

Наталія Морзе, Ольга Барна

Оріон



ІНФОРМАТИКА



Глосарій

Байт — послідовність із восьми бітів.

Біт (англ. *binary digit* — двійкова цифра) — цифра 0 або 1 у двійковому коді.

Відеопам'ять (VRAM — від англ. *Video Random Access Memory*) — швидка оперативна пам'ять для зберігання коду зображення, що відображається на екрані монітора.

Відеоредактор — це програма, що містить набір інструментів, за допомогою яких створюють і редагують відеофайли на комп'ютері.

Двійковий код — набір даних, що отримується в результаті двійкового кодування.

Дискретизація (від англ. *discretisation*) — процес усунення неперервності (часової і просторової) аналогових сигналів — носіїв даних.

Драйвери (від англ. *to drive* — управляти, вести) — програми, які управляють роботою зовнішніх пристроїв комп'ютера.

Зовнішня пам'ять, або носії даних, — пам'ять комп'ютера, яка використовується для тривалого зберігання даних.

Інтерпретатор — програма, яка зчитує вихідний код програми по одній команді й одразу намагається їх «перекладати» та виконувати.

Елементи керування — це графічні об'єкти, розташовані на вікні програми для показу або введення даних, виконання дій або полегшення роботи: текстові поля, списки, перемикачі, кнопки, прапорці тощо.

Кишеньковий комп'ютер — це вид портативних комп'ютерів, які вирізняються маленьким розміром і потужною батареєю, що дає змогу переносити та використовувати пристрій тривалий час. Засіб введення в таких пристроях — або сенсорний екран, або висувна клавіатура.

Код — це система правил для перетворення повідомлень, що містять текст, звук, зображення, жести тощо.

Кодування повідомлень — це процес перетворення за певними правилами одного подання набору даних в інший. Під час кодування повідомлення відбувається зміна вигляду повідомлення без зміни його змісту.

Колонтитул — об'єкт, що розміщується над текстом (верхній колонтитул) або під текстом (нижній колонтитул) кожної сторінки книги, газети, журналу, документа.

Коментар — це будь-який текст для пояснення роботи програми, який не виконується інтерпретатором.

Компілятор — програма, яка зчитує одразу всю програму й переписує її машинним кодом або мовою асемблеру. Процес трансляції, який за такого підходу називається **компіляцією**, відбувається один раз, і результат перекладу зберігається в окремому файлі.

Конвертер — програма, яка перетворює у файлі дані з одного формату в інший.

Магістраль (шина) — набір електронних ліній, за допомогою яких процесор з'єднується з іншими пристроями пам'яті та пристроями для передавання даних і службових сигналів.

Методи — це команди, які об'єкт розуміє і може виконувати.

Мова програмування — це система позначень для точного опису алгоритму, який потрібно виконати за допомогою комп'ютера.

Моноблок — вид комп'ютера, у якому системний блок та монітор виглядають як єдине ціле.

Мультимедіа — це поєднання різних способів подання повідомлень.

Настільні комп'ютери — це комп'ютери, які потребують постійного підключення до живлення, зазвичай розташовують на столі й не переносяться під час роботи.

Наталія Морзе
Ольга Барна



Підручник для 8 класу
закладів загальної середньої освіти

*Рекомендовано
Міністерством освіти і науки України*



УДК 004*кл8(075.3)
М79

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

**Видано за рахунок державних коштів.
Продаж заборонено**

Навчальне видання
МОРЗЕ Наталія Вікторівна
БАРНА Ольга Василівна

ІНФОРМАТИКА

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

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

Головна редакторка *І. В. Красуцька*

Редакторка *В. М. Ліченко*

Головна художниця

І. П. Медведовська

Технічний редактор *Е. А. Авраменко*

Коректорка *С. В. Войтенко*

Малюнки *Олени Харченко*

Використано ілюстрації з ресурсів
shutterstock.com і *depositphotos.com*

Формат 70х100 ¹/₁₆.

Ум. друк. арк. 18,144 + 0,324 форзац.

Обл.-вид. арк. 16,33 + 0,55 форзац.

Зам. №

Наклад 131 630 пр.

**ТОВ «Український освітянський
видавничий центр «Оріон»**

Свідоцтво про внесення суб'єкта видавничої
справи до Державного реєстру видавців,
виговлювачів і розповсюджувачів
видавничої продукції

Серія ДК № 4918 від 17.06.2015 р.

Адреса видавництва: 03061,

м. Київ, вул. Миколи Шепелева, 2

Віддруковано у ТОВ «КОНВІ ПРІНТ».

Свідоцтво про внесення суб'єкта видавничої
справи до Державного реєстру видавців,
виговлювачів і розповсюджувачів
видавничої продукції серія ДК № 6115,
від 29.03.2018 р.

03680, м. Київ, вул. Антона Цедіка, 12,

Н. В. Морзе

М79 Інформатика. Підручник для 8 кл. закладів загальної серед-
ньої освіти / Н. В. Морзе, О. В. Барна. — Київ : УОБЦ «Оріон»,
2021. — 224 с. : іл.

ISBN 978-966-991-119-3.

УДК 004*кл8(075.3)

ISBN 978-966-991-119-3

© Н. Морзе, О. Барна, 2021

© УОБЦ «Оріон», 2021

Дорогі друзі!

Цей підручник допоможе вам продовжити вивчення інформатики. Він складається з п'яти розділів. Ви дізнаєтесь, як кодують дані в комп'ютері, ознайомитеся з характеристиками складових сучасних комп'ютерів. Удосконаліть та розширите свої вміння роботи з текстовими документами у другому розділі. Поринете в неймовірний світ творчості разом із програмами для роботи з мультимедіа у третьому розділі. За матеріалами четвертого розділу продовжите вивчення мови програмування *Python* і навчитесь складати справжні програми з графічним інтерфейсом. Останній розділ підручника допоможе вам створювати сайти.

Тематичні блоки об'єднано в розділ, який завершується рубрикою **Оцінюємо себе**. Простий інструмент для самооцінювання допоможе вам визначити свій поступ у навчанні.

Щоб досягти очікуваних результатів навчання та бути компетентними в цифровому світі, радимо:

- напередодні уроку читати матеріали рубрики **Вивчаємо**, щоб разом з учнями та вчителем або вчителькою обговорити незрозумілі й поділитись вивченням;
- виконати всі завдання з рубрики **Діємо** й бути готовими надати допомогу тим, хто не може оволодіти базовими вміннями без сторонньої допомоги;
- постійно експериментувати під час виконання завдань рубрики **Досліджуємо**;
- постійно співпрацювати з іншими, навчати їх і навчатись у них під час роботи в парах і малих групах рубрики **Працюємо в парах**;
- планувати свою навчальну діяльність, мислити логічно, шукати незвичні розв'язки завдань, робити висновки та самостійно приймати рішення, виконуючи завдання з рубрики **Працюємо самостійно**;
- обирати не тільки завдання базового рівня, а й ті, що потребують креативного мислення, здатності аналізувати, оцінювати,

творити, — саме вони позначені  та .

Файли-заготовки до практичних завдань завантажуйте за посиланням <https://cutt.ly/NvMCytY>.



Ви вже дорослі. Тому маєте розуміти, що саме від вас залежить, чого ви можете досягти у своєму житті. Використовуйте цей підручник та різноманітні навчальні ресурси, щоб пізнавати нове, навчатися завжди і скрізь.

Успіхів вам у пізнанні цікавих загадок цього світу!

Розділ 1

Кодування даних
та апаратне забезпечення

1



Кодування даних

ПРИГАДАЙТЕ:

- ▶ як пов'язані між собою поняття інформація, повідомлення, дані;
- ▶ способи подання повідомлень;
- ▶ пристрої для роботи з даними.

ВИ ДІЗНАЄТЕСЬ:

- ▶ що означає «опрацювання даних»;
- ▶ що таке кодування повідомлень та для чого воно потрібне;
- ▶ які особливості має двійкове кодування;
- ▶ як кодуються в комп'ютері текстові повідомлення;
- ▶ як визначити довжину двійкового коду повідомлення;
- ▶ як кодують графічні дані.



Вивчаємо

1. Що означає «опрацювання даних»?

Людина постійно отримує, шукає, зберігає і передає повідомлення, коли пізнає навколишній світ і спілкується. Відомості, зафіксовані на деякому носії, перетворюються в **дані**. З повідомленнями та даними вона може виконувати різні дії залежно від поставленої мети. Такі дії називають **інформаційними процесами** (мал. 1.1).

Серед інформаційних процесів тільки один процес передбачає отримання нових даних з наявних. Це відбувається під час їх опрацювання.

Опрацювання даних — це інформаційний процес отримання нових даних з наявних.



Мал. 1.1

Наприклад, опрацьовуючи дані про кількість захворювань населення на Covid-19, можна зробити висновок про тенденцію (зростання чи спадання) захворюваності та порівняти з рівнем захворюваності в інших країнах (мал. 1.2).

Місце	Усього випадків	Нові випадки (за 1 день*)	Нові випадки за останній час (60 дн.)	Випадків на 1 млн осіб	Смерті
Польща	843 475	24 213		21 978	13 288
Великобританія	1 493 383	19 875		22 479	54 626
Франція	2 127 672	17 651		31 720	48 184
Україна	629 850	14 864		15 039	11 149
Німеччина	927 990	13 872		11 161	14 061

Мал. 1.2

А виконуючи обчислення над вхідними даними в поставленій задачі, ми отримуємо нові дані — розв'язок. Наприклад, маючи покази термометра (за Цельсієм) в Україні, можна отримати дані про цю само температуру (за Фаренгейтом) у США. Для цього використовують формулу переведення градусів Цельсія в градуси Фаренгейта:

$$\text{Цельсій} \cdot 1,8 + 32 = \text{Фаренгейт}.$$

2. Що таке кодування повідомлень та для чого воно потрібне?

Повідомлення можна подати текстовим, графічним, звуковим способами, а також за допомогою відео, жестами та декількома способами одночасно — комбінованим повідомленням. Для подання усних повідомлень ми використовуємо мову. А будь-яке письмове повідомлення складається з набору різних знаків (символів), які належать певному алфавіту. Наприклад, для подання числових даних використовують алфавіт, що складається з 11 знаків: цифр 0–9 і десяткової коми для розділення цілої та дробової частини в десятковому дробі. Використання пристроїв для подання повідомлень зумовлює необхідність використання спеціальних знаків.

Код — це система правил для перетворення повідомлень, що містять текст, звук, зображення, жести тощо.

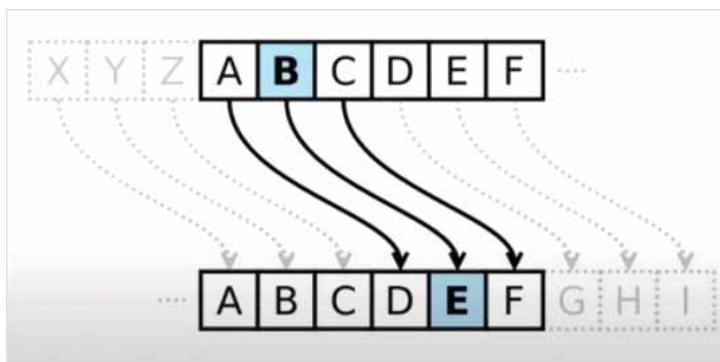
Код Морзе	Штрихкод	QR-код
... - - - -		

Кодування повідомлень — це процес перетворення за певними правилами одного подання набору даних в інший. Під час кодування повідомлення відбувається зміна вигляду повідомлення без зміни його змісту.



Це цікаво

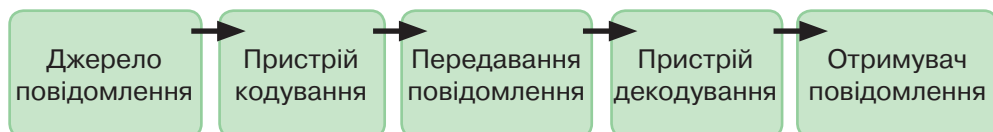
Для кодування повідомлень римський полководець Юлій Цезар придумав власний шифр. У ньому кожен літеру слова замінювали на ту, яка в алфавіті зсунута на деяке число-ключ.



У процесі обміну повідомленнями ми здійснюємо дві операції: кодування і декодування. Перша пов'язана з переходом від вихідної форми подання повідомлення у форму, яка є зручною для збереження, передавання або опрацювання. А друга — зі зворотним переходом до вихідного подання повідомлення.

Процес прийому й передавання повідомлень можна зобразити схемою (мал. 1.3).

Мал. 1.3



Кодування і декодування належать до процесу опрацювання повідомлень, який можна здійснювати за допомогою комп'ютера.



Діємо

Вправа 1. QR-код.

Завдання. Створіть QR-код із фрагментом вірша улюбленого українського поета чи поетеси. Завантажте код і запропонуйте його комусь у класі, щоб можна було відгадати автора чи авторку поезії.

1. Відкрийте вікно браузера. У рядку пошуку задайте пошуковий запит: створити QR-код безкоштовно.
2. У списку доступних сервісів оберіть один з них, наприклад:

qr9.me › free-qr-code-generator ›

Безкоштовний онлайн генератор QR кодів / динамічні ...

Слово QR code являє собою абревіатуру від англ. Quick Response code (код Швидкого Відгуку). Ці квадратні штрих-коди вперше були розроблені і ...

3. Ознайомтеся з інтерфейсом сервісу. Визначте, на який вміст за допомогою цього сервісу можна створити QR-код.
4. Оберіть вміст: текст. Введіть фрагмент поезії обсягом відповідно до встановлених обмежень.
5. Завершіть створення коду. Завантажте зображення коду та збережіть файл з іменем *Прізвище_Код* у своїй структурі папок.
6. За можливості роздрукуйте його або розмістіть файл у спільній папці класу.
7. Установіть на смартфон програму для зчитування QR-кодів або скористайтесь установленою. Розкодуйте інформацію в QR-коді, який створено кимось у класі, просканувавши зображення коду. Якщо автор вам не відомий, то застосуйте інструмент пошуку під текстом.
8. Зробіть висновок про інформаційні процеси, що застосовані під час виконання даного завдання.

3. Які особливості має двійкове кодування?

Повідомлення можуть кодуватися різними способами. Вибір способу кодування залежить від виду повідомлення, яке необхідно кодувати: текст, число, графічне зображення або звук чи відео. Для опрацювання за допомогою комп'ютера повідомлення подаються (кодуються) у вигляді послідовностей електричних або магнітних сигналів двох видів. Кожний сигнал одного виду умовно позначають цифрою 0, а іншого виду — 1.

Кодування повідомлень з використанням двох сигналів називається **двійковим**. Набір даних, що отримується в результаті двійкового кодування, називається **двійковим кодом**.

