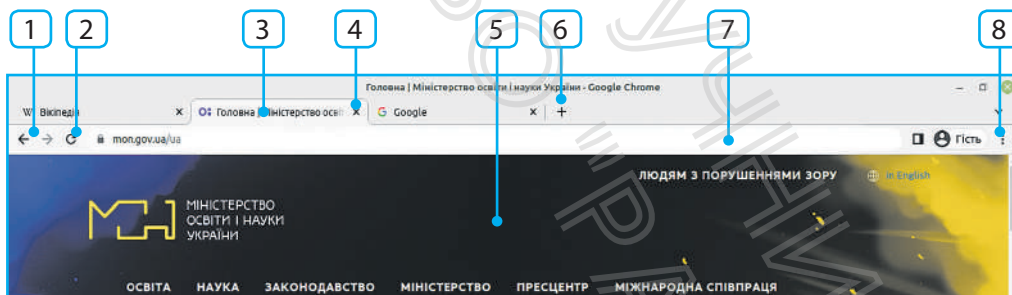


Рис. 8.6

Тут 1 — кнопки переходу між сторінками; 2 — кнопка оновлення сторінки; 3 — вкладка вебсторінки; 4 — кнопка закриття вкладки; 5 — робоче поле; 6 — кнопка додавання нової вкладки; 7 — рядок адреси; 8 — кнопка Меню.

Робота із сайтами

Приватні особи, організації, навчальні заклади розміщують на сайтах багато різноманітної інформації.



Sinoptik.ua — прогноз погоди; ukrposhta.ua — пошта України; megogo.net — розваги й ін.

Щоб перейти на вебсторінку, потрібно ввести її адресу в рядок адреси браузера. Це можна зробити різними способами, наприклад:

- набрати з клавіатури;
- вибрати зі списку меню Закладки або Історія;
- на відкритій вебсторінці клацнути на гіперпосиланні.

Після введення адреси вебсторінки автоматично встановлюється зв'язок із сервером і вебсторінка відкривається.



Якщо в полі адреси браузера ввести abetka.ukrlife.org, у робочому полі відкриється головна сторінка сайту Весела абетка (рис. 8.7). Якщо клацнути гіперпосилання Абетки, Казки, Читанка, Усмішки тощо, то потрапимо на інші вебсторінки з гіперпосиланнями для подальших переходів.



Рис. 8.7



Питання для самоперевірки

1. Розкажіть, що ви знаєте про інтернет.
2. Яку роботу виконує провайдер?
3. Поясніть структуру доменного імені ranok.com.ua.
4. Що таке гіпертекст? Наведіть приклади.
5. Для чого призначений браузер?
6. Назвіть поширені браузери.
7. Якими способами можна ввести адресу сайту?



Вправа 8

■ Відкрити сайт у браузері, переглянути його сторінки, переходячи за гіперпосиланнями.

1. Запустіть браузер і текстовий процесор.
2. Наберіть в адресному рядку браузера `abetka.ukrlife.org` і натисніть клавішу Enter. Опишіть, що відбулося після цього. Зробіть скриншот (клавіша Print Screen) і вставте його в текстовий документ.
3. Перейдіть в інший розділ сайту (Казки, Усмішки тощо) і додайте в документ скриншот.
4. Скопіюйте з адресного рядка браузера адресу сторінки та вставте в документ.
5. Перейдіть в інший розділ сайту та додайте в документ скриншот і адресу сторінки.
6. Збережіть файл у власній папці (див. вправу 7) із назвою Вправа 8. Завершіть роботу з комп'ютером.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням `rnk.com.ua/102351`).



§ 9. БЕЗПЕЧНЕ ВИКОРИСТАННЯ ІНТЕРНЕТУ. СПІЛКУВАННЯ В ІНТЕРНЕТІ

Інтернет — це майже необмежений ресурс, який можна використовувати для навчання, відпочинку та спілкування з друзями тощо. В інтернеті є багато цікавих і корисних матеріалів, проте він може бути й небезпечним, особливо для дітей.

Ризики, пов'язані з користуванням інтернетом

Кожна особа, яка користується інтернетом, має знати про загрози, що існують у ньому. Наприклад, про комп'ютерні віруси, троянські програми тощо.

Комп'ютерні віруси — програми, здатні самостійно створювати власні копії. Вони можуть завдавати шкоди: знищувати, пошкоджувати дані або порушувати роботу комп'ютера, поширюватися на інші пристрої.

Троянські програми — шкідливі програми, які можуть викрадати 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, зазвичай із метою отримання секретної інформації.

Спам — масове розсилання небажаних повідомлень рекламного чи іншого характеру.

Далі розглянемо програми, які допомагають користувачам і користувачкам уникнути перелічених та інших загроз.



Програма	Призначення
Брандмауер	Програмне й апаратне забезпечення, що захищає локальну мережу від небезпек, які є в інтернеті
Антивіруси	Службові програми для пошуку вірусів та відновлення заражених файлів
Спам-фільтр	Програма для фільтрації електронної пошти та посилань, яка блокує листи від небажаних адресатів



Антивірус українського виробництва Zillya! (рис. 9.1) захищає комп'ютер від вірусів, троянських, шпигунських та інших шкідливих програм.



Рис. 9.1

► Безпечне користування інтернетом

Інтернет — публічне місце. Працюючи онлайн, слід дотримуватися правил безпеки так само, як ви, наприклад, дотримуетесь правил дорожнього руху.

Основні правила безпечної роботи в інтернеті такі.

- Не надавайте незнайомцям особистих даних рідних.
- Уникайте поширення своїх особистих даних.
- Не реєструйтеся на сумнівних сайтах.
- Не погоджуйтеся на зустріч із незнайомими людьми.
- Повідомте старших, якщо вас залякують.
- Пам'ятайте, що не вся інформація є правдивою.

Спілкування в інтернеті

Один із найдавніших засобів спілкування між людьми за допомогою комп'ютерів — **електронна пошта** (англ. *E-mail* — Е-пошта). Вона моделює звичайну пошту: знаючи адресу потрібної особи, їй можна надіслати листа. (Докладніше про електронну пошту ви дізнаєтеся в наступних класах.)

Існує ціла низка засобів для миттєвого обміну повідомленнями, серед яких Viber, Facebook Messenger (рис. 9.2) та багато інших.



Рис. 9.2

Форум — засіб спілкування на сайті, де можна відкрито обговорювати новини та різні проблеми.

Соціальна мережа — «місце» в інтернеті, де спілкується одночасно дуже багато людей: створюють групи за інтересами, обговорюють новини, обмінюються світлинами, відео, музикою, публікують оголошення.



До найпопулярніших соціальних мереж (рис. 9.3) належать Facebook (спілкується до 3 млрд людей), Instagram — призначена для обміну фотографіями, та TikTok — для обміну відеороликами. Месенджер Telegram також нагадує соціальну мережу.

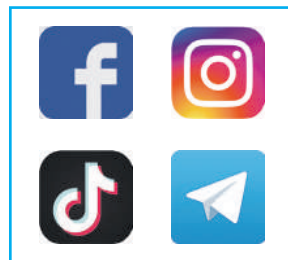


Рис. 9.3

Збільшення швидкості передавання даних в інтернеті робить електронне спілкування надзвичайно популярним.

! Деякі засоби для спілкування мають обмеження щодо віку користувачів/користувачок. На це слід зважати під час реєстрації.



Питання для самоперевірки

1. Які небезпеки пов'язані з використанням інтернету?
2. Яке призначення антивірусних програм?
3. Назвіть основні правила безпечної роботи в інтернеті.
4. Які ви знаєте засоби для електронного спілкування?
5. Назвіть кілька соціальних мереж.



Вправа 9

- Зберегти на комп'ютері тексти та малюнки з інтернету.
- 1. Запустіть браузер і перейдіть на сайт abetka.ukrlife.org.
- 👤 2. Відшукайте на сайті загадку про зиму.
- 3. Запустіть текстовий редактор і скопіюйте загадку в документ. Збережіть файл із назвою Загадка в папку Тексти, що міститься у власній папці.
- 4. Знайдіть в інтернеті малюнок на тему «Україна».
- ⚙️ 5. Збережіть малюнок у власній папці.
- 👤 6. Знайдіть вебсторінку про свій населений пункт. Збережіть текст у файлі з назвою Вправа 9.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102352).



§ 10. ПОШУК ВІДОМОСТЕЙ В ІНТЕРНЕТІ. АВТОРСЬКЕ ПРАВО

Інтернет містить сайти, які часто складаються з багатьох вебсторінок. Так, український розділ вікіпедії має понад 1 200 000 статей. Як відшукати те, що нас цікавить?

Пошукові системи

Відшукати інформацію можна за допомогою спеціальної програми — **пошукової системи**, яка за запитами користувачів і користувачок переглядає мільйони вебсторінок.

Цю систему встановлюють на пошуковому сервері, на якому зберігаються також дані про зміст вебсторінок, серед яких і здійснюється пошук. Спеціальні програми регулярно оновлюють і доповнюють ці дані.

До найвідоміших пошукових систем належать DuckDuckGo, Google, Yahoo, Lycos тощо (рис. 10.1).



Рис. 10.1

Отримавши пошуковий запит, програма сама знаходить відповідні сторінки й повертає список із гіперпосиланнями.

► Пошукова система Google

Щоб скористатися пошуковою системою Google, потрібно в браузері набрати її адресу `google.com.ua` і натиснути клавішу Enter — з'явиться початкова сторінка Google (рис. 10.2). Далі необхідно ввести в текстове поле пошуковий запит, натиснути кнопку Пошук Google та дочекатися результатів.

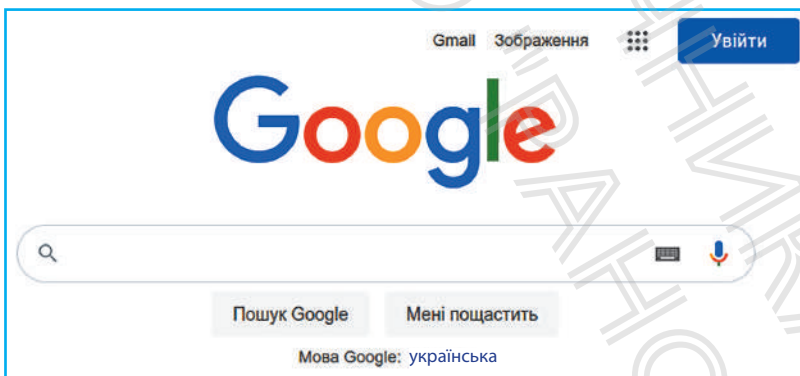


Рис. 10.2

Стратегія пошуку інформації

Успіх пошуку залежить від набору ключових слів пошукового запиту, які варто добирати за такими правилами.

- Шукайте більше ніж за одним словом: краще написати *птахи України*, ніж *птахи*.
- Найважливіші ключові слова пишуть першими.
- Пишіть без помилок.
- Якщо список знайдених сторінок занадто малий, використайте синоніми.
- Не починайте слова з великої літери, крім власних назв.
- Щоб знайти певну фразу, візьміть її в лапки: «*Людина нібито не літає, а крила має*». Лапки означають, що слід шукати саме таку послідовність слів.
- Дефіс перед ключовим словом означає, що цього слова не повинно бути на шуканій сторінці.



За запитом «Котигорошко» серед перших результатів отримали декілька посилань на український мультфільм. Коли ж запит записали так: «Котигорошко **-мультфільм**», то в перших рядках з'явилися посилання на сторінки, не пов'язані з мультфільмом.

Завантаження файлів з інтернету та авторське право

Деякі гіперпосилання вказують не на інші вебсторінки, а на файли, які можна завантажити та зберегти на своєму комп'ютері. У разі клацання на такому гіперпосиланні відкривається вікно Зберегти як...

У зазначеному вікні потрібно уточнити, яку папку слід вибрати для збереження, яке ім'я файлу, а потім вибрати команду Зберегти та розпочати завантаження файлу.

Усі матеріали, розміщені в інтернеті, мають власників/власниць. Лише ці особи мають право підписувати матеріали своїм іменем і розповсюджувати. Таке право називають **авторським**. Його охороняє Закон України «Про авторське право».

Плагіат (від латин. *plagium* — викрадати) — оприлюднення повністю або частково чужого твору під іменем особи, яка не є його автором/авторкою.

Матеріали з інтернету можна використовувати для особистих потреб, а поширювати як власні — ні. Так, до матеріалів у своїй презентації слід додати посилання на джерело.

Критичне оцінювання інформації

На багатьох сайтах публікують неперевірену, а отже, не завжди надійну інформацію. Тому, користуючись інтернетом, варто навчитися критично аналізувати знайдену інформацію (рис. 10.3).



Рис. 10.3

Фейк (англ. *fake* — підробка) — неправдива інформація, яку поширюють як правдиву. На противагу фейку, **факт** — це подія або явище, правдивість якого доведено. Наприклад, фактом є подія, що вже відбулася. **Судження** — це особиста думка людини, яку можна підтвердити або спростувати. Звичай судження має емоційну забарвленість.



Твердження «Місяць тому вийшла нова серія мультфільму» описує факт, а твердження «Неперевершена графіка сприяла надзвичайній популярності мультфільму» містить судження. У другому випадку слова «неперевершена» і «надзвичайній» передають емоції особи, покликані привернути увагу аудиторії до об'єкта обговорення.

Використання інтернету для навчання

Нині набули популярності **онлайн-ресурси** (рис. 10.4), з якими можна працювати в режимі реального часу.

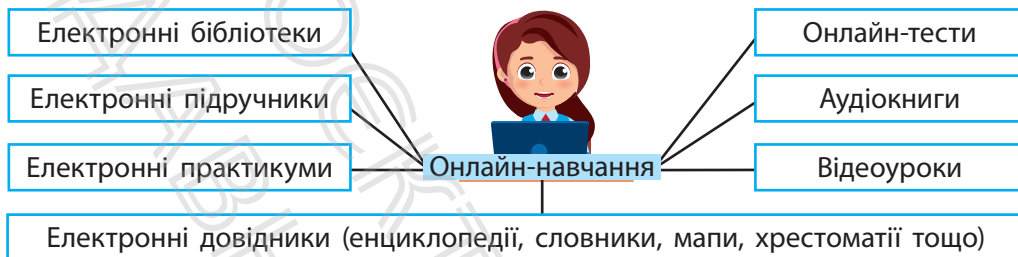


Рис. 10.4

Навчання в інтернеті має *низку переваг*: навчальні матеріали доступні в будь-який час, створюючи можливість планувати час роботи; спілкування дає змогу зрозуміти складні місця; в інтернеті можна відшукати додаткові матеріали.

Проте є й *недоліки*: доводиться багато часу проводити перед екраном комп'ютера; не кожна людина може змусити себе працювати, якщо трапиться складне завдання.

Існують сайти, що допомагають учням і ученицям у вивченні навчальних предметів (рис. 10.5). Не варто засмучуватися, якщо не вдається опанувати якийсь інтернет-курс. Можливо, він розрахований на старших дітей, тож слід порадитися з дорослими чи вибрати інший навчальний сайт.



Рис. 10.5

Серед учнів і учениць набули популярності інтернет-олімпіади. Найчастіше для участі в такому змаганні необхідно зареєструватися на відповідному сайті, а потім виконати завдання й посісти певне місце в таблиці результатів.



На сайті code.org вже майже 2 мільйони школярів і школярок з усього світу вивчають основи програмування.



Питання для самоперевірки

1. Назвіть засоби пошуку інформації в інтернеті.
2. Як потрапити на вебсторінку пошукового сервера?
3. У якому вигляді пошукова система подає результати?
4. Назвіть найпопулярніші пошукові системи.
5. Що ви розумієте під авторським правом?
6. Поясніть різницю між фейком, фактом та судженням.



Вправа 10

- Знайти в інтернеті файли і зберегти на комп'ютері.
1. Запустіть браузер. За допомогою пошукової системи знайдіть пісню-щедрівку у файлі формату MP3.
 2. Завантажте пісню і збережіть, не змінюючи назви файлу, в папку Музика у власній папці. Запишіть адресу сайту, з якого завантажено файл.
 3. Прослухайте пісню з папки Музика за допомогою навушників.
 - ⚙ 4. Перейдіть на сайт дитячого графічного редактора Tux Paint (tuxpaint.org), а потім — на сторінку завантаження програми.
Запишіть назви чотирьох операційних систем, для яких розробляється Tux Paint.
 - ⚙ 5. Завантажте програму для операційної системи, встановленої на вашому комп'ютері (або іншої, зазначеної вчителем/вчителькою).
 - 6. Відшукайте в інтернеті текст пісні, яку знайшли раніше (див. п. 1–3).
Збережіть текст у файлі з назвою Щедрівка.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102353).





ПРАКТИЧНА РОБОТА 2 ПОШУК ІНФОРМАЦІЇ В ІНТЕРНЕТІ

- Завдання:** навчитися шукати інформацію в інтернеті, за-
своїти основні правила безпечної роботи.
- Обладнання:** комп'ютер, приєднаний до інтернету.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть браузер. Перейдіть на сайт пошукової системи. Запишіть назву та адресу сайту.
2. Відшукайте в інтернеті текст української казки (наприклад, «Колобок», «Рукавичка», «Ріпка» тощо).
3. Запустіть текстовий процесор, створіть новий документ і скопіюйте текст казки. Запишіть адресу сайту.
4. Збережіть текстовий документ у власній папці. Запишіть, яка назва файлу підійде найкраще.
5. Відшукайте в інтернеті дві ілюстрації до казки й збережіть у власній папці. Запишіть назви сайтів.
6. Перейдіть на сайт <https://казка.укр/> і завантажте файл із шуканою казкою (або іншою). Перегляньте цей файл.



Зробіть висновок: як шукати в інтернеті потрібні інформаційні матеріали та зберігати їх на комп'ютері.

РОЗДІЛ 3

КОМП'ЮТЕРНІ ПРЕЗЕНТАЦІЇ



§ 11. Створення та опрацювання комп'ютерних презентацій

§ 12. Оформлення комп'ютерних презентацій

§ 13. Виступи з показом презентації

Практична робота 3. Проектування та розроблення власної презентації

ПОВТОРЮЄМО



У попередніх класах ви навчилися працювати з програмою для створення й опрацювання комп'ютерних презентацій — PowerPoint із пакунка Microsoft Office (або Impress із пакунка LibreOffice тощо).

Ви знаєте, що *комп'ютерна презентація* — це електронний документ, який складається з окремих слайдів, призначених для почергового показу на екрані монітора, а за наявності проєктора — на великому екрані.

Як вам відомо, створенню презентації передує *підготовча робота*: продумування мети та сценарію виступу, обмірковування структури презентації тощо. Ви вмiєте додавати слайди, вставляти на них тексти, малюнки тощо.

1. Назвіть програми для створення презентацій.
2. Яке призначення комп'ютерної презентації?
3. Із чого складається презентація?
4. Назвіть пристрої для показу презентацій.
5. Що передує створенню презентації?
6. Об'єкти яких типів може містити слайд?



У цьому розділі ви ознайомитеся з особливостями створення та оформлення комп'ютерних презентацій, дізнаєтеся, як правильно підготуватися до виступу.

§ 11. СТВОРЕННЯ ТА ОПРАЦЮВАННЯ КОМП'ЮТЕРНИХ ПРЕЗЕНТАЦІЙ

Чи доводилося вам розповідати друзям і подругам про літній відпочинок, домашнього улюбленця, своє захоплення? Напевно, ви не обмежувалися розповіддю, а показували фотографії, малюнки тощо (рис. 11.1).




Рис. 11.1

Поняття презентації

Як ви знаєте, **презентація** (від латин. *praesento* — представлення, подання) — це сукупність дій (виступів, доповідей тощо) і документів (фотографій, плакатів, схем, аудіо- й відеозаписів, текстів тощо), призначених для донесення до аудиторії певної інформації (про ідею, проєкт, результат роботи, товар, винахід та ін.).

Нині в презентаціях замість традиційних паперових засобів наочності здебільшого використовують можливості сучасної комп'ютерної техніки: цифрові фотографії, відеофільми, звукозаписи й ін.

 **Комп'ютерна презентація** — це набір електронних документів, підготовлених для перегляду на екрані монітора або на великому проєкційному чи іншому екрані.

Системи опрацювання презентацій

Для створення і демонстрування презентацій необхідні певні технічні та програмні засоби. Доповідач чи доповідачка може вибрати, які їхні можливості доцільно використати.



Застосунок для створення та редагування презентацій на комп'ютері називають **системою опрацювання комп'ютерних презентацій**.

Яке апаратне забезпечення потрібне для демонстрування презентації? Для показу в класі або більшій аудиторії зазвичай використовують стаціонарний комп'ютер, проєктор, екран та акустичні колонки (рис. 11.2).

Кілька осіб можуть переглянути презентацію на моніторі стаціонарного комп'ютера чи ноутбука або навіть на своїх планшетах або смартфонах у застосунку Google Презентації або в браузері (рис. 11.3).



Рис. 11.2



Рис. 11.3

Яке програмне забезпечення потрібне для створення комп'ютерної презентації? Для цього на комп'ютер слід установити відповідні програми або скористатися онлайновими застосунками, які працюють у браузері. Нині їх перелік дуже великий. Можна вибрати програмні засоби, що створюють можливості розкрити задум автора / авторки презентації.




До програмної складової систем опрацювання слайдових презентацій належать PowerPoint (Microsoft Office), Impress (LibreOffice), Apple Keynote, Google Презентації, Prezi та ін. Це далеко не повний перелік програм для новачків і професіоналів.

Редактор презентацій PowerPoint

Відомий вам застосунок Microsoft PowerPoint (рис. 11.4) входить до складу комерційного пакунка Microsoft Office для Windows і призначений для роботи зі слайдами. Доступна й безкоштовна веб-версія в складі пакунка програм Office 365.



Рис. 11.4

Пригадаймо, що редактор презентацій PowerPoint можна запустити як із головного меню, так і клацанням значка .

Для швидкого запуску програми PowerPoint її ярлики розміщують у головному меню, на панелі завдань, робочому столі, а також додають команди в контекстне меню файлів презентацій. Команди в PowerPoint 2010–2019 років випуску різняться несуттєво. Опанувавши PowerPoint 2013, ви орієнтуватиметеся в решті версій програми.

► Створення презентації

Після запуску програми на екрані з'являється вікно, в якому можна вибрати шаблон (рис. 11.5) з інтернету; створити нову презентацію; відкрити один з останніх збережених файлів комп'ютерної презентації.

Створюють нові презентації та редагують уже наявні в головному вікні.

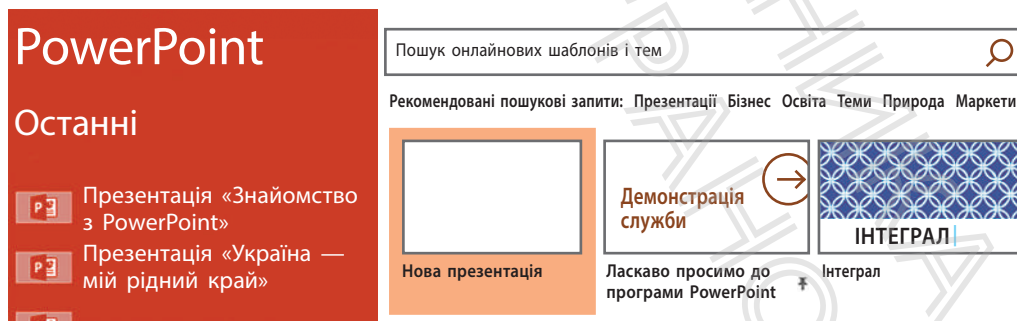


Рис. 11.5

Пригадаймо основні елементи головного вікна (рис. 11.6).

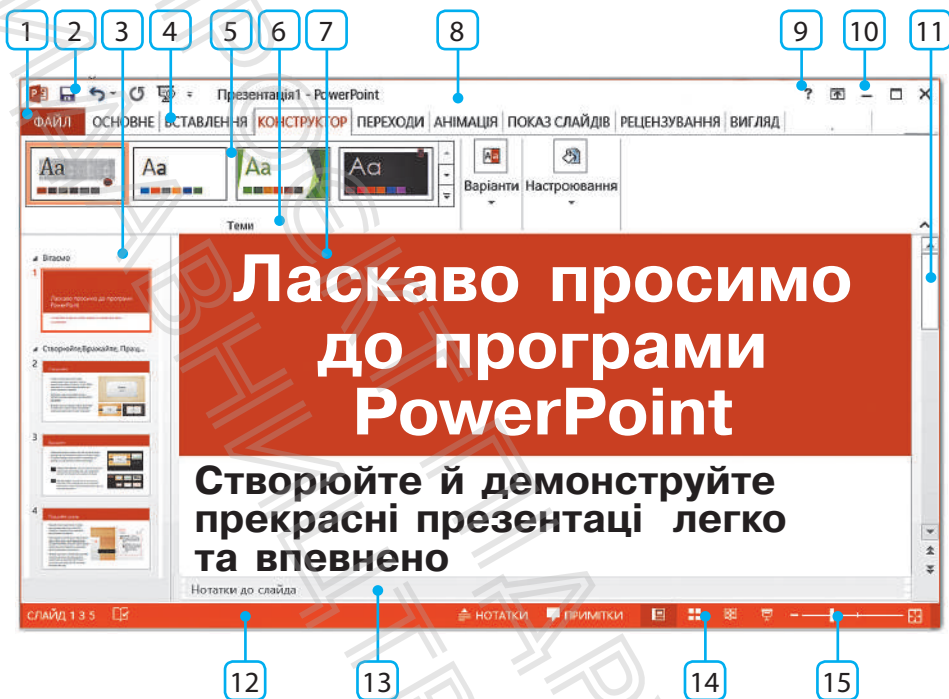


Рис. 11.6

Тут 1 — кнопка Файл; 2 — панель швидкого доступу; 3 — панель перегляду слайдів; 4 — меню вкладок; 5 — стрічка інструментів; 6 — групи інструментів; 7 — область відображення слайда; 8 — рядок заголовка; 9 — кнопка довідки; 10 — кнопки керування вікном; 11 — смуга прокручування; 12 — рядок стану; 13 — область нотаток; 14 — кнопки режиму перегляду презентації; 15 — засоби масштабування.

► Робота зі слайдами

Основним об'єктом презентації є **слайд**; розрізняють *титульний* і *звичайний* слайди. Якщо вибрати команду Нова презентація, в області відображення слайда побачимо титульний слайд із місцем для заголовка й підзаголовка (рис. 11.7).

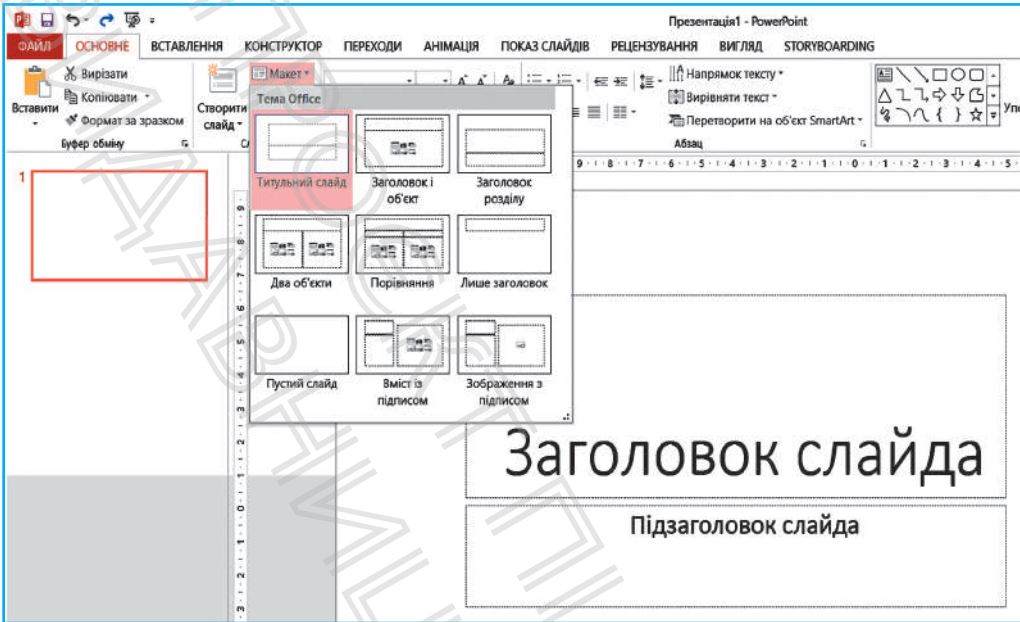


Рис 11.7

Для додавання слайда до презентації можна скористатися кнопкою Створити слайд або вибрати потрібний макет слайда у списку Макет (див. рис. 11.7).


Порядок розташування слайдів презентації можна змінити перетягуванням як окремих слайдів, так і групи на панелі перегляду слайдів.


Слайди презентації також можна виділяти, вставляти, копіювати, вилучати тощо.

► Збереження презентації

Як і інші офісні документи, презентацію для подальшого використання слід **зберегти**. Для цього потрібно:

- 1) на панелі швидкого доступу клацнути значок Зберегти або на вкладці ФАЙЛ вибрати команду Зберегти (для збереження наявного документа в іншу папку та/або з іншим іменем слід вибрати команду Зберегти як...);

- 2) указати як місце збереження одну з папок у списку або вибрати іншу папку, скориставшись кнопкою  ;
- 3) увести назву файлу в поле Ім'я файлу і вибрати команду Зберегти (Зберегти як...).

У режим редагування можна повернутися з будь-якого кроку збереження, натиснувши кнопку .

Навчившись створювати власні презентації, ви зможете зацікавити будь-яку аудиторію своїми проектами, поділитися з нею розповідями про подорожі, захоплення тощо.

- Високопрофесійними вважаються презентації Стіва Джобса (рис. 11.8), що є засновником всесвітньо відомої корпорації Apple (США). Його виступи з використанням презентацій завжди привертати увагу публіки, захоплювали продуманістю, якістю та простотою.




Рис. 11.8

Питання для самоперевірки

1. Що таке комп'ютерна презентація?
2. Яке апаратне забезпечення необхідне для її показу?
3. Назвіть програму для створення й редагування комп'ютерних презентацій.
4. Як запустити редактор презентацій PowerPoint?
5. Як створити та зберегти презентацію на комп'ютері?
6. Як змінити порядок розташування слайдів?



Вправа 11

- Створити презентацію з вибраної теми (наприклад, «Мій котик Мурчик»).
- 1. Запустіть PowerPoint і створіть нову презентацію. Кнопкою  на вкладці ОСНОВНЕ створіть *чотири* слайди.
- 2. Придумайте назву презентації (наприклад, «Мій котик Мурчик») і її вміст (наприклад, ви маєте намір розповісти про те, як Мурчик розважається і що їсть).
- 3. На *першому* слайді додайте до заголовка назву презентації (наприклад, «Мій котик Мурчик»).
У тексті слайда опишіть у підзаголовку, хто є автором/ авторкою презентації, наприклад «презентація учениці 5-го класу Сумлінної Ліни».
- 4. На *другому* слайді додайте до заголовка назву слайда (наприклад, «Розваги Мурчика»)
У тексті стисло опишіть розваги Мурчика (наприклад, «Мурчик любить гратися із м'ячем та клубком ниток»).
- 5. На *третьому* слайді додайте до заголовка назву слайда (наприклад, «Раціон Мурчика»)
У тексті стисло опишіть, чим любить ласувати Мурчик (наприклад, ковбаса, сметана).
- ⚙ 6. На *четвертому* слайді вмістіть побажання аудиторії. Наприклад, заголовок: «Дякуємо за увагу». Текст слайда: «Я і мій кіт Мурчик бажаємо вам гарного дня». Перегляньте презентацію, відкоригуйте її вміст. Збережіть файл у форматі PPTX із назвою Вправа 11.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102354).





§ 12. ОФОРМЛЕННЯ КОМП'ЮТЕРНИХ ПРЕЗЕНТАЦІЙ

Робота над презентацією складається з кількох етапів: планування, розроблення структури, добирання тексту й малюнків, створення власне презентації, підготовка виступу та виступ. Важливим для сприйняття презентації є оформлення.

Об'єкти на слайдах

Більшість слайдів містять *заголовок слайда* і *текст слайда*. Засоби для додавання інших об'єктів містить вкладка Вставлення: текстових — у групі Текст, а графічних — у групі Зображення.

Об'єкти слайда можна редагувати, тобто виправляти помилки, вносити зміни до тексту та форматувати — змінювати зовнішній вигляд об'єкта.

Шаблони презентацій

Слайди презентації кожний автор/авторка оформлюють залежно від теми, змісту, власних уподобань тощо. Для прискорення роботи можна скористатися готовими проектами презентацій — так званими шаблонами.



Шаблон комп'ютерної презентації — це схема (проект) презентації з одного чи кількох слайдів, які оформлені з дотриманням певного задуму.

Зазвичай шаблони зберігаються у файлах із розширенням .potx. Це дає змогу користувачам і користувачкам створювати на їх основі комп'ютерні презентації та зберігати їх в інших форматах, не змінюючи файлу шаблону. Небажано використовувати як шаблон презентації файл у форматі PPTX.

Потрібний шаблон можна відшукати у вікні PowerPoint. Його надає програма одразу після її запуску (рис. 12.1). Шаблони й навіть готові презентації різного оформлення та призначення можна безкоштовно завантажити із сайтів, наприклад, безкоштовні шаблони для PowerPoint містяться на сайті slidehunter.com.

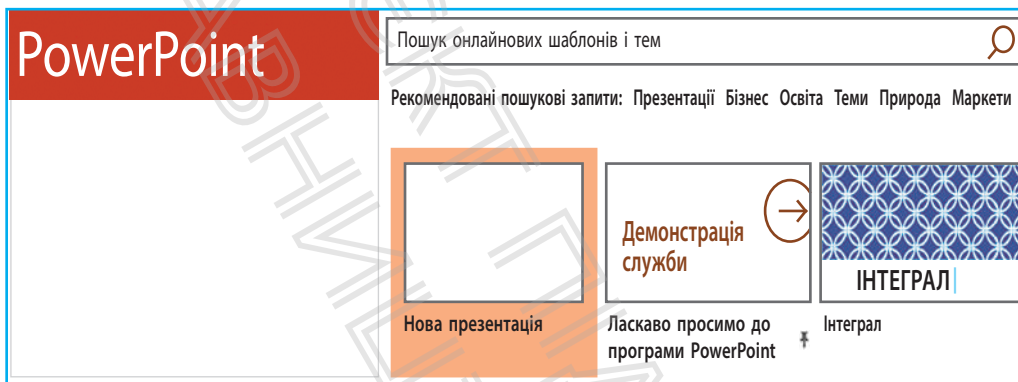



Рис. 12.1

Макет слайда

На слайди презентації можна додавати різні об'єкти. При цьому не завжди вдається швидко дібрати вдалий варіант розміщення. Розв'язати це питання допоможуть готові макети слайдів, пропоновані в PowerPoint.

 **Макет** — це схема слайда, що визначає спосіб розташування об'єктів на ньому.

Макет має такі властивості, як макет, формат тла, оформлення слайда.

Щоб створити слайд із певним макетом, потрібно:

- 1) розгорнути список Створити слайд на вкладці ОСНОВНЕ у групі Слайди;
- 2) клацнути вибраний макет (рис. 12.2, с. 64).

Розгляньмо, які макети є в наявності (див. рис. 12.2).

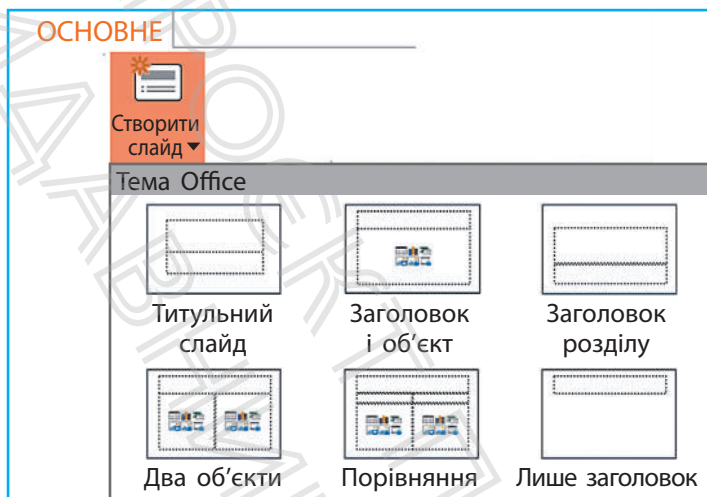


Рис. 12.2

Щоб змінити макет наявних слайдів, потрібно:

- 1) виділити один або кілька слайдів;
- 2) розкрити на вкладці Основне, група Слайди список Макет;
- 3) клацнути один зі зразків макета — в області відображення слайда з'являться ділянки, обведені прямокутником, так звані *контейнери* (рис. 12.3).

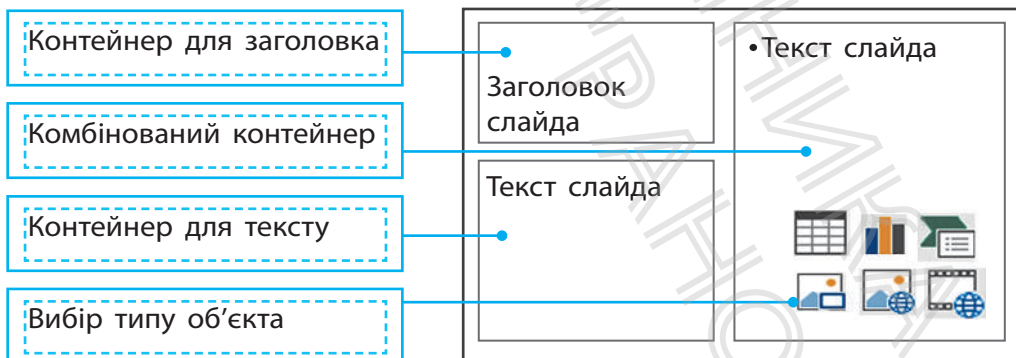


Рис. 12.3

Контейнери різних типів призначені для розміщення в них тих чи інших об'єктів слайда: заголовка, тексту, графічних та інших об'єктів із можливістю автоматично їх форматувати під час зміни макета та/або оформлення слайда.

Пам'ятайте, що об'єкти, розміщені поза контейнерами, не форматуватимуться під час автоматичного оформлення слайдів засобами PowerPoint (рис. 12.4).



Порожній макет слайда

Заповнений слайд
із доданим написом

Застосовано шаблон.
Доданий поза
контейнером напис
«загубився»

Рис. 12.4

Для введення тексту в **контейнер будь-якого типу** слід клацнути у вільному місці всередині нього та після появи курсора ввести текст.

Контейнер можна вилучити, клацнувши його прямокутну рамку, а потім натиснувши клавішу Delete.

Контейнер для заголовка вилучати небажано, оскільки пізніше, під час налаштування гіперпосилань (про це далі), заголовок слайда з'явиться у списку слайдів, що значно полегшує роботу.

Комбінований контейнер дає змогу розміщувати об'єкти різних типів. Для введення тексту достатньо клацнути на вільному місці контейнера й розпочати набирати текст.



Щоб додати в контейнер новий об'єкт, необхідно спочатку вилучити об'єкт, який вставлено раніше.

Теми оформлення слайдів

У табличному процесорі значну частину роботи допомагають автоматизувати стилі. Подібні засоби є і в програмі PowerPoint. Шаблони презентацій можуть містити різні стилі оформлення слайдів (теми).

Тема слайда (презентації) — це іменований стиль оформлення об'єктів на слайді: тла, шрифтів, графічних об'єктів тощо.

На вкладці Конструктор доступна стандартна колекція тем оформлення слайдів (рис. 12.5) — наборів властивостей об'єктів. Піктограми в колекції ілюструють тло слайда, колірну гаму та стиль тексту кожної з тем оформлення. Достатньо навести вказівник на піктограму, щоб в області відображення слайда побачити відповідне його оформлення.

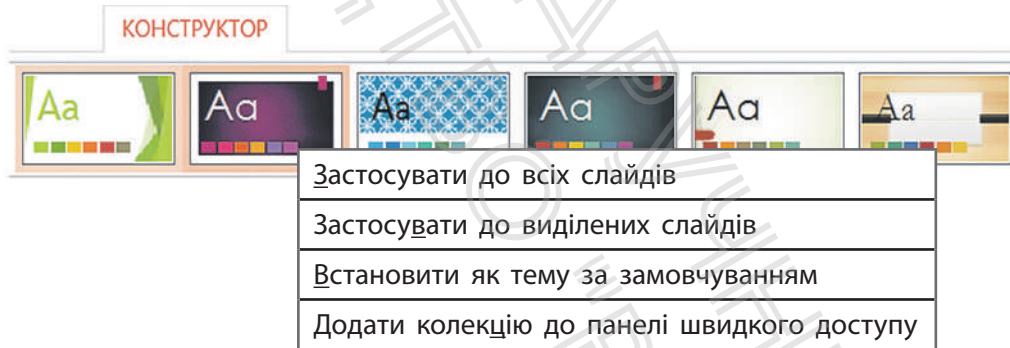


Рис. 12.5

! Якщо клацнути одну з піктограм — усі слайди презентації набудуть вибраної теми (стилю) оформлення.

Щоб змінити тему для окремих слайдів, необхідно:

- 1) відкрити контекстне меню;
- 2) вибрати команду Застосувати до виділених слайдів.

Кожну тему презентації чи слайда можна налаштувати, дібравши варіанти колірної гами слайда, заповнення тла тощо на свій розсуд. Для цього слід на вкладці КОНСТРУКТОР розгорнути список групи Варіанти та вибрати потрібне.

Одна й та сама тема оформлення слайда (наприклад, Ретроспектива) може мати різний вигляд залежно від налаштувань (рис. 12.6).

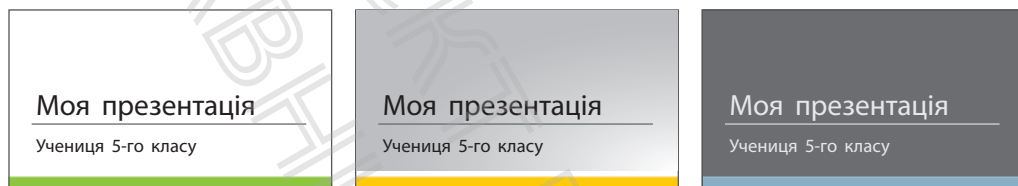


Рис. 12.6

Питання для самоперевірки



1. Що таке шаблон презентації?
2. Як вибрати готовий шаблон презентації?
3. Як змінити макет створеного слайда?
4. Чому не варто додавати до слайда об'єкти поза контейнерами?
5. Чому небажано вилучати контейнер для заголовка?
6. Що таке тема слайда?



Вправа 12

- Налаштувати загальне оформлення презентації та додати графічні об'єкти.
1. Відкрийте файл Вправа 11. Обміркуйте, які зображення варто додати, до яких слайдів і скільки. Наприклад, до *другого* слайда («Розваги Мурчика») доцільно додати фото котика, що грається з м'ячем чи клубком ниток, а до *третього* слайда — зображення двох його улюблених страв тощо (рис. 12.7, с. 68).



Рис. 12.7

2. Змініть макет *другого* слайда на такий, що містить заголовок, один контейнер для тексту й один комбінований контейнер (назва макета Вміст із підписом). На *другому* слайді перенесіть текст із комбінованого контейнера до контейнера для тексту. Комбінований контейнер має стати порожнім.
3. До комбінованого контейнера *другого* слайда додайте фото чи малюнок котика, скориставшись піктограмами  або . Під час онлайн-пошуку варто ввести до рядка пошуку ключові слова котик, клубок.
4. Змініть макет *третього* слайда на такий, що містить заголовок, два контейнери для тексту та два комбіновані контейнери (назва макета Порівняння). У *третьому* слайді перенесіть тексти із комбінованого контейнера до двох контейнерів для тексту. Комбіновані контейнери мають стати порожніми.

5. До кожного з комбінованих контейнерів *третього* слайда додайте фото чи малюнок відповідної страви, скориставшись піктограмами  або .
6. Застосуйте до *всіх* слайдів презентації одну з тем оформлення, що на вкладці КОНСТРУКТОР. Задайте однакове налаштування відтінків кольору вибраної теми на *першому* й *останньому* слайдах.
Збережіть файл у форматі PPTX із назвою Вправа 12.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102355).



§ 13. ВИСТУПИ З ПОКАЗОМ ПРЕЗЕНТАЦІЇ







Прості комп'ютерні презентації демонструвати нескладно. Існує кілька режимів показу презентацій: звичайний, режим доповідача, довільний перегляд.

Звичайний режим

Вам відомо, що для повноекранного показу слайдів презентації у звичайному режимі слід натиснути кнопку Із початку (клавіша F5) або Із поточного слайда (сполучення клавіш Shift + F5), що містяться на вкладці ПОКАЗ СЛАЙДІВ.

Якщо під час показу презентації навести вказівник на лівий нижній куток слайда, то з'являться приховані елементи керування показом презентації.

Розгляньмо таблицю з елементами керування.

Піктограма	Назва
	Повернення до попереднього слайда (ефекту)
	Перехід до наступного слайда (ефекту)
	Використання вказівника, маркера або пера
	Відображення піктограм слайдів для вибору показуваного
	Масштабування показу окремої частини слайда
	Інші елементи: режим доповідача, режим екрана тощо

Гіперпосилання

До складу багатослайдової презентації варто додати слайд зі змістом, звідки можна перейти на потрібний слайд. Для таких переходів створюють гіперпосилання (посилання).



Гіперпосилання — це виділений кольором або в інший спосіб текст чи зображення, клацання яких викликає перехід на перегляд інших даних.

Гіперпосилання дають змогу переходити до певних частин або на вебсторінки, відкривати інші документи. Їх можна додавати до текстових і графічних об'єктів.

Щоб створити гіперпосилання, слід:

- 1) виділити фрагмент тексту або малюнок;
- 2) вибрати команду Гіперпосилання, що міститься на вкладці ВСТАВЛЕННЯ в групі Посилання;

- 3) задати у вікні, з чим буде пов'язане гіперпосилання: з файлом, вебсторінкою; місцем у документі; новим документом; електронною поштою;
- 4) увести параметри і клацнути кнопку ОК.

Щоб додати гіперпосилання на слайд, потрібно вибрати варіант місцем у документі. У вікні (рис. 13.1) для першого пункту на слайді ЗМІСТ слід вибрати назву *третього* слайда, який відкриватиметься після клацання посилання.

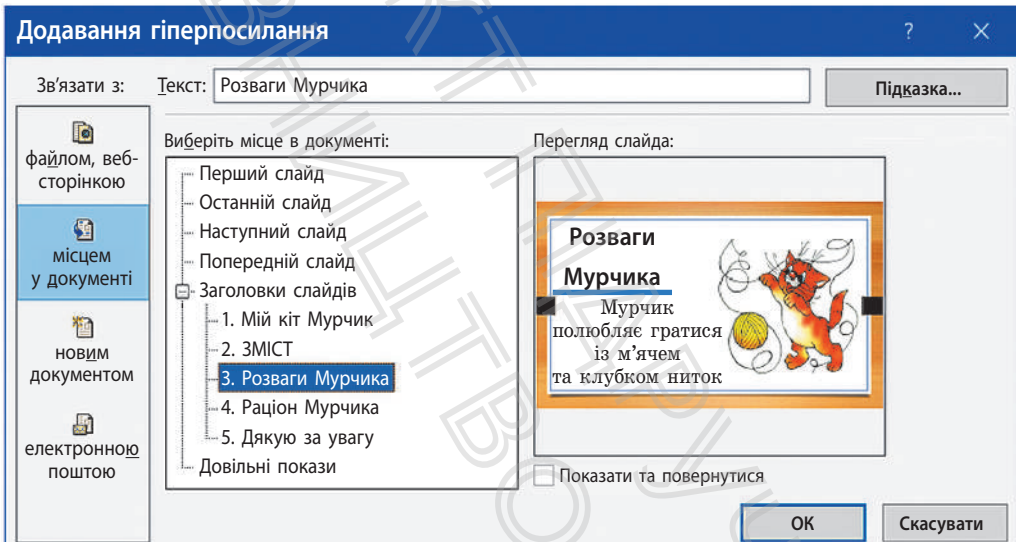


Рис. 13.1

Щоб після перегляду довільного слайда презентація продовжувалася зі слайда, на якому клацнули гіперпосилання (у нас — повернутися до слайда ЗМІСТ), необхідно встановити прапорець Показати та повернутися (див. рис. 13.1).

- У 1984 році Р. Гаскінс разом із Д. Остіном розпочали роботу над програмою Presenter, а 1987 року Д. Остін і Т. Рудкін створили оригінальну версію програми. Пізніше Р. Гаскінс запропонував назвати програму PowerPoint.

Щоб із будь-якого слайда повернутися на слайд змісту, можна на цих слайдах створити *кнопку дії* за алгоритмом: ВСТАВЛЕННЯ → Фігури → Кнопки дії → Кнопка дії: Додому.

Кнопки дії — це зображення кнопок, для яких уже призначено одну із часто вживаних дій.

Наприклад, для кнопки дії Додому за замовчуванням призначено перехід до першого слайда після клацання миші. Його можна замінити гіперпосиланням на слайд Зміст, який заздалегідь було додано до презентації.

Для цього в списку Перейти за гіперпосиланням потрібно вибрати команду Слайд... (рис. 13.2, а) і в переліку, що з'явиться, вибрати потрібний слайд — у нас Зміст (рис. 13.2, б).

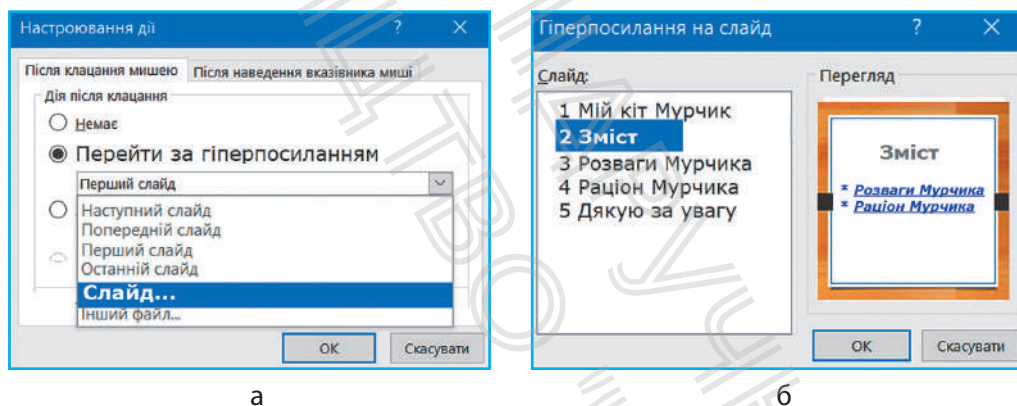


Рис. 13.2



Питання для самоперевірки

1. У якому режимі доцільно запускати показ презентації?
2. Як запустити звичайний повноекранний показ презентації з *першого* слайда; з *поточного* слайда?
3. Що таке гіперпосилання?
4. Як додати гіперпосилання на слайд?
5. Що таке кнопка дії і як її додати до слайда?

Вправа 13

- Додати гіперпосилання до комп'ютерної презентації.
- 1. Відкрийте файл презентації Вправа 12. Додайте до презентації *другий* слайд із макетом: Заголовок і об'єкт.
- 2. Додайте до заголовка *другого* слайда слово Зміст. Додайте два рядки тексту, що є заголовками *третього* та *четвертого* слайдів. Наприклад, «Розваги Мурчика» та «Раціон Мурчика» (рис. 13.3). Додайте до кожного з них гіперпосилання на слайди презентації.

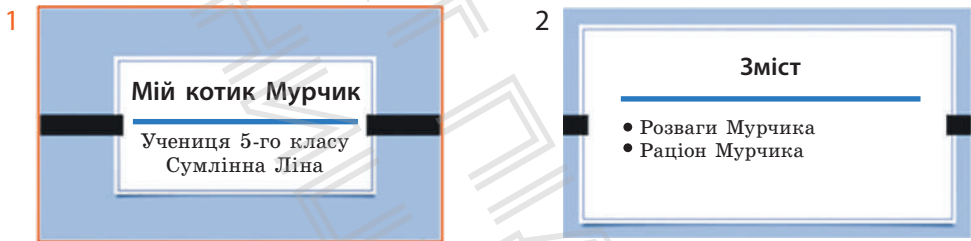


Рис. 13.3

- 3. Додайте до пунктів списку на *другому* слайді гіперпосилання на відповідні слайди презентації.
- 4. Додайте на *третьому* слайді «Розваги Мурчика» у правому нижньому кутку кнопку Додому.
- ⚙️ 5. Додайте до кнопки дію на перехід до слайда Зміст у разі клацання мишею. Скопіюйте кнопку дії на *третьому* слайді та додайте її до *четвертого* слайда.
- 👤 6. Перегляньте презентацію, перевірте роботу гіперпосилань і кнопок дій. Збережіть файл у форматі PPTX із назвою Вправа 13.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102356).





ПРАКТИЧНА РОБОТА 3 ПРОЄКТУВАННЯ ТА РОЗРОБЛЕННЯ ВЛАСНОЇ ПРЕЗЕНТАЦІЇ

- Завдання:** спроектувати та розробити презентацію для виступу тривалістю 5 хв.
- Обладнання:** комп'ютер із програмою PowerPoint.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запишіть тему презентації про своє захоплення.
2. Запишіть текст свого виступу (до 100 слів).
3. Позначте в тексті те, якими елементами презентації супроводжуватиметься ваш виступ.
4. Обміркуйте, яка кількість слайдів потрібна для супроводу виступу (не забудьте додати до основних три *традиційні* слайди: Титульний, Слайд змісту, Завершальний).
5. Сплануйте, які об'єкти будуть розташовані на кожному зі слайдів, проставте в тексті номери слайдів.
6. Завантажте та збережіть файли із зображеннями.
7. Запустіть PowerPoint та створіть нову презентацію. Створіть заплановану кількість слайдів із відповідними параметрами макета.
8. Додайте до кожного слайда заголовки й тексти.
9. Додайте до слайдів графічні об'єкти.
10. Налаштуйте загальне оформлення презентації.
11. Налаштуйте переходи слайдів.
12. Збережіть файл з іменем Практична робота 3.pptx.

Зробіть висновок: для кого призначена створена презентація, у якому режимі її краще демонструвати.

РОЗДІЛ 4

ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ



§ 14. Основні об'єкти текстового документа

§ 15. Робота з фрагментами тексту. Редагування та форматування символів

Практична робота 4. Форматування символів текстового документа

§ 16. Форматування абзаців

Практична робота 5. Форматування абзаців текстового документа

Практична робота 6. Створення текстового документа

§ 17. Опрацювання об'єктів текстового документа

ПОВТОРЮЄМО



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

Властивостями *символу* є шрифт, накреслення, розмір і колір; *абзацу* — вирівнювання, міжрядковий інтервал, відступи тощо. Значення властивостей можна змінювати.

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

1. Із яким текстовим процесором ви знайомі?
2. Назвіть об'єкти текстового документа.
3. Назвіть операції над текстовими об'єктами.
4. Які властивості має символ?
5. Які властивості має абзац?
6. Що таке форматування текстового документа?



Опрацювавши цей розділ, ви навчитеся здійснювати редагування та форматування документа за допомогою текстового процесора, а також додавати в нього таблиці та рисунки, готувати документ до друку.