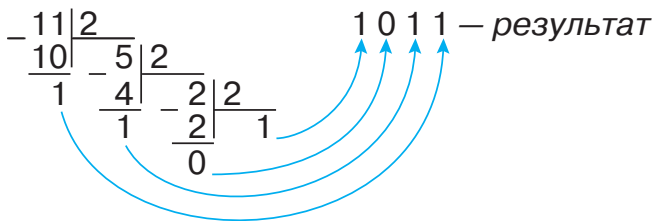
Двійковий код використовують у *двійковій системі числення*. Ця система числення є позиційною, її база дорівнює двом і використовує для запису чисел тільки два символи: зазвичай 0 (нуль) та 1 (одиницю). Числа, представлені в цій системі, часто називають **двійковими**, або **бінарними**, числами. Для запису числа у двійковій системі числення використовують подання цього числа за допомогою степенів числа 2. Наприклад,

$$110001_2 = 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 49_{10}$$

Зверніть увагу, що в записі нижній індекс позначає систему числення.

Переведення числа з десяткової системи числення у двійкову відбувається послідовним діленням з остачею (мал. 1.4).

 Мал. 1.4



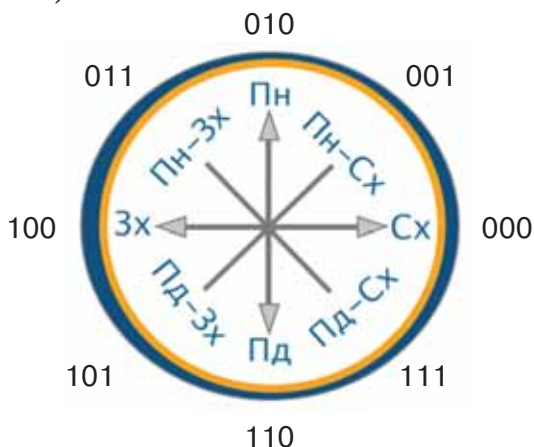
Цифра 0 або 1 у двійковому коді повідомлення називається **біт** (англ. *binary digit* — двійкова цифра).

Однією з двох цифр 0 або 1 можна закодувати, наприклад:

- правильність твердження: неправильно (0) або правильно (1);
- стан вимикача: вимкнено (0) або ввімкнено (1) тощо.

З двох бітів можна скласти 4 ($4 = 2^2$) коди (00, 01, 10 і 11). Ними можна закодувати, наприклад, чотири чверті координатної площини: 00 — ліва верхня, 01 — права верхня, 10 — ліва нижня, 11 — права нижня.

З трьох бітів можна скласти вже 8 ($8 = 2^3$) кодів (000, 001, 010, 011, 100, 101, 110, 111). Ними можна закодувати, наприклад, сторони горизонту (мал. 1.5).



Мал. 1.5

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

З восьми бітів можна скласти $2^8 = 256$ кодів, і цієї кількості кодів достатньо, щоб закодувати всі літери англійського та українського (або якогось іншого) алфавіту, арабські цифри, розділові знаки, знаки арифметичних дій, а також деякі інші символи.

Послідовність із восьми бітів називається байтом.

1 Б (байт) = 8 бітів

Таблиця степенів числа 2 демонструє кількість можливих різних повідомлень, які можна закодувати за допомогою відповідної кількості бітів:

i	0	1	2	3	4	5	6	7	8	9	10
$N = 2^i$	1	2	4	8	16	32	64	128	256	512	1024

4. Як кодуються в комп'ютері текстові повідомлення?

Традиційно для кодування одного символу використовують двійковий код, довжина якого дорівнює 1 байту, тобто 1 символ = 1 байт = 8 бітів. Тоді можна обчислити, яку кількість різних символів можна закодувати: $N = 2^8 = 256$. Така кількість символів є цілком достатньою

Таблиця 1.1

Код	Символ	Код	Символ	Код	Символ	Код	Символ
32	про-біл	56	8	80	P	104	h
33	!	57	9	81	Q	105	i
34	«	58	:	82	R	106	j
35	#	59	;	83	S	107	k
36	\$	60	<	84	T	108	l
37	%	61	=	85	U	109	m
38	&	62	>	86	V	110	n
39	'	63	?	87	W	111	o
40	(	64	@	88	X	112	p
41	)	65	A	89	Y	113	q
42	*	66	B	90	Z	114	r
43	+	67	C	91	[	115	s
44	,	68	D	92	\	116	t
45	-	69	E	93	]	117	u
46	.	70	F	94	^	118	v
47	/	71	G	95	_	119	w
48	0	72	H	96	'	120	x
49	1	73	I	97	a	121	y
50	2	74	J	98	b	122	z
51	3	75	K	99	c	123	{
52	4	76	L	100	d	124	
53	5	77	M	101	e	125	}
54	6	78	N	102	f	126	~
55	7	79	O	103	g	127	DEL

для подання текстових повідомлень, включаючи великі та малі літери українського й латинського алфавітів, цифри, знаки, графічні символи тощо. Кодування полягає в тому, що кожному символу ставиться у відповідність унікальний десятковий код від 0 до 255 або відповідний йому двійковий код від 00000000 до 11111111.

Користувач натискає на клавіатурі клавішу із символом, і в комп'ютер надходить певна послідовність із восьми електричних імпульсів (двійковий код символу). Код символу зберігається в оперативній пам'яті комп'ютера, де займає один байт. У процесі виведення символу на екран комп'ютера проводиться зворотний процес — декодування, тобто перетворення коду символу в його зображення.



Коди символів зберігаються у спеціальній таблиці. Кодова таблиця встановлює відповідність між символами та їхніми двійковими кодами для подання текстових даних у комп'ютері. Для того щоб весь світ однаково кодував текстові дані, потрібні єдині таблиці кодування.

У всьому світі за стандарт прийнято таблицю ASCII (*American Standard Code for Information Interchange* — Американський стандартний код для обміну повідомленнями). Створена в 1963 р. система кодування ASCII передбачала кодування 128 символів, коди яких склалися із 7 бітів ($2^7 = 128$). Із часом кодування було розширене до 256 символів ($2^8 = 256$), при цьому коди перших 128 символів не змінилися.

Система кодування ASCII (табл. 1.1) складається з базової (значення кодів від 0 до 31) і розширеної (значення кодів від 32 до 255) таблиць. Коди з 33 по 127 є інтернаціональними й відповідають символам латинського алфавіту, цифрам, знакам арифметичних операцій

і знакам пунктуації. Коди зі 128 по 255 є національними, тобто в національних кодуваннях тому самому коду відповідають різні символи.

У світі існують й інші системи кодування. У різних таблицях кодування ті само символи можуть мати різні коди. Останнім часом серед таблиць кодування, які містять українські літери, найпоширенішими є KOI8-U і Windows-1251. Довжина коду кожного символу в них — 1 байт.

Широкого поширення набув міжнародний стандарт *Unicode* — *Unicode Consortium* (UTF 32, UTF 16 і UTF 8), який відводить на кожний символ не один байт, а два, тобто 16 бітів, тому за його допомогою можна закодувати не 256, а $2^{16} = 65\,536$ різних символів.



ДІЄМО

Вправа 2. Таблиця *Unicode*.

Завдання. Закодуйте за таблицею *Unicode* слово *rease*. Запишіть розв'язок у текстовий документ. Збережіть його з іменем *Слово_Прізвище* у своїй структурі папок.

1. Відкрийте текстовий процесор і створіть документ з іменем *Слово_Прізвище*.
2. У головному меню операційної системи оберіть програму *Таблиця символів*.
3. У вікні, що відкриється, встановіть додаткові параметри перегляду (мал. 1.6).



Мал. 1.6

4. Послідовно обирайте літери слова та в нижній частині вікна визначайте код літери. Записуйте відповідний код у текстовий документ.
5. Збережіть файл.

5. Як визначити довжину двійкового коду повідомлення?

Довжина двійкового коду текстового повідомлення — це кількість бітів чи байтів у двійковому коді цього повідомлення.

Для позначення довжин двійкового коду повідомлень використовують і більші одиниці вимірювання, які, згідно з Міжнародною системою одиниць (СИ), утворюються за допомогою префіксів кіло-, мега-, гіга-, тера- і т. д. Історично склалося так, що ці префікси (кіло-, мега-, гіга-, тера-) в інформатиці трактувалися інакше, не так, як, наприклад, у математиці, і мають у своїй основі степені числа 2, а саме:

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

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

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

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

Щоб перевести біти в байти, треба число бітів поділити на 8. Наприклад: 32 біти — це 4 байти. Щоб перевести байти в кілобайти, треба число байтів поділити на 1024. Наприклад: у 2048 байтах буде 2 кілобайти. І так далі за наступними одиницями вимірювання.

Щоб перевести байти в біти, треба число байтів помножити на 8. Наприклад: у 3 байтах буде 24 біти.

Щоб перевести кілобайти в байти, треба число кілобайтів помножити на 1024. Наприклад: у 3 кілобайтах буде 3072 байти й відповідно 24 576 бітів.

Щоб обчислити довжину двійкового коду повідомлення, треба кількість символів у тексті помножити на кількість бітів, потрібних для кодування одного символу. Наприклад: двійкове число 01010111 займає в пам'яті 8 бітів. Якщо його записати у вигляді тексту в кодуванні ASCII, довжина коду буде 8 байтів, або 64 біти, оскільки кожний символ кодується за допомогою 8 бітів. Довжина двійкового коду цього ж тексту в кодуванні *Unicode* становитиме 16 байтів, або 128 бітів.

Не слід забувати, що пропуски треба теж вважати за символи, оскільки вони також набираються на клавіатурі, мають свій код і зберігаються в пам'яті.



ДІЄМО

Вправа 3. Довжина двійкового коду тексту.

Завдання. Книжка містить 150 сторінок, на кожній сторінці — 40 рядків, у кожному рядку — 60 символів (включаючи пропуски). Знайдіть довжину двійкового коду тексту книжки, якщо для кодування кожного символу використано 8 бітів.

Розв'язання. Обчислимо кількість символів у книжці: $60 \cdot 40 \cdot 150 = 360\,000$ (символів).

Оскільки довжина двійкового коду 1 символу — 8 біт = 1 Б, то довжина двійкового коду книжки дорівнює 360 000 Б.

Переведемо байти в більші одиниці: $360\,000 \text{ Б} : 1024 = 351,56 \text{ кБ}$;
 $351,56 \text{ кБ} : 1024 = 0,34 \text{ МБ}$.

Вправа 4. Перекодування.

Завдання. Автоматичний пристрій здійснив перекодування текстового повідомлення довжиною 48 символів, спочатку записаного в 7-бітному коді ASCII, в 16-бітне кодування *Unicode*. Визначте, на скільки збільшилася довжина двійкового коду повідомлення.

Розв'язання. Зміна кодування із 7 бітів на 16 бітів збільшує довжину коду кожного символу на $16 - 7 = 9$ біт. Оскільки повідомлення містить 48 символів, то його обсяг збільшився на $48 \cdot 9 = 432$ (біта). Переведемо біти в байти: $432 : 8 = 54$ (Б).

Вправа 5. Модем.

Завдання. Визначте, скільки часу модем, що працює зі швидкістю 1200 біт/с, буде передавати дані десяти сторінок тексту із 40 рядків по 80 символів у рядку.

Розв'язання. Обчислимо загальну кількість символів на сторінці. Це $40 \cdot 80 = 3200$ (символів).

Оскільки в кодуванні ASCII один символ має довжину коду 1 Б, загальна довжина двійкового коду даних на сторінці — 3200 Б. Переведемо 3200 Б в біти, оскільки швидкість модема вказана в біт/с. Отримаємо $3200 \text{ Б} = 25\,600$ біт.

Розділимо 25 600 біт на 1200 біт/с, отримаємо 21,33 с. Таким чином, 10 сторінок тексту будуть передані за 213,3 с, або 3 хв 33,3 с.



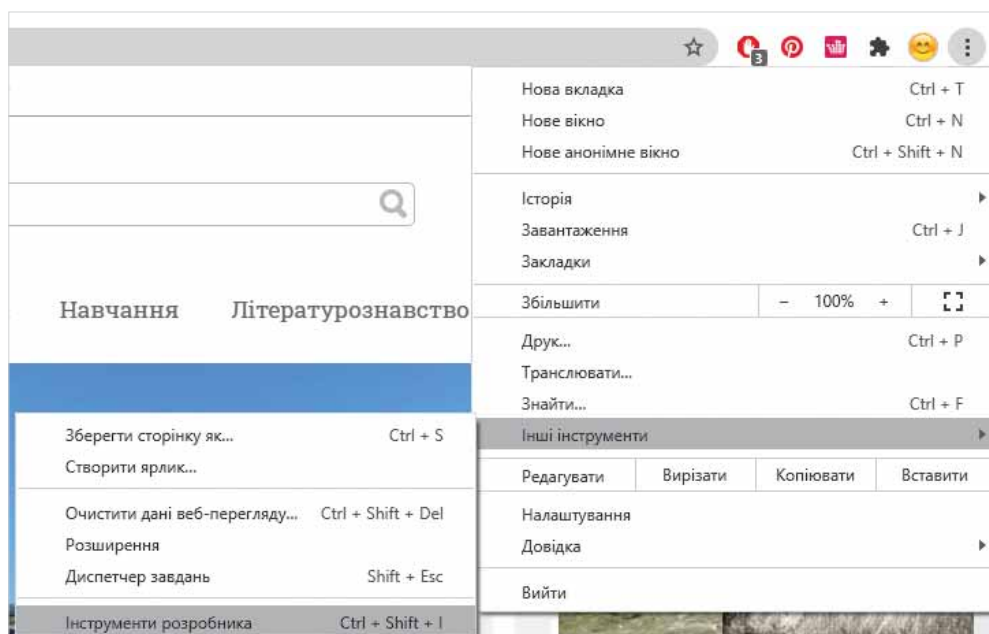
Досліджуємо

Вправа 6. Кодування вебсторінок.

Завдання. Дослідіть, як зміняться текстові дані на головній сторінці вебпорталу *Бібліотека української літератури* (<http://www.ukrlib.com.ua/>), якщо змінити систему кодування, що використана під час її створення.

1. Відкрийте вікно браузера, наприклад, *GoogleChrome*. У полі адреси введіть адресу вебпорталу *Бібліотека української літератури* (<http://www.ukrlib.com.ua/>). Перегляньте, чи правильно відображаються текстові написи на сторінці.

2. Оберіть у меню, що відкрилося, вказівку *Інші інструменти/Інструменти розробника* (мал. 1.7).



Мал. 1.7

3. Розгляньте, як виглядає код сторінки.

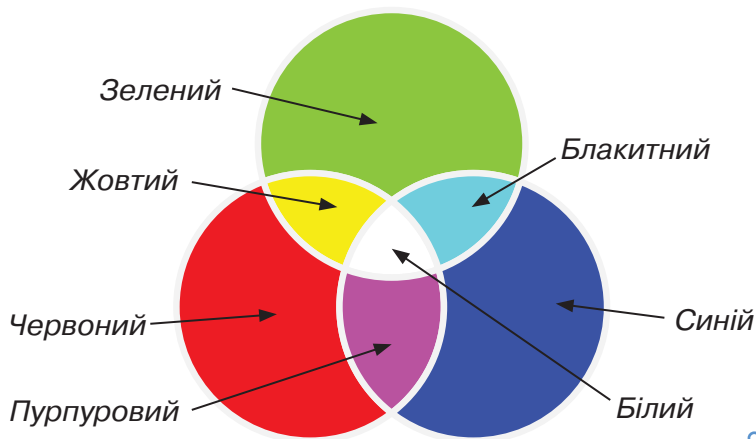
6. Як кодують графічні дані?