§ 14. ОСНОВНІ ОБ'ЄКТИ ТЕКСТОВОГО ДОКУМЕНТА

Документ, розроблений за допомогою програм для опрацювання тексту, може містити текст, графічні зображення.

Програмне забезпечення для опрацювання текстів

Для підготовки складних текстових документів — статей, рефератів тощо — призначений **текстовий процесор**.

Ви вже знайомі з текстовим процесором Microsoft Word (далі — Word), що входить до складу програмного пакунка Microsoft Office. Його можна запустити командою з розділу головного меню Всі програми або значком (ярликом) на робочому столі, панелі задач тощо — з'явиться вікно з можливістю створення нового документа або відкриття інтернет-шаблонів чи останніх збережених файлів.

Як бачимо з рис. 14.1, вікно Word містить такі об'єкти:

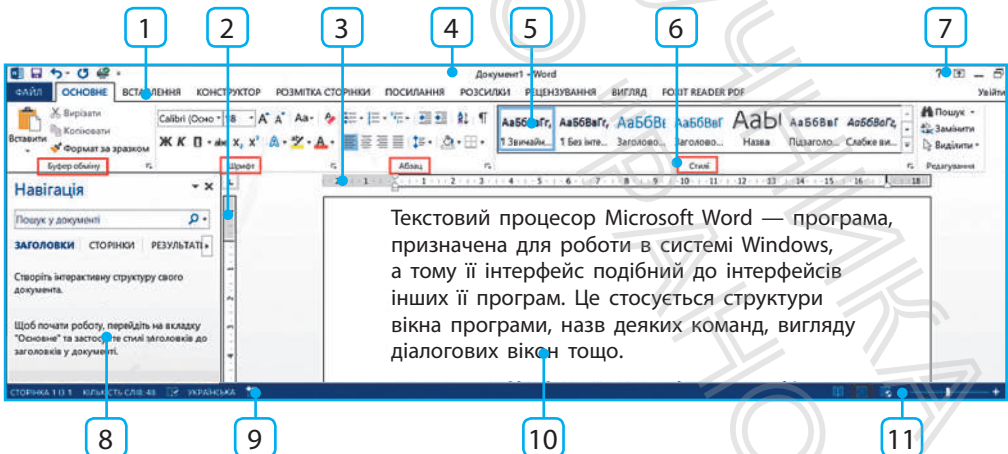


Рис. 14.1

Тут 1 — вкладки; 2, 3 — лінійки; 4 — рядок заголовка; 5 — команди; 6 — групи команд; 7 — кнопки керування вікном; 8 — панель навігації; 9 — рядок стану; 10 — робоча область; 11 — масштаб.

Об'єкти текстового документа

Розгляньте сторінки свого підручника з інформатики. Які об'єкти текстового документа ви на них бачите? Проаналізуємо схему (рис. 14.2).

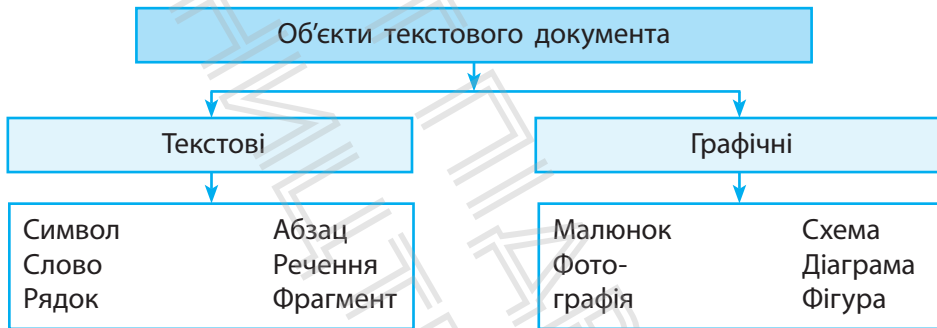


Рис. 14.2

Отже, найменшим об'єктом документа є **символ** — літера, цифра, знак пунктуації, знак арифметичної операції чи спеціальний знак.

Один або більше символів утворюють *слово*, зі слів формуються *речення*, з яких складаються *абзаци* (на рис. 14.3 — фрагмент казки про ріпку).

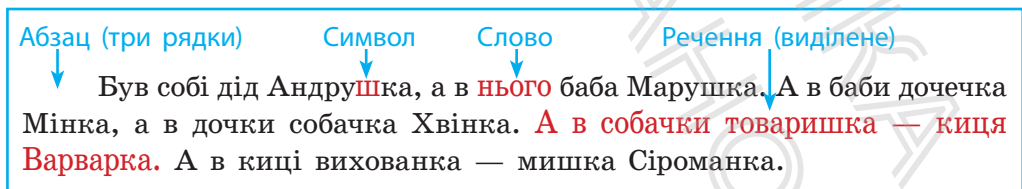


Рис. 14.3

Слово — послідовність символів, відділена від інших символів пропуском.

Речення — послідовність слів, яка починається з великої літери та закінчується крапкою, знаком питання, знаком оклику або трьома крапками.

Абзац — частина тексту, набирання якої закінчується натисканням клавіші Enter. Абзац може займати один або кілька рядків.



Операції над об'єктами текстового документа

Над об'єктами текстового документа можна виконувати певні дії. Пригадаймо, які саме, та розгляньмо таблицю:

Операція	Опис
Уведення та зберігання	Уведення з клавіатури символів у текстовий документ та збереження у вигляді файлу на диску
Редагування	Виправлення помилок, видалення, переміщення, копіювання, вставлення символів, слів, абзаців, рядків тощо
Форматування	Зміна вигляду документа: встановлення кольорів тексту, вирівнювання абзаців, зміна орієнтації сторінки тощо
Вставлення	Додавання рисунків, таблиць, діаграм, схем тощо
Макетування	Оформлення заголовків, розбивка на сторінки, нумерація їх, опрацювання рисунків, створення змісту тощо
Друкування	Виведення на папір усіх або вибраних сторінок створеного текстового документа, а також його фрагментів

Властивості сторінки

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

На рис. 14.4 ми бачимо: 1 — верхнє поле; 2 — абзац; 3 — ліве поле; 4 — сторінка; 5 — нижнє поле; 6 — верхній колонтитул; 7 — праве поле; 8 — таблиця; 9 — рисунок; 10 — нижній колонтитул.

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

Поля — це ділянки сторінки вздовж її країв, які не містять основного тексту документа.

На сторінці є *верхнє*, *нижнє*, *ліве* та *праве* поля.

Колонтитули (від франц. *colonne* — стовпець і лат. *titulus* — напис, заголовок) — повідомлення на полях сторінки.

Розрізняють *верхній* та *нижній* колонтитули. Вони можуть містити номери сторінок, назву документа або поточного розділу, прізвище автора/авторки, 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.

Окрім текстових об'єктів, сторінка документа може містити графічні об'єкти: рисунки, таблиці, діаграми, схеми тощо.

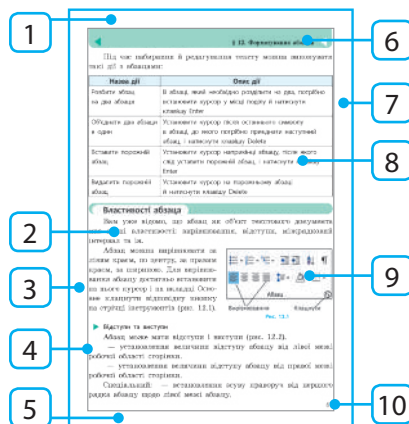


Рис. 14.4

Щоб установити розміри полів, орієнтацію та розмір сторінки (рис. 14.5), слід відкрити вкладку Розмітка сторінки і потрібний інструмент групи Параметри сторінки, де вибрати з колекції один із варіантів налаштувань або встановити користувацькі налаштування.

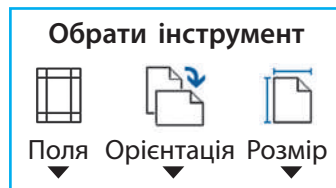


Рис. 14.5

Установлення масштабу перегляду

Масштаб відображення документа на екрані монітора встановлюють за допомогою регулятора в правому нижньому куті вікна або на вкладці Вигляд.

Щоб установити масштаб, потрібно вибрати команду Масштаб — відкриється вікно Масштаб (рис. 14.6), задати масштаб або встановити число на лічильнику Відсоток.

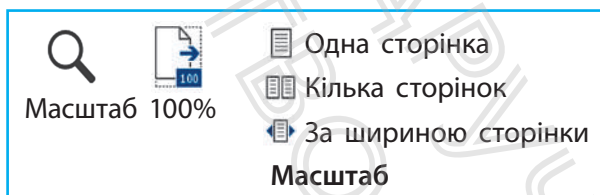


Рис. 14.6

Збереження текстового документа

Після завершення роботи документ можна зберегти.

Для збереження текстового документа потрібно:

- 1) виконати команду Файл → Зберегти → Огляд — відкриється діалогове вікно Збереження документа (рис. 14.7);
- 2) відкрити папку, в якій має зберігатися новий документ (тут Робочий стіл → Радзівілюк);
- 3) увести назву файлу та клацнути Зберегти.

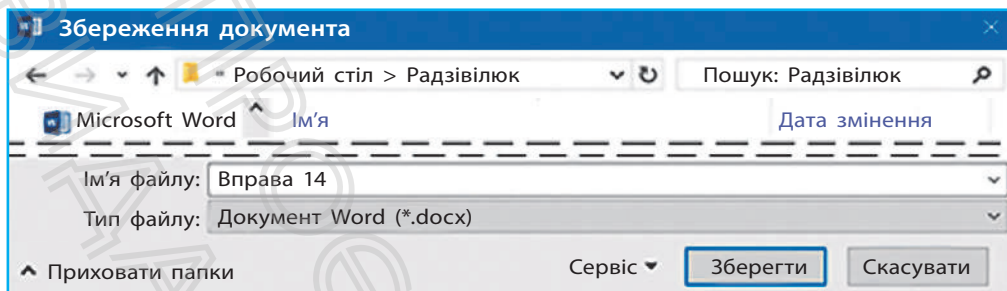


Рис. 14.7



Питання для самоперевірки

1. Опишіть вікно текстового процесора Word.
2. Назвіть текстові об'єкти документа.
3. Які поля має сторінка?
4. Яке призначення колонтитулів?
5. Як установити параметри сторінки; масштаб?



Вправа 14

- Налаштувати параметри сторінки та зберегти документ.
1. Запустіть текстовий процесор і створіть документ.
 2. Задайте такі параметри:
 - розмір паперу — А5, полів — по 2 см;
 - орієнтацію — книжкову.
 3. Установіть масштаб — За шириною сторінки.
 4. Установіть такі параметри:
 - шрифт — Times New Roman;
 - розмір — 14; накреслення — звичайний.
 5. Уведіть у документ свої ім'я та прізвище, клас і школу.
 6. Збережіть файл із назвою Вправа 14 у власній папці.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102357).



§ 15. РОБОТА З ФРАГМЕНТАМИ ТЕКСТУ. РЕДАГУВАННЯ ТА ФОРМАТУВАННЯ СИМВОЛІВ

Будь-який текст зазвичай потребує подальшого редагування та форматування. Тому розгляньмо, як це зробити за допомогою текстового процесора.

Операції над фрагментом тексту

Текстовий процесор дозволяє вирізати, копіювати або переносити на інше місце виділену частину тексту (символ, слово, речення, абзац), тобто **фрагмент тексту**.

! Перш ніж виконувати дії з фрагментом тексту, його необхідно виділити за допомогою клавіатури або миші.

► Виділення за допомогою клавіатури

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

Щоб **виділити фрагмент тексту**, потрібно:

- 1) установити курсор на початок фрагмента тексту;
- 2) утримуючи клавішу Shift, переміщувати курсор клавішами керування курсором (рис. 15.1).

Виділений фрагмент тексту підсвічується кольором (рис. 15.2). Щоб зняти виділення, потрібно натиснути будь-яку клавішу керування курсором (див. рис. 15.1).

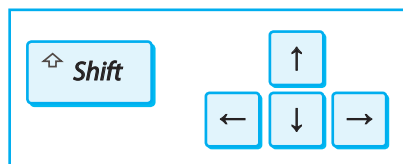


Рис. 15.1

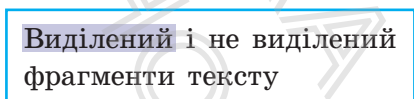


Рис. 15.2

► Виділення за допомогою миші

Для **виділення фрагмента тексту** слід установити вказівник на початок і, утримуючи натиснутою ЛКМ, протягнути вказівник до кінця фрагмента. Для **виділення кількох фрагментів** ці дії повторюють із натиснутою клавішею Ctrl.

Існує кілька *швидких способів* виділення мишею:

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

Останній зі способів проілюстровано на рис. 15.3.

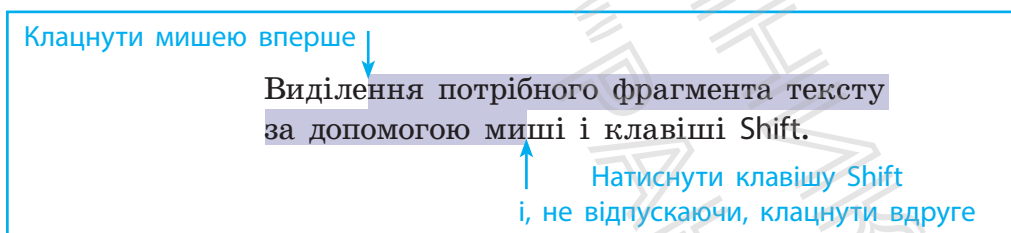


Рис. 15.3

Щоб зняти виділення фрагмента тексту, необхідно клацнути в будь-якому місці на сторінці документа.

► Копіювання та переміщення фрагментів тексту

До виділених фрагментів тексту можна застосовувати низку операцій, які виконуються за допомогою команд контекстного меню або сполучень клавіш.

Під час копіювання (сполучення клавіш Ctrl+C) фрагмент потрапляє в **буфер обміну** — ділянку пам'яті, призначену для тимчасового зберігання даних різних типів. Потім цей фрагмент можна вставити (Ctrl+V) з буфера обміну в інше місце.

Щоб **перемістити виділений фрагмент** з одного місця документа в інше, потрібно:

- 1) вирізати його (Ctrl+X);
- 2) так само вставити в інше місце (Ctrl+V).

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



Існує спосіб переміщення текстового фрагмента без його зберігання в буфері обміну. Для цього достатньо виділити текстовий фрагмент і перетягнути мишею на інше місце в документі (рис. 15.4).



Рис. 15.4

Редагування символів

У тексті зазвичай трапляються помилки трьох видів:

- *зайвий символ* — його потрібно видалити;
- *пропущений символ* — його необхідно вставити;
- *неправильний символ* — його слід замінити.

Розгляньмо рис. 15.5 та способи виправлення помилок.

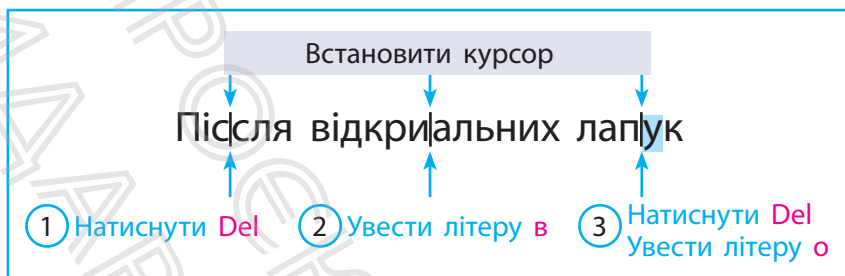


Рис. 15.5

- Для **видалення зайвого символу**, розміщеного *праворуч* від курсора, слід натиснути клавішу Delete, а розміщеного *ліворуч* від курсора — клавішу Backspace.
- Для **вставлення пропущеного символу** слід установити курсор на місце, де він має бути, і ввести з клавіатури потрібний символ (на рис. 15.5 — літера в).
- Для **замінення символу** слід видалити неправильний символ клавішею Delete (на рис. 15.5 — літера у) і ввести потрібний символ (на рис. 15.5 — літера о). На місці неправильного символу з'явиться введений.

Властивості символів

Символ як об'єкт текстового документа має такі властивості, як шрифт, розмір, накреслення, колір та ін.

! Властивості можна встановити одночасно для всіх символів виділеного фрагмента.

► Добір і зміна шрифту

Основні властивості шрифту можна вибрати та встановити за допомогою відповідних інструментів вкладки Основне або діалогового вікна Шрифт.

Розгляньмо інструменти, наведені на рис. 15.6.



Рис. 15.6

Для **викликання діалогового вікна** слід клацнути значок . Після цього у вікні Шрифт можна вибрати вид, накреслення, розмір, колір, підкреслення, спецефекти тощо.

! Розмір шрифту й інтервали між символами та рядками вимірюють у типографських пунктах (пт):
1 пт = 0,376 мм.

► Накреслення символів

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

До виділеного тексту можна застосувати такі ефекти.



Закреслений;

подвійне підкреслення;

контур тощо.

нарядковий знак $A^2 + B^2 = C^2$;

пірядковий знак C_2H_5OH ;

На стрічці інструментів можуть міститися кнопки для перетворення малих літер на великі і навпаки, а також кнопки для виконання інших команд форматування.

► Міжсимвольний інтервал

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

- для *збільшення* інтервалу між символами слід збільшити значення лічильника 3 кроком: (тут — 1,3 пт). У полі Інтервал автоматично встановиться режим Розріджений (рис. 15.7);
- для *зменшення* інтервалу між символами необхідно зменшити значення лічильника 3 кроком: У полі Інтервал устанowitzя режим Ущільнений.

Для підтвердження зміни слід натиснути кнопку ОК.

Міжсимвольний інтервал

Масштаб: 100%

Інтервал: Розріджений 3 кроком: 1,3 пт

Розташування: Немає 3 кроком:

Кернінг для знаків розміром: пунктів і більше

Рис. 15.7

Питання для самоперевірки

1. Які операції можна виконувати над фрагментом тексту?
2. Назвіть властивості символу як текстового об'єкта.
3. Опишіть способи редагування символів.
4. Як виділити в документі слово, рядок, речення, абзац?
5. Як скопіювати, вирізати, видалити, перемістити в текстовому документі виділений об'єкт?
6. Що таке буфер обміну і як із нього вставити фрагмент в інше місце?

Вправа 15

- Змінити властивості символів у текстовому документі.
- Запустіть текстовий процесор і створіть новий документ. Задайте такі параметри:
 - розмір паперу — А4; полів — по 1,5 см;
 - орієнтацію — альбомна;
 - шрифт — Arial; розмір — 10; накреслення — звичайний;
 - масштаб — За шириною сторінки;
 - вирівнювання — за лівим краєм.
 - Уведіть у документ наведений текст, наприкінці кожного абзацу натискаючи клавішу Enter.

Україна — найбільша за площею країна Європи.
Територію України омивають Чорне та Азовське моря.
Державними символами України є Гімн, Прапор і малий Герб.

- Для *першого* абзацу встановіть такі параметри:
 - шрифт — Arial; розмір — 12;
 - накреслення — курсив;
 - колір — червоний.
- Для *другого* абзацу встановіть такі параметри:
 - шрифт — Times New Roman;
 - розмір — 14;
 - накреслення — підкреслений.
- Для *третього* абзацу встановіть такі параметри:
 - розмір — 14; накреслення — жирний;
 - інтервал між символами — ущільнений 0,8 пт.
- Збережіть файл із назвою Вправа 15 у власну папку. Завершіть роботу з текстовим процесором.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102358).





ПРАКТИЧНА РОБОТА 4 ФОРМАТУВАННЯ СИМВОЛІВ ТЕКСТОВОГО ДОКУМЕНТА

Завдання: виконати форматування тексту за переліком параметрів.

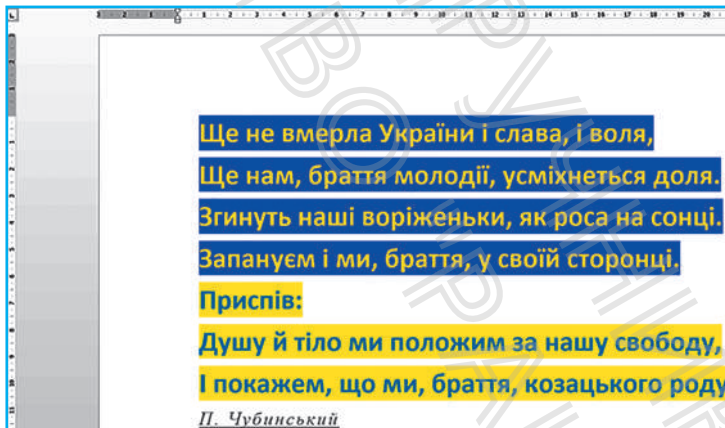
Обладнання: комп'ютер зі встановленим текстовим процесором.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть браузер.
2. Знайдіть в інтернеті текст Державного Гімну України.
3. Скопіюйте знайдений текст у буфер обміну.
4. Запустіть текстовий процесор і створіть документ.
5. Вставте з буфера обміну в документ текст Державного Гімну України.
6. Розташуйте кожен рядок в окремому абзаці.
7. Під текстом напишіть ініціали, прізвище автора слів.
8. Відформатуйте сторінку за такими параметрами:
 - розмір — А4;
 - поля — по 3 см;
 - орієнтація — альбомна.
9. Відформатуйте шрифт куплету за такими параметрами:
 - тип — Calibri; розмір — 30;
 - накреслення — жирний;
 - колір шрифту — жовтий;
 - колір виділення шрифту — синій;
 - міжсимвольний інтервал — звичайний.

10. Відформатуйте шрифт приспіву за такими параметрами:
- тип — Calibri;
 - розмір — 30;
 - накреслення — жирний;
 - колір шрифту — синій;
 - колір виділення шрифту — жовтий;
 - міжсимвольний інтервал — звичайний.
11. Відформатуйте шрифт прізвища за такими параметрами:
- тип — Times New Roman;
 - розмір — 20;
 - накреслення — курсив і підкреслення;
 - колір шрифту — чорний;
 - виділення шрифту — без кольору;
 - міжсимвольний інтервал — розріджений із кроком 2 пт.
12. За необхідності виправте помилки написання та форматування тексту відповідно до зразка (див. рисунок).



Збережіть файл із назвою Практична робота 4 у зазначеному місці (папці). Завершіть роботу за комп'ютером.

Зробіть висновок: як здійснювати форматування тексту.

§ 16. ФОРМАТУВАННЯ АБЗАЦІВ

Текст зазвичай поділяється на абзаци. Абзац завершується недрукованим символом, який вводиться натисканням клавіші **Enter**. Щоб побачити або приховати його розташування, необхідно на вкладці **Основне** в групі **Абзац** натиснути кнопку ¶.

Як ви знаєте, з абзацами можна виконувати певні дії. Розгляньмо таблицю:

Дія	Опис дії
Розбити абзац на два абзаци	Установити курсор в абзаці в місці поділу й натиснути клавішу Enter
Об'єднати два абзаци в один	Установити курсор наприкінці абзацу, 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 абзацу

Абзац як об'єкт текстового документа має такі властивості: вирівнювання, відступи, міжрядковий інтервал та ін.

Рядки абзацу можна вирівнювати за лівим краєм, по центру, за правим краєм, за шириною. Для **вирівнювання абзацу** достатньо встановити на нього курсор і на вкладці **Основне** клацнути відповідну кнопку, розташовану на стрічці інструментів (рис. 16.1, с. 93).



Рис. 16.1

Швидко встановити відступи абзацу допоможе горизонтальна лінійка (рис. 16.2). Щоб змінити величини відступів і виступів, достатньо перетягти на нове місце відповідний маркер.

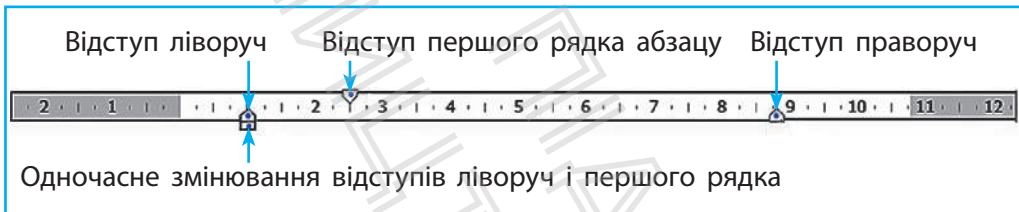


Рис. 16.2

Відступи та виступи абзацу можна також установити відповідними інструментами вкладки Основне або діалогового вікна Абзац, для виклику якого слід клацнути спеціальний значок (рис. 16.3).

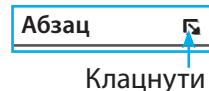


Рис. 16.3

Діалогове вікно Абзац можна також викликати командою Абзац... із контекстного меню, клацнувши правою кнопкою на абзаці.

Для **точного встановлення відступів і виступів** слід:

- 1) установити курсор на абзац і викликати діалогове вікно Абзац;
- 2) у діалоговому вікні (на рис. 16.4 подано фрагмент) за допомогою лічильників Ліворуч (тут 0,5 см) і Праворуч (тут 0,7 см) установити величини лівого і правого відступів.

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

Відступ

Ліворуч: 0,5 см

Праворуч: 0,7 см

Дзеркальні відступи

Інтервал

Перед: 12 пт

Після: 6 пт

Спеціальний: Перший рядок

На: 1 см

Міжрядковий інтервал: Множинний

На: 1,25

Рис. 16.4

► Міжабзацний і міжрядковий інтервали

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

Міжабзацний інтервал встановлюють за допомогою лічильників Перед: і Після: (на рис. 16.4 Перед: — 12 пт, Після: — 6 пт).

Для встановлення міжрядкового інтервалу необхідно:

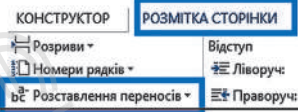
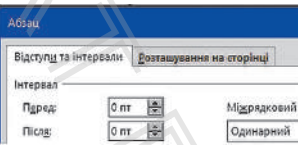
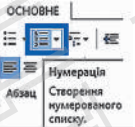
- 1) виділити один або кілька абзаців;
- 2) відкрити діалогове вікно Абзац;
- 3) відкрити список Міжрядковий інтервал у діалоговому вікні Абзац;
- 4) установити потрібний міжрядковий інтервал;
- 5) натиснути кнопку ОК.

Правила комп'ютерного набору тексту

Уміння швидко набирати текст на комп'ютері дуже важливе. Разом із тим текст потрібно набирати грамотно, а для цього необхідно дотримуватися певних правил.

Розгляньмо деякі правила набору тексту:

Правило	Приклад	
	правильно	неправильно
Під час друку між словами ставте лише один пропуск (проміжок)	Правила набору тексту	Правила...набору... тексту
Розділовий знак (крапку, кому, двокрапку тощо) не відокремлюйте пропуском від попереднього слова	Привіт, друже!	Привіт, друже!
Після кожного розділового знака ставте пропуск	3 повагою, Марійка.	3 повагою,Марійка.
Текст, який береться в лапки або дужки, пропуском від них не відділяйте	(правила набору) «цифровий»	(:правила набору:) «:цифровий:»
Біля дефіса не робіть пропусків	Людина-робот	Людина- робот
Тире відділяйте пропусками	Книга — джерело знань	Книга-джерело знань
Не використовуйте клавішу Enter для переходу на новий рядок у межах одного абзацу (перехід відбувається автоматично)	Не використовуйте клавішу Enter для переходу на новий рядок у межах одного абзацу¶	Не використовуйте¶ клавішу Enter для¶ переходу на новий¶ рядок у межах¶ одного абзацу¶

Правило	Приклад	
	правильно	неправильно
Не ставте наприкінці рядка знак переносу (це можна налаштувати)		Не ставте знаків переносу
Не вставляйте порожні рядки (для цього існують міжабзацні та міжрядкові інтервали)		Не вставляйте рядки. ¶ ¶ Існують інтервали. ¶
Не нумеруйте вручну списки та сторінки (для цього є спеціальні засоби)	 <p>1. → Січень ¶ 2. → Лютий ¶ 3. → Березень ¶</p>	1. Січень ¶ 2. Лютий ¶ 3. Березень ¶

Тут маємо такі позначення: ¶ — кінець абзацу (натискання клавіші Enter); • — пропуск; → — знак табуляції (під час автонумерації).

Отже, грамотний набір тексту сприяє правильній автоматичній перевірці правопису, полегшує читання та сприймання тексту, спрощує подальшу роботу над документом.



Питання для самоперевірки

1. Які операції виконують над абзацами?
2. Опишіть порядок дій для розбиття одного абзацу на два.
3. Наведіть послідовність дій для об'єднання двох абзаців.
4. Як можна змінити вирівнювання рядків абзацу?
5. Які відступи може мати абзац; виступи?
6. Як установити інтервали перед абзацом і після нього?

Вправа 16

■ Налаштувати властивості абзаців у заданому документі.

- Запустіть текстовий процесор, створіть новий документ. Задайте такі параметри:
 - розмір паперу — А5 (якщо такого розміру немає, то встановіть вручну: ширина — 14,8 см, висота — 21 см);
 - поля — по 2 см; орієнтацію — книжкова;
 - шрифт — Arial; розмір — 14;
 - вирівнювання — за лівим краєм; інтервали Перед: і Після: — 0 пт; міжрядковий інтервал — одинарний;
 - масштаб — За шириною сторінки.
- Уведіть наведений текст, кожне речення — з абзацу.

Правила введення тексту. Робіть пропуск між словами. Після розділового знака робіть пропуск. До і після дефіса пропуск не робіть, а до і після тире — робіть.

- Другий абзац розбийте на два. Після *третього* вставте порожній абзац, після *п'ятого* — два порожні абзаци.
- Установіть відступи для *першого* абзацу:
 - ліворуч — 1 см; праворуч — 2 см;
 - для першого рядка — 1 см.
- Установіть відступи для інших абзаців:
 - ліворуч — 0 см; праворуч — 3 см;
 - виступ навислий для першого рядка — 1 см.
- Допишіть імена п'ятох друзів і подруг (або казкових героїв). Для всього тексту встановіть міжрядковий інтервал із множником 1,1. Збережіть файл із назвою Вправа 16. Завершіть роботу з текстовим процесором.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102359).



ПРАКТИЧНА РОБОТА 5 ФОРМАТУВАННЯ АБЗАЦІВ ТЕКСТОВОГО ДОКУМЕНТА

Завдання: виконати форматування абзаців документа за заданими параметрами.

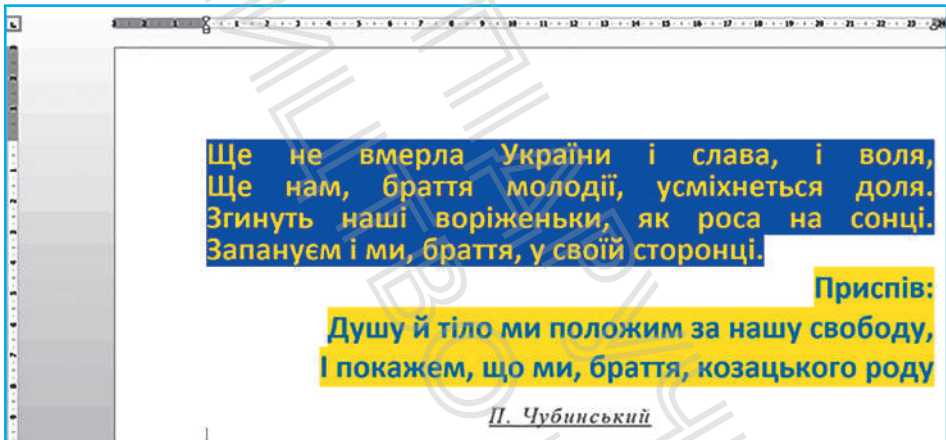
Обладнання: комп'ютер зі встановленим текстовим процесором.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Відкрийте файл Практична робота 4.
2. Об'єднайте рядки тексту Державного Гімну України у два абзаци: *перший* — куплет, *другий* — приспів.
3. Відформатуйте *перший* абзац за такими параметрами:
 - вирівнювання — За шириною;
 - відступи:
 - ліворуч — 0 см;
 - праворуч — 0 см;
 - перший рядок — немає;
 - інтервал: Перед: — 0 пт, Після: — 0 пт;
 - міжрядковий інтервал — 0,8.
4. Відформатуйте *другий* абзац (приспів) за параметрами:
 - вирівнювання — За правим краєм;
 - відступи:
 - ліворуч — 0 см;
 - праворуч — 0 см;
 - перший рядок — немає;
 - інтервал: Перед: — 0 пт, Після: — 0 пт;
 - міжрядковий інтервал — одинарний.

5. Відформатуйте *третій* абзац (прізвище автора тексту) за такими параметрами:
- вирівнювання — По центру;
 - відступи:
 - ліворуч — 0 см;
 - праворуч — 0 см;
 - перший рядок — немає;
 - інтервал: Перед: — 15 пт, Після: — 0 пт;
 - міжрядковий інтервал — одинарний.
6. За необхідності виправте помилки форматування відповідно до зразка (див. рисунок).



Збережіть файл із назвою Практична робота 5 у зазначеному вчителем/учительською місці (папці).

Завершіть роботу за комп'ютером.



Зробіть висновок: як виконувати форматування абзаців текстового документа за заданими параметрами.



ПРАКТИЧНА РОБОТА 6 СТВОРЕННЯ ТЕКСТОВОГО ДОКУМЕНТА

Завдання: створити простий текстовий документ і виконати його форматування.

Обладнання: комп'ютер зі встановленим текстовим процесором.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Запустіть текстовий процесор і створіть новий документ. Задайте такі параметри:
 - розмір паперу — А5 (якщо такого розміру паперу в списку не існує, то встановіть розмір паперу вручну: ширина — 14,8 см, висота — 21 см),
 - полів — по 1,5 см;
 - орієнтацію — книжкова;
 - шрифт — Arial (або інший, зазначений вчителем/учителькою);
 - розмір — 12;
 - накреслення — звичайний;
 - масштаб — За шириною сторінки;
 - вирівнювання — За лівим краєм;
 - інтервали абзацу:
Перед: — 0 пт;
Після: — 0 пт;
 - міжрядковий інтервал — 1,5 рядка.
2. Наберіть текст заяви на ім'я директора/директорки школи від імені матері, батька або людини, яка опікується вами, за наведеним зразком.



Директорці ЗОШ № 9
Марчук Майї Іванівні
Сумлінної Катерини Михайлівни
Заява

Оскільки моя дочка, учениця 5-А класу Сумлінна Ліна, під час навчальних занять брала участь у змаганнях із шахів, прошу надати їй можливість підвищити результати семестрового оцінювання з інформатики.

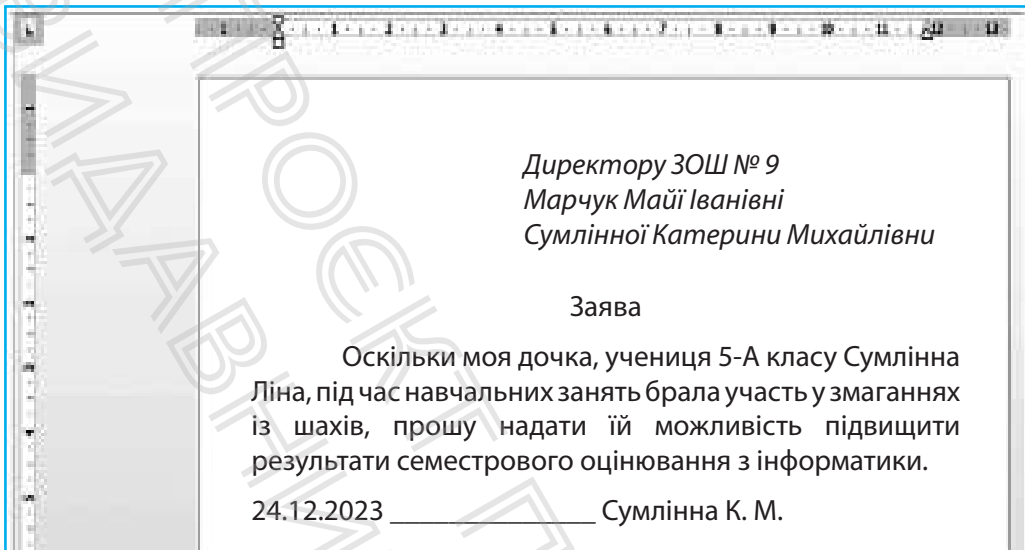
24.12.2023 _____ Сумлінна К. М.

3. Для перших трьох рядків установіть такі параметри:
 - накреслення — курсив;
 - відступ ліворуч — 7 см.
4. Для абзацу зі словом «Заява» встановіть такі параметри:
 - шрифт — Arial, розмір — 14,
 - накреслення — жирний;
 - вирівнювання — По центру;
 - міжсимвольний інтервал: розріджений — 5 пт;
 - інтервали абзацу:
Перед: — 24 пт;
Після: — 6 пт.
5. Для змістової частини заяви встановіть такі параметри:
 - вирівнювання — За шириною;
 - відступ першого рядка — 1 см.
6. Для прізвища заявника/заявниці встановіть такі параметри:
 - вирівнювання — За правим краєм;
 - інтервал Перед: — 18 пт.

Збережіть файл із назвою Практична робота 6 у власній папці.

Порівняйте результат із наведеним зразком (с. 102).

Завершіть роботу за комп'ютером.



Зробіть висновок: які засоби для створення й форматування документа надає текстовий процесор.

§ 17. ОПРАЦЮВАННЯ ОБ'ЄКТІВ ТЕКСТОВОГО ДОКУМЕНТА

У великих за обсягом документах буває досить складно знайти вручну потрібні фрагменти (наприклад, повторювані).

Автоматичний пошук і заміна фрагментів тексту

Текстовий процесор оснащений засобом для автоматичного пошуку об'єктів документа (слів, речень, рисунків тощо).

Для пошуку фрагмента тексту потрібно:

- 1) на вкладці Основне в групі інструментів Редагування клацнути команду Пошук (рис. 17.1) або натиснути сполучення клавіш Ctrl+F — відкриється бокова вкладка Навігація;

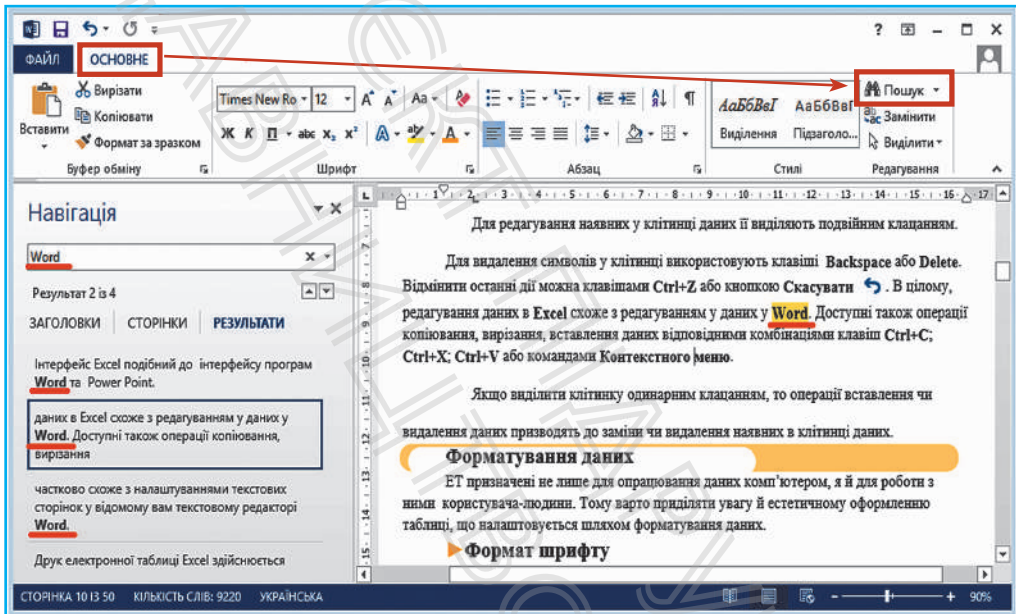


Рис. 17.1

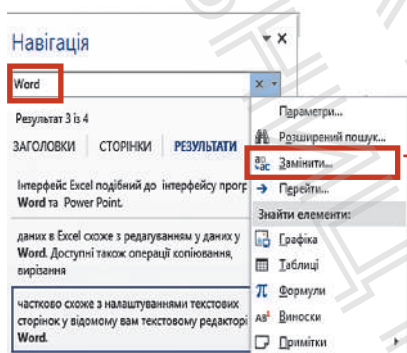
- 2) у текстове поле пошуку ввести текст, який потрібно відшукати, — на вкладці Навігація з'являться уривки документа зі знайденим фрагментом тексту;
- 3) для переходу до відповідного місця в документі клацнути один з уривків (див. рис. 17.1).

! Якщо текстовий фрагмент перед пошуком виділити та скопіювати в буфер обміну (Ctrl+C), то в такому випадку пошук фрагмента можна здійснити за допомогою сполучення клавіш Ctrl+F.

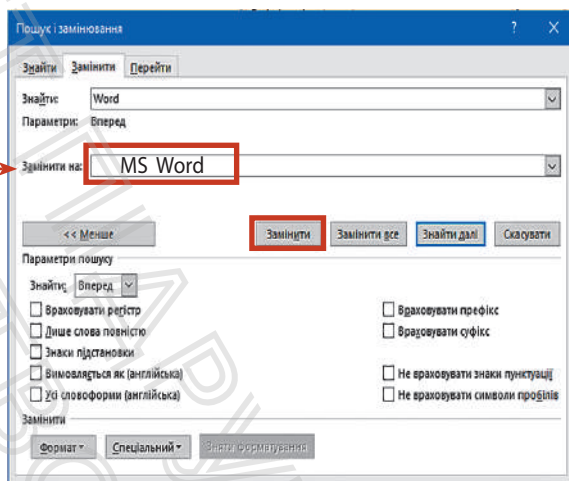
► Замінення текстових фрагментів

Щоб **замінити фрагмент**, слід виконати такі дії:

- 1) на боковій вкладці Навігація у списку, що розгортається, вибрати команду Замінити (рис. 17.2, а);
- 2) у вікно Пошук і замінювання, що відкриється, у полі Замінити на: ввести текст, яким потрібно замінити знайдений текстовий фрагмент, наприклад MS Word (рис. 17.2, б);
- 3) натиснути кнопку Замінити (або Замінити все).



а



б

Рис. 17.2

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

Діалогове вікно Пошук і замінювання (див. рис. 17.2, б) можна відкрити також сполученням клавіш Ctrl+N.



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

Для пошуку інших об'єктів (графічних зображень, таблиць тощо) слід вибрати відповідну команду зі списку, що розгортається біля поля пошуку на вкладці Навігація (див. рис. 17.2, а).

Перевірка правопису

Текстовий процесор оснащений засобом, який дозволяє перевірити й виправити написання слів у тексті. Якщо слово підкреслене червоною хвилястою лінією, це свідчить про його відсутність у словнику (зазвичай через помилку).

► Автоматична перевірка правопису

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

Виклик контекстного меню для слова з помилкою (клацання по ньому правою кнопкою миші) дає змогу:

- вибрати один із варіантів автоматичного виправлення помилки;
- знехтувати нею (Пропустити все);
- зберегти як правильний варіант для подальших перевірок (Додати до словника) (рис. 17.3).

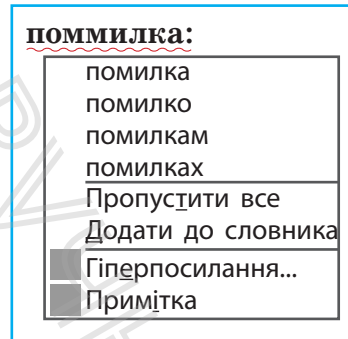


Рис. 17.3

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

► Вибір мови перевірки правопису

Мову перевірки правопису за потреби можна змінити. Для цього в рядку стану необхідно клацнути команду Мова, а в діалоговому вікні вибрати потрібну мову.




Питання для самоперевірки

1. Опишіть алгоритм пошуку фрагмента тексту.
2. Назвіть сполучення клавіш для швидкого пошуку текстового фрагмента.
3. Поясніть, як здійснити автоматичний пошук текстового фрагмента в документі.
4. Опишіть, як можна виконати автоматичний пошук мажорків, таблиць тощо.
5. Як змінити мову перевірки правопису?
6. Як виконати автоматичну заміну фрагмента тексту в документі?



Вправа 17

■ Ознайомитися з особливостями копіювання, вставлення, пошуку й заміни фрагментів тексту.

1. Запустіть текстовий процесор і створіть новий документ.
2. Запустіть браузер. Знайдіть в інтернеті опис однієї з визначних пам'яток історії або природи області, де ви мешкаєте.
3. Скопіюйте в буфер обміну кілька абзаців опису і вставте в документ.
4. За потреби відредагуйте та відформатуйте текст.
5.  Виберіть у тексті одне з повторюваних слів або словосполучень і замініть його в документі синонімом. За потреби налаштуйте параметри розширеного пошуку.
6. Збережіть документ із назвою Вправа 17.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102360).



РОЗДІЛ 5

АЛГОРИТМИ ТА ПРОГРАМИ



- § 18. Алгоритм і його властивості. Виконавці алгоритмів
- § 19. Способи опису алгоритму. Алгоритмічні структури
- § 20. Середовище опису та виконання алгоритмів
- § 21. Основні поняття мови програмування Python
- § 22. Лінійні алгоритми
- § 23. Математичні оператори мови Python
Практична робота 7. Складання лінійних алгоритмів
- § 24. «Черепашача» графіка
- § 25. Алгоритми створення зображень
Практична робота 8. Створення зображень за алгоритмами
- § 26. Логічні вирази
- § 27. Алгоритми з розгалуженнями. Команда if
- § 28. Алгоритми з розгалуженнями. Команда if...else
Практична робота 9. Складання та виконання алгоритмів із розгалуженнями
- § 29. Алгоритми з повтореннями. Цикл із параметром
- § 30. Використання циклів для побудови зображень
Практична робота 10. Складання та виконання алгоритмів із повтореннями

ПОВТОРЮЄМО



У 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 — мова, якою записують алгоритми для виконання комп'ютером. Для розроблення програм існують *системи програмування*. Усе, що ми робимо за допомогою комп'ютера, можливе завдяки програмному коду, написаному програмістами. Ви ознайомилися з основними поняттями програмування, працюючи в середовищі Scratch.

1. Що таке алгоритм?
2. Хто або що може бути виконавцем?
3. Що таке програмування?
4. Що таке мова програмування?
5. Які способи опису алгоритму ви знаєте?
6. Із якими мовами програмування ви вже знайомі?



Опрацювавши цей розділ, ви ознайомитеся з сучасною мовою програмування Python, якою можна розробляти застосунки (наприклад, ігри), створювати вебсайти, автоматизувати обчислення тощо.

§ 18. АЛГОРИТМ І ЙОГО ВЛАСТИВОСТІ. ВИКОНАВЦІ АЛГОРИТМІВ

Як ви вже знаєте, людям досить часто доводиться розв'язувати завдання, котрі постають у їхньому повсякденному житті, навчанні або роботі, діючи за певними алгоритмами.

Що таке алгоритм



Алгоритм — скінченна послідовність указівок щодо виконання дій, спрямованих на розв'язування певної задачі.



Слово «алгоритм» походить від імені арабського математика XI ст. Аль-Хорезмі (рис. 18.1). Він сформулював правила виконання 4 арифметичних дій над багатозначними числами. Латиною ім'я вченого європейці писали як *Algorithmi* і спочатку алгоритмами називали саме ці 4 правила.



Рис. 18.1



Алгоритмами є рецепти приготування страв, порядок підготовки нового смартфона до використання, ранковий розпорядок дня школяра чи школярки (рис. 18.2).



Рис. 18.2

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

2 Гра Баше. Є 11 предметів. За один хід можна взяти 1, 2, 3 предмети. Програє особа, яка візьме останній предмет. Алгоритм виграшу для першого гравця/гравчині:

1-й хід: взяти два предмети.

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

Уміння складати алгоритми важливе для всіх професій.

Властивості алгоритму

Як вам відомо, алгоритм складається з окремих указівок, які потрібно виконувати в певному порядку.

! Якщо порушити порядок виконання або пропустити якусь указівку, то, найпевніше, буде отримано неправильний результат.

3 Розгляньмо алгоритм відмикання дверей.

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

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



! Алгоритм має певні властивості, які відрізняють його від інших наборів указівок. Щоб алгоритм виконав своє призначення, його необхідно складати за певними правилами.

Розгляньмо властивості алгоритму, наведені на рис. 18.3.

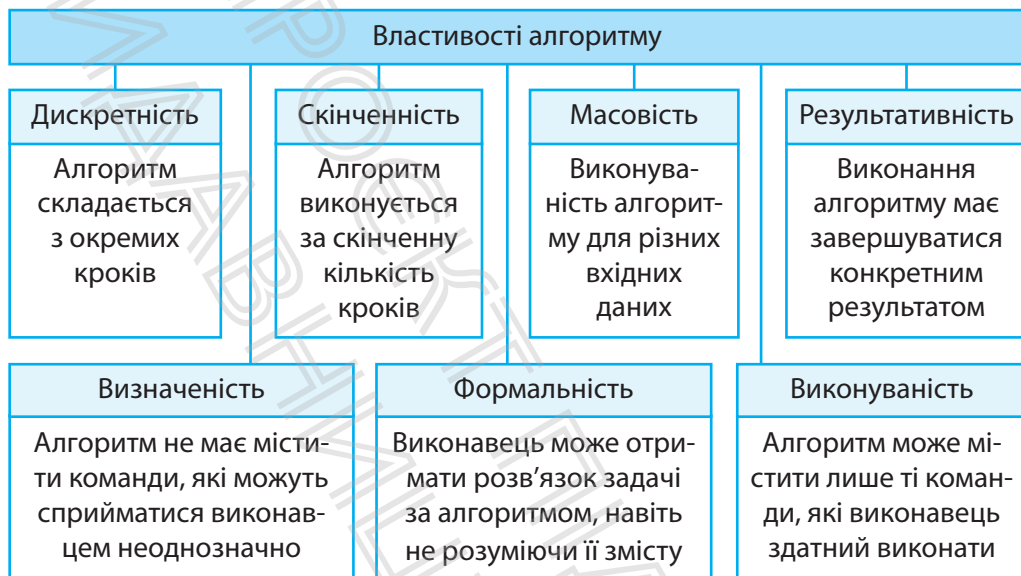


Рис. 18.3

4

Наведена послідовність указівок є скінченною.

1. Помножимо задане двоцифрове число на 10.
2. Додамо до результату 5.
3. Якщо одержано більше ніж 100, то перейдемо до команди 2, якщо ні, то припинимо виконання вказівок.

Виконавці алгоритмів

Кожен алгоритм орієнтовано на певного виконавця. Виконавцем алгоритму може бути людина, тварина, комп'ютер, автоматичний верстат, система людина–машина тощо, які «вміють» виконувати вказівки цього алгоритму.

Якщо виконавцем алгоритму є деякий пристрій, то речення «виконавець уміє виконувати вказівку» означає, що цей пристрій може виконати задану вказівку автоматично, тобто без зовнішнього втручання.

Виконавець — це об’єкт, здатний діяти за заданим алгоритмом. Щоб скласти алгоритм, орієнтований на конкретного виконавця, необхідно знати характеристики саме цього виконавця.

Характеристики виконавця

Кожен виконавець існує в якомусь середовищі і може виконувати певний набір дій. Для опису виконавця слід вказати середовище, припустимі дії, систему команд (рис. 18.4).

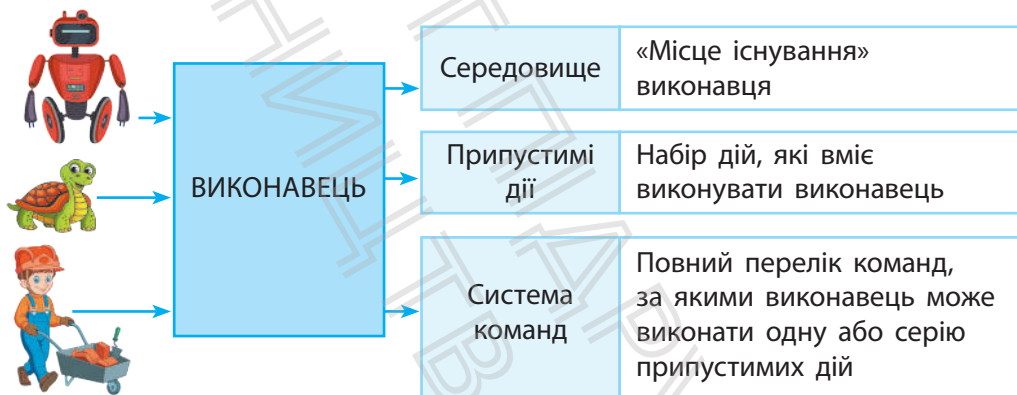


Рис. 18.4

Виконавця можна уявити як пристрій із кнопковим керуванням. Натискання кнопки означає виклик команди.

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

Познайомимося з виконавцем Кресляр, який призначений для побудови малюнків і середовищем якого є поле розміром 5×5 клітинок (рис. 18.5).

Кресляр вміє виконувати команди: Підніми перо, Опустити перо, Перейди до точки (x, y) .

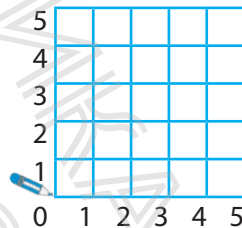


Рис. 18.5

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

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

Відмова виникає, якщо Кресляр отримує команду перейти до точки, розташованої за межами поля.



Складемо для виконавця Кресляр алгоритм побудови даху будиночка (рис. 18.6).

1. Перейди до точки $(0, 3)$.
2. Опустити 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.

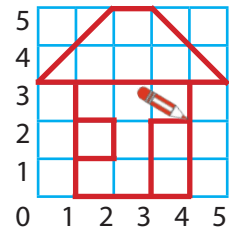



Рис. 18.6

Виконуючи алгоритм, виконавець може не розуміти змісту того, що він робить, і попри це отримувати потрібний результат. Комп'ютер — формальний виконавець алгоритмів. Отже, помилки робить не комп'ютер, а розробник алгоритму.



Питання для самоперевірки

1. Чи має послідовність указівок із прикладу 4 властивості масовості та визначеності?
2. Чи можна скласти алгоритми розв'язання таких задач: знайти корінь рівняння $ax + b = c$; відвідати театр; сконструювати машину для виконання домашніх завдань?

3. Як виконується додавання двох трицифрових чисел?
4. Що таке система команд виконавця?
5. Якими припустимими діями ви оснастили б автомат, що заміняє: а) касирку; б) вахтерку; в) двірника?
6. Складіть алгоритм для побудови решти зображення будиночка (див. приклад 5).
7.  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 18

- Скласти алгоритм, використовуючи систему команд виконавця Художник.
1. Художник зорієнтований до нижнього краю аркуша:

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

Складіть для виконавця Художник алгоритм малювання таких фігур (довжина всіх відрізків по 1 см):



2. Запропонуйте однокласникам і однокласницям виконати складені алгоритми. Чи правильно відтворюються малюнки за вашими алгоритмами?



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102361).



§ 19. СПОСОБИ ОПИСУ АЛГОРИТМУ. АЛГОРИТМІЧНІ СТРУКТУРИ

Існують *різні форми подання алгоритмів* — словесна, словесно-формульна, графічна, у вигляді програми та інші. Спосіб подання алгоритму залежить від вибору виконавця.