Розрізняють два основні методи створення і кодування графічних зображень: **растровий** і **векторний**.

Векторний малюнок складається з переліку об'єктів і наданих їм властивостей, зокрема форми, місця розміщення, розміру, контуру, вмісту. Усі ці властивості мають свій цифровий код.

Растрове зображення складається з точок — пікселів відповідних кольорів і нагадує аркуш паперу в клітинку, на якому кожна клітинка зафарбована певним кольором. Для кожного пікселя такого зображення вказують координату й колір. Під час кодування для прямокутного растра характеристикам кожного пікселя ставиться у відповідність цифровий код. Обсяг файлу, що містить растрове графічне зображення, — це переважно добуток розмірів (кількість пікселів за шириною і висотою) цього зображення на глибину кольору — кількість бітів, що необхідна для кодування кольору пікселя.

Найпоширеніший спосіб кодування кольору — RGB (від англ. *Red, Green, Blue* — червоний, зелений, синій). Використовується під час відображення зображення на екрані монітора комп'ютера чи телевізора, демонстрації презентацій, відеофрагментів, графіки (мал. 1.8).



Мал. 1.8

Будь-який колір можна отримати внаслідок зміни яскравості основних кольорів — червоного, зеленого та синього. У моделі RGB кожний з основних кольорів може набувати значень від 0 до 255. Наприклад, RGB (255,0,0) відображатиметься як червоний, оскільки параметр червоного кольору задано його найбільшим значенням, а інші дорівнюють 0.



Діємо

Вправа 7. Кількість кольорів.

Завдання. Для зберігання фото на сайті соціальної мережі пропонують завантажити зображення розміром 32×64 пікселі обсягом 2 кБ. Визначте, скільки кольорів можна використати при цьому.

Розв'язання.

1. Визначимо кількість пікселів у зображенні: $32 \cdot 64 = 2^5 \cdot 2^6 = 2^{11}$ (точок).
2. Визначимо обсяг файлу: $2\text{кБ} = 2 \cdot 2^{10} = 2^{11} \text{Б}$.
3. Визначимо обсяг байтів на одну точку: $2^{11} \text{Б} : 2^{11} \text{точок} = 1 \text{Б} = 8 \text{бітів}$.
4. Висновок: кількість кольорів $= 2^8 = 256$ (кольорів).



Працюємо в парях

1. Обговоріть, який інформаційний процес відбувається в поданих ситуаціях:
 - лікарка на основі аналізу крові визначає діагноз захворювання пацієнта;
 - бухгалтер на основі даних про виробіток працівника нараховує заробітну плату;
 - метеорологиня на основі даних про стан довкілля прогнозує погоду на наступний день;

• хімік на основі даних хімічного рівняння визначає результат хімічної реакції.

Наведіть власний приклад подібної ситуації, що реалізує визначений вами інформаційний процес.

- ★ 2. Запропонуйте одне одному текстове повідомлення, закодоване за допомогою однієї із систем кодування. Хтось подає придумане повідомлення, використовуючи таблицю *Unicode*, інший чи інша — *Windows-1251*.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	0
С	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
Д	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	І	Ї	Ь	Є	Ю	Я

Перевірте правильність кодування. Для цього обміняйтесь кодами та декодуйте повідомлення. Визначте довжину двійкового коду кожного з повідомлень. Зробіть висновок.

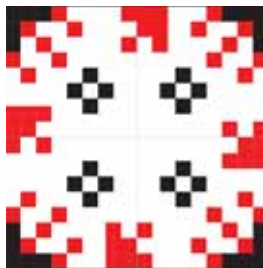
- ★ 3. Створіть у табличному процесорі таблицю переведення одиниць довжини двійкового коду повідомлення в такому вигляді:

Кількість символів двійкового коду, біт	Б	кБ	МБ	ГБ	ТБ

Запропонуйте одне одному та перевірте на практиці інші таблиці переведення, наприклад, кілобайтів у біти, байти, мегабайти, гігабайти, терабайти. Обговоріть, як при цьому будуть змінюватись формули в клітинках електронної таблиці.

- ★★ 4. На сайті <https://ornament.name/creator#> запропоновано таблицю кодів для літер української абетки. Її можна використати для створення орнаментів вишивання. Наприклад, для слова ВОЛЯ отримаємо орнамент.

Поясніть, який спосіб кодування графічних зображень використано. Запропонуйте свої варіанти «вишивання» цього слова.





Працюємо самостійно

5. Створіть власний спосіб кодування даних. Презентуйте його в класі. Поясніть, кому такий спосіб кодування може бути корисним.
- ★ 6. Дослідіть, які дані можна отримати зі штрихкоду товару. Які дані мають бути на штрихкоді, щоб взяти участь в акції з підтримки вітчизняних товаровиробників «Купляймо українське». Складіть листівку-довідник з питань придбання товарів.
- ★ 7. Перевірте, чи правильно записано рівності, якщо нижній індекс позначає систему числення:

$$1110011_2 = 115_{10}$$

$$43_{10} = 101010_2$$

$$1100001_2 = 98_{10}$$

$$36_{10} = 100100_2$$

Виконайте відповідні обчислення.

8. Автоматичний пристрій здійснив перекодування текстового повідомлення українською мовою, спочатку записаного в 16-бітному коді *Unicode*, у 8-бітне кодування KOI-8. При цьому довжина двійкового коду повідомлення зменшилася на 480 біт. Скільки символів містить повідомлення?
9. Довжина двійкового коду текстового повідомлення, підготовленого за допомогою комп'ютера, становить 3,5 кБ. Скільки символів містить цей текст? Чи достатньо даних має задача, щоб можна було дати однозначну відповідь?
- ★ 10. Відомо, що на кожній сторінці документа — 128 рядків, що містять по 48 символів кожний. Скільки сторінок у документі, якщо довжина його двійкового коду становить 720 кБ, за умови, що кожний символ кодувався 2-байтовим кодуванням *Unicode*?
11. Довжина двійкового коду текстового повідомлення становить 8192 біти. Виразіть це значення в кілобайтах.
12. Довжина двійкового коду текстового повідомлення становить 2 097 152 Б. Виразіть це значення в мегабайтах.



Апаратне забезпечення комп'ютера

ПРИГАДАЙТЕ:

- ▶ як у давнину люди здійснювали обчислення, обмінювались інформацією чи зберігали її;
- ▶ які є види сучасних персональних комп'ютерів.

ВИ ДІЗНАЄТЕСЬ:

- ▶ як технічний прогрес вплинув на розвиток комп'ютерної техніки;
- ▶ якими бувають сучасні комп'ютери;
- ▶ яку архітектуру мають сучасні комп'ютери.



Вивчаємо

1. Як технічний прогрес вплинув на розвиток комп'ютерної техніки?

Комп'ютерна техніка використовується для обчислень й опрацювання інформаційних даних різного типу. Першими пристосуваннями для обчислень були, ймовірно, зарубки, вузлики, лічильні палички. Розвиваючись, ці пристосування ставали складнішими. Поступово з найпростіших пристосувань для лічби народжувалися складніші пристрої: абак (рахівниця), логарифмічна лінійка, механічний арифмометр, електронний комп'ютер. Засоби для обчислень постійно змінювались і пройшли кілька етапів розвитку (мал. 2.1).

Ранні
пристосування
і пристрої
для лічби

Немеханічні
обчислювальні
пристрої

Механічні
обчислювальні
пристрої

Електронні
обчислювальні
машини

 **Мал. 2.1**

Обчислювальна техніка поступово стала використовуватися не лише для обчислень, а й для вирішення інших завдань, наприклад, для автоматизації різних процесів, використання електронних засобів зв'язку, контролю обладнання, виконання офісних завдань, комп'ютерних ігор, навчання тощо. Кожна галузь, у свою чергу,

запровадила додаткові вимоги до комп'ютерного обладнання, яке постійно розвивається відповідно до цих вимог.

Удосконалення комп'ютерів постійно відбувається в кількох напрямках. По-перше, змінюють або ж використовують нові основні елементи, з яких виготовляють комп'ютер, — змінюється елементна база комп'ютерів. По-друге, змінюється програмне забезпечення. Крім того, вдосконалюються пристрої введення-виведення даних (периферія комп'ютера) та організація і взаємозв'язок його окремих складових.

Початком ери комп'ютерів вважають 1945–1946 рр., коли американські вчені Проспер Еккерт і Джон Моучлі сконструювали в Пенсильванському університеті (США) першу ЕОМ ENIAC (*Electronic Numerical Integrator and Calculator*). ENIAC мала 1800 електронних ламп, 150 000 електромеханічних реле, а її потужність становила 150 кВт. Зрозуміло, що ця машина була дуже громіздкою, складною в управлінні — щоб змінити програму, необхідно було перепаявати схему, що була ненадійною в роботі й мала низку інших недоліків. ENIAC відносять до першого покоління електронних обчислювальних машин (ЕОМ).

Під **поколіннями ЕОМ** розуміють усі типи й моделі електронних обчислювальних машин, розроблені конструкторськими колективами, але побудовані на тих само наукових і технічних принципах. Кожне наступне покоління визначається новими електронними елементами, технологія виготовлення яких є принципово іншою, обчислювальними можливостями, швидкодією та іншими властивостями.

Покоління ЕОМ (англ. *computer generation*) — один із класів у класифікації обчислювальних систем за ступенем розвитку апаратних і програмних засобів.

Зазвичай вирізняють чотири покоління електронної обчислювальної техніки (табл. 2.1).

Швидке розповсюдження комп'ютерів почалося з третього покоління.

Таблиця 2.1

Покоління ЕОМ	Роки	Електронні елементи	Швидкодія
I	1950–1960	Електровакуумні лампи (ENIAC, MECSM)	10–20 тис. оп./с
II	1960–1965	Транзистори	100–500 тис. оп./с
III	1965–1970	Інтегральні схеми	1 млн оп./с
IV	з 1970	Мікропроцесори	сотні млн оп./с



Електровакуумна
лампа



Транзистор



Інтегральні
схеми



Мікропроцесор

У нашій країні розробки ЕОМ починаються також у 1940-х рр. У 1951 р. в Києві під керівництвом професора С. А. Лебедева вводиться в експлуатацію ЕОМ, яку назвали МЭСМ (російською «Малая электронно-счётная машина»). Перші програми для цієї машини написала програмістка К. Л. Ющенко.

Значний внесок у розвиток вітчизняної комп'ютерної техніки зробив Віктор Михайлович Глушков (1923–1982) — фундатор наукової школи кібернетики, ініціатор й організатор реалізації науково-дослідних програм створення проблемно-орієнтованих програмно-технічних комплексів для інформатизації, комп'ютеризації і автоматизації господарської та оборонної діяльності країни, почесний член багатьох іноземних академій.

Починаючи з 1950-х рр., бурхливо розвивається обчислювальна техніка і за кордоном, і в Україні.



С. О. Лебедєв



К. Л. Ющенко



В. М. Глушков



Діємо

Вправа 1. Історія розвитку обчислювальної техніки.

Завдання. Доповніть таблицю у файлі *Покоління комп'ютерів*, що збережений у папці *Забезпечення комп'ютера*, зображеннями комп'ютерів кожного з поколінь і їхньої елементної бази, відомостями про видатних людей, які причетні до розвитку кожного з поколінь.

1. Створіть папку *Комп'ютери та програми* у своїй структурі папок.
2. Відкрийте файл *Музеї комп'ютерної техніки*, що збережений у папці *Забезпечення комп'ютера*, та оберіть одне з наведених посилань. Знайдіть зображення комп'ютерів кожного з поколінь, їх елементної бази й відомості про видатних людей, які причетні до розвитку кожного з поколінь.
3. Відкрийте файл *Покоління комп'ютерів*, що збережений у папці *Комп'ютери та програми*. Додайте до клітинок таблиці знайдені відомості.

4. Збережіть файл з іменем *Покоління комп'ютерів_Прізвище* в папці *Комп'ютери та програми* у своїй структурі папок.

2. Якими бувають сучасні комп'ютери?

У сучасному суспільстві комп'ютери використовуються в усіх сферах: від домашнього використання до складних космічних станцій.

Найпоширенішими є персональні комп'ютери, які поділяють на декілька видів залежно від їхніх функцій, призначення та будови.

Настільні комп'ютери — це комп'ютери, які потребують постійного підключення до живлення; зазвичай їх розташовують на столі й не переносять під час роботи.

До них належать:



настільні (десктопи) — складаються з окремого системного блоку, до якого підключаються монітор, клавіатура, миша й інші пристрої;



неттопи — вид стаціонарних персональних комп'ютерів (ПК). Системний блок компактних розмірів, як правило, з невеликими енерговитратами й шумом. Через невеликі розміри неттопи мають більш низьку продуктивність, зате чудово вписуються в домашній інтер'єр і не займають багато місця в офісах;



моноблоки — вид комп'ютерів, у яких системний блок і монітор мають вигляд єдиного цілого.

Портативні комп'ютери — це комп'ютери, у яких є батарея, що дає змогу пристрою певний час працювати без підключення до електромережі. Вони менші за розміром і вагою, тому їх можна легко переносити із собою.

До портативних комп'ютерів належать:



ноутбуки — це комп'ютери, у корпусі яких поєднуються стандартні комплектуючі портативного комп'ютера (дисплей, клавіатура, тачпад, батарея). Вони можуть виконувати всі функції настільного комп'ютера;



нетбуки — пристрої, схожі на ноутбук, тільки менші за розміром та з низькою продуктивністю, тому їх використовують для виконання обмеженого набору завдань;



планшети — портативні комп'ютери із сенсорним екраном;



планшетні ноутбуки — це різновид ноутбуків, у яких є і сенсорний екран, і клавіатура для введення даних. Залежно від моделі клавіатура в таких пристроях може складатися або висуватися.

Кишенькові комп'ютери — це вид портативних комп'ютерів, які вирізняються маленьким розміром і потужною батареєю, що дає змогу переносити та використовувати пристрій протягом тривалого часу. Засіб введення в таких пристроях — або сенсорний екран, або висувна клавіатура.

До кишенькових комп'ютерів належать:



комунікатори — це мобільні телефони, доповнені деякими функціями комп'ютера;



кишенькові персональні комп'ютери — це мініатюрні персональні комп'ютери з клавіатурою або без неї;



смартфони — це компактні персональні комп'ютери, доповнені функціями мобільного телефона.

До комп'ютерів, які використовують в особливих випадках, відносять суперкомп'ютери, сервери, спеціалізовані комп'ютери та мікрокомп'ютери.

Суперкомп'ютери (Ельбрус, моделі серії *Cray*) — це багатопроцесорні системи, які виконують мільярди дій за секунду. Вартість такої машини — декілька мільйонів доларів. Їх використовують у косміч-

них і військових дослідженнях, для опрацювання геодезичної інформації, для моделювання глобальних і регіональних процесів в атмосфері, океанах і земній корі, моделювання фармацевтичних сполук і роботи штучного інтелекту. Також існують комп'ютери, націлені на максимально точне відтворення «архітектури» головного мозку людини, як, наприклад, розробка вчених Манчестерського університету — суперкомп'ютер *SpiNNaker*.