Форми подання алгоритмів

Словесна форма подання найзручніша для опису інструкцій із користування приладами, поведінки в разі виникнення надзвичайної ситуації тощо. Алгоритм подається як послідовність пронумерованих команд. Команди виконують у порядку зростання їх номерів, якщо немає вказівки для переходу до команди з іншим номером.



Подамо у словесній формі алгоритм приготування «швидкої» піци (рис. 19.1).

1. Для отримання соусу змішайте сметану з томатною пастою, додайте спеції.
2. Соусом намастіть основу піци.
3. Зверху викладіть шматочки помідорів.
4. Посипте натертим сиром.
5. Поставте в духову шафу на 5 хв для запікання.



Рис. 19.1

Алгоритми обчислень зручно подавати як **формули**.



Алгоритм обчислення площі прямокутного трикутника (рис. 19.2) для виконавця

Учень можна подати у вигляді « $S = \frac{a \cdot b}{2}$ »,

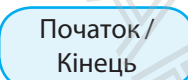
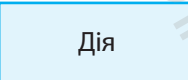
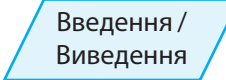

де a , b — довжини двох менших сторін трикутника».



Рис. 19.2

Для **графічного подання** алгоритмів використовують **блок-схеми**. При цьому команди подають в окремих блоках, а послідовність виконання команд позначають стрілками.

Кожна команда міститься в блоці певної форми:

Назва блока	Опис дії
 Початок / Кінець	Позначає початок або кінець алгоритму
 Дія	Позначає дію, яку потрібно виконати, тобто одну або більше вказівок (додати два числа, накреслити відрізок, намалювати будинок)
 Введення / Виведення	Позначає введення вхідних даних або виведення вихідних даних
 Умова	Позначає перевірку деякої умови. Умова може набувати одного з двох значень — істина (Так) або хибність (Ні)



Розгляньмо приклад графічного подання алгоритму.

На рис. 19.3 наведено блок-схему алгоритму знаходження середнього арифметичного трьох чисел, де:

a, b, c — вхідні дані;

S — проміжний результат;

Sr — вихідні дані (результат виконання алгоритму).

```
>>> a, b, c = 12, 7, 14
>>> S = a+b+c
>>> Sr = S/3
>>> Sr
11.0
```

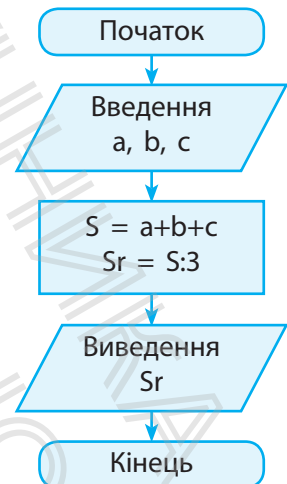


Рис. 19.3

Базові алгоритмічні структури

Як ви знаєте, в алгоритмах використовують три базові алгоритмічні структури: *слідування*, *розгалуження*, *повторення*. Згадаймо означення цих алгоритмічних структур.

Слідування — це така форма організації указівок в алгоритмі, за якої дії виконуються послідовно одна за одною, без пропусків або повторень (рис. 19.4).



Рис. 19.4

Таку структуру використано в алгоритмі на рис. 19.3.

Розгалуження — це така форма організації дій, коли залежно від виконання або невиконання певної умови виконується одна з двох дій (рис. 19.5).

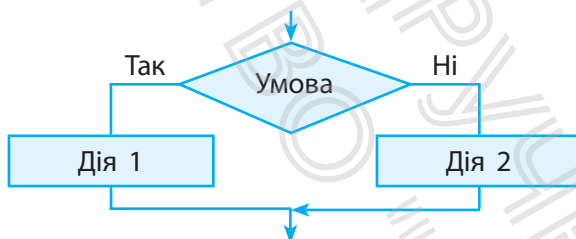


Рис. 19.5

Умова — це запитання, сформульоване так, щоб допускалася лише одна з двох відповідей: «так» або «ні». Перевірка умови має бути припустимою дією виконавця.

Якщо умова *істинна*, то виконується Дія 1 (гілка Так); якщо умова *хибна*, то виконується Дія 2 (гілка Ні). Після виконання однієї з дій виконавець переходить до дії, наступної після розгалуження.

4

Складемо блок-схему алгоритму вибору розваги: якщо в касі є квитки, то придбати квиток і переглянути фільм, інакше — піти на прогулянку до парку (рис. 19.6).

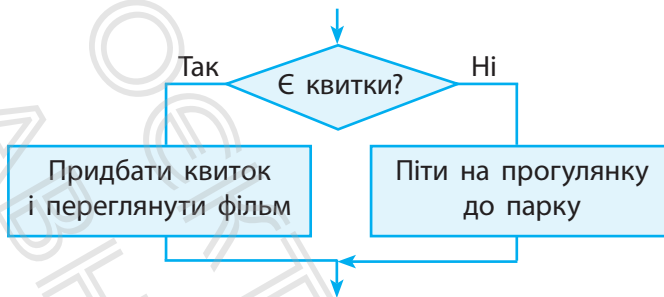


Рис. 19.6



Повторення (цикл) — це така форма організації дій, за якої одна й та сама дія залежно від певної умови виконується кілька разів.

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

Наприклад, на рис. 19.7 подано блок-схему циклу, в якому певна дія повторюється доти, поки задана умова є істинною. Під час виконання такого циклу спочатку перевіряється умова.

Якщо умова *істинна*, то тіло циклу виконується й відбувається повернення на перевірку умови.

Якщо умова *хибна*, то виконання циклу припиняється.

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

Якщо під час повторення циклу умова незмінно залишається *істинною*, то цикл може повторюватися нескінченно.

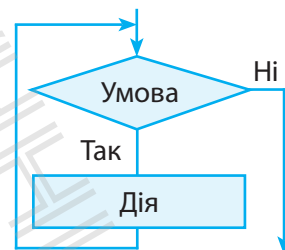


Рис. 19.7



На рис. 19.8 подано блок-схему алгоритму збирання врожаю яблук.

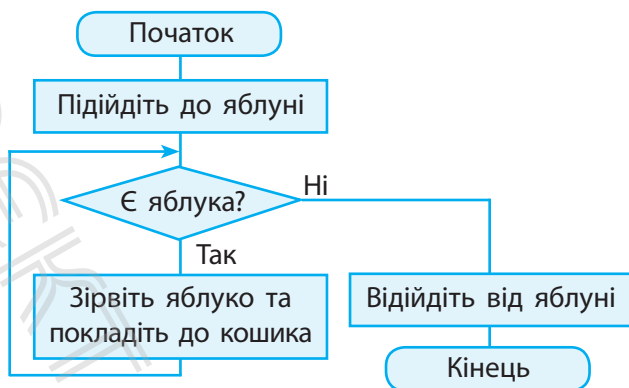


Рис. 19.8

Як побудувати алгоритм

Під час складання алгоритму розв'язування задачі необхідно дотримуватися **чіткого плану дій**. Повторимо його.

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 алгоритму для різних значень аргументів.



Питання для самоперевірки

1. Як подати базові алгоритмічні структури на блок-схемі?
2. Поясніть переваги графічного подання алгоритму.
3. Наведіть приклад алгоритму з розгалуженням.
4. Яким буде результат виконання алгоритму, наведеного на рис. 19.8, якщо врожай занадто великий?

Вправа 19

■ Виконати завдання.

1. Складіть блок-схему алгоритму обчислення площі прямокутного трикутника за прикладом 2.
2. Визначте істинність умови $a > 5$, якщо значення a дорівнює: а) 8; б) 5; в) 2.
3. Знайдіть значення величини c після виконання команди розгалуження (рис. 19.9), якщо початкове значення b дорівнює: а) 4; б) 6; в) 10.
4. Знайдіть значення величини x після виконання команди повторення (рис. 19.10), якщо початкове значення x дорівнює: а) 4; б) 9; в) 10.

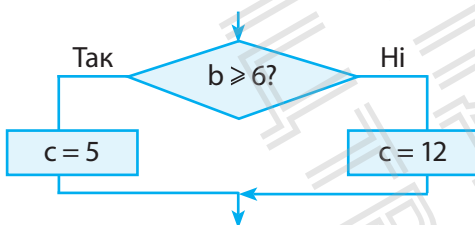


Рис. 19.9

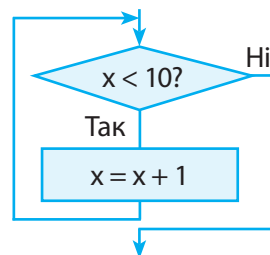


Рис. 19.10

5. Складіть блок-схему алгоритму для розв'язання задачі.
Задача. Якщо ціна книжки не перевищує n гривень, купити цю книжку й сувенір, в іншому випадку купити тільки книжку.
6. Складіть блок-схему виграшного алгоритму в грі Баше для першого гравця/гравчині (див. § 18).



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102363).



§ 20. СЕРЕДОВИЩЕ ОПИСУ ТА ВИКОНАННЯ АЛГОРИТМІВ

Пригадаймо, що *програмою* називають алгоритм розв'язування певного завдання, записаний мовою програмування. **Мова програмування** — штучна мова, призначена для запису алгоритмів у формі, придатній для їх виконання комп'ютером. Ви вже знайомі зі Scratch і основними поняттями, які будуть корисними під час вивчення інших мов програмування.

Починаймо знайомство з Python, сучасною мовою програмування, яка застосовується для розв'язування різних задач: написання прикладних програм, створення ігор тощо.

Мову програмування Python створив 1991 року нідерландський програміст Гвідо ван Россум (рис. 20.1) і назвав на честь скетч-серіалу «Летючий цирк Монті Пайтона» (англ. *Monty Python's Flying Circus*).



Рис. 20.1

Мова Python доступна практично для всіх операційних систем. Наразі використовують дві версії 2.x і 3.x. У нашому підручнику ми розглядатимемо версію Python 3.

Середовище програмування IDLE

Перш ніж почати роботу, потрібно встановити на комп'ютер систему програмування. Завантажити файл для встановлення можна із сайту [python.org](http://www.python.org) (рис. 20.2).



Рис. 20.2

Разом із версією Python 3 на комп'ютер буде встановлено IDLE — середовище програмування, в якому є всі засоби для написання та налагодження програм. Для запуску IDLE виберіть команди Пуск → Всі програми → Python → IDLE.

Після запуску IDLE відкриється вікно консолі Python Shell (рис. 20.3). Можна починати програмувати, записуючи команди після позначки «>>>». Рядок >>> називається *запрошенням*, і його наявність означає, що комп'ютер готовий прийняти вашу команду.

Якщо після запрошення набрати команду і натиснути Enter, комп'ютер її одразу виконає і виведе результат. Також у вікні консолі виводяться результати виконання програми й повідомлення про помилки в програмному коді (див. далі).

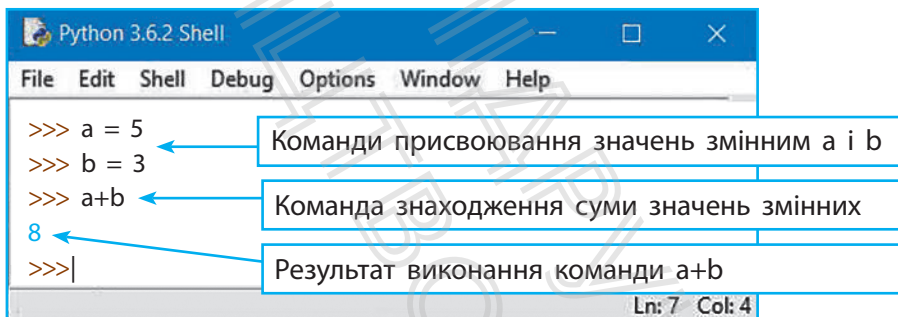


Рис. 20.3

Вікно консолі зручно використовувати для того, щоб зрозуміти, як виконується та чи інша команда. Але для створення великих програм, які необхідно зберігати й редагувати, використовують вікно програми.

Вікно програми

Вікно програми призначене для введення та редагування тексту програми. Щоб відкрити вікно програми, необхідно вибрати в меню вікна консолі команду File → New File.

Відкриється вікно програми із заголовком Untitled (рис. 20.4):

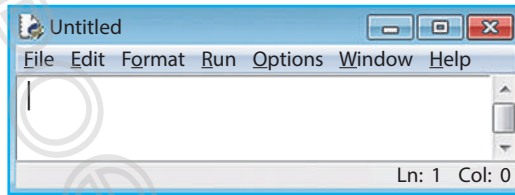


Рис. 20.4

На рис. 20.5 наведено алгоритм роботи з програмою в середовищі IDLE Python у вікні програми.

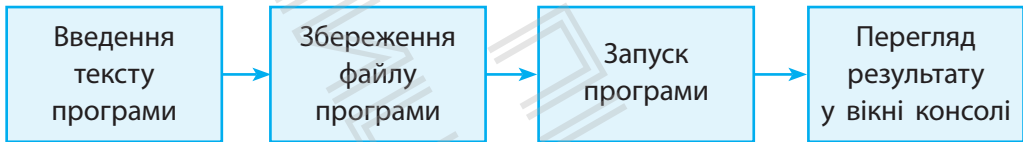


Рис. 20.5



Створимо програму розв'язування задачі за поданим раніше алгоритмом роботи з програмою.

Задача. Знайти суму і добуток двох чисел.

1. Уведемо код у вікні програми (рис. 20.6).
2. Виберемо команду File → Save As. Уведемо ім'я файлу Перша програма і натиснемо Save. Файл збережено з розширенням .py.

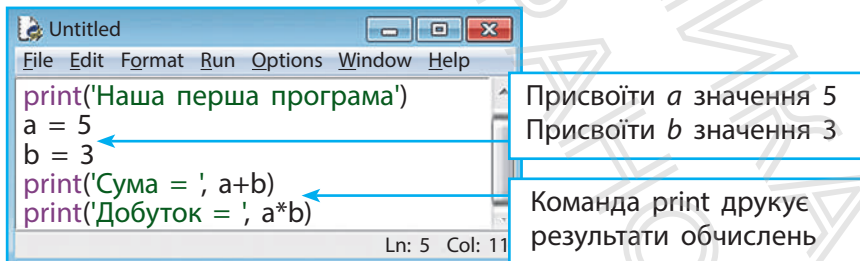


Рис. 20.6

3. Виберемо команду Run → Run Module (або натиснемо F5).
4. Розглянемо результат роботи у вікні консолі (рис. 20.7).

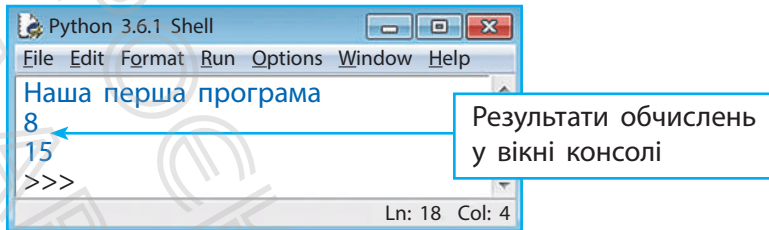


Рис. 20.7

Програму не можна запустити на виконання, поки її не збережено. У разі спроби виконати програму без збереження з'явиться попередження.

Щоб уникати помилок під час набору команд, в IDLE складові програмного коду виділяються певними кольорами:

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



Питання для самоперевірки

1. Які види вікон є в середовищі програмування Python?
2. Які повідомлення виводяться у вікні консолі?
3. Як відкрити вікно програми?
4. Для чого призначене вікно програми?
5. Який алгоритм роботи з програмою в середовищі Python?
6. Чому складові коду виділяються різними кольорами?

Вправа 20

■ Написати програму додавання двох чисел.

1. Завантажте програму: Python → IDLE.

Виберіть команду File → New File (Файл → Новий файл), щоб відкрити вікно програми.

Наберіть текст програми додавання і множення двох чисел, наведений на рис. 20.6 (с. 123).

2. Для збереження файлу виберіть команду File → Save As (Файл → Зберегти як).

У вікні збереження файлу виберіть власну папку, наберіть назву файлу Vprava20 та 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. Почнеться виконання програми.

Якщо з'явиться повідомлення про помилку, перевірте, чи правильно набрано текст програми.

Перегляньте результат виконання програми у вікні консолі. Закрийте всі вікна IDLE.

4. Повторно запусіть програму. Для цього у власній папці на значку програми відкрийте контекстне меню і виберіть команду Edit with IDLE (Змінити в IDLE).

⚙ 5. У вікні програми змініть значення змінних a і b.

6. Збережіть файл і запусіть програму на виконання.

Поясніть результат виконання програми.

Закрийте вікна IDLE.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102364).



§ 21. ОСНОВНІ ПОНЯТТЯ МОВИ ПРОГРАМУВАННЯ PYTHON

Для написання комп'ютерної програми необхідно знати алфавіт мови програмування та правила запису команд і вміти їх використовувати.

Алфавіт мови Python

Розгляньмо алфавіт, синтаксис і семантику мови Python.



Алфавіт мови — це набір символів і їх комбінацій, що може використовуватися в 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 застосовують також:

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



Синтаксис мови — це сукупність правил побудови команд мовою програмування.



Розгляньмо програмний код:

```
>>> (x+5))
```

```
SyntaxError: invalid syntax
```

У команді надруковано зайву дужку, тому в консолі з'являється повідомлення про синтаксичну помилку.

Уважно читайте такі повідомлення — це допоможе вам зрозуміти, в чому помилка, і виправити її.

Семантика мови — це опис того, що означає і як виконується кожна команда.

Із семантикою мови Python ми будемо знайомитися в міру вивчення команд.

Величини в мові Python

Окремий інформаційний об'єкт (число, символ, рядок тощо) називають **величиною**. Кожна величина характеризується *обсягом виділеної для її зберігання пам'яті, типом і значенням*. Вид **величини** визначає, чи можна змінювати значення цієї величини в програмі. Величина може бути *константою* (тобто *сталю*) або *змінною*.

Константа — величина, значення якої не може змінюватися в ході виконання програми.

Прикладом константи може бути число (5, 1.23) або рядок ("Це рядок!").

Змінна — величина, значення якої може змінюватися в ході виконання програми.

Змінні потрібні для зберігання даних. Змінним надають назви (ідентифікатори). З'ясуємо правила надання назв.

- Першим символом назви має бути літера чи знак нижнього підкреслювання «_».
- Решта назви може складатися з літер, цифр або знаків нижнього підкреслювання.
- Не можна використовувати пробіли, знаки арифметичних операцій та спеціальні символи.
- Назви змінних «чутливі» до регістру. Наприклад, `myname` і `myName` — це різні змінні.
- Не можна використовувати як назви змінних службові слова, наприклад `import`.



Правильні назви: `i`, `__my_name`, `name_23`, `a1`, `b2`.
Неправильні назви: `name with spaces`, `7books`, `my-level`.

Типи величин у мові Python

Тип величини визначається *множиною припустимих значень* величини та *операціями*, які можна виконувати. Основними типами величин є *числа* й *рядки* (рис. 21.1).

```
Python 3.6.1 Shell
File Edit Format Run Options Window Help
>>> a = 12          # Ціле число
>>> b = 3.14        # Дробове число
>>> c = 'Hello!'    # Рядок
>>> print(a, b, c)
12 3.14 Hello!
```

Рис. 21.1

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



Цілими числами називають натуральні числа (1, 2, 3...), протилежні їм числа (-1, -2, -3...) і число 0.

Кількість учнів та учениць у класі, температуру на вулиці ми зазвичай указуємо за допомогою цілих чисел. Дробові числа потрібні, коли ми хочемо вказати частину чогось, наприклад 3.5 м, 1.25 грн.



Як роздільник між цілою та дробовою частиною в Python використовується крапка.

Величини рядкового типу зберігають фрагменти тексту.



Рядок — це взята в одинарні або подвійні лапки послідовність будь-яких символів: цифр, літер, розділових знаків та ін. Рядки належать до типу `str`.

Указівка присвоювання

Значення — характеристика змінної, яка може багаторазово змінюватися в процесі опрацювання інформації.

Щоб створити змінну, необхідно дати їй назву та присвоїти значення.

Загальний вигляд указівки присвоювання (рис. 21.2): $A = B$, де A — назва змінної, $=$ — оператор присвоювання, B — константа, змінна або вираз (докладніше — далі).

Схема виконання вказівки присвоювання: спочатку обчислюється значення виразу праворуч від знака «=», потім воно надається змінній, назву якої записано ліворуч (рис. 21.3).

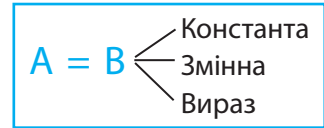


Рис. 21.2

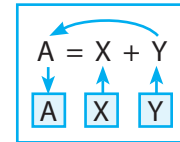


Рис. 21.3

3

Наведені вказівки присвоюють змінній `rabbits` значення 5, потім те саме значення присвоюють змінній `hats`:

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

Змінні можна використовувати у виразах.

4

Нехай $a = 10$. Тоді після виконання вказівки присвоювання $a = a + 5$ змінна a набуде значення 15.

При спробі виконати команду $a + 2 = b$ буде виведено повідомлення про синтаксичну помилку (рис. 21.4).

```
>>> a = 5
>>> a + 2 = b
SyntaxError: cannot assign to operator
>>> |
```

Рис. 21.4



Питання для самоперевірки

1. Назвіть основні характеристики величини.
2. Які послідовності символів неприпустимо використовувати як назви (ідентифікатори) й чому?
а) `suma`; б) `w1`; в) `result 1`; г) `(sum)`; ґ) `a-4`; д) `if`.
3. До яких типів належать величини: а) кількість учениць у класі; б) маса дині в кілограмах; в) прізвище учня.
4. Чому дорівнює значення `x` після виконання послідовності присвоювань: а) `y = 2; x = y`; б) `x = 10; x = x+3`?



Вправа 21

■ Виконати дії над величинами в середовищі Python.

1. Завантажте програму: Python → IDLE.

Після запрошення `>>>` запишіть команду створення змінної `a`: `>>> a = 10` і натисніть клавішу Enter. Інтерпретатор створить змінну `a` і надасть їй значення 10.

2. Щоб вивести значення змінної `a`, після запрошення `>>>` запишіть назву змінної і натисніть Enter.

3. Уведіть команду Збільшити значення змінної `a` на 3:

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

Виведіть значення змінної `a`.



4. Уведіть команду Збільшити значення змінної `a` вдвічі і виведіть значення змінної `a`.

5. Створіть змінну `s` рядкового типу: `>>> s = 'Вітаю!'`.



6. Уведіть команду Подвоїти значення змінної `s`:

```
>>> s = s*2
```

Виведіть значення змінної `s`. Поясніть результат.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102365).



§ 22. ЛІНІЙНІ АЛГОРИТМИ

У багатьох алгоритмах дії виконуються послідовно, одна за одною, тобто утворюють структуру *слідування*.

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

Загальна структура будь-якої програми є лінійною: програма має прийняти вхідні дані (*введення*), опрацювати їх і повернути результат (*виведення*) (рис. 22.1). Проте кожен із блоків може складатися з багатьох команд.

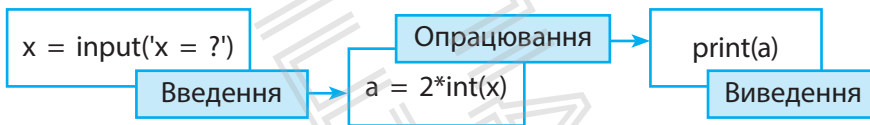


Рис. 22.1

Введення даних

Функція `input()` призначена для введення даних із клавіатури. Коли в програмі зустрічається функція `input()`, комп'ютер призупиняє виконання й очікує, поки користувач/користувачка введе дані й натисне `Enter`. Якщо в дужках записано рядок-підказку, що саме потрібно ввести, то вона з'явиться на екрані.



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

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

Значення, отримане від команди `input()`, Python сприймає як рядок, навіть якщо введено число.

2

При спробі додати 3 до значення змінної `a` виникне помилка: Python «не знає», як додати число до рядка.

```
>>> a = input('a = ?')
a = ? 5
>>> a+3
TypeError: must be str, not int
```

Перетворити введене значення на ціле число можна за допомогою функції `int()`, а на дробове — `float()`.

3

Тепер помилки немає:

<pre>>>> a = input('a = ?') a = ? 3 >>> int(a)+5 8</pre>	<pre>>>> x = float(input('x = ?')) x = ? 5.25 >>> x+3 8.25</pre>
--	--

У другому випадку функцію перетворення типу застосовано до значення, яке повернула функція `input()`, а вже результат перетворення присвоєно змінній `x`.

Виведення значень змінних

Команда `print()` виводить текст у вікно консолі:
`print(<список виведення>)`

!

Тут і далі кутовими дужками `<>` позначено назву того, що насправді записують на цьому місці.

4

Виведемо значення змінної `rabbits`:

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

У списку виведення можуть бути записані константи, змінні, вирази. Значення в списку відокремлюються комами.



Виведемо кілька результатів обчислень:

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

За допомогою команд `input()` і `print()` у програмі можна організувати діалог користувача з комп'ютером:

```
name = input("Як тебе звати?")
print(name, 'працює на
комп'ютері')
```

```
=====RESTART:
Як тебе звати? Іринка
Іринка працює на комп'ютері.
>>>
```

Коментарі в програмі

Коментар — це пояснення до програми, призначене для людини, а не для комп'ютера. Щоб комп'ютер відрізняв коментар від команди, на його початку ставиться знак `#`. Редактор IDLE виділяє коментарі червоним кольором, нагадуючи про те, що ці фрагменти коду комп'ютер знехтує.



У коментарі пояснено призначення наступної команди:

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



Питання для самоперевірки

- Чому виникає помилка під час виконання коду?


```
a = input("Уведіть значення a:")
b = a+10
```
- Для чого призначена команда `print()`?
- Що буде виведено у вікно консолі після виконання команд, якщо $a = 5$, $b = 7$?

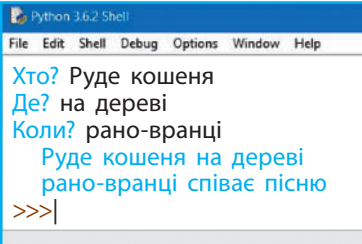

```
a) print("a+b = ", a+b); б) print("Площа дорівнює", a*b, "кв. м")
```
- Поясніть помилки в записах команд:


```
a) a = int(input('a = ?')) б) print("s = ", "v*t", "км")
```
- Яке призначення коментарів у програмі?

Вправа 22

- Скласти програму-гру Нісенітниця.

Алгоритм гри. Програма просить увести відповіді на запитання «Хто?», «Де?», «Коли?» і зберігає введені дані в трьох різних змінних. З них із додаванням жартівливої кінцівки формується речення та виводиться на екран (рис. 22.2).



```
Python 3.6.2 Shell
File Edit Shell Debug Options Window Help
Хто? Руде кошеня
Де? на дереві
Коли? рано-вранці
Руде кошеня на дереві
рано-вранці співає пісню
>>>|
```

Рис. 22.2

1. Завантажте програму: Python → 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 назвою Вправа 22. Запустіть програму на виконання. Дайте відповіді на питання програми. Проаналізуйте результат виконання програми.
- ⚙ 4. Додайте до програмного коду команду для введення відповіді на питання «Що робить?». Збережіть відповідь у змінній c4.
- ⚙ 5. Додайте значення змінної c4 до рядка phrase.
6. Збережіть програму й перевірте її роботу.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102366).



§ 23. МАТЕМАТИЧНІ ОПЕРАТОРИ МОВИ PYTHON

Майже в усіх програмах вам доведеться використовувати математичні операції. Як і в математиці, для обчислень записують арифметичні вирази з назв змінних, констант, знаків арифметичних операцій і круглих дужок.

З арифметичними діями додавання і множення, доступними в Python, ми познайомилися в попередніх параграфах.

Оператори — знаки, якими позначають арифметичні операції. Вам уже відомий оператор присвоєння, про інші операції та відповідні оператори дізнаєтеся далі.

Операнди — величини, над якими виконуються операції (рис. 23.1).



Рис. 23.1

Математичні оператори Python

Розгляньмо математичні операції та оператори:

Операція	Оператор	Приклад	Результат
Додавання	+	$a = 15+3$	$a = 18$
Віднімання	-	$b = 15-3$	$b = 12$
Множення	*	$c = 8*2$	$c = 16$
Ділення	/	$d = 7/2$	$d = 3.5$
Цілочисельне ділення (обчислення неповної частки від ділення)	//	$e = 7//2$	$e = 3$
Обчислення остачі від ділення	%	$f = 7\%2$	$f = 1$
Піднесення до степеня	**	$g = 7**2$	$g = 49$

У разі виконання кількох операцій в одному виразі враховується їх пріоритет (порядок виконання):

- 1) піднесення до степеня (**);
- 2) множення (*) і ділення (/), цілочисельне ділення (//), обчислення остачі (%);
- 3) додавання (+) і віднімання (-).

Операції з однаковим пріоритетом виконуються по черзі зліва направо. Змінити порядок виконання операцій можна за допомогою дужок.



Обчислимо значення змінних a і b:

$$a = 10 + 2 * 3 ** 2 \quad \# a = 28$$

$$b = (10 + 2) * 3 ** 2 \quad \# b = 108$$

Арифметичні вирази в тексті програми записують у рядок (рис. 23.2). При цьому не можна опускати знак операції множення.



Рис. 23.2



Присвоїмо змінній a значення виразу

$$\frac{2x - 5}{3 + x};$$

$$a = (2 * x - 5) / (3 + x)$$



Розв'яжімо задачу.

Задача. Відомо, що деяка подія відбулася x секунд тому. Подати цей проміжок часу в годинах, хвилинах і секундах.

```
x = int(input('Кількість секунд?'))
```

```
h = x // 3600
```

```
print(h, 'год')
```

```
m = (x % 3600) // 60
```

```
print(m, 'хв')
```

```
s = (x % 3600) % 60
```

```
print(s, 'с')
```

```
Кількість секунд? 10000
```

```
2 год
```

```
46 хв
```

```
40 с
```

Випадкові числа

Під час створення ігрових або тестових програм буває потрібно вибрати число з певного проміжку випадковим чином.

Щоб користуватися в програмі випадковими числами, слід за допомогою команди `import` завантажити модуль `random`:

```
from random import*
```

Тепер можна викликати функцію `randint(x1, x2)`, яка повертає ціле випадкове число в діапазоні від x_1 до x_2 .

4

Отримаємо ціле випадкове число в діапазоні від 1 до 10 (рис. 23.3).

```
Python 3.6.1 Shell
File Edit Format Run Options Window Help
>>> from random import*
>>> randint(1, 10)
8
>>> randint(1, 10)
6
```

Результат першого виклику

Результат другого виклику

Рис. 23.3

5

Для програми-тренажера «Таблиця множення» згенеруємо вираз із випадковими значеннями множників:

```
from random import*
x = randint(2, 9)
y = randint(2, 9)
print(x, '*', y, ' = ?')
vidp = int(input('Добуток: '))
```

```
2*3 = ?
Добуток: 6
>>> |
```



Питання для самоперевірки

1. Обчисліть значення виразів:

а) $7/2$;

в) $7\%2$;

г) $123\%10$;

б) $7//2$;

г) $123//100$;

д) $(123//10)\%10$.

2. Обчисліть значення, якого набуває змінна a в результаті виконання команди присвоєвання, якщо $b = 4$:
 - а) $a = 37 - b^{**}2$;
 - б) $a = 100 - 2 * b^{**}2$;
 - в) $a = (b + 12) / 2$;
 - г) $a = b + 12 / 2$.
3. Запишіть математичні вирази за правилами Python:
 - а) $\frac{x-5}{2x}$;
 - б) x^5 ;
 - в) $2x^2 - 3x + 5$.
4. Як отримати випадкове число в діапазоні від 1 до 100?
5. Наведіть приклади задач, для яких потрібні випадкові числа.
6. Напишіть програму для обчислення площі та периметра прямокутної кімнати, ширина якої a м, довжина b м.

Вправа 23

- Розв'язати задачу в середовищі Python.

Задача. У скарбничці є a копійок. Обчисліть, скільки гривень становить ця сума.

1. Завантажте програму: Python → IDLE. Виберіть команду File → New File, щоб відкрити вікно програми.
2. У вікні програми наберіть текст програми:


```
a = 250
a1 = a//100
a2 = a%100
print(a1, a2)
```
3. Для збереження файлу виберіть команду File → Save As. У вікні збереження файлу виберіть власну папку, уведіть назву файлу Вправа 23 і натисніть Save.
4. Для запуску програми на виконання виберіть команду Run → Run Module або натисніть F5. Перегляньте результат виконання програми. Запустіть програму кілька разів, змінюючи значення a .
5. Змініть список виведення команди print так, щоб результат виводився з поясненнями, наприклад: 2 грн 50 к.

6. Змініть програму так, щоб значення змінної a вводилося з клавіатури. Запустіть програму. Введіть значення $a = 185$. Збережіть програму, закрийте вікно IDLE.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102367).



ПРАКТИЧНА РОБОТА 7 СКЛАДАННЯ ЛІНІЙНИХ АЛГОРИТМІВ

Завдання: створити програму Калькулятор піци.
Програма має: запитати, скільки піц замовляється; запитати ціну піци, зазначену в меню; обчислити підсумкову вартість, ураховуючи знижку 10% на честь ювілею піцерії; вивести ціну покупки.

Обладнання: комп'ютер зі встановленим середовищем Python.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Завантажте програму: Python → IDLE і створіть нове вікно програми.
2. Запишіть команду для введення кількості піц:
`number = int(input('Скільки піц замовляєте?'))`
3. Запишіть команду для введення ціни однієї піци. Вартість однієї піци може бути дробовим числом, тому введене значення слід перетворити до дробового типу (float).
`cost = float(input('Скільки коштує одна піца?'))`

4. Запишіть команди для обчислення ціни покупки без урахування знижки:


```
total = number*cost
print('Ціна без знижки', total)
```
5. Запишіть команди для обчислення величини знижки та ціни покупки з урахуванням знижки.


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

 Виведіть значення змінних `discount` і `total`.
6. Збережіть файл із назвою `Pizza`. Запустіть програму на виконання, перевірте її для різних початкових даних.

```
Скільки піц замовляєте? 2
Скільки коштує одна піца? 10.50
Ціна без знижки 21.00
Знижка 2.10
Ціна зі знижкою 18.90
```

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

§ 24. «ЧЕРЕПАШАЧА» ГРАФІКА

У середовищі Python Черепашкою зветься виконавець, який переміщується екраном за алгоритмом, залишаючи (або не залишаючи) за собою намальований слід. Спочатку положення та напрямок руху Черепашки показує чорна стрілка, що повільно пересувається по екрану. Це дає змогу зрозуміти, як кожен рядок коду впливає на рух Черепашки.

Черепашка допоможе 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 Graphics (рис. 24.1).

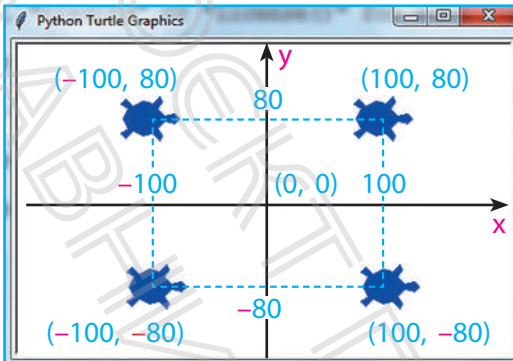


Рис. 24.1

Під час керування рухом Черепашки слід указувати її положення у вікні. Як це зробити? Проведемо вертикальну і горизонтальну лінії, які ділять вікно на рівні частини. Виміримо відстані від Черепашки до кожної з ліній. Якщо Черепашка лівіше від вертикальної або нижче від горизонтальної лінії, то перед відповідним числом ставимо знак мінус.

Отримані два числа, які задають положення Черепашки у вікні, називають її **координатами**. Першою вказують відстань від вертикальної осі, другою — від горизонтальної.

Команди Черепашки

Для використання Черепашки потрібен модуль turtle:
`from turtle import*`

Тепер у програмі можна використовувати команди керування Черепашкою. Спочатку вона перебуває в центрі вікна — точці з координатами (0; 0). Команда `goto(55, 80)` змушує її перейти в точку з координатами (55; 80).

Спочатку перо Черепашки опущене і під час пересування вона залишає слід.

Для керування пером призначені такі команди:

- `down()` — Опустити перо;
- `up()` — Підняти перо.

Ознайомімося детальніше з іншими командами та їхнім призначенням. Розгляньмо таблицю:

Команда	Призначення
<code>width(n)</code>	Установити ширину сліду Черепашки <code>n</code> пікселів
<code>forward(n)</code>	Проповзти вперед <code>n</code> кроків (пікселів)
<code>left(angle)</code>	Повернути ліворуч на <code>angle</code> градусів
<code>right(angle)</code>	Повернути праворуч на <code>angle</code> градусів
<code>circle(r)</code>	Намалювати коло радіуса <code>r</code> ліворуч від Черепашки
<code>circle(r, angle)</code>	Намалювати дугу радіуса <code>r</code> з градусною мірою <code>angle</code>
<code>dot(size, color)</code>	Намалювати круг діаметра <code>size</code> кольору <code>color</code>
<code>home()</code>	Перейти в центр вікна і повернутися вправо
<code>clear()</code>	Очистити графічне вікно



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. 24.2.

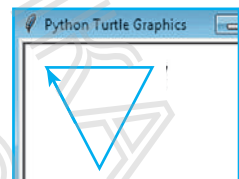


Рис. 24.2

2

Намалюємо коло радіуса 50 із центром у точці $(0; 0)$ і товщиною лінії 3. Команда `circle(50)` малює коло, розміщене ліворуч від Черепашки. Оскільки Черепашка спочатку дивиться вправо, то точка С (початок малювання) буде найнижчою точкою кола.

Оскільки радіус кола за умовою задачі дорівнює 50 пікселів, а координати точки $O(0; 0)$, то точка С міститься на 50 пікселів нижче від початку координат. Отже, її координати $(0; -50)$ (рис. 24.3).

Алгоритм малювання:

```
from turtle import*
width(3)
```

```
up() # Підняти перо, щоб при пересуванні не залишати сліду
goto(0, -50) # Пересунути черепашку в точку С
down() # Опустити перо, щоб почати залишати слід
circle(50) # Намалювати коло
```

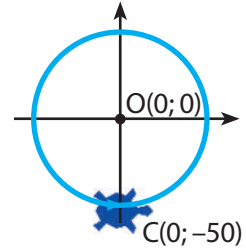


Рис. 24.3

3

Намалюємо смайлик (рис. 24.4).

```
from turtle import*
circle(50)
```

Малювання обличчя

```
up()
```

```
goto(-20, 60)
```

```
down()
```

```
dot(20)
```

Малювання лівого ока

```
up()
```

```
goto(20, 60)
```

```
down()
```

```
dot(20)
```

Малювання правого ока

```
up()
```

```
goto(20, 40)
```

```
left(90); down()
```

```
circle(20, -160)
```

Усмішка



Рис. 24.4

Як видно з передостаннього рядка прикладу 3, в Python можна записувати кілька команд в одному рядку, відокремлюючи їх крапкою з комою.



Питання для самоперевірки

1. Яку команду потрібно застосувати для завантаження команд роботи з Черепашкою?
2. Що означає команда `goto(-75; 40)`?
3. Поясніть призначення команд `down()`, `up()`.
4. Черепашка перебуває у точці (100; 100). Які координати вона матиме після виконання команд малювання?

а) <code>forward(50)</code>	б) <code>forward(50)</code>
<code>right(90)</code>	<code>forward(50)</code>
<code>forward(50)</code>	<code>left(90)</code>
	<code>forward(50)</code>
5. Де буде намальовано коло за командою `circle(60)`, якщо Черепашка «дивиться» вгору? вліво?



Вправа 24

- Скласти програму для побудови малюнка (рис. 24.5).

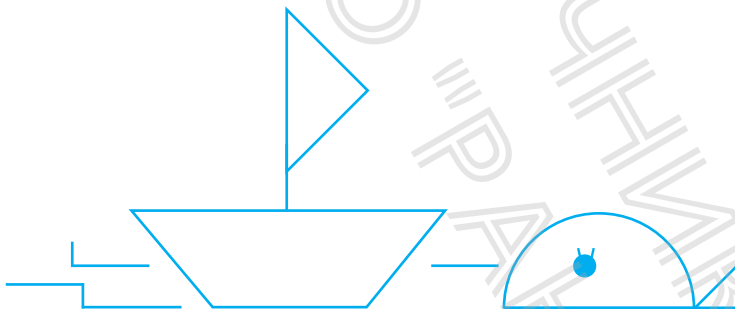


Рис. 24.5

1. Завантажте програму: Python → IDLE і створіть нове вікно програми. Завантажте модуль `turtle`.

Установіть ширину сліду Черепашки 3 пікселі.

Збережіть програму у файлі Вправа 24.

- Визначте координати кінців відрізків, які утворюють зображення човника. Наприклад, якщо почати малювання з точки $O(0; 0)$, точка A може мати координати $(-40; 50)$, а точка $B(120; 50)$ (рис. 24.6).

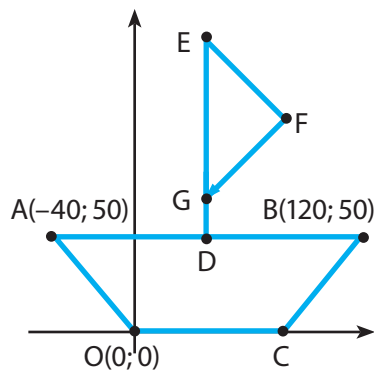


Рис. 24.6

- Запрограмуйте рух Черепашки від точки до точки:

```
goto(-40, 50)      # Побудова відрізка OA
goto(120, 50)     # Побудова відрізка AB
```

і т. д.

- Додайте зображення кита. Перейдіть у точку $(250, 0)$, поверніть Черепашку вліво на 90° (щоб дуга була ліворуч від неї) і намалюйте дугу з градусною мірою 180° .

```
up()
goto(250, 0)
down()
left(90)
circle(50, 180)      # Малювання спини кита
goto(270, 0)
goto(270, 20)       # Малювання хвоста кита
goto(250, 0)
```

- За допомогою команди `dot()` намалюйте око кита.
- Додайте кілька ліній, що зображують хвилі.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102368).



§ 25. АЛГОРИТМИ СТВОРЕННЯ ЗОБРАЖЕНЬ

Ви навчилися створювати малюнки чорного кольору. Модуль `turtle` дає змогу створювати кольорові малюнки, а також змінювати форму і розмір самої Черепашки.

Створення кольорового малюнка

Черепашка може залишати слід не лише чорного, але й інших кольорів. У відповідних командах назва кольору (англійською мовою) береться в лапки, наприклад `'red'`, `'yellow'`, `'green'` тощо.

Щоб переглянути весь список кольорів, відвідайте сайт www.tcl.tk/man/tcl8.4/TkCmd/colors.htm.

Команди для встановлення кольорів:

Команда	Призначення
<code>color(<колір>)</code>	Установити колір сліду Черепашки
<code>color(<колір1>, <колір2>)</code>	Установити для сліду <code>колір1</code> , а для заповнення замкненої фігури — <code>колір2</code>

Команди для зафарбовування замкнених фігур:

Команда	Призначення
<code>fillcolor(<колір>)</code>	Установити колір заповнення
<code>begin_fill()</code>	Почати стежити за Черепашкою для заповнення ділянки
<code>end_fill()</code>	Заповнити кольором <code>колір2</code> ділянку, пройдену Черепашкою, починаючи від <code>begin_fill()</code>



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

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

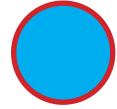


Рис. 25.1

Результат виконання команд наведено на рис. 25.1.

Зміна вигляду Черепашки

Щоб змінити форму Черепашки, слід скористатися командою `shape(<форма>)`, де `<форма>` — назва форми, узятя в лапки. У модулі `turtle` існують такі форми Черепашки:

Команда	arrow	circle	square	triangle	turtle
Форма					

Розмір Черепашки можна змінити командою `shapeseize(n)`, де `n` — коефіцієнт змінення розміру Черепашки. Так, команда `shapeseize(2)` збільшить її вдвічі.

Черепашка може залишати відбитки за допомогою команди `stamp()`: де вона була, залишиться її зображення.



Накажемо Черепашці креслити лінію синього кольору, залишаючи відбитки (рис. 25.2).

```
from turtle import*
shape('turtle')
color('blue' )
stamp()
forward(50)
shapeseize(2)
stamp()
forward(100)
```



Рис. 25.2

Змінювати можна не тільки вигляд Черепашки, але й деякі властивості графічного вікна:

Команда	Призначення
<code>bgcolor(<колір>)</code>	Задати колір фону графічного вікна
<code>setup(<ширина>, <висота>)</code>	Установити розміри вікна в пікселях

3

Задамо розміри вікна 200×100 пікселів, колір тла вікна — помаранчевий (рис. 25.3).

```
from turtle import*
setup(200, 100)
bgcolor('orange')
```

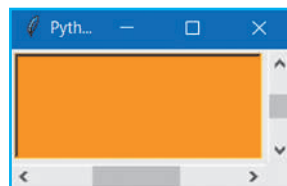


Рис. 25.3

Додавання написів

У вікні для графіки модуля turtle можна також виводити написи; використовується команда `write` — вивести текст:

```
write(text, align, font),
```

де `text` — текст (у лапках), який буде відображатися у вікні для графіки turtle Python;

`align` — параметр, що відповідає за розташування тексту щодо Черепашки; набуває значення 'left', 'right', 'center';

`font` — параметри шрифту.

4

Додамо текст у графічне вікно, застосовуючи різні значення параметрів тексту (рис. 25.4).

```
write('Це квадрат!', align = 'left',
      font = 'Arial 16')
```

```
write('Це квадрат!', align = 'right',
      font = 'Arial 16')
```

Рис. 25.4



Питання для самоперевірки

1. Запишіть для Черепашки команду встановити: а) зелений колір сліду; б) жовтий колір тла вікна; в) синій колір заповнення замкненої фігури.
2. Як побудувати фігуру, заповнену заданим кольором?
3. Складіть програму для малювання прапорів (рис. 25.5).



Рис. 25.5

4. Запишіть фрагмент програми для побудови трикутника з вершинами в точках (100, 100), (150, 100), (80, 70); колір фону — сірий; колір ліній — червоний.



Вправа 25

- Розфарбувати зображення човника і кита (рис. 25.6).

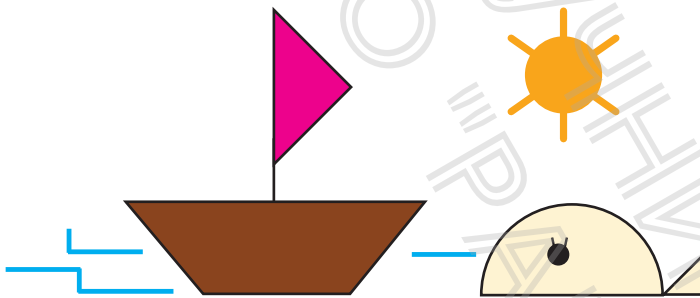


Рис. 25.6

1. Завантажте програму: Python → IDLE і відкрийте файл Вправа 24.
2. Додайте заповнення до корпусу човника.


```
fillcolor('brown')
begin_fill()
goto(-40, 50)
goto(120, 50)
goto(80, 0)
goto(0, 0)
end_fill()
```



Додайте заповнення до вітрила човника.

- 3. Змініть колір хвиль на синій. Зафарбуйте кита.
- 4. Намалуйте сонце.

```
color('orange', 'orange')
up()
goto(200, 200)
down()
begin_fill()
circle(30)
end_fill()
```

- 5. Додайте промені. Запустіть програму на виконання.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102369).



ПРАКТИЧНА РОБОТА 8 СТВОРЕННЯ ЗОБРАЖЕНЬ ЗА АЛГОРИТМАМИ

Завдання: скласти програму малювання за зразком.

Обладнання: комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Завантажте програму: Python → IDLE і створіть нове вікно програми.
2. Збережіть файл у папці з назвою Практична робота 8.
3. Розгляньте рисунок. Поміркуйте над алгоритмом малювання одного кільця: задати колір, яким будемо малювати; підняти перо; перейти на точку початку малювання кільця; опустити перо; малювати коло радіуса 45.
4. Завантажте модуль turtle, задайте ширину лінії 3 пікселі.
5. Запишіть програму малювання синього кола:


```
color('blue')      # Встановлюємо синій колір
up()              # Піднімаємо перо
goto(-110, -25)   # Переходимо в потрібну точку
down()           # Опускаємо перо
circle(45)        # Малюємо коло радіуса 45
```
6. Додайте команди підняти перо і перейти в точку з координатами (0, -25).
7. Запишіть код для малювання кола чорного кольору.
8. Додайте код для малювання кіл: третє — 'red', (110, -25); четверте — 'yellow', (-55, -75); п'яте — 'green', (55, -75).
9. Переведіть Черепашку в точку з координатами (0, 80).
10. Установіть червоний колір малювання.
11. Над олімпійською емблемою введіть власне прізвище та ім'я.
12. Запустіть програму, проаналізуйте результат її виконання у вікні Python Turtle Graphics.
Закінчіть роботу.



Зробіть висновок: як складати та виконувати лінійні алгоритми для створення зображень.

§ 26. ЛОГІЧНІ ВИРАЗИ

Чим відрізняються такі два твердження: « $2 + 3 = 5$ » і « $2 + 2 = 5$ »? Ви одразу ж скажете: перше з них — істинне, а друге — хибне. Величини, які можуть набувати тільки двох значень — істина (так) або хибність (ні) — називають **логічними величинами**.

Логічний тип даних

Окрім числових 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).

Якщо змінній присвоїти значення `True`, це буде змінна типу `bool`:

```
>>> a = True
>>> print(a)
True
```

Інша назва логічного типу даних — булів, або булівський, — на честь англійського математика Джорджа Буля (рис. 26.1), засновника *математичної логіки* — розділу математики, що розглядає застосування математичних методів для розв'язування логічних задач. Тепер ідеї Дж. Буля використовуються у всіх сучасних цифрових пристроях.

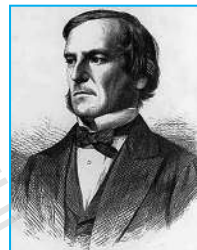


Рис. 26.1

Логічні вирази



Логічними є вирази, значенням яких є `True` або `False`.

Умови в програмі записують у вигляді логічних виразів.

Проста умова (простий логічний вираз) утворюється за допомогою операцій порівняння:

Оператор	Значення	Логічний вираз	Результат
==	Дорівнює	8 == 9	False
>	Більше	8>9	False
<	Менше	8<9	True
>=	Більше або дорівнює	5 >= 5	True
<=	Менше або дорівнює	5 <= 2	False
!=	Не дорівнює	2 != 5	True



Проаналізуємо результати обчислення логічних виразів у вікні консолі:

```
>>> books = 10
>>> books == 5      # Перевірка, чи дорівнює books 5
False
>>> books < 10     # Перевірка, чи є books меншим за 10
False
>>> books >= 5     # Перевірка, чи books більше або
                   # дорівнює 5
True
>>> books != 10    # Перевірка, чи не дорівнює books 10
False
```

Значення логічного виразу можна зберегти у змінній.



Обчислимо значення логічних виразів:

```
x = 5
y = 2
a = x <= y          # a = False
b = x-4.5 < y*2     # b = True
```

Логічні операції

Бувають ситуації, коли одночасно необхідно перевірити виконання кількох умов. У такому разі до простих умов застосовують логічні операції.

Складена умова — це кілька простих умов, з'єднаних логічними операціями: логічне І (оператор and), логічне АБО (оператор or), логічне заперечення НЕ (оператор not).

Нехай A і B — два логічні значення. Складена умова A and B істинна, тільки якщо й A , і B істинні.

Складена умова A or B хибна, якщо й A , і B хибні.

3

Розгляньмо приклади складених умов:

- $\text{not } a \leq 3$ — рівнозначне виразу $a > 3$;
- $\text{age} \geq 10$ and $\text{age} \leq 18$ — істинне, якщо значення age належить проміжку від 10 до 18 включно;
- $\text{age} < 10$ or $\text{age} > 18$ — істинне для всіх значень age , які не належать проміжку від 10 до 18.

4

Проаналізуємо результати обчислення логічних виразів у вікні консолі.

```
>>> books = 10
>>> 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
```

5

Визначимо, чи належить точка з координатою x відріжку $[4; 15]$ (рис. 26.2). Точка належить відріжку, якщо справджується нерівність $4 \leq x \leq 15$.

У програмі таку подвійну нерівність записують як складену умову: $x \geq 4$ and $x \leq 15$.



Рис. 26.2



Обчислимо значення логічних виразів, якщо $x = 1$; $y = 2$; $z = 3$.

$a = (x < y) \text{ and } (y < z)$

$a = \text{True}$, бо обидві частини істинні

$a = (x > y) \text{ or } (y > z)$

$a = \text{False}$, бо обидві частини хибні

$a = \text{not}(x > y)$

$a = \text{True}$, бо вираз у дужках хибний



Біля витоків української школи програмування стояла Катерина Ющенко, докторка фізико-математичних наук, авторка однієї з перших у світі мов програмування високого рівня, в якій застосовуються операції над адресами об'єктів у пам'яті комп'ютера (рис. 26.3).



Рис. 26.3



Питання для самоперевірки

1. Яких значень може набувати змінна логічного типу?
2. Назвіть операції над величинами логічного типу.
3. Запишіть мовою програмування прості умови:
 - а) x більше 10;
 - б) x не більше y ;
 - в) x — парне число;
 - г) x дорівнює y .
4. Запишіть мовою програмування складені умови:
 - а) $2 < x < 10$;
 - б) x не належить проміжку (2, 10).
5. Обчисліть значення логічних виразів:
 - а) $(A \leq B) \text{ and } (A == B-1)$, якщо $A = 2$, $B = 4$;
 - б) $(A \leq B) \text{ or } (A == B-1)$, якщо $A = 2$, $B = 4$.



Вправа 26

- Обчислити логічні вирази.
Написати програму, в якій залежно від значень змінних a та b обчислюються значення логічних виразів, а результат обчислення виводиться у вікно консолі.

1. Завантажте програму: Python → IDLE і створіть файл Логічні операції. Завантажте модуль random для генерації випадкових чисел.
2. Надайте змінній a випадкове значення з проміжку від 1 до 20, виведіть значення a у вікно консолі:

```
a = randint(1, 20)
print("a = ", a)
```

Надайте змінній b випадкове значення з проміжку від 1 до 20, виведіть значення b у вікно консолі.

3. Змінній x присвойте значення логічного виразу $a > b$, виведіть значення x :

```
x = a > b
print("a > b = ", x)
```

4. Змінній y присвойте значення логічного виразу $b \leq 10$, виведіть значення y :

```
y = b <= 10
print("b <= 10 = ", y)
```

5. Виведіть у консоль значення складеної умови x and y :

```
print("x and y = ", x and y)
```

6. Виведіть у консоль значення логічних виразів x or y , not x and y ; not(x and y).

Приклад виконання програми наведено на рис. 26.4.

Запустіть програму кілька разів, проаналізуйте результати.

```
a = 18
b = 12
a > b = True
b <= 10 = False
x and y = False
x or y = True
not x and y = False
not(x and y) = True
>>>
```

Рис. 26.4



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102370).



§ 27. АЛГОРИТМИ З РОЗГАЛУЖЕННЯМИ. КОМАНДА IF

Ви вже знаєте, що однією з базових алгоритмічних структур є розгалуження. Для організації розгалужень у програмах, написаних мовою Python, призначені команди перевірки умови if та if...else.