Сервер — це потужний комп'ютер, чи вартісна багатопроцесорна система великої продуктивності, що надає багатьом користувачам доступ до спільної оперативної і дискової пам'яті великої ємності.



Мікрокомп'ютери — це комп'ютери, які є складовими різних пристроїв промислового й побутового призначення (фотоапаратів, автомобілів, літаків тощо). Їх використовують також для підтримки технології інтернету речей.

3. Яку архітектуру мають сучасні комп'ютери?

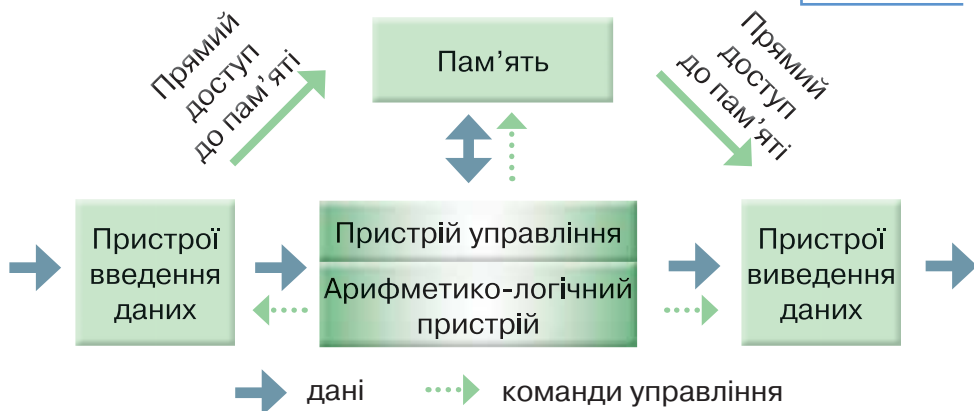
Попри те, що сфери застосування сучасних комп'ютерів практично необмежені, в їхню основу покладено єдину логічну схему (архітектуру комп'ютера) та єдині принципи роботи. Вони визначають загальну структуру, без урахування особливостей тих чи інших моделей, і відображають основні зв'язки між пристроями комп'ютера, основні потоки даних, які циркулюють між ними, та принципи їх опрацювання. Уніфікація архітектури ПК забезпечує їхню сумісність із точки зору користувача.

Архітектура ПК — принципи роботи та взаємодії основних пристроїв комп'ютера: процесора, внутрішньої і зовнішньої пам'яті та пристроїв введення-виведення.

В основу більшості моделей сучасних комп'ютерів покладено архітектуру Джона фон Неймана — опис логічної організації ЕОМ.

Основні складові фон-нейманівської машини такі: пристрій управління, арифметико-логічний пристрій, пам'ять, пристрої введення та виведення даних (мал. 2.2).

Мал. 2.2



 Це цікаво

Принципи, згідно з якими функціонує більшість сучасних комп'ютерів, опубліковано в 1946 р. американським математиком Джоном фон Нейманом (1903–1957). Він також описав машину, яка може бути універсальним засобом опрацювання даних.

Принципи роботи комп'ютера за фон Нейманом:

1. За допомогою пристроїв введення дані та програми їх опрацювання потрапляють у пам'ять комп'ютера.
2. З пам'яті комп'ютера дані надсилають до процесора.
3. Арифметико-логічний пристрій здійснює опрацювання даних.
4. Пристрій управління керує процесами опрацювання даних, їх збереженням і передаванням.
5. Пристрої виведення даних здійснюють подання результатів опрацювання даних у вигляді, зручному для користувача.



Джон фон
Нейман

Працюємо в парах

1. Наведіть три приклади різних видів комп'ютерів, з якими ви маєте справу в повсякденному житті. Визначте їхні спільні ознаки. Результати подайте діаграмою Венна. Порівняйте отриманий результат з діаграмою іншої пари.
2. Знайдіть в інтернеті інформацію про найпотужніші комп'ютери світу. Визначте призначення цих комп'ютерів. Перевірте, чи входить до Топ-5 таких комп'ютерів найпотужніший комп'ютер в Україні, передбачте можливу причину. Порівняйте причину, визначену у вашій парі, з причиною, що названа іншою парою. Зробіть висновки.
3. Розгляньте складові, які використовували в різних поколіннях комп'ютерів (<https://cutt.ly/dkPOvf7>). Обговоріть, чому зміна елементної бази комп'ютерів приводила до технологічного стрибка та більш широкого розповсюдження комп'ютерної техніки у світі.



Працюємо самостійно

4. Здійсніть віртуальну екскурсію музеєм ІТ в Сумському державному університеті <https://cutt.ly/0kntJ4j>. У текстовому документі складіть план екскурсії.
5. Побудуйте схему класифікації сучасних комп'ютерів засобами текстового процесора. Надішліть схему на спільний ресурс класу. Порівняйте її зі схемою іншого учня чи учениці.
6. Сплануйте та складіть презентацію «Україна в історії обчислювальної техніки». Використайте матеріали сайту *Інформаційні технології в Україні: історії та особистості* (<http://ua.uacomputing.com/>).
7. Попрацюйте в мініпроекті «Історія комп'ютерної техніки: про що варто знати сучасникам?». Доберіть спосіб подання результатів проекту. Використайте для підготовки проекту матеріали сайту <https://cutt.ly/Oknyelo>.





Призначення і характеристики складових комп'ютера

ПРИГАДАЙТЕ:

- ▶ класифікацію пристроїв комп'ютера;
- ▶ інформаційні процеси та пристрої, що їх реалізують.

ВИ ДІЗНАЄТЕСЬ:

- ▶ чи може комп'ютер працювати без процесора;
- ▶ які пристрої комп'ютера розміщуються всередині системного блока;
- ▶ що і як можна зберігати в пам'яті комп'ютера;
- ▶ які особливості мають пристрої введення-виведення даних;
- ▶ які пристрої входять до складу мультимедійного обладнання;
- ▶ як обрати найкращий комп'ютер для роботи.



Вивчаємо

1. Чи може комп'ютер працювати без процесора?

Процесор називають електронним «мозком» комп'ютера. Він призначений для автоматичного опрацювання і перетворення даних за наперед введеними програмами, а також для управління роботою всіх пристроїв комп'ютера. Від його обчислювальної потужності здебільшого й залежить продуктивність комп'ютера.

Процесор — це мікросхема, яка створюється на напівпровідниковому кристалі (або кількох кристалах) шляхом застосування складної мікроелектронної технології. Різноманітні операції в процесорі виконуються за спеціальними вказівками. Вказівки для процесора записують у комп'ютерній програмі (мал. 3.1).



Мал. 3.1

Під час роботи процесор може сильно нагріватися, тому на нього встановлюють систему охолодження — вентилятор, який називають кулером.



Це цікаво

Процесор ще іноді називають *CPU* — від англ. *Central Processing Unit* — модуль центрального процесора.

Кулер — від англ. *cooler* — охолоджувач.

Процесор складається з:

- арифметико-логічного пристрою — для виконання арифметичних і логічних операцій з даними;
- пристрою управління — для управління функціонуванням усіх складових комп'ютера;
- регістрів власної пам'яті.

Під час роботи процесор опрацьовує дані. Частина даних інтерпретується як власне дані, частина даних — як адресні дані, а частина — як вказівки. Сукупність різноманітних вказівок, які може виконати процесор над даними, утворює так звану систему вказівок процесора.

Основними характеристиками процесорів є:

- *mun* — відповідно до фірми-виробника розрізняють процесори *Intel (Pentium, Celeron, Core2 Duo, Core i series* тощо), *AMD (AMD64, Duron, Athlon, Ryzen* тощо) та інші;
- *тактова частота* — визначає кількість виконуваних елементарних операцій за одну секунду, тобто швидкодію процесора; тактова частота сучасних процесорів вимірюється в гігагерцах (ГГц); нині пікова частота процесора становить 8,429 ГГц;
- *розрядність* — максимальна довжина двійкового коду, який може опрацьовуватись або передаватись процесором; що вищою є розрядність, то вища потужність процесора;
- *кеш-пам'ять* — це внутрішня пам'ять процесора, яка дає змогу зберігати проміжні дані.

Режим роботи процесора задається спеціальною мікросхемою, яка має назву генератор тактової частоти. Цей пристрій визначає ритм і швидкість роботи — на виконання процесором кожної операції відводиться певна кількість тактів.

Кеш процесора — кеш, який використовується центральним процесором для скорочення середнього часу доступу до пам'яті комп'ютера. Коли процесору потрібно звернутися в пам'ять для читання або запису даних, він спочатку перевіряє, чи є їхні копії в кеші.

Більшість сучасних мікропроцесорів для настільних комп'ютерів і серверів мають щонайменше три незалежні кеші: кеш інструкцій для прискорення завантаження машинного коду, кеш даних для прискорення читання і запису даних та буфер асоціативної трансляції для прискорення трансляції віртуальних (логічних) адрес у фізичні, як для інструкцій, так і для даних. Кеш даних часто реалізується у вигляді багаторівневого кешу.

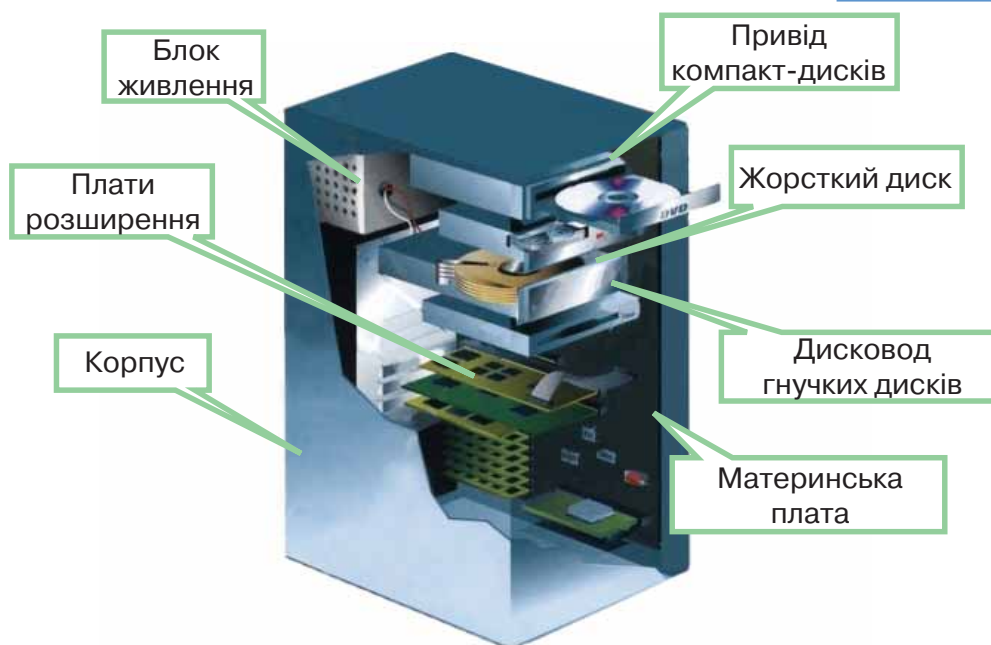
Обсяг кеш-пам'яті третього рівня сучасних процесорів може досягати 256 МБ.

Розрядність пов'язана з розмірами спеціальних комірок пам'яті, що містяться в самому процесорі й називаються регістрами. Процесор з регістром 1 байт (8 бітів) називають 8-розрядним, 2 байти — 16-розрядним, 4 байти — 32-розрядним. Найпотужніші комп'ютери на сьогодні мають 8-байтові регістри (64 розряди).

2. Які пристрої комп'ютера розміщуються всередині системного блока?

Деякі пристрої розміщуються всередині системного блока комп'ютера (мал. 3.2), а інші приєднують до системного блока, тому вони належать до зовнішніх.

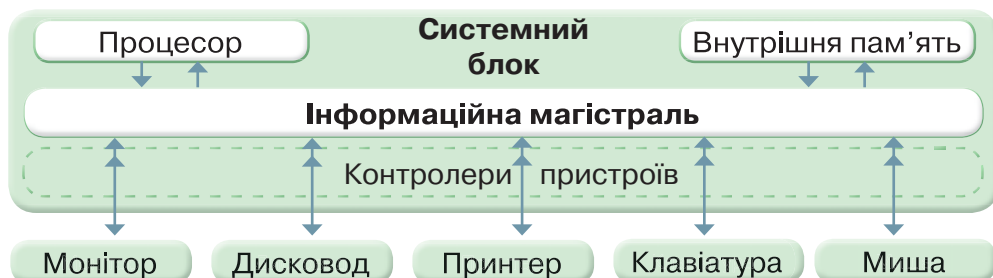
Мал. 3.2



Усередині комп'ютера розміщується **системна плата**, яку ще називають **материнською**. На ній встановлюються процесор, внутрішня пам'ять комп'ютера та інші пристрої. Процесор з'єднується з іншими пристроями пам'яті та пристроями для передавання даних і службо-

вих сигналів за допомогою набору електронних ліній, які називають **магістраллю (шиною)**. Користувач може створювати різні конфігурації комп'ютера, приєднуючи до магістралі окремі модулі різних пристроїв введення-виведення, пам'яті тощо (мал. 3.3).

Мал. 3.3



Для магістралі характерна така організація: через одну групу проводів (шину даних) передаються дані, що опрацьовуються, через іншу (шину адрес) — адреси пам'яті або зовнішніх пристроїв, до яких «звертається» процесор. Через третю частину магістралі (шину управління) передаються керуючі сигнали (наприклад, перевірка готовності пристрою до роботи, сигнал до початку роботи пристрою та ін.).

Користувач може змінювати набір пристроїв комп'ютера. Апаратне під'єднання зовнішніх пристроїв до магістралі здійснюється через контролери й адаптери — електронні мікросхеми, за допомогою яких узгоджують роботу зовнішніх пристроїв. Вони призначені для перетворення даних, що надходять із процесора, на відповідні сигнали, за допомогою яких здійснюється управління роботою пристрою. Їхні роз'єми виведені на задній панелі системного блока, і за допомогою відповідних кабелів до них приєднують зовнішні пристрої.



Діємо

Вправа 1. Системний блок.

Завдання. В емуляторі системного блока розмістіть пристрої в правильному місці та правильному порядку.

1. Відкрийте програму *Системний блок*, збережену в папці *Забезпечення комп'ютера*.
2. Послідовно розміщуйте блок живлення, материнську плату та інші пристрої в місце їх правильного розташування.
3. Перевірте, чи отримано правильний результат.

3. Що і як можна зберігати в пам'яті комп'ютера?

Пам'ять комп'ютера призначена для зберігання даних і програм. Її можна поділити на внутрішню і зовнішню (мал. 3.4).

Пристрої внутрішньої пам'яті виготовляють у вигляді мікросхем (модулів), які вставляються в спеціальні роз'єми на материнській платі.



Мал. 3.4

Внутрішня пам'ять комп'ютера поділяється на оперативний запам'ятовуючий пристрій (ОЗП), постійний запам'ятовуючий пристрій (ПЗП), напівпостійний програмований запам'ятовуючий пристрій (НПЗП), відеопам'ять і кеш-пам'ять.

Розглянемо пристрої внутрішньої пам'яті.

Оперативний запам'ятовуючий пристрій — ОЗП (RAM — від англ. *Random Access Memory* — пам'ять з довільним доступом) — швидка й енергозалежна пам'ять (мал. 3.5). Оперативна пам'ять призначена для тимчасового зберігання вхідних даних, проміжних і кінцевих результатів обчислень, програм опрацювання даних. Це своєрідний робочий простір для комп'ютера. ОЗП може використовуватися як для читання даних, так і для записування. Дані в ОЗП зберігаються доти, поки на їхнє місце не будуть записані нові дані. При вимкненні електроживлення дані в ОЗП втрачаються. Планки оперативної пам'яті, які серійно виробляються для комп'ютерів, мають обсяги до 64 ГБ.



Мал. 3.5

Постійний запам'ятовуючий пристрій — ПЗП (ROM — від англ. *Read Only Memory* — пам'ять тільки для читання) — швидка й енергонезалежна пам'ять (мал. 3.6). Дані заносяться до неї один раз назавжди (як правило, в заводських умовах) і зберігаються постійно (при ввімкненому й вимкненому живленні). Постійна пам'ять — мікросхема, у якій містяться програми для управління роботою комп'ютера та програми тестування основних складових комп'ютера, а також набір програм для управління всіма його пристроями (BIOS — від англ. *Basic Input/Output System* — базова система введення-виведення). Постійна пам'ять також розміщена на материнській платі.



Мал. 3.6

Дані, що зберігаються в **напівпостійному програмованому запам'ятовуючому пристрої** — НПЗП (пам'ять, виконана за технологією CMOS — від англ. *Complementary Metal-Oxide Semiconductor* — технологія виготовлення мікросхем), можуть бути замінені в режимі програмування, коли користувач має спеціальні знання та може написати спеціальні програми для управління комп'ютером. До таких даних належать дані щодо зберігання і зміни конфігурації комп'ютера, календаря і годинника. НПЗП також називають пам'яттю автономного живлення, або пам'яттю «на батарейках», оскільки дані зберігаються за допомогою акумуляторної батарейки, за своїми функціями подібні до батарейок кварцових годинників.

Відеопам'ять — швидка оперативна пам'ять для зберігання коду зображення, що відображається на екрані монітора. Відеопам'ять (VRAM — від англ. *Video Random Access Memory*) може бути різної ємності; розміщена на відеокарті (мал. 3.7). Найпродуктивніші відеокарти застосовують для комп'ютерних ігор або для роботи з просторовими зображеннями.



Мал. 3.7

Що більшою є ємність відеопам'яті комп'ютера, то більшими є можливості відображення на моніторі зображень з високою роздільною здатністю і великою кількістю кольорів. Ємність відеопам'яті сучасних відеокарт здебільшого становить 4–26 ГБ і більше.

Для тривалого зберігання даних призначено **зовнішню пам'ять**, або носії даних:

- жорсткий магнітний диск (англ. HDD — від англ. *Hard Disk Drive*), або вінчестер (мал. 3.8). Як правило, він вбудований разом з дисководом у корпус системного блока (але може бути розміщений і зовні);
- твердотілий накопичувач (англ. SSD, *Solid-State Drive*) — комп'ютерний запам'ятовувальний пристрій на основі мікросхем пам'яті та контролера керування ними, що не містить рухомих механічних частин (мал. 3.9);
- лазерні диски (CD-ROM, CD-R, CD-RW чи DVD) (мал. 3.10):
— диски CD-ROM (від англ. *Compact Disk Read Only Memory* — компакт-диски тільки для читання) — високонадійні диски для зберігання даних, довговічні (прогнозований термін придатності — до 50 років). Принцип запису і зчитування — оптичний;



Мал. 3.8



Мал. 3.9



Мал. 3.10

— диски CD-R (від англ. *Compact Disc Recordable* — компакт-диск із одноразовим записуванням) — це різновид оптичного диска, на якому можна записати файли за допомогою записуючого пристрою.

Записані дані можна прочитати з диска за допомогою приводу CD-ROM або відтворити в програвачі компакт-дисків (якщо це музика).

На цей вид оптичних дисків можна записувати додаткові дані, аж доки закінчиться вільне місце. Зазвичай на диску CD-R можна помістити 650 МБ даних, або 74 хв музики. На CD-R дисках новішого покоління можна додатково збільшити цей ліміт до 737 МБ, або 80 хв музики;

— диски CD-RW (від англ. *Compact Disc ReWritable* — компакт-диски з можливістю перезаписування) — це інший вид оптичного диска, на якому можна не тільки записувати дані, а й вилучати й перезаписувати їх. Вони мають таку само ємність, як CD-R;

— диски DVD (від англ. *Digital Video Disc* — цифровий відеодиск або *Digital Versatile Disc* — цифровий багатофункціональний диск) —

це різновид носія даних, який зовні нагадує диск CD-ROM. Однак на DVD-диску можна записати значно більше даних.

Стандартна ємність цих носіїв становить 4,7 ГБ, хоча трапляються диски вдвічі більшої ємності. На DVD-диску можна записати з досконалою якістю повнометражний фільм у кількох мовних версіях;

— флеш-пам'ять, або USB-накопичувачі (від англ. *Universal Serial Bus* — універсальна послідовна шина), під'єднуються безпосередньо до порту USB на комп'ютері (мал. 3.11). Вони є мікросхемами й можуть зберігати до кількох терабайтів даних.

Мал. 3.11



4. Які особливості мають пристрої введення-виведення даних?

Пристрої введення-виведення призначені для введення даних до комп'ютера та виведення результатів їх опрацювання у вигляді, зручному для користувача (мал. 3.12).

Мал. 3.12



З основними пристроями введення даних ви вже ознайомились раніше.

Основним пристроєм виведення даних є **монітор (дисплей)**. Монітори мають такі характеристики:

- якість відображення кольорових зображень, тобто кількість кольорів для відображення;
- роздільна здатність, що визначається кількістю точок (пікселів) на екрані, які використовуються для створення зображення.

Роздільна здатність подається як добуток кількості пікселів по горизонталі на кількість пікселів по вертикалі; наприклад, 1920×1080 або 1680×1050 . Що вищою є роздільна здатність, то більше деталей зображення можна відобразити;

- довжина діагоналі в дюймах, розмір зерна (відстань на екрані між двома точками однакового кольору);
- максимальна частота відновлення зображення, на яку здатен монітор (її вимірюють у герцах), наприклад, 120 Гц чи 240 Гц. Що вищою є частота відновлення, то більшою буде плавність відтворення зображення.

За принципом виготовлення розрізняють монітори на основі **електронно-променевої трубки** (мал. 3.13) та **рідкокристалічні монітори** (мал. 3.14, 3.15).

У рідкокристалічних моніторів — LCD (від англ. *Liquid Crystal Display*) — повністю відсутнє шкідливе електромагнітне випромінювання, використовується менше електроенергії, не створюється ефект мерехтіння, не спотворюються зображення. Вони мають плоский екран (мал. 3.14) або новітні моделі — вигнутий екран (мал. 3.15), і займають менше місця на столі. У свою чергу, рідкокристалічні монітори поділяють на TFT, IPS, TN, VA та ін.



Мал. 3.13



Мал. 3.14



Мал. 3.15

Для виведення зображень на монітор використовують відеопам'ять, що розташовується на відеокарті. Усе, що користувач бачить на екрані монітора, міститься у відеопам'яті, з якої надходять відеосигнали до монітора. **Відеокарта**, яку також називають **відеоадаптером**, установлюється всередині комп'ютера та використовується для підключення монітора й надсилання до нього графічних даних.

Параметри відеокарти визначають максимальні роздільну здатність, кількість кольорів і частоту відновлення зображення. Що більшим є обсяг відеопам'яті, то більшою кількістю пікселів на екрані може керувати відеоадаптер, тобто мати більш високу роздільну здатність екрана. Що більший обсяг відеопам'яті використовується для керування одним пікселем, то більшою буде кількість відтворюваних кольорів, отже, багатшою — кольорова палітра монітора.

Сучасні комп'ютери мають монітори здебільшого з такими характеристиками:

- роздільна здатність — 1920×1080 (2560×1600 , 3996×2160);
- розмір екрана — 15 – 42 дюйма.

Для виведення даних на папір, плівку чи інший носій використовують **принтери** (мал. 3.16). Основними характеристиками принтерів є якість друку (dpi — точок на дюйм) та швидкість друку (сторінок за хвилину).



Мал. 3.16

Для домашнього використання зазвичай використовують струменеві принтери, які дають змогу друкувати як чорно-білі документи, так і кольорові. У струменевих принтерах для формування зображення використовують спеціальні сопла, через які на папір подаються чорнила. Тонкі, як волосся, сопла розміщені на головці принтера, де встановлено резервуар з рідким чорнилом, яке, як мікрочастинки, переноситься через сопла на матеріал носія. Кількість сопел залежить від моделі принтера та його виробника. Зазвичай їх буває від 16 до 64. Деякі останні моделі мають набагато більше число сопел, наприклад, головка принтера DeskJet 1600 має 300 сопел для чорних чорнил і 416 — для кольорових. Такі принтери здійснюють якісний малощумний друк. Але вони вимогливі до паперу, потребують дорогих розхідних матеріалів, чорнило може розпливатись у воді.

Невисока якість чорно-білого друку до 300 dpi, можливість друку кількох копій одночасно під копіювальний папір і достатня дешевизна забезпечують використання матричних принтерів у банках і для друку чеків. Працює матричний принтер шумно та з низькою швидкістю. В офісах, де є потреба в друці великої кількості документів з невеликим шумом, використовують лазерні принтери. Якість друку

на таких принтерах сягає до 1200 dpi 25–50 сторінок за хвилину. Розхідні матеріали для них відносно недорогі, хоча є вимоги до якості паперу.

3D-принтер — пристрій, що використовує метод пошарового створення фізичного об'єкта за цифровою 3D-моделлю (об'ємною). Існує декілька технологій 3D-друку від формування об'єкта з порошку до поступового його нашарування зі спеціального полімеру. Вони використовуються як у малосерійній інженерії, так і у виробництві складних систем, навіть біологічних. Ці технології останнім часом дуже швидко розвиваються, поступово стають дешевшими й доступнішими користувачам.

5. Які пристрої входять до складу мультимедійного обладнання?

Використовувати мультимедійні програми й опрацьовувати мультимедійні дані можна лише за наявності в комп'ютері відповідного обладнання.

Мінімальний набір мультимедійного обладнання складається зі **звукової карти** (плати, що приєднується до материнської плати), до якої через відповідну панель системного блока під'єднується **акустична система (колонки)**, та **накопичувача для оптичних дисків**. Звук, який чує користувач комп'ютера, — результат роботи двох взаємопов'язаних компонентів: звукової карти (мал. 3.17) та акустичної системи. Їхній вибір залежить від потрібної якості звуку та сфери використання ПК (ігри, домашній мультимедійний центр, домашній кінотеатр для перегляду DVD-відео тощо). Проте якість відтворення звуку залежить не лише від пристроїв, а й від програмного забезпечення.



Мал. 3.17

Важливими характеристиками акустичних систем є такі.

- **Діапазон частот.** Зазвичай — у межах від 20 Гц до 20 кГц. Це доволі широкий діапазон, і для його відтворення потрібно кілька динаміків.
- **Кількість динаміків.** Кожний динамік відтворює свій вузький діапазон частот.
- **Потужність.** Становить від 2 до 180 Вт.

Замість колонок можна використовувати навушники (мал. 3.18). Роз'єм для навушників є в багатьох колонках або на задній панелі системного блока. Популярними на сьогодні є колонки й навушники з просторовим звуком.



Мал. 3.18

До ширшого комплекту мультимедійної системи належать **мікрофон, відеокамера, відеоманітофон, цифрова фотокамера** тощо.

За допомогою мікрофона можна записати звуковий фрагмент і зберегти його як файл. Зазвичай мікрофон використовують для спілкування в інтернеті засобами IP-телефонії.

Мікрофон не є базовим мультимедійним пристроєм, тому перед покупкою варто дізнатися про сумісність мікрофона зі встановленою звуковою платою.

Мікрофон під'єднується до відповідного роз'єму звукової плати або до лінійного входу. У більшості мікрофонів є вимикач для вимикання вихідного сигналу (замість від'єднання від звукової плати).

Фізичне під'єднання зазначених пристроїв має супроводжуватися встановленням відповідних програм — **драйверів** (від англ. *to drive* — управляти, вести), які управляють роботою зовнішніх пристроїв комп'ютера. Як правило, відповідні драйвери розміщуються на CD-дисках і входять до комплекту під час продажу пристрою, оскільки кожний тип зовнішнього пристрою має індивідуальний драйвер. Створюючи мультимедійний центр, слід пам'ятати, що ефективна робота на комп'ютері з відео та графікою потребує особливих характеристик процесора, оперативної пам'яті, жорсткого диска.

6. Як обрати найкращий комп'ютер для роботи?

Основні характеристики персонального комп'ютера залежать від характеристик його складових. До них належать:

- тип, швидкодія процесора (тактова частота, розрядність);
- ємність оперативної пам'яті;
- тип і ємність жорсткого диска (або SSD) та час доступу до нього;
- розмір, тип і швидкість відеокarti та ємність відеопам'яті;
- тип монітора та його характеристики — роздільна здатність, розмір по діагоналі, кількість точок;
- наявність мультимедійних компонентів: CD-ROM, звукова плата (частота сигналу, кількість голосів, які можна одночасно відтворювати за допомогою плати), гучномовці, TV-тюнер, мікрофон, навушники;
- роздільна здатність сканера, види операцій, які можна за його допомогою виконувати (сканування малюнків, тексту, плівок, слайдів тощо);
- тип принтера — якість і швидкість друку, підтримка кольору.

Треба також зважати на те, що деякі складові комп'ютера впливають на швидкість перетворення даних, тобто поліпшують продуктивність комп'ютера (наприклад, пам'ять RAM, відеокarta), а інші поліпшують комфорт роботи користувача (ємність жорсткого диска, розмір і яскравість монітора, якість звукової карти). Але в будь-якому разі для здійснення зваженого вибору комп'ютера необхідно бути обізнаними щодо призначення, функціональності й особливостей роботи основних його складових. **Офісні** персональні комп'ютери призначені для опрацювання і зберігання насамперед текстових та

числових даних. **Домашні** комп'ютери використовуються не тільки для розв'язування задач, а й для відтворення музики, відеофільмів, створення комп'ютерних фотоальбомів тощо. А **ігрові** персональні комп'ютери — для комп'ютерних ігор. Вони можуть мати додаткові ігрові пристрої введення і виведення даних: джойстик, руль і педалі для комп'ютерних перегонів. Якщо користувач не збирається працювати з професійними графічними програмами або опрацьовувати велику кількість даних, то йому не потрібні найдорожчі моделі, й можна вибрати «повільніший» процесор.