Команди перевірки умови if

Команда if відповідає алгоритмічній структурі **неповне розгалуження** (рис. 27.1).

Синтаксис команди:

```
if <умова>:
```

```
    ___ <команди>
```

де ___ — обов'язковий відступ від лівого краю.

Після запису умови слід поставити двокрапку.

У наступних рядках з обов'язковим однаковим відступом від лівого краю записуються команди, які буде виконано, якщо умова *істинна*. Однакові відступи зручно робити, натискаючи клавішу Tab.

Якщо ж умова *хибна*, то ці команди буде пропущено і керування перейде до команди, що йде після команди if.

Розгляньмо приклади використання неповного розгалуження у програмах.



Після виконання цієї програми, в разі правильної відповіді на питання, отримаємо результат **Правильно**.

```
result = int(input("Скільки буде 28+4?"))
```

```
if result == 32:
```

```
    print("Правильно!")
```

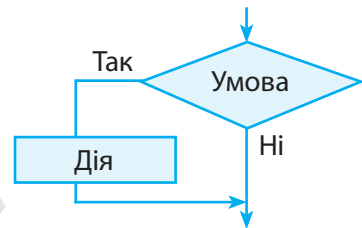


Рис. 27.1

2

Вітання виводиться, тільки якщо введено літеру «т».

```
answer = input('Сьогодні твій день народження? (т/н)')
```

```
if answer == 'т':
```

```
    print('Вітаю!')
```

```
print('Гарного дня!')
```

Приклад виконання команди наведено на рис. 27.2.

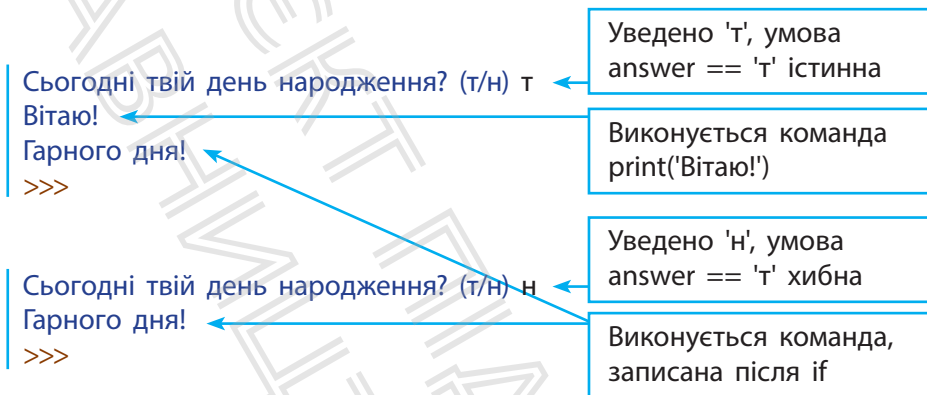


Рис. 27.2

3

Збільшимо значення змінної `a` на одиницю, якщо її поточне значення менше 5:

```
if a < 5:
```

```
    a = a + 1
```

4

Визначимо, чи є число `n` кратним 2, 3 або 5 (рис. 27.3).

```
n = int(input('n = ? '))
```

```
if n % 2 == 0: print(n, 'парне число')
```

```
if n % 3 == 0: print(n, 'кратне 3')
```

```
if n % 5 == 0: print(n, 'кратне 5')
```

```
n = ? 24
24 парне число
24 кратне 3
```

Рис. 27.3

У програмі використано послідовні розгалуження, тобто по чергово перевіряються кілька умов. Перевірка наступної умови не залежить від результату перевірки попередньої.

Питання для самоперевірки

- Як записується й виконується команда `if`?
- Яких значень набудуть змінні a і b після перевірки умов для початкових значень $a = 3$, $b = 10$?
 - `if a%2 == 0:` `if a != b:` `if a < b:`
 $a = a//2$ $a = b$ $b = b - a$
- Виконання яких команд не викличе повідомлення про помилку?
 - `if a > b:` `if a%3:` `if a < 10:` `if b > 10:`
`print ("a>b")` $a = a//3$ $a = a + 5$ $b = b - 10$
- Запишіть команди, які реалізують такі дії:
 - якщо a більше b , замінити a нулем;
 - якщо ціле число a парне, поділити його на 2.
- Складіть програму, яка запитує значення числа n , $1 < n < 10\,000$, і повідомляє кількість цифр у числі n .

Вправа 27

- Скласти програму для малювання багатокутника, кількість кутів n якого вводить користувач/користувачка.
- Завантажте програму: Python → IDLE і створіть файл Вправа 27. Завантажте модуль для роботи з графікою.
 - Запишіть команду для введення значення n .
 - Запишіть команду неповного розгалуження з перевіркою умови $n == 3$. Якщо умова істинна, виконуються команди малювання трикутника.


```
if n == 3:
    forward(50)
    right(120)
    forward(50)
    right(120)
    forward(50)
    right(120)
```



4. Якщо $n = 4$, слід намалювати квадрат. Запишіть команду для перевірки умови $n == 4$ і команди малювання квадрата.
5. Запрограмуйте побудову п'ятикутника, якщо введено 5.
6. Запишіть команду розгалуження, яка виконує таку дію: якщо $n < 3$ або $n > 5$, вивести повідомлення. Такої фігури не передбачено. Перевірте роботу програми.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102371).



§ 28. АЛГОРИТМИ З РОЗГАЛУЖЕННЯМИ. КОМАНДА IF...ELSE

Для виконання однієї дії, якщо деяка умова істинна, та іншої дії, якщо ця умова хибна, використовується алгоритмічна структура **повне розгалуження** (див. рис. 19.5). Цій структурі відповідає команда перевірки умови `if...else`.

Синтаксис команди:

```
if <умова>:  
    <команди 1>  
else:  
    <команди 2>
```

Якщо умова істинна (True), то буде виконано блок дій команди 1; якщо умова хибна (False) — виконуються дії блоку команди 2, записані після службового слова `else`.



Якщо вводиться літера «т», виводиться Вітаю!, інакше — Щасти тобі!. Далі в обох випадках — Гарного дня!

```
answer = input('Сьогодні твій день народження? (т/н)')
if answer == 'т':
    print('Вітаю!')
else:
    print('Щасти тобі!')
print('Гарного дня!')
```

Команди, вкладені в гілки `if` і `else`, пишуть з однаковим відступом. Одну команду можна записати після двокрапки.



Команда `if...else` реалізує дію: якщо $a > b$, то змінні міняються значеннями так, щоб виявилось $a < b$ (рис. 28.1).

```
if a > b:
    c = a
    a = b
    b = c
    print('a і b помінялися значеннями')
else: print('обмін значеннями не потрібен')
print('a = ', a, ', b = ', b)
```

```
a = 5 b = 2
a і b помінялися значеннями
a = 2, b = 5
>>>
a = 3 b = 8
обмін значеннями не потрібен
a = 3, b = 8
>>>
```



Рис. 28.1



Чи можна скласти трикутник із відрізків a , b , c ?

if $(a < b + c)$ and $(b < a + c)$ and $(c < a + b)$:

```
print('Трикутник з такими сторонами існує')
else: print('Трикутника з такими сторонами не існує')
```

! У програмі можуть послідовно виконуватися команди повного і неповного розгалуження.

4

Складемо програму пошуку найбільшого з трьох чисел a , b , c . Блок-схему алгоритму наведено на рис. 28.2.

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

```
a = ? 3
b = ? 12
c = ? 8
m = 12
>>>
```

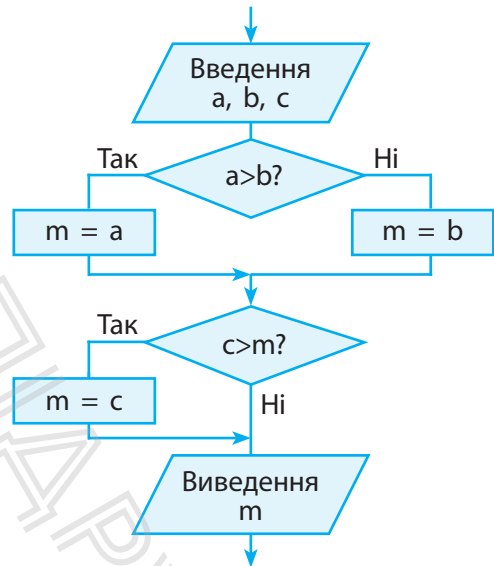


Рис. 28.2

Отже, команди перевірки умов `if` та `if...else` є важливими інструментами програмування, які стануть вам у пригоді під час створення тестових та ігрових програм.



Питання для самоперевірки

- Як виконується команда `if...else`?
- Поясніть, як виконано обмін значеннями в прикладі 2.
- Яких значень набудуть змінні a і b після виконання таких команд, якщо спочатку $a = 3$; $b = 5$?
 - `if a>b: a = 0`
`else: b = 0`
 - `if a>b: a = a+10`
`else: b = b+10`
 - `if a<b: a = 2*a`
`else: b = b*a`

4. Запишіть команди розгалуження, які реалізують дії:
 - а) якщо ціле число a парне, поділити його на 2, інакше — збільшити a на 1;
 - б) значення більшого з чисел a і b замінити нулем.
5. Складіть програму: виводиться вираз на додавання двох випадкових чисел і перевіряється введена відповідь.

Вправа 28

- Скласти програму для визначення, чи достатньо користувачеві років, щоб кермувати автомобілем.

Алгоритм: увести вік особи; якщо вік перевищує 18 років, то повідомити, що кермувати автомобілем можна, інакше — повідомити, через який час буде можна.

1. Завантажте програму: Python → 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 кермувати авто')
else:
    r = 18-age
    print('Доведеться почекати', r, 'років')
```
3. Збережіть файл у власній папці з назвою Вправа 28.
4. Запустіть програму, проаналізуйте результат. Випробуйте програму для різних значень віку.
- ⚙️ 5. Змініть програмний код так, щоб у разі ствердної відповіді виводилося повідомлення, скільки років людина вже могла б кермувати автомобілем.
6. Закінчіть роботу, заклавши вікна IDLE.



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102372).





ПРАКТИЧНА РОБОТА 9 СКЛАДАННЯ ТА ВИКОНАННЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМИ

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

Обладнання: комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- Проаналізуйте алгоритм розв'язування рівняння, поданий словесно:
 - якщо $a = 0$, $b = c$, то коренем є будь-яке число;
 - якщо $a = 0$, $b \neq c$, то коренів немає;
 - якщо $a \neq 0$, то $x = (c - b) / a$.
- Завантажте програму: Python \rightarrow IDLE і створіть нове вікно програми. Збережіть файл у власній папці з назвою Практична робота 9.
- Складіть програму на основі блок-схеми (рис. 1, с. 165).
Програма розв'язування рівняння

```
a = int(input('a = ?'))  
b = int(input('b = ?'))  
c = int(input('c = ?'))  
if a != 0:  
    x = (c-b)/a  
    print('x = ', x)  
if a == 0 and b == c: print('Безліч коренів')  
if a == 0 and b != c: print('Коренів немає')
```
- Запустіть програму. Перевірте її роботу для тестового набору значень (рис. 2, с. 165).

5. Випробуйте програму для таких наборів даних:

- 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 вікна IDLE.

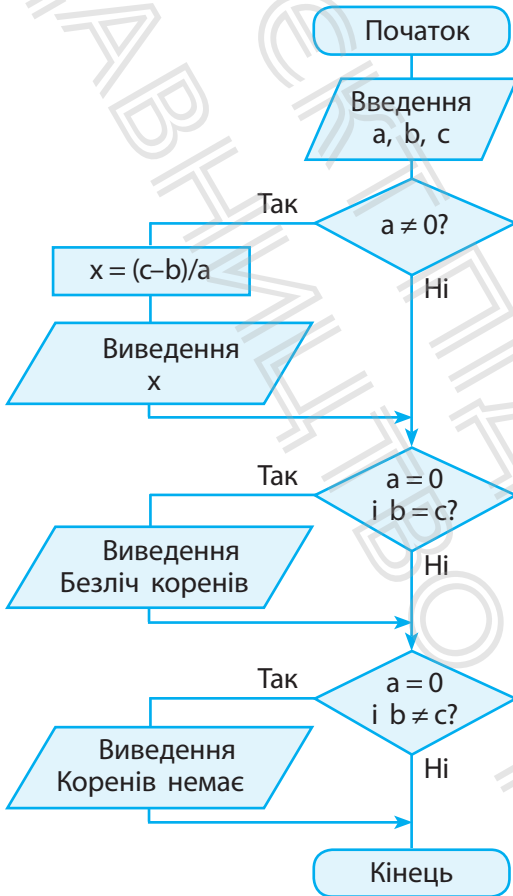


Рис. 1



```

=== RESTART:
a = ? 5
b = ? 2
c = ? 17
x = 3.0
    
```

Рис. 2

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

§ 29. АЛГОРИТМИ З ПОВТОРЕННЯМИ. ЦИКЛ ІЗ ПАРАМЕТРОМ

Пригадаймо, що повторення (цикл) — це алгоритмічна структура, за допомогою якої та сама послідовність дій виконується кілька разів. І розглянемо правила запису алгоритмів із повторенням. У мові Python для запису повторень (див. с. 118) призначена команда циклу з параметром `for`.

Повторення команд

У циклі з параметром блок команд (тіло циклу) повторюється *задану кількість разів*.

Синтаксис команди циклу `for`:

```
for i in <послідовність>:  
    ___ <тіло циклу>
```

Насамперед ми вказуємо ключове слово `for`, після цього — назву змінної (*i*), яка буде параметром циклу. Ключове слово `in` наказує почергово надати змінній *i* кожне значення із зазначеної далі послідовності.

Рядок, що починається з `for`, називають **заголовком циклу**, а повторювані в циклі команди — **тілом циклу**. Команди тіла циклу записують з однаковим відступом. Кожне виконання тіла циклу називають **ітерацією**.



Проаналізуємо зміну значень параметра *i* в циклі `for`.

```
for i in range(4):  
    print(i)
```

Буде виведено числа 0, 1, 2, 3.

Отже, `range(n)` — це послідовність цілих чисел від 0 до $n-1$, які почергово присвоюються змінній *i*. Її можна використовувати у виразах всередині циклу.

2

У прикладі 1 (§ 24) для малювання трикутника Черепашка тричі виконує команди `forward(100)` і `right(120)`.

Замість цього можна повторити ці команди в циклі:

```
for i in range(3):
```

```
    forward(100)
```

```
    right(120)
```

Результат виконання циклу подано на рис. 24.2 (с. 142).

Параметр циклу `for`

Розгляньмо ще деякі способи задання послідовності значень для змінної циклу.

3

Значеннями параметра по чергово стають літери зазначеного рядка (рис. 29.1).

```
a = 'Рядок'
```

```
for x in a:
```

```
    print('->', x)
```

```
-> Р
-> я
-> Д
-> О
-> К
>>>
```

Рис. 29.1

Послідовність значень можна задати як список.



Список — це набір значень, розділених комами, взятий у квадратні дужки.

У списках можна зберігати значення будь-якого типу — числа, рядки тощо. Можна сказати, що список — це контейнер для збереження набору значень.

4

Лічильник по чергово набуває заданих у списку значень (рис. 29.2).

```
b = ['Пн', 'Вт', 'Ср', 'Чт', 'Пт']
```

```
for x in b:
```

```
    print('->', x)
```

```
-> Пн
-> Вт
-> Ср
-> Чт
-> Пт
>>>
```

Рис. 29.2

Отримати послідовність значень від n_1 до n_2 із кроком k можна, викликавши `range()` із такими параметрами: `range(n1, n2, k)`

Відлік завершується на 1 раніше від значення n_2 . Якщо крок дорівнює 1, значення k можна не вказувати: `range(n1, n2)`.

5

Виклик `range(2, 11, 3)` надає змінній x значення від 2 до $(11-1)$ із кроком 3, тому, хоча $8 + 3 = 11$, значення 11 до послідовності значень не входить (рис. 29.3).

```
>>> for x in range(2, 11, 3):
      print('->', x)
-> 2
-> 5
-> 8
>>>
```

Рис. 29.3

Розгляньмо приклади розв'язування задач із циклом `for`.

6

Виведемо квадрати непарних чисел із проміжку від 11 до 19 (рис. 29.4).

```
for i in range(11, 20, 2):
    print(i, '**2 = ', i**2)
```

```
11**2 = 121
13**2 = 169
15**2 = 225
17**2 = 289
19**2 = 361
```

Рис. 29.4

7

Уведемо з клавіатури 5 цілих чисел і визначимо, скільки серед них було парних. Створимо змінну k , у якій зберігатимемо кількість парних чисел. Значення змінної a на кожній ітерації вводитимемо з клавіатури (рис. 29.5).

```
0? 10
1? 3
2? 12
3? 7
4? 25
k = 2
```

Рис. 29.5

```
k = 0          # Початкове значення лічильника парних чисел
for i in range(5):
    a = int(input(str(i)+'?')) # Уведення чергового числа
    if a%2 == 0: k = k+1      # Якщо число парне, лічильник k
    print("k = ", k)         # збільшується на 1
```

Якщо слід вести зворотний відлік (у бік зменшення значень), тобто $n_1 > n_2$, то укажіть значення кроку k зі знаком «-».

На рис. 29.6 наведено зміну значень лічильника з від'ємним кроком.

Отже, цикл із параметром є потужним інструментом програмування, він позбавляє програміста необхідності писати однакові блоки коду кілька разів.

```
>>> for x in range(5, 0, -1):
      print(i)
      print('Старт!')
```

5
4
3
2
1
Старт!

Рис. 29.6

Віктор Михайлович Глушков — засновник Інституту кібернетики Національної академії наук України, активний учасник світового процесу становлення та розвитку обчислювальної техніки, інформатики, кібернетики (рис. 29.7). Під його керівництвом розроблено перший в Україні мінікомп'ютер «МИР-1».



Рис. 29.7

Питання для самоперевірки

- Поясніть правила виконання циклу for.
- Яких значень набуває змінна x у ході виконання циклу з таким заголовком: for x in range(5)?
- Яких значень набуває змінна x у ході виконання циклу з таким заголовком: for x in range(0, 10, 2)?
- Що буде надруковано в ході виконання циклу?

а) s = 'цикл'	б) b = [1, 3, 9, 8]	в) for x in range(2, 8, 2):
for x in s:	for x in b:	print(x*2)
print(x)	print(x)	
- Чому дорівнює s після виконання циклу?

а) s = 0	б) s = 0
for a in range(5, 7):	for a in range(5, 11, 2):
s = s + 1	s = s + a

Вправа 29

■ Створити рисунки, скориставшись циклом for.

1. Завантажте програму: Python → IDLE і створіть нове вікно програми. Збережіть файл із назвою Вправа 29.
2. Запишіть програму для побудови малюнка (рис. 29.8).

```
from turtle import*
color('blue')
for i in range(1, 100, 2):
    up()
    goto(i*2, 0)
    down()
    circle(i)
```



Рис. 29.8

3. Запустіть програму. Скільки кіл побудовано?
4. Поекспериментуйте з наведеним програмним кодом. Змініть параметри range() так: range(1, 100, 6). Запустіть програму. Скільки кіл побудовано?

5. Поставте на початку рядка goto(i*2, 0) символ #. Запустіть програму. Як змінився малюнок?

6. Змініть команду goto(i*2, 0), щоб малюнок мав такий вигляд, як наведено на рис. 29.9.

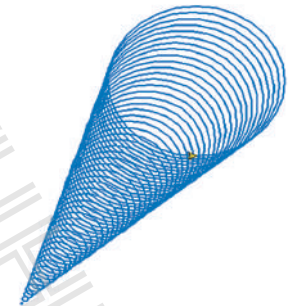


Рис. 29.9



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102373).

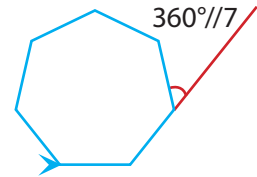


§ 30. ВИКОРИСТАННЯ ЦИКЛІВ ДЛЯ ПОБУДОВИ ЗОБРАЖЕНЬ

Повторення команд малювання з використанням циклу for допоможе створювати цікаві малюнки. Переконаймося?

Як намалювати багатокутник

Програмуючи малювання три-, чотири- і п'ятикутника (див. вправу 27), ви могли помітити, що в кожному випадку Черепашка після малювання сторони поверталася на кут $360/n$ градусів, де n — число кутів (рис. 30.1). Це дозволяє скласти одну програму для малювання будь-якого багатокутника.



Скільки кутів? 7
Довжина сторони? 50
>>>

Рис. 30.1



Складемо програму для малювання n -кутників.

```
from turtle import*
n = int(input('Скільки кутів? '))
x = int(input('Довжина сторони? '))
for i in range(n):
    forward(x)
    left(360/n)
```



Черепашка малює візерунки

Розгляньмо алгоритми малювання фігур на прикладі. Проаналізуємо виконання програми:

```
for x in range(1, 100, 2):
    forward(x)
    left(90)
```

У подальшій роботі спиратимемося на рис. 30.2.

Під час кожної ітерації Черепашка малює лінію довжиною x пікселів і повертається ліворуч на 90° (рис. 30.2, а).

Отже, за 4 ітерації циклу Черепашка повертається до початкового напрямку ($4 \cdot 90^\circ = 360^\circ$). Програма малює спіраль, оскільки кожен відрізок на 2 пікселі довший за попередній: перший відрізок має довжину 1 піксель, другий — 3, останній (п'ятдесятий) — 99.

Поворот на 90° створює квадратну спіраль. Змінимо кут повороту на 2° : `left(92)`. Зміна лише одного значення викликає зміни у вигляді спіралі (рис. 30.2, б).

Змінимо команду `forward(x)` на `circle(x)`, яка малює коло радіуса 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 ліворуч на 90° (рис. 30.2, в). Щоб намалювати n наборів кіл, необхідно повертати Черепашку на $360/n$ градусів.

Змінимо програму так, щоб малювалося 6 наборів кіл ($360/6 = 60$) (рис. 30.2, г):

```
for x in range(1, 100, 2):
    circle(x)
    left(60)
```

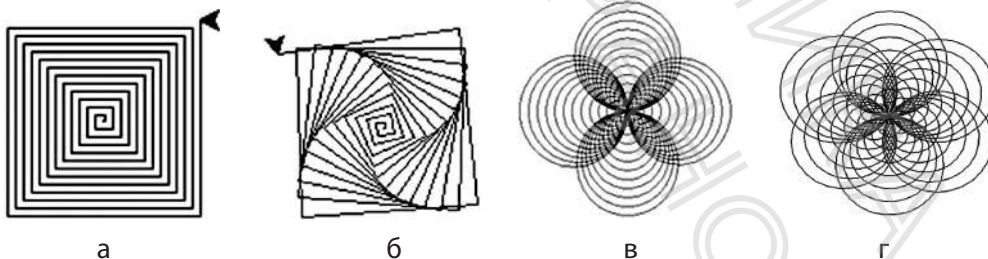


Рис. 30.2

Зміна кольорів у циклі

З'ясуймо, як зробити спіралі різнокольоровими. Для цього нам знадобиться список кольорів, а не один колір.

Створімо змінну `colors`, у яку помістимо список із назв чотирьох кольорів:

```
colors = ['red', 'yellow', 'green', 'blue']
```

! Акцентуймо увагу на тому, що звернутися до значення зі списку з номером `n` можна як до `colors[n]`. Наприклад, `colors[0] = 'red'`, `colors[3] = 'blue'`.

Але як змусити Черепашку під час кожної ітерації, коли `x` змінюється від 1 до 99, обирати одне з 4 значень списку? Використаймо як номер кольору в списку `colors` остачу від ділення `x` на 4 — `x%4`: адже вона може дорівнювати тільки 0, 1, 2, 3, що відповідає номерам кольорів у списку.

Повернімося до програми малювання квадратної спіралі, але крок циклу зробімо таким, що дорівнює 1 (рис. 30.3):

```
colors = ['red', 'yellow', 'green', 'blue']
for x in range(100):
    color(colors[x%4])
    forward(x)
    left(90)
```

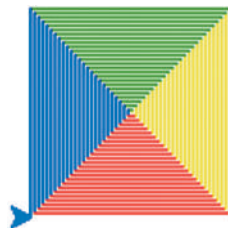


Рис. 30.3

Таким чином, вносячи незначні зміни до команд у тілі циклу `for`, ви можете створювати яскраві зображення.

2

Збільшимо кут повороту на 2° і зробимо товщину лінії залежною від параметра циклу (рис. 30.4):

```
color(colors[x%4])
forward(x)
left(360/4+2)
width(x*4/100)
```

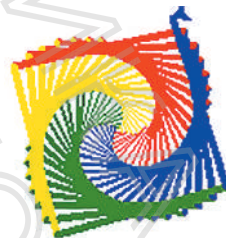


Рис. 30.4



Питання для самоперевірки

1. Поясніть алгоритм зміни кольору лінії за кожної ітерації циклу (див. с. 173).
2. Яку відстань пройде Черепашка під час виконання циклу?
 - а) `for x in range(5): forward(x)`
 - б) `for x in range(2, 10, 2): forward(10)`
3. Визначте градусну міру кута, на який повернеться Черепашка в ході виконання циклу.
 - а) `for x in range(4): left(20)`
 - б) `for x in range(5, 9): left(10)`
4. Створіть список `cls`, елементами якого будуть назви червоного, жовтого, зеленого кольорів.
5. Обговоріть алгоритм установлення Черепашки в точку з випадковими координатами.
6. Як намалювати орнамент, поданий у формі зубчастої лінії (рис. 30.5)?



Рис. 30.5



Вправа 30

- Створити програму Зоряне небо.
Завдання. Заповнити область вікна зображеннями зірок із випадковими координатами.
1. Завантажте програму: Python → IDLE і створіть нове вікно програми. Збережіть файл з іменем Вправа 30. Завантажте модулі, які потрібні для роботи з Черепашкою та випадковими числами.
 2. Задайте темно-синій колір фону вікна:
`bgcolor('darkblue')`
 3. Створіть список `colors`, елементами якого будуть назви білого, жовтого, помаранчевого кольорів.
`colors = ['white', 'yellow', 'orange']`

4. Запишіть команду циклу `for` для малювання 100 точок, координати і радіуси яких випадкові, а кольори по чергово вибираються зі списку `colors` (рис. 30.6).