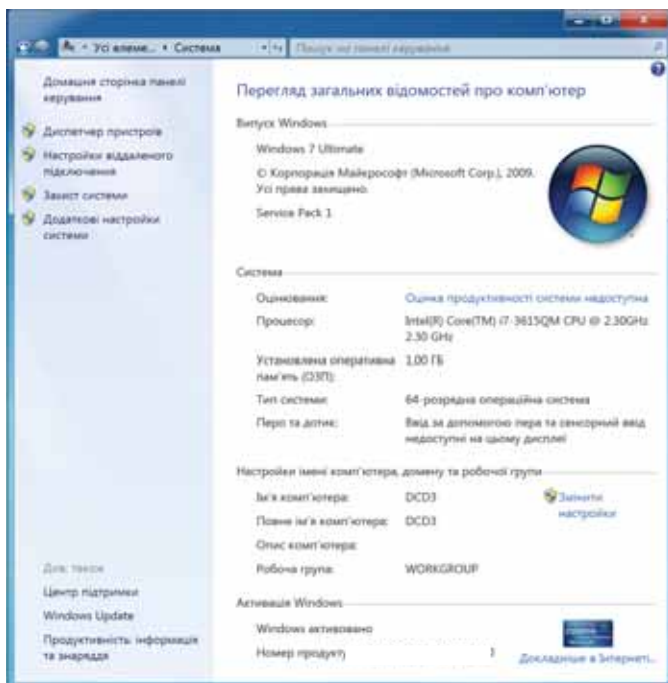


Досліджуємо

Вправа 2. Конфігурація комп'ютера.

Завдання. Визначте характеристики комп'ютера, за яким ви працюєте.

1. Клацніть правою кнопкою миші на значку *Комп'ютер* на *Робочому столі* та оберіть вказівку *Властивості*.
2. Перегляньте та проаналізуйте такі характеристики комп'ютера в області *Перегляд загальних відомостей про комп'ютер* (мал. 3.19): версія операційної системи, характеристики процесора, обсяг оперативної пам'яті, тип системи.
3. Порівняйте визначені характеристики із запропонованими на малюнку 3.19. Зробіть висновок про призначення досліджуваних комп'ютерів і про один з них.



Мал. 3.19

Працюємо в парах

1. Для чого призначені окремі пристрої комп'ютера? Обговоріть у парах: хтось називає пристрій, що належить до комп'ютера, а сусід чи сусідка по парті — його призначення; потім поміняйтеся ролями та стежте, щоб назви пристроїв не повторювались.
- ★ 2. Упорядкуйте носії даних за ступенем надійності, вартості, швидкості доступу, новизни тощо. Результати обговоріть у парах.
3. Чи тільки в комп'ютерах є процесори? Назвіть інші технічні пристрої, які можуть містити процесор. Де використовуються ці пристрої? Які функції виконують процесори в наведених вами прикладах? Чи схожі їхні функції на функції процесора ПК? Чи можуть існувати ці пристрої без процесорів? Чи зміняться при цьому їхні призначення та ефективність?
4. Обговоріть, без яких пристроїв не можна слухати музику на комп'ютері. За яких умов можна прослухати на комп'ютері концерт класичної музики, що записаний на CD? За яких умов на комп'ютері можна переглянути відеофільм? А грати в комп'ютерну гру?
- ★★ 5. Попрацюйте в групах у мініпроекті «Пристрої комп'ютера: як технічний прогрес впливає на їх удосконалення?». Побудуйте лінію часу для обраного пристрою, на якій продемонструйте етапи зміни характеристик пристрою та спроєктуйте їх на процеси зміни в технологіях. Підготуйте виступ із презентацією свого проекту перед класом.

Працюємо самостійно

6. Доберіть з інтернету зображення пристроїв для моделювання комп'ютера для: а) навчання і досліджень; б) ігор і розваг; в) бюджетної організації.
- ★ 7. Складіть карту знань класифікації пристроїв комп'ютера. Передбачте в ній стандартні та додаткові пристрої введення, виведення, запам'ятовування і процесор. Скористайтесь для довідки файлом *Додаткові пристрої*, що збережений у папці *Забезпечення комп'ютерів*.
- ★ 8. Визначте конфігурацію домашнього комп'ютера. Скористайтесь цінами в інтернет-магазині та визначте теперішню вартість подібного комп'ютера. Зробіть висновок.
- ★★ 9. Знайдіть у довідковій літературі відомості про небезпеки, на які можуть наражатись користувачі комп'ютерів. Як уникати цих небезпек? Обговоріть, як краще подати отримані відомості, щоб поділитися з ними у класі. Складіть план виступу.

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Умію кодувати й декодувати повідомлення за певними правилами.
- ☐ Розумію поняття двійкового коду.
- ☐ Можу назвати одиниці вимірювання довжини двійкового коду та пояснити їх співвідношення.
- ☐ Можу описати загальний принцип побудови таблиці кодів символів.
- ☐ Розумію та можу пояснити деякі принципи кодування графічних даних.
- ☐ Можу навести приклади застосування сучасних пристроїв у різних галузях.
- ☐ Розумію та можу пояснити зв'язок між інформаційними технологіями та потребами й етапами розвитку людського суспільства.
- ☐ Розумію та можу пояснити основні характеристики складових обчислювальних пристроїв і сутність їх взаємодії.
- ☐ Розумію та можу пояснити процес обробки даних комп'ютерними пристроями.
- ☐ Умію визначати характеристики складових персонального комп'ютера залежно від його призначення.

Розділ 2

Опрацювання текстових даних

4



Редагування і форматування текстового документа

ПРИГАДАЙТЕ:

- ▶ призначення текстових процесорів;
- ▶ з якими об'єктами працюють у середовищі текстового процесора;
- ▶ дії з фрагментами тексту;
- ▶ форматування об'єктів текстового документа.

ВИ ДІЗНАЄТЕСЬ:

- ▶ як виконувати заміну фрагментів тексту в *Microsoft Word*;
- ▶ які особливості заміни фрагментів тексту в *LibreOffice Writer*;
- ▶ як до тексту вставити символи, яких немає на клавіатурі;
- ▶ як застосувати стиліове форматування для оформлення документа;
- ▶ як додати формули до текстового документа.



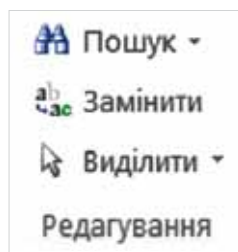
Вивчаємо

1. Як виконувати заміну фрагментів тексту в *Microsoft Word*?

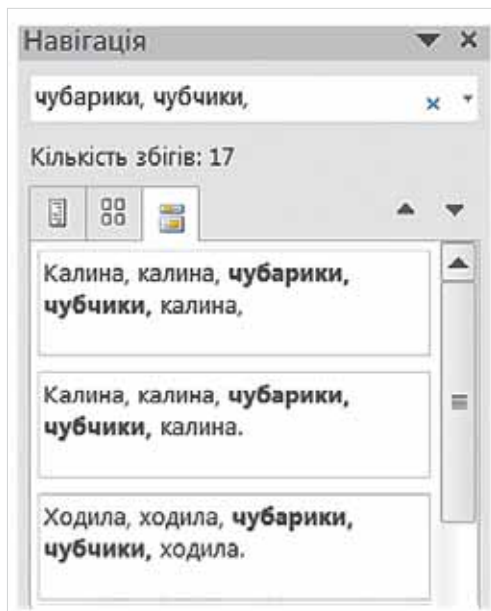
Ви вже знаєте, що внесення змін до змісту текстового документа називають його редагуванням. Цей процес передбачає також пошук і заміну фрагментів тексту та інших об'єктів.

Пошук і заміну фрагментів тексту можна здійснювати автоматизовано. Для цього використовується група інструментів *Редагування* на вкладці *Основне* (мал. 4.1).

Для пошуку потрібного фрагмента тексту в документі слід виконати вказівку *Пошук*. У полі пошуку панелі *Навігація* потрібно ввести шукану фразу, наприклад, *чубарики, чубчики* (мал. 4.2).



Мал. 4.1



Мал. 4.2

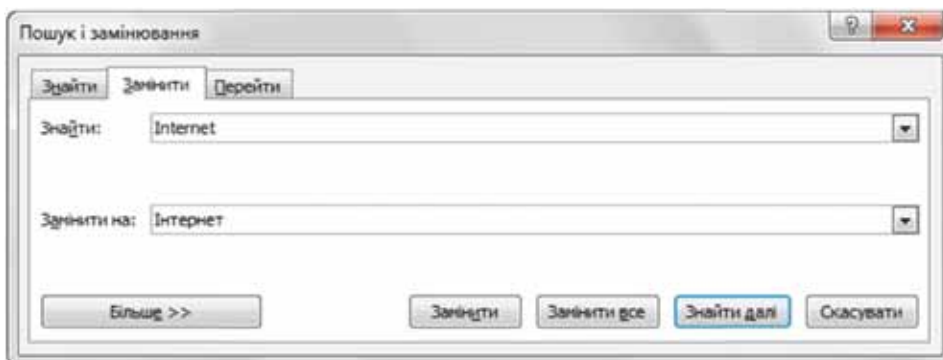
У результаті знайдена фраза буде виділена в тексті іншим кольором. Якщо потрібна фраза трапляється в тексті не один раз, усі збіги відображаються на панелі навігації. Для переходу слід натиснути на потрібному фрагменті.

Якщо в документі необхідно замінити слово (або фразу) на інше, можна використати засіб автоматичної заміни за таким алгоритмом.

1. Установити текстовий курсор на початок документа або в місце, починаючи з якого необхідно шукати слово, яке треба замінити.

2. Виконати вказівку *Замінити* за допомогою однойменного інструмента.

3. У діалоговому вікні *Пошук і замінування* в поле *Знайти* ввести слово, яке необхідно шукати в документі, а в поле *Замінити на* ввести слово, на яке необхідно замінити (мал. 4.3).



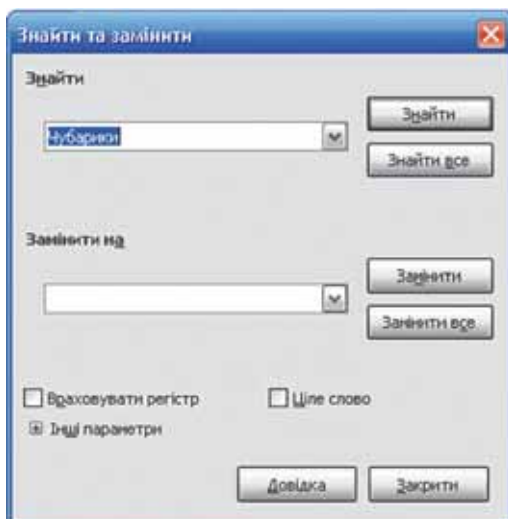
Мал. 4.3

2. Які особливості заміни фрагментів тексту в LibreOffice Writer?

У текстовому процесорі *LibreOffice Writer* режим редагування буде доступний, якщо на панелі інструментів *Стандартна* обрати інструмент . Для пошуку потрібного фрагмента в меню *Зміни* обирають вказівку *Знайти* (*Ctrl + F*), яка викликає панель інструментів *Знайти*.



У полі *Знайти* вказують фрагмент пошуку, наприклад, *Чубарики*. Стрілки  (*Ctrl + H*) у вікні, що відкривається, задають умови пошуку та заміни (мал. 4.4).



Мал. 4.4



Діємо

Вправа 1. Заміна фрагментів тексту.

Завдання. Виконайте заміну фрагмента тексту *Чубарики*, *чубчики* в документі *Ой у лузі калина* на текстовий фрагмент *Комарики*, *дзюбрики*.

1. Відкрийте документ *Ой у лузі калина*, що зберігається в папці *Текстовий процесор*.
2. Виділіть фрагмент тексту *Чубарики*, *чубчики*, що міститься в четвертому рядку документа, і скопіюйте його до буфера обміну.
3. Виконайте вказівку *Замінити*. Для цього:
 - установіть текстовий курсор у поле *Знайти* й натисніть комбінацію клавіш *Ctrl + V* для вставляння з буфера обміну;

• у поле *Замінити на* введіть фразу *Комарики, дзюбрики*. Натисніть кнопку *Замінити все*.

4. Збережіть документ із тим само ім'ям у папці *Тексти* своєї структури папок.

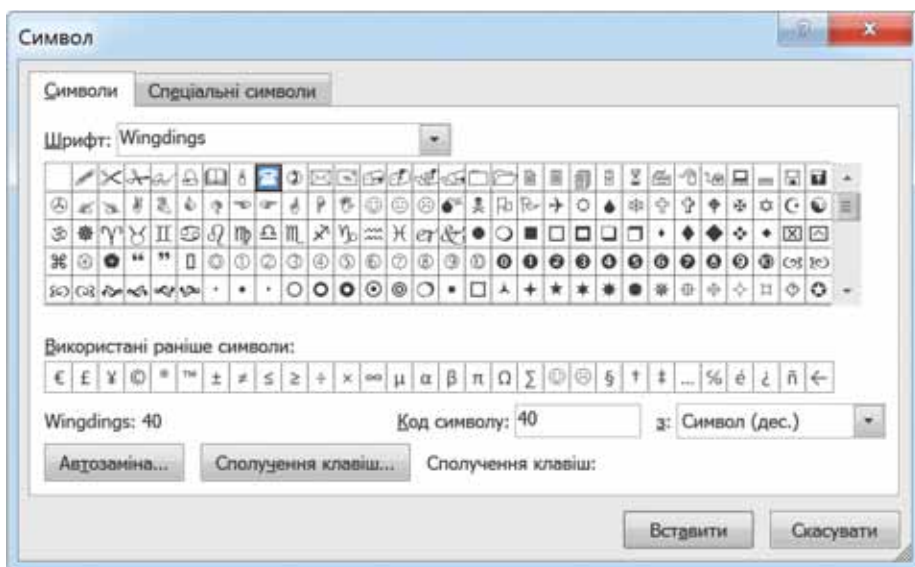
3. Як до тексту вставити символи, яких немає на клавіатурі?

Іноді до тексту необхідно додати символи, яких немає на клавіатурі. Це можуть бути знаки авторського права ©, математичних операцій порівняння $\neq$, $\leq$, $\geq$, $\approx$, грецькі літери α , β й інші символи.

Щоб додати до документа такі символи, в *Microsoft Word* використовують інструмент *Символ* на вкладці *Вставлення* групи *Символи* (мал. 4.5). Деякі часто вживані символи відображені у списку, що розкривається. За допомогою вказівки *Інші символи* відкривається вікно *Символ*, у якому можна вибирати різні символи. Список *Шрифт* містить перелік шрифтів, які можна обирати. Наприклад, шрифт *Wingdings* містить символи у вигляді малюнків книжки, телефона, листа, папки тощо (мал. 4.6).



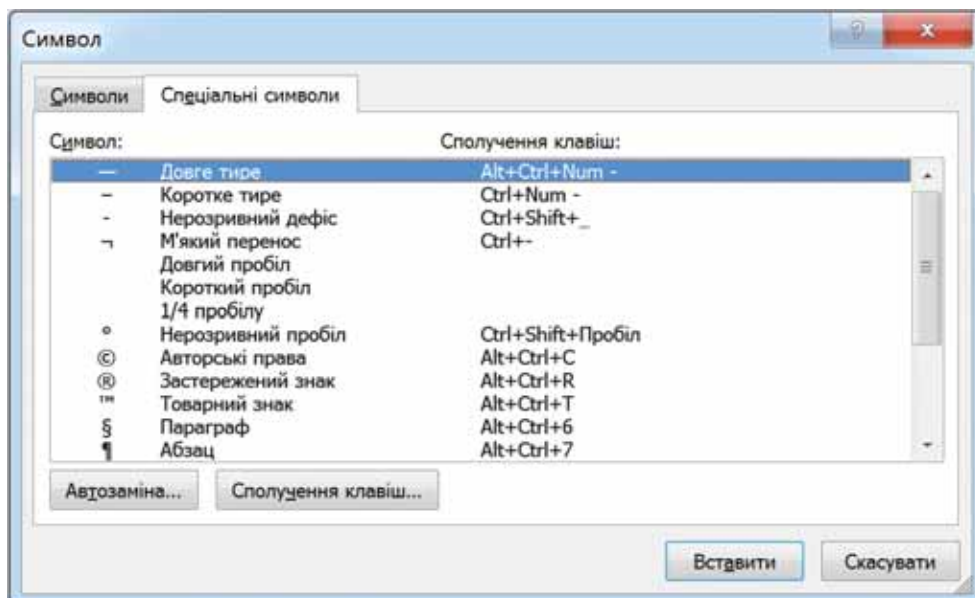
Мал. 4.5



Мал. 4.6

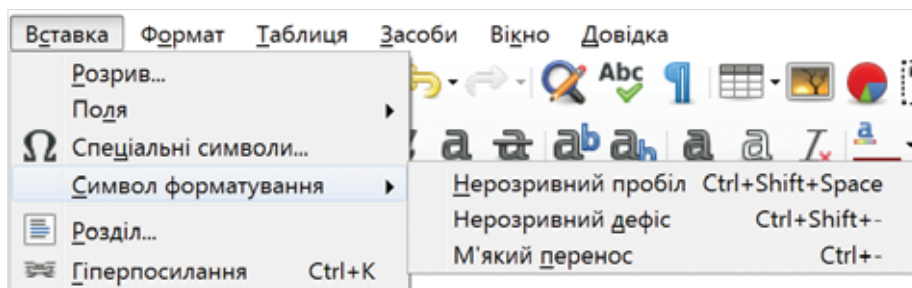
Деякі символи можна вводити за допомогою комбінацій клавіш або обрати на вкладці *Спеціальні символи* вікна *Символ* (мал. 4.7). Зокрема символи *нерозривний дефіс* ($\text{Ctrl}+\text{Shift}+ -$) і *нерозривний пропуск*

(*Ctrl+Shift+Пропуск*) використовують тоді, коли слово або фразу, що містить дефіс або пропуск, слід сприймати як одне слово й не розривати на два рядки. Наприклад, щоб ініціали та прізвище *І. В. Карпенко* завжди були записані в одному рядку, слід після ініціалів використати не звичайний пропуск, а нерозривний пропуск.



Мал. 4.7

У текстовому процесорі *LibreOffice Writer* вставити символи можна за допомогою інструмента *Спеціальні символи*  на панелі інструментів *Стандартна* або вказівок меню *Вставка/Спеціальні символи*. Вікно *Вибір символу* схоже на вікно *Символ* у *Microsoft Word*. Нерозривний дефіс чи нерозривний пропуск можна вставити за допомогою комбінацій клавіш або вказівки *Вставка/Символ форматування* (мал. 4.8).



Мал. 4.8

Для вставлених символів, як і для символів, що вводяться з клавіатури, можна змінювати значення параметрів форматування — колір, розмір, накреслення тощо.



Досліджуємо

Вправа 2. Шифрування.

Завдання. Дослідіть, як засобами текстового процесора «зашифрувати» текст повідомлення про призначення комусь таємної зустрічі.

1. У текстовому документі введіть текст повідомлення: *Чекатиму тебе сьогодні після уроків на подвір'ї школи біля спортивного майданчика. Потрібно поговорити наодинці.*
2. Дослідіть, чи можна виконати заміну голосних літер у тексті на деякі графічні символи.
3. Придумайте свій код для шифрування. Складіть свій ключ для дешифрування.
4. Виконайте шифрування. Збережіть зашифрований текст у документі *Шифровка_Прізвище*. Запишіть у цей текст також ключ для дешифрування. Зробіть висновок, чи може такий спосіб подання тексту зберегти персональні відомості.
5. Надішліть файл *Шифровка_Прізвище* на навчальну електронну скриньку.

4. Як застосувати стильове форматування для оформлення документа?

Тексту можна надавати різного зовнішнього вигляду, змінюючи значення параметрів форматування символів, абзаців і сторінок документа.

Ви вже вмієте змінювати шрифт, колір і розмір символів, їх накреслення, вирівнювання абзаців, відступи абзацу, міжрядковий інтервал й інші параметри форматування символів та абзаців. Полегшити та прискорити процес форматування тексту в документі можна за допомогою стилів.

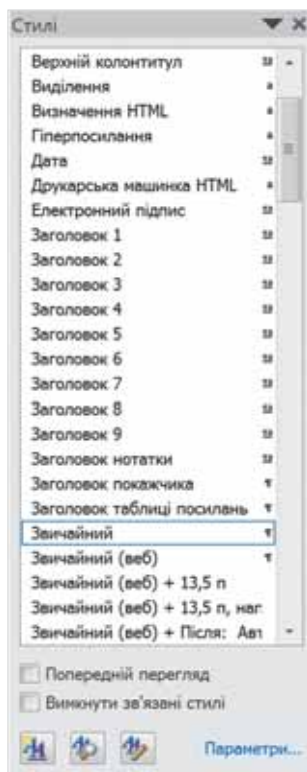
Стилем називається сукупність параметрів форматування, що позначається унікальним іменем і може застосовуватися для форматування текстових фрагментів.



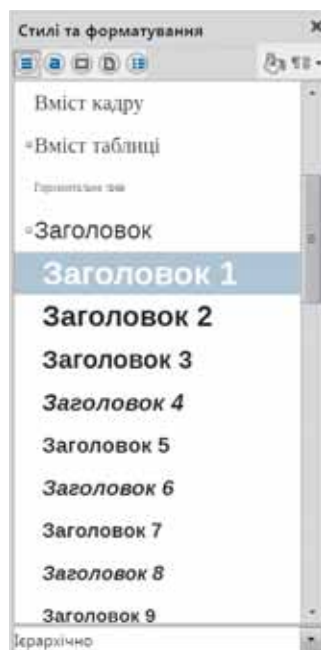
Це цікаво

Стиль (у загальному розумінні) — характерний вид, різновид чогонебудь, який виражається в деяких особливих ознаках, властивостях художнього оформлення.

Щоб застосувати стиль до введеного тексту, достатньо виділити текстовий фрагмент, який потрібно відформатувати, й у списку *Стиль* обрати потрібний. Якщо назви потрібних стилів не відображаються у списку *Стиль*, то можна відкрити бічну панель *Стилі* (*Стилі та форматування*), яка містить повний набір вбудованих стилів, а також інструменти для внесення змін до існуючих стилів і створення нових (мал. 4.9, 4.10). Щоб відкрити панель *Стилі*, у *Microsoft Word* на вкладці *Основне* в групі *Стилі* слід натиснути кнопку , розташовану в правому нижньому куті групи (мал. 4.11). У *LibreOffice Writer* панель *Стилі та форматування* з'являється, якщо обрати вказівку *Ще стилі* наприкінці списку стилів (мал. 4.12).



Мал. 4.11



Мал. 4.12



Діємо

Вправа 3. Форматування документа з використанням стилю *Заголовок*.

Завдання. У документі *Кореспонденція*, що зберігається в папці *Текстовий процесор*, застосуйте до назви документа стиль *Заголовок 1*, до підзаголовків — стиль *Заголовок 3*.

1. Відкрийте документ *Кореспонденція*, що зберігається в папці *Текстовий процесор*.
2. Виділіть абзац, що містить назву документа. На вкладці *Основне* (на панелі інструментів *Форматування*) відкрийте список *Стиль* і виберіть стиль *Заголовок 1*.
3. Аналогічно застосуйте до підзаголовків у документі стиль *Заголовок 3*. Якщо назва стилю *Заголовок 3* не відображається в списку *Стиль*, відкрийте область *Стилі (Стилі та форматування)*.
4. Знайдіть в основному тексті слова, що взяті в лапки. Послідовно виділяйте кожне з таких слів чи словосполучень та застосуйте до них стиль *Виділення*.
5. Збережіть результати у файлі з тим само іменем у папці *Тексти* своєї структури папок.

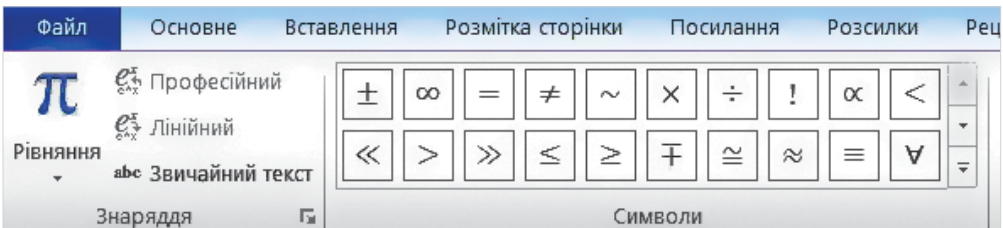
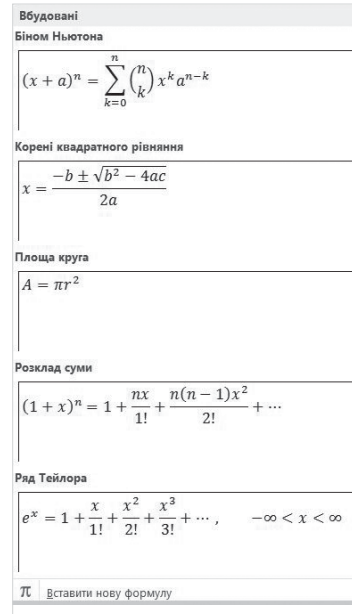
5. Як додати формули до текстового документа?

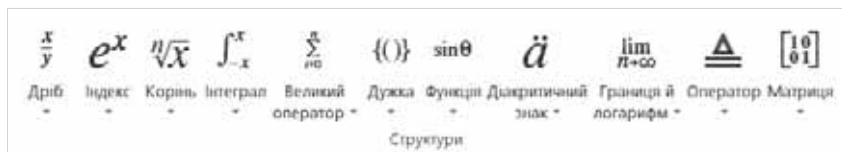
Текстові процесори містять вбудовані засоби для створення математичних, фізичних чи хімічних формул, які не можна ввести з клавіатури. Це спрощує процес створення навчальних та наукових документів, що містять формули.

У *Microsoft Word* для цього призначе-

ний інструмент  на вкладці *Вставлення* в групі *Символи*. У списку, що розкривається, наведені деякі вбудовані формули, а також доступна вказівка *Вставити нову формулу* (мал. 4.13), за допомогою якої можна створювати власні формули.

Після вибору цієї вказівки стають доступними інструменти *Конструктора* для роботи з формулами, за допомогою яких можна вставляти до поля формули спеціальні символи (мал. 4.14) та структури: дробі, індекси, корені, дужки тощо (мал. 4.15).





Мал. 4.15

Список, що розкривається при виборі інструмента для додавання певної структури, містить набір різних шаблонів — зразків фрагмента формули з використанням обраної структури. Наприклад, за допомогою інструмента *Корінь* можна обрати шаблони для запису квадратного чи кубічного кореня, а також кореня, степінь якого необхідно вказати з клавіатури (мал. 4.16).

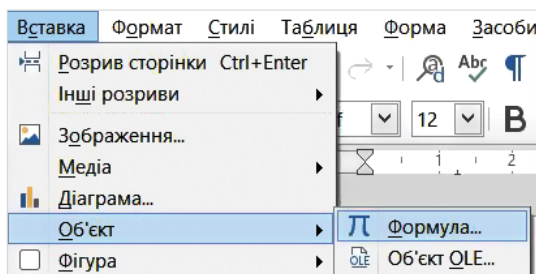


Мал. 4.16

Для конструювання формули слід обрати необхідні структури й у виділені поля ввести потрібні символи. Символи можна вводити з клавіатури або вибирати серед символів на вкладці *Конструктор*. Для завершення створення формули слід клацнути лівою кнопкою миші за її межами — тобто завершити використання спеціального засобу.

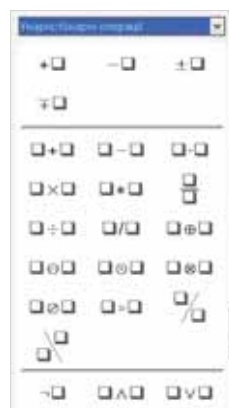
Щоб відредагувати створену формулу, необхідно двічі клацнути мишею в її межах, при цьому стануть доступними інструменти створення і редагування формули на вкладці *Конструктор*.

Для додавання формули в текстовий документ у *LibreOffice Writer* використовують вказівку *Вставка/Об'єкт/Формула* (мал. 4.17).



Мал. 4.17

Символи та структури, які можна використовувати у формулах, зібрані у список, що відкривається в лівій частині вікна (мал. 4.18), і згруповані за категоріями: відношення, функції, фігурні дужки тощо.



Мал. 4.18



Діємо

Вправа 4. Створення формули.

Завдання. Створіть текстовий документ *Формула*, до якого вставте

формулу $y = \frac{x^2 - 1}{|x| - 1}$.

1. Завантажте текстовий процесор і створіть новий текстовий документ.
2. На вкладці *Вставлення* оберіть інструмент *Рівняння* в групі *Символи*.
3. Введіть із клавіатури початок формули: $y =$.
4. На вкладці *Конструктор* у групі *Структури* оберіть інструмент *Дріб* і шаблон дробу з горизонтальною рисою .
5. Виділіть чисельник дробу, на вкладці *Конструктор* у групі *Структури* оберіть інструмент *Індекс* і шаблон у вигляді степеня.
6. Виділіть основу степеня, введіть символ x , виділіть показник степеня, введіть символ 2, установіть вказівник миші в чисельнику дробу та введіть -1 .
7. Виділіть знаменник дробу. На вкладці *Конструктор* у групі *Структури* оберіть *Дужка та шаблон* у вигляді модуля . Виділіть об'єкт під модулем, введіть символ x , установіть вказівник миші у знаменнику після модуля, введіть -1 .
8. Клацніть мишею за межами формули. Перевірте, що створена формула відповідає зразку.
9. Збережіть результат у файлі з іменем *Формула* в папці *Тексти* своєї структури папок.



Працюємо в парах

1. У яких випадках під час виконання автоматизованої заміни фрагмента тексту доцільно використовувати вказівки *Замінити*, *Замінити все*, *Знайти далі*? Відповідь аргументуйте.
2. Як здійснити пошук потрібних фрагментів тексту в середовищі текстового процесора? У яких режимах роботи з текстовим документом це можна виконати?
- ★ 3. Учні знайшли текст із помилками. У ньому між словами подекуди було два, три й чотири пропуски. Обговоріть, як швидко виправити помилки.
- ★ 4. У школі щороку випускають збірку творів учнів за новелами *Української хартії вільної людини* (<https://cutt.ly/ykUJmoB>). Запропонуйте редакторам збірки рекомендації щодо форматування творів.