```
for i in range(1, 100):
    color(colors[i%3])
    up()
    goto(randint(-200, 200), randint(-200, 200))
    down()
    dot(randint(1, 8))
```

5. Додайте до списку `colors` ще 4 назви кольорів. Змініть програмний код так, щоб були намальовані 100 кіл. Малювання кіл починається з точок із випадковими координатами в діапазоні від -200 до 200 по обох осях, а радіуси є випадковими числами в діапазоні від 5 до 50 (рис. 30.7).
6. Додайте до програми команди, потрібні для того, щоб кола були зафарбовані випадковим кольором. Перевірте роботу програми. Закінчіть роботу, закривши вікно IDLE.

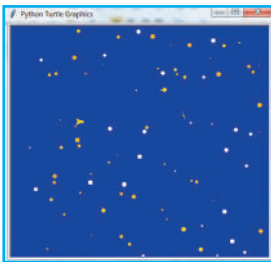


Рис. 30.6

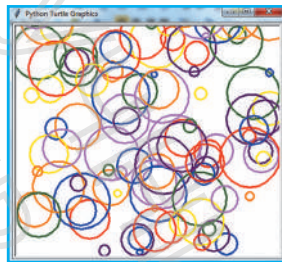


Рис. 30.7



Комп'ютерне тестування

Виконайте тестове завдання з автоматичною перевіркою відповідей (за QR-кодом або посиланням rnk.com.ua/102374).





ПРАКТИЧНА РОБОТА 10 СКЛАДАННЯ ТА ВИКОНАННЯ АЛГОРИТМІВ ІЗ ПОВТОРЕННЯМИ

Завдання: скласти програму для перевірки знання таблиці множення.

Обладнання: комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Завантажте програму: Python → IDLE, створіть файл із назвою Практична робота 10.
2. Завантажте модуль random.
3. Створіть змінну $k = 0$, лічильник правильних відповідей.
4. Запишіть команди виведення та перевірки відповіді для одного виразу:


```

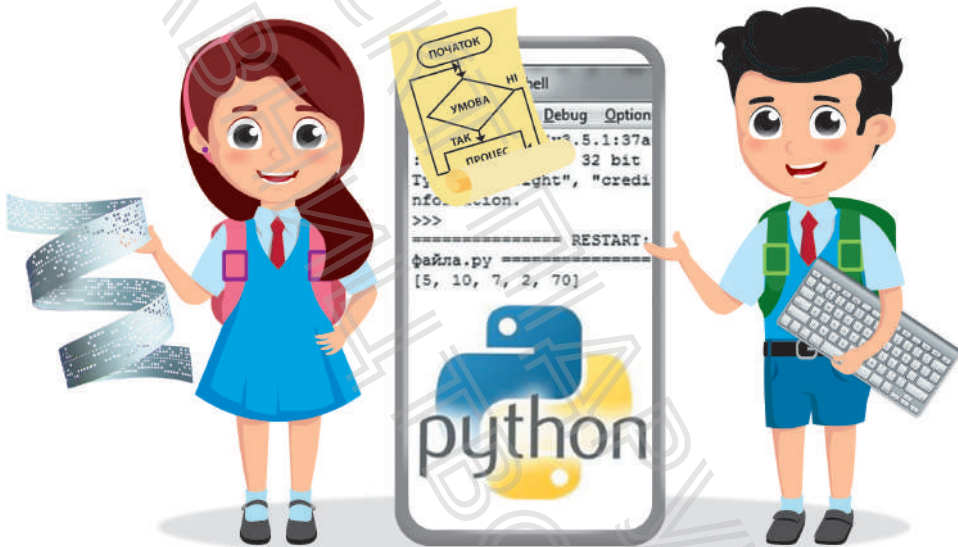
a = randint(1, 10)      # Присвоювання змінним a і b
b = randint(1, 10)      # випадкових значень
print(a, "*", b, " = ") # Виведення виразу
c = int(input("?"))     # Введення відповіді
if a*b == c:            # Перевірка відповіді
    print("Так!")
    k = k+1
else: print("Hi!")

```
5. Допишіть заголовок циклу, щоб 10 разів повторити команди з п. 4.
6. Допишіть команду для виведення значення лічильника правильних відповідей. Перевірте роботу програми.

Зробіть висновок: як складати та виконувати циклічні алгоритми для розв'язування задач.

РОЗДІЛ 6

ПРАКТИКУМ З ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ



Практична робота 11. Створення словника комп'ютерних термінів

Практична робота 12. Передавання інформації за допомогою графічних символів

Практична робота 13. Створення зображень засобами «черепашчої» графіки

Практична робота 14. Складання алгоритмів із розгалуженнями та повтореннями

Практична робота 15. Складання алгоритмів із повтореннями

ПОВТОРЮЄМО



- Ви вже здобули певні знання та навички:
- ознайомилися з будовою та роботою комп'ютера; навчилися безпечно працювати з операційною системою, вмiєте виконувати операції над файлами та папками;
 - дізналися про особливості будови та роботи мереж, навчилися шукати та опрацьовувати потрібну інформацію;
 - навчилися редагувати та форматувати текст за допомогою текстового процесора, створювати комп'ютерну презентацію;
 - ознайомилися з прийомами роботи в середовищі програмування мовою Python, навчилися працювати в інтерактивному режимі, створювати й зберігати програмний код, а також вмiєте створювати малюнки за допомогою «черепашачої» графіки.

1. Наведіть приклади опрацювання інформації.
2. Наведіть приклади інформаційних технологій.
3. Назвіть правила безпечної роботи в інтернеті.
4. Опишіть інструменти форматування тексту.
5. Які об'єкти може містити презентація?
6. Які види вікон існують у середовищі Python?



У цьому розділі ви створите інформаційні продукти, користуючись інструментами програмних середовищ, із якими ознайомилися в 5 класі.

ПРАКТИЧНА РОБОТА 11

СТВОРЕННЯ СЛОВНИКА

КОМП'ЮТЕРНИХ ТЕРМІНІВ

Завдання: створити словник комп'ютерних термінів засобами текстового процесора.

Обладнання: комп'ютер із текстовим процесором.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- Завантажте текстовий процесор і створіть документ. Задайте такі параметри:
 - розмір паперу — А4; поля — по 2 см;
 - орієнтацію — книжкова;
 - шрифт — Arial; розмір — 36; накреслення — жирний;
 - масштаб — одна сторінка; вирівнювання — по центру;
 - інтервали: Перед: — 6 пт; міжрядковий — 1,5 рядка.
- Оформіть першу (титульну) сторінку за зразком (рис. 1).
- Виберіть у комп'ютерному словнику підручника терміни, що стосуються апаратної частини комп'ютера. Наберіть 6 означень.
- Відформатуйте шрифт тексту *другої* сторінки за такими параметрами:
 - розмір шрифту — 16 пт;
 - шрифт — Times New Roman;
 - накреслення — жирний;
 - колір шрифту термінів — темно-синій.

Апаратна частина комп'ютера

Словник
комп'ютерних термінів

Укладач:
учень 5-А класу
Ігор Петренко



Рис. 1

5. Установіть формат абзаців *другої* сторінки (рис. 2):
- вирівнювання — за шириною;
 - перший рядок — навислий, 2 см;
 - інтервали: Перед:, Після: — по 6 пт, міжрядковий — 1,5 рядка.

Апаратна складова комп'ютера (hardware) — це комплекс технічних засобів, який містить пристрої опрацювання і зберігання даних, пристрої введення і виведення, засоби комунікації.

Рис. 2

6. Знайдіть в інтернеті зображення складових комп'ютера, скопіюйте їх та додайте на *титульну* сторінку.
7. Збережіть документ у файлі з назвою Комп'ютерний словник у власній папці. Закінчіть роботу за комп'ютером.

Зробіть висновок: які засоби для створення та форматування документа надає текстовий процесор.



ПРАКТИЧНА РОБОТА 12 ПЕРЕДАВАННЯ ІНФОРМАЦІЇ ЗА ДОПОМОГОЮ ГРАФІЧНИХ СИМВОЛІВ

Завдання: створити презентацію засобами редактора презентацій, у якій дати пояснення, що символізують певні зображення.

Обладнання: комп'ютер із програмою PowerPoint.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.


1. Завантажте програму PowerPoint і створіть нову презентацію. Кнопкою  на вкладці ОСНОВНЕ створіть 4 слайди майбутньої презентації. Змініть макет слайдів на такий, що містить заголовок, два контейнери для тексту та два комбіновані контейнери. Застосуйте до всіх слайдів одну з тем оформлення, що на вкладці Конструктор.
2. На *першому* слайді додайте до заголовка назву презентації: Передавання інформації за допомогою графічних символів. У тексті слайда опишіть у підзаголовку, хто є автором презентації, наприклад Презентація учениці 5-Б класу Сумлінної Ліни.
3. Знайдіть в інтернеті інформацію за ключовими словами «Національний стандарт України», «графічні символи, кольори та знаки безпеки» і з'ясуйте, що означають символи безпеки, наведені на рис. 1.



Рис. 1

Додайте до комбінованих контейнерів зображення двох знаків безпеки, до Контейнерів тексту уведіть пояснення використання цих знаків.

4. Знайдіть в інтернеті інформацію за ключовими словами «символіка рослин». Додайте до заголовка *четвертого* слайда текст «Що символізують рослини?», до комбінованих контейнерів — зображення рослин, які мають символічне значення.

Наприклад, за віруваннями українців верба — це священне дерево життя. Освяченою гілочкою верби вперше виганяли худобу на випас, з верби робили колиски. «Без верби і калини нема України», — говорить прислів'я.

Додайте написи з поясненням символіки цих зображень. Приклад оформлення слайда наведено на рис. 2.



Рис. 2

5. Знайдіть в інтернеті інформацію за ключовими словами «символіка тварин». Додайте до заголовка *четвертого* слайда текст «Що символізують тварини?», до комбінованих контейнерів — зображення тварин, які символізують людські якості: мудрість, хоробрість тощо.

Наприклад, лев — символ відваги, безстрашності, мужності, а голуб — символ миру і добра, любові, дружби.

Додайте написи з поясненням символіки цих зображень.

6. Збережіть файл у форматі PPTX із назвою Символіка у власній папці.

Зробіть висновок: як змінювати розмір, колір та розташування текстів і зображень на слайдах комп'ютерної презентації.

ПРАКТИЧНА РОБОТА 13

СТВОРЕННЯ ЗОБРАЖЕНЬ

ЗАСОБАМИ «ЧЕРЕПАШАЧОЇ» ГРАФІКИ

Завдання: скласти програму малювання символу виду спорту.

Обладнання: комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Завантажте програму: Python → IDLE і створіть нове вікно програми. Збережіть файл у власній папці з назвою Спорт. Розробіть на папері у клітинку схематичне зображення, яке символізує певний вид спорту (рис. 1).

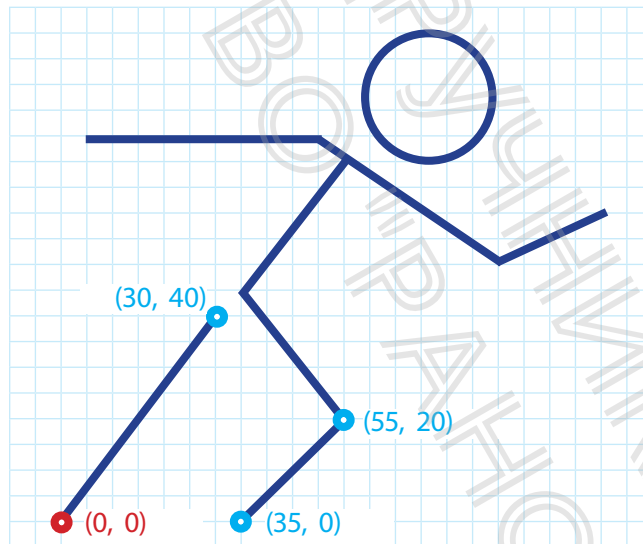


Рис. 1

Визначте координати кінців відрізків (на рис. 1 сторона клітинки дорівнює 5 пікселів).

- Завантажте модуль turtle і встановіть ширину лінії 5 пікселів, колір ліній — синій.

```
from turtle import*
width(5)
color('blue')
```

- Запишіть програму малювання символу (див. рис. 1).

```
goto(30, 40)
up() # Піднімаємо перо
goto(35, 0) # Переходимо в потрібну точку
down() # Опускаємо перо
goto(55, 20)
goto(35, 45)
goto(55, 70)
up()
goto(5, 75)
down()
goto(50, 75)
goto(75, 50)
goto(105, 60)
up()
goto(70, 75)
down()
circle(15) # Малюємо коло радіуса 15
```



- Додайте у графічне вікно назву виду спорту:

```
up()
goto(0, -25)
down()
write('Легка атлетика', font = 'Arial 16')
```

- Запустіть програму на виконання.

Проаналізуйте результат виконання програми у вікні Python Turtle Grafics.

6. Розробіть програму малювання одного з символів видів спорту (рис. 2).



Рис. 2

Зробіть висновок: як передати інформацію мовою символів; яку роль відіграють символи на мапах, вказівниках, у путівниках тощо.



ПРАКТИЧНА РОБОТА 14 СКЛАДАННЯ АЛГОРИТМІВ ІЗ РОЗГАЛУЖЕННЯМИ ТА ПОВТОРЕННЯМИ

Завдання: скласти програму, яка реалізує алгоритм прислів'я.

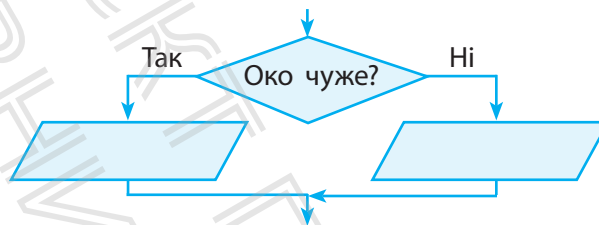
Обладнання: комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Проаналізуйте наведені прислів'я та приказки. Визначте тип алгоритму, описаного в них.
 - Після дощу і сонце засяє.
 - Посієш вітер — пожнеш бурю.

- Як випадуть у травні три дощі добрих, то дадуть хліба на три роки.
 - Змазуй підводу, Клим, якщо зібрався в млин.
 - Сім разів відміряй — один відріж.
2. Впишіть потрібні написи в блок-схему команди розгалуження відповідно до прислів'я «В чужому оці й порошинку видно, а у своєму і колоди не побачиш».



3. Завантажте програму: Python → IDLE і створіть нове вікно програми. Наберіть код, що реалізує алгоритм прислів'я.
- ```

vidp = input('Око чуже? (Так/Hi)')
if vidp == 'Так': print('Видно порошинку')
else: print('Не бачу колоду')

```
4. Складіть блок-схему команди повторення за приказкою «Раз, два, три, чотири, кицю грамоти учили».
5. Створіть нове вікно програми. Наберіть код, що реалізує алгоритм прислів'я.
- ```

for N in range(4):
    print('Кицю грамоти учили')
    print('Навчання закінчилось')
  
```
6. Побудуйте блок-схему команди розгалуження відповідно до приказки «Немазаний віз скрипить, а мазаний сам біжить» і напишіть програму для реалізації алгоритму.

Зробіть висновок: які базові алгоритмічні структури ви знайшли; як створити і реалізувати алгоритми з розгалуженнями та повтореннями.

ПРАКТИЧНА РОБОТА 15

СКЛАДАННЯ АЛГОРИТМІВ

ІЗ ПОВТОРЕННЯМИ

Завдання: скласти програму для перевірки уміння виконувати операції цілочисельного ділення (//) і знаходження остачі від ділення (%).

Обладнання: комп'ютер із середовищем програмування Python.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

1. Завантажте програму: Python → IDLE і створіть нове вікно програми. Збережіть файл із назвою Тест. Завантажте модуль для роботи з випадковими числами.
2. Створіть змінну $k = 0$, лічильник правильних відповідей.
3. Запишіть команду for, параметр якої набуває значень із послідовності 0...10. for i in range(___):
4. Запишіть команду тіла циклу:


```
b = randint(1, 10)
print(a, "/", b, "=", end = ' ') # Виведення виразу
c1 = int(input("?"))           # Введення відповіді на питання
if a//b == c1:                 # Перевірка, чи правильна
    print("Так!")              # відповідь на питання
    k = k+1
else: print("Hi!")
```
5. Додайте до тіла циклу блок для перевірки вміння обчислювати остачу від ділення.
6. Напишіть команду для виведення значення лічильника правильних відповідей. Перевірте роботу програми.

Зробіть висновок: як складати програми для перевірки знань.

КОМП'ЮТЕРНИЙ СЛОВНИК

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

Гіперпосилання — виділений кольором або в інший спосіб текст, зображення чи кнопка, клацання яких викликає перехід на перегляд інших даних.

Змінна — величина, значення якої може змінюватися в ході виконання програми.

Інтерфейс користувача — сукупність засобів для взаємодії між користувачем і комп'ютером.

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

Інформаційна технологія — сукупність методів, процесів опрацювання інформації із застосуванням відповідних засобів.

Локальна мережа — мережа, що об'єднує комп'ютери та інші пристрої, розташовані близько один від одного.

Макет слайда — схема слайда, що визначає спосіб розташування об'єктів на слайді.

Мова програмування — штучна мова, що використовується для запису і виконання алгоритмів за допомогою комп'ютера.

Повторення — форма організації дій, за якої та сама послідовність дій виконується кілька разів залежно від умови.

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

Розгалуження — форма організації дій, коли залежно від виконання (невиконання) умови виконується одна з двох дій.

Рядок — послідовність будь-яких символів, узятя в оди-нарні або подвійні лапки.

Слідування — форма організації дій, за якої команди виконуються одна за одною без пропусків або повторень.

Тіло циклу — сукупність указівок, що повторюється під час кожного проходження (ітерації) циклу.

АЛФАВІТНИЙ ПОКАЖЧИК

А

Алгоритм 109

Алфавіт мови 126

Антивірус 44

Б

Блок-схема алгоритму 116

В

Виконавець алгоритму 112

Властивості алгоритму 111

Властивості сторінки 80

Г

Гіперпосилання 39, 70

Гіпертекст 39

Д

Дані 12

Доменне ім'я 39

З

Змінна 127

Значення 129

І

Інтерфейс користувача 25

Інформаційна система 13

Інформаційна технологія 14

Інформаційні процеси 11

К

Клієнт (робоча станція) 36

Ключове слово 48

Коментар 133

Комп'ютерний вірус 43

Константа 127

Контейнер 64

Л

Лінійний алгоритм 131

Логічний вираз 152

М

Мова програмування 121

О

Оператор 135

П

Папка 26

Повторення (цикл) 118

Припустимі дії 112

Провайдер 38

Програмне забезпечення 22

Р

Розгалуження 117

Рядок 128

С

Семантика мови 127

Середовище виконавця 112

Синтаксис мови 126

Система команд

виконавця 112

Складена умова 154

Слідування 117

Список 167

Ф

Файл 25

Ш

Шаблон презентації 62

Я

Ярлик 27

ЗМІСТ

Передмова.....	3
----------------	---

Розділ 1. ІНФОРМАЦІЙНІ ПРОЦЕСИ ТА СИСТЕМИ

§ 1. Інформація та повідомлення. Правила роботи за комп'ютером.....	7
§ 2. Інформаційні процеси. Інформаційні технології.....	11
§ 3. Використання комп'ютерів	17
§ 4. Комп'ютер як інформаційна система. Складові комп'ютера	20
§ 5. Операційна система та її інтерфейс	24
§ 6. Операції над файлами та папками.....	28
Практична робота 1. Виконання операцій над файлами та папками.....	32

Розділ 2. МЕРЕЖЕВІ ТЕХНОЛОГІЇ ТА ІНТЕРНЕТ

§ 7. Комп'ютерні мережі. Локальна мережа.....	35
§ 8. Глобальна комп'ютерна мережа.....	38
§ 9. Безпечне використання інтернету. Спілкування в інтернеті.....	43
§ 10. Пошук відомостей в інтернеті. Авторське право	46
Практична робота 2. Пошук інформації в інтернеті	52

Розділ 3. КОМП'ЮТЕРНІ ПРЕЗЕНТАЦІЇ

§ 11. Створення та опрацювання комп'ютерних презентацій	55
§ 12. Оформлення комп'ютерних презентацій	62
§ 13. Виступи з показом презентації	69
Практична робота 3. Проектування та розроблення власної презентації.....	74

Розділ 4. ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ

§ 14. Основні об'єкти текстового документа	77
§ 15. Робота з фрагментами тексту. Редагування та форматування символів	83
Практична робота 4. Форматування символів текстового документа.....	90
§ 16. Форматування абзаців	92

Практична робота 5. Форматування абзаців текстового документа.....	98
Практична робота 6. Створення текстового документа	100
§ 17. Опрацювання об'єктів текстового документа.....	102

Розділ 5. АЛГОРИТМИ ТА ПРОГРАМИ

§ 18. Алгоритм і його властивості. Виконавці алгоритмів.....	109
§ 19. Способи опису алгоритму. Алгоритмічні структури.....	115
§ 20. Середовище опису та виконання алгоритмів.....	121
§ 21. Основні поняття мови програмування Python.....	126
§ 22. Лінійні алгоритми.....	131
§ 23. Математичні оператори мови Python	135
Практична робота 7. Складання лінійних алгоритмів	139
§ 24. «Черепашача» графіка	140
§ 25. Алгоритми створення зображень	146
Практична робота 8. Створення зображень за алгоритмами	150
§ 26. Логічні вирази	152
§ 27. Алгоритми з розгалуженнями. Команда if.....	157
§ 28. Алгоритми з розгалуженнями. Команда if...else	160
Практична робота 9. Складання та виконання алгоритмів із розгалуженнями	164
§ 29. Алгоритми з повтореннями. Цикл із параметром	166
§ 30. Використання циклів для побудови зображень	171
Практична робота 10. Складання та виконання алгоритмів із повтореннями	176

Розділ 6. ПРАКТИКУМ З ВИКОРИСТАННЯ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Практична робота 11. Створення словника комп'ютерних термінів	179
Практична робота 12. Передавання інформації за допомогою графічних символів.....	180
Практична робота 13. Створення зображень засобами «черепашачої» графіки	183
Практична робота 14. Складання алгоритмів із розгалуженнями та повтореннями	185
Практична робота 15. Складання алгоритмів із повтореннями	187
Комп'ютерний словник.....	188
Алфавітний покажчик	189

Відомості про користування підручником

№ з/п	Прізвище та ім'я учня/учениці	Навчальний рік	Стан підручника	
			на початку року	у кінці року
1				
2				
3				
4				
5				

Навчальне видання

БОНДАРЕНКО Олена Олександрівна
ЛАСОВЕЦЬКИЙ Василь Васильович
ПИЛИПЧУК Олександр Павлович
ШЕСТОПАЛОВ Євген Анатолійович

«ІНФОРМАТИКА»

підручник для 5 класу закладів загальної середньої освіти

Рекомендовано Міністерством освіти і науки України

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

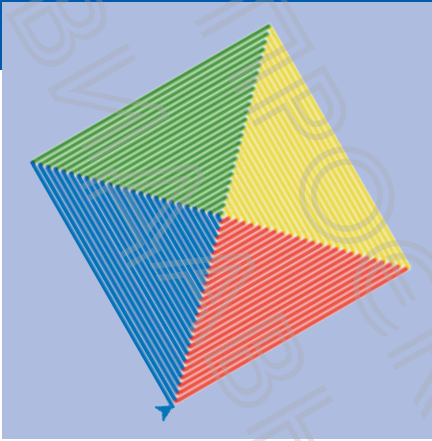
Редактор *Л. Каюда*. Технічний редактор *А. Пліско*.
Художнє оформлення *В. Труфена*, *Т. Задорожної*.
Комп'ютерна верстка *О. Піковець*. Коректор *Н. Красна*.

Підписано до друку _____. Формат 70×90/16.
Папір офсетний. Гарнітура Шкільна. Друк офсетний.
Ум. друк. арк. 5,27. Обл.-вид. арк. 17,8.
Наклад _____ прим. Зам. № _____.

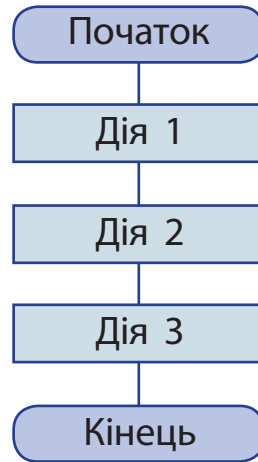
ТОВ Видавництво «Ранок»,
вул. Космічна 21-а, Харків 61145.
Свідоцтво суб'єкта видавничої справи ДК № 7548 від 16.12.2021.
Адреса редакції: вул. Космічна 21-а, Харків, 61165.
E-mail: office@ranok.com.ua. Тел. (057)701-11-22

Підручник надруковано на папері українського виробництва

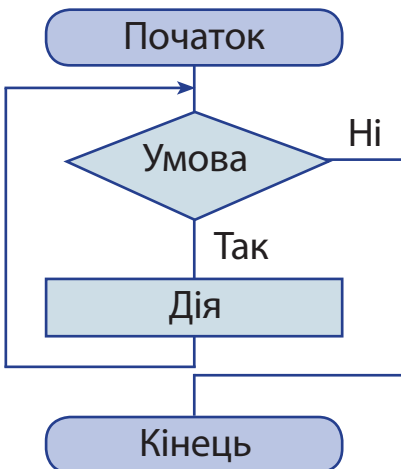
Надруковано у друкарні _____



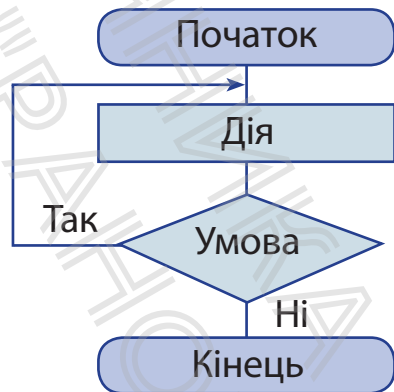
Лінійний алгоритм



Алгоритм із повторенням (цикл із передумовою)

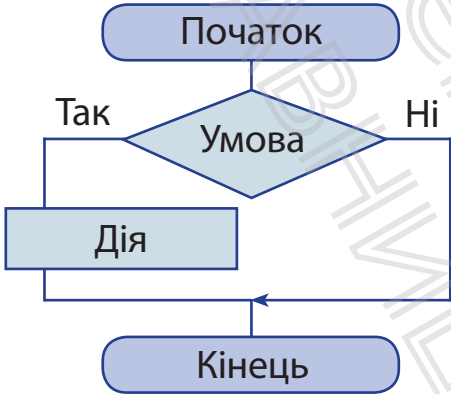


Алгоритм із повторенням (цикл із післяумовою)

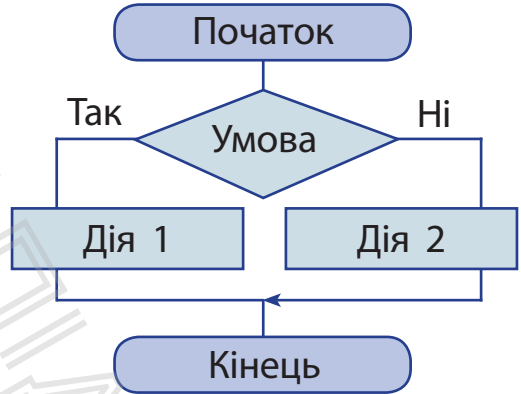


АЛГОРИТМІЧНІ СТРУКТУРИ

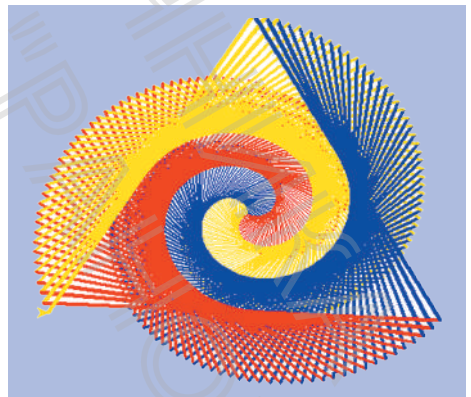
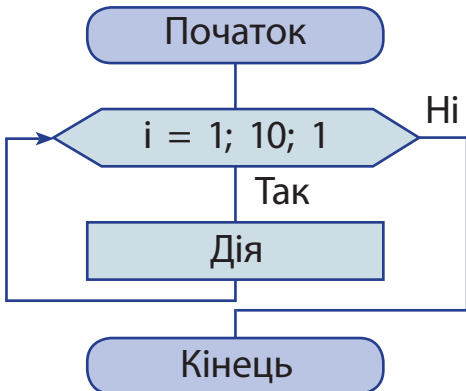
Алгоритм із розгалуженням (неповне розгалуження)



Алгоритм із розгалуженням (повне розгалуження)



Алгоритм із повторенням (цикл із параметром)



підручник для 5 класу
закладів загальної
середньої освіти

ІНФОРМАТИКА

Особливості підручника:

- ▶ зорієнтованість матеріалу на життєвий досвід дітей, підготовку до життя в інформаційному суспільстві
- ▶ покроковий опис виконання практичних робіт
- ▶ численні приклади з різних галузей людської діяльності

Електронний інтерактивний додаток містить:

- ▶ тести за всіма темами курсу
- ▶ підсумкові роботи до кожного розділу
- ▶ додаткові матеріали до практичних робіт



i Інтернет-
підтримка

Електронний інтерактивний
додаток до підручника
доступний за QR- кодом
або посиланням
rnk.com.ua/101524