5. Попрацюйте в мініпроекті «Молодь обирає здоров'я». Складіть сценарій виступу агітбригади. За потреби використайте матеріали з інтернету. Відформатуйте документ так, щоб слова різних учасників виділялись. Обговоріть, які засоби текстового процесора ви використовуєте.
6. Пограйте в гру «Формула»: по черзі наведіть приклади математичних, хімічних і фізичних формул, які не можна ввести з клавіатури, а можна — за допомогою засобу створення формул.



Працюємо самостійно

7. У текстовому документі *Характер та особливості знаків зодіаку*, який збережено в папці *Текстовий процесор*, виконайте заміну знаків зодіаку на відповідні назви.

♈	♉	♊	♋	♌	♍	♎	♏	♐	♑	♒	♓
Овен	Телець	Близнюки	Рак	Лев	Діва	Терези	Скорпіон	Стрілець	Козеріг	Водолій	Риби

8. Інструкція з правил безпечної роботи в кабінеті інформатики має розділи: *Загальні положення*, *Вимоги безпеки перед початком роботи в кабінеті інформатики*, *Вимоги охорони праці під час роботи в кабінеті інформатики*, *Вимоги безпеки після завершення роботи в кабінеті інформатики*. Якими засобами можна швидко виділити заголовки й відформатувати однаково зміст розділів? Виконайте таке форматування у файлі *Інструкція* з папки *Текстовий процесор*. Збережіть файл та надішліть його з іменем *Інструкція_Прізвище* на навчальну скриньку.
9. Створіть текстовий документ *Поезія*. Знайдіть в інтернеті повний текст вірша Володимира Сосюри «Любіть Україну». Доберіть і застосуйте різні стилі для строф вірша так, щоб передати інтонацію його прочитання: підвищення тону, зниження тону, рівний тон. Розмістіть текстовий документ на спільному ресурсі класу. Порівняйте своє «прочитання» вірша з пропозиціями інших.
10. Створіть текстовий документ, що містить такі формули:

$$y = \frac{1}{x - \frac{1}{x}}; \quad (x_2 - x_1)^2 \geq 0; \quad \text{Cu(OH)}_2 \xrightarrow{t} \text{CuO} + \text{H}_2\text{O}; \quad \begin{cases} 2x + 5y = 13, \\ 3x - 5y = -3; \end{cases} \quad \rho = \frac{m}{V}.$$



5

Структурування текстового документа

ПРИГАДАЙТЕ:

- ▶ що передбачає форматування документа;
- ▶ які параметри сторінки можна змінювати в текстовому процесорі;
- ▶ об'єкти текстового документа, формат яких можна змінювати;
- ▶ які використовують стилі під час роботи з документом.

ВИ ДІЗНАЄТЕСЬ:

- ▶ як розмістити текст у кілька колонок;
- ▶ як у документі створити кілька розділів і відформатувати їх;
- ▶ що таке колонтитули та як їх додати до текстового документа;
- ▶ які особливості роботи з колонтитулами в середовищі *Microsoft Word*;
- ▶ які особливості роботи з колонтитулами в середовищі *LibreOffice Writer*.



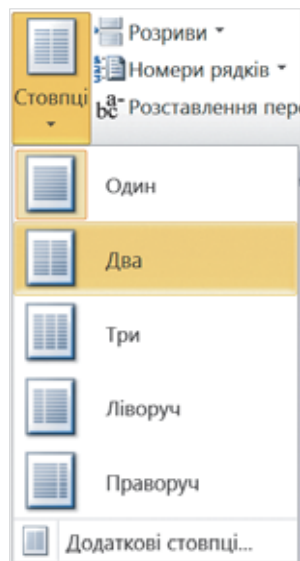
Вивчаємо

1. Як розмістити текст у кілька колонок?

У газетах і журналах вам траплявся текст, розбитий на кілька колонок. Такий текст можна підготувати в середовищі текстового процесора, при цьому різні абзаци можна розбити на різну кількість колонок.

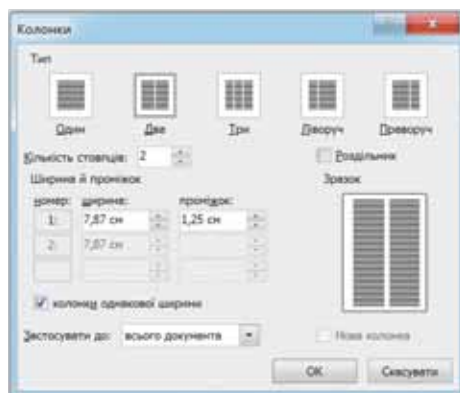
Процес підготовки документа, що містить колонки, складається з двох етапів: спочатку вводять текст в одну колонку, а потім виділяють потрібні абзаци та вказують кількість колонок, у які слід розмістити виділений фрагмент.

У *Microsoft Word* для цього використовують інструмент *Стовпці* на вкладці *Розмітка сторінки* в групі *Параметри сторінки*. Користувачеві пропонується п'ять варіантів розташування виділеного тексту в шпальти (мал. 5.1).



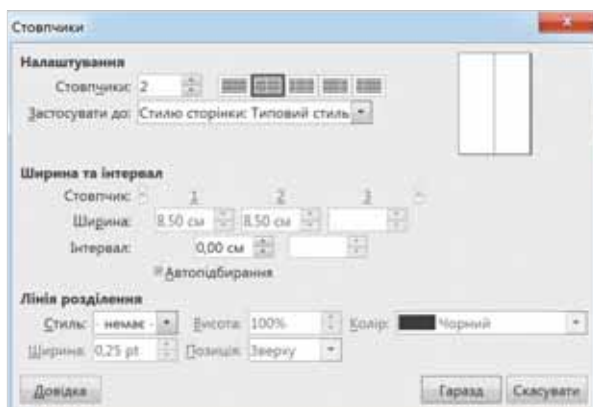
Мал. 5.1

Для форматування колонок використовують вказівку *Додаткові стовпці*, при цьому відкривається вікно *Колонки* (мал. 5.2). Усі властивості налаштування інтуїтивно зрозумілі, до того ж в області *Зразок* відображається схематичний вигляд сторінки.



Мал. 5.2

У *LibreOffice Writer* для створення колонок використовують вказівку *Формат/Колонки*. При цьому з'являється вікно *Стовпчики* (мал. 5.3), у якому можна вибрати кількість колонок і параметри їх форматування.



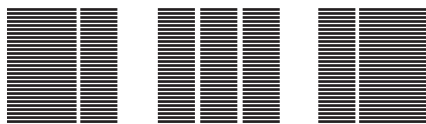
Мал. 5.3



ДІЕМО

Вправа 1. Стінна газета.

Завдання. Подайте текст *Пам'яті героїв Крут*, що зберігається у файлі *Історія* папки *Текстовий процесор* для розміщення на інформаційному історичному стенді у школі за таким макетом:



1. Завантажте текстовий документ *Історія*, що зберігається в папці *Текстовий процесор*.
2. Виділіть текст на першій сторінці. Скористайтесь інструментом *Стовпці* на вкладці *Розмітка сторінки* в групі *Параметри сторінки* текстового процесора *Microsoft Word* або вказівкою *Формат/Колонки* в *LibreOffice Writer*. Оберіть спосіб формування шпальт — *Ліворуч*. Завершіть дію натискуванням кнопки *ОК* (*Гаразд*).
3. Виконайте попередню дію для другої і третьої сторінки тексту, щоразу обираючи інший спосіб формування шпальт, як показано в макеті.
4. Збережіть файл з іменем *Стенд* папки *Тексти* у стандартному форматі текстового процесора та форматах PDF і RTF. Перевірте, чи впливає обраний формат на спосіб розміщення тексту в колонки та як зміна формату впливає на розмір файлу. Зробіть висновки.

2. Як у документі створити кілька розділів і відформатувати їх?

За замовчуванням текстовий документ складається з одного розділу. Всі сторінки одного розділу завжди мають однакові значення параметрів форматування сторінки. Однак часто доводиться опрацьовувати великі документи, які розбиті на кілька розділів. Наприклад, електронний примірник книжки чи підручника, збірка правил чи інструкцій тощо.

Як правило, новий розділ створюється, якщо потрібно для різних сторінок документа задати різні параметри сторінки або щоб певна частина документа починалася з нової сторінки. Наприклад, нову главу в романі прийнято розпочинати з нової сторінки. Під час розбиття тексту на колонки автоматично створюються розділи, оскільки кількість колонок — це одна з властивостей розділу.

Розділ — частина документа, яка має задані параметри форматування сторінки чи колонок.

Щоб розбити документ на кілька розділів, слід вставити розриви розділів, а потім задати значення параметрів форматування для кожного розділу.

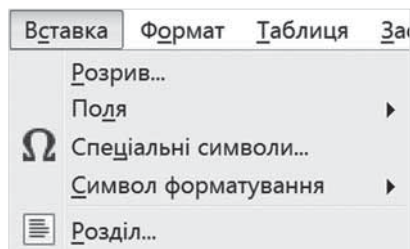
У *Microsoft Word* для створення нового розділу в документі потрібно встановити текстовий курсор до того місця в документі, де потрібно вставити розрив розділу, і на вкладці *Розмітка сторінки* в групі *Параметри сторінки* обрати інструмент *Розриви*. У списку, що розкривається, в області *Розриви розділів* слід вибрати параметр, який вказує, звідки слід почати новий розділ (мал. 5.4). При цьому в документ вставляється мітка ~Розрив розділу (з наступної сторінки)~, яку можна побачити в режимі перегляду недрукованих знаків. Ця мітка позначає

кінець розділу та містить відомості про такі параметри форматування розділу, як розміри полів, орієнтація сторінки, послідовність номерів сторінок тощо.

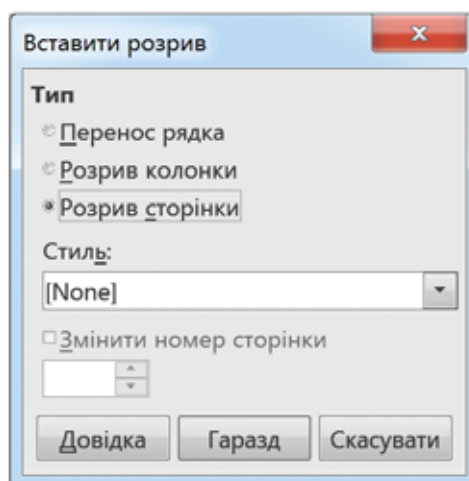


Мал. 5.4

Для створення нового розділу в *LibreOffice Writer* використовують вказівку *Вставка/Розділ* (мал. 5.5). Якщо потрібно, щоб розділ розпочинався з нової сторінки, виконують вказівку *Вставка/Розрив* й у вікні *Вставити розрив* обирають тип *Розрив сторінки* (мал. 5.6).



Мал. 5.5

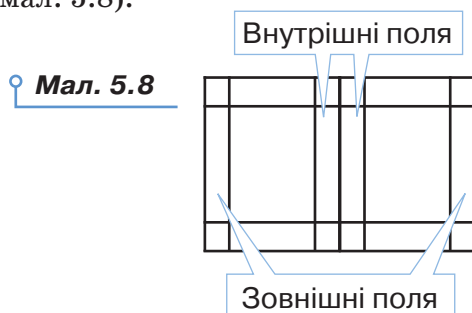
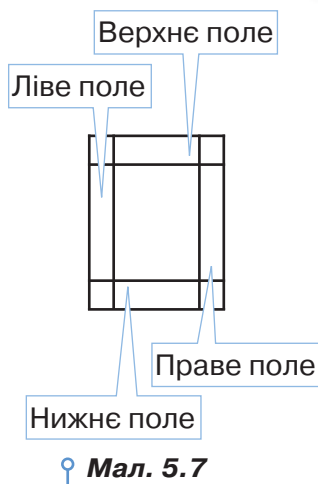


Мал. 5.6

У текстовому документі або його розділі можна змінювати параметри форматування сторінок, зокрема поля, орієнтацію сторінки, розмір паперу тощо.

Поля сторінки визначають відстань від краю аркуша паперу до початку відображення тексту.

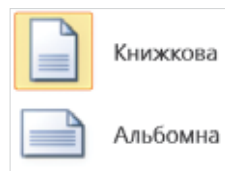
За замовчуванням документ друкується з однієї сторони аркуша паперу. У цьому випадку розрізняють верхнє, нижнє, лівє та правє поля (мал. 5.7). Під час підготовки журнальних і книжкових видань застосовують двосторонній друк. При цьому замість лівого та правого полів використовують поняття внутрішніх і зовнішніх полів (мал. 5.8).



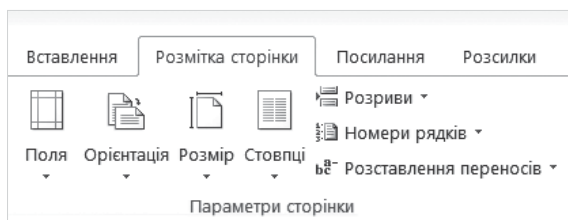
Якщо передбачається зшивання надрукованого документа, то можна вибрати *Розташування корінця* для переплетення — зліва чи зверху, і задати значення додаткової відстані.

Добираючи *Розмір* паперу, слід враховувати, на якому принтері документ буде надруковано. Так, аркуш формату А4 можна роздрукувати на будь-якому сучасному принтері, тоді як формат паперу А3 підтримується лише спеціальними принтерами.

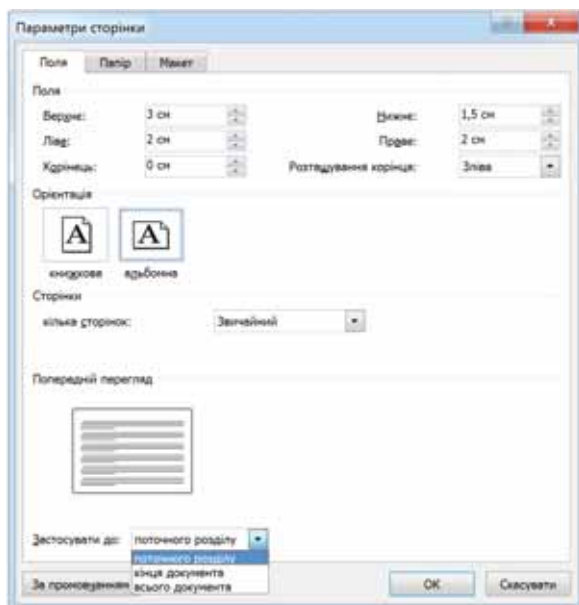
Орієнтація сторінки може бути *книжкова* — якщо сторінка розташовується вертикально, або *альбомна* — якщо горизонтально (мал. 5.9).



Щоб змінити параметри форматування сторінок для певного розділу в *Microsoft Word*, слід установити текстовий курсор у межах цього розділу, на вкладці *Розмітка сторінки* в області *Параметри сторінки* обрати відповідні інструменти (мал. 5.10) або натиснути кнопку , щоб викликати вікно *Параметри сторінки*, на відповідних вкладках якого можна вказати потрібні значення параметрів. Щоб вказані зміни впливали не на весь документ у цілому, а лише на обраний розділ, слід на будь-якій із вкладок *Поля*, *Папір*, *Макет*, на якій змінювалися значення параметрів, у списку *Застосувати до* вибрати значення поточного розділу (мал. 5.11).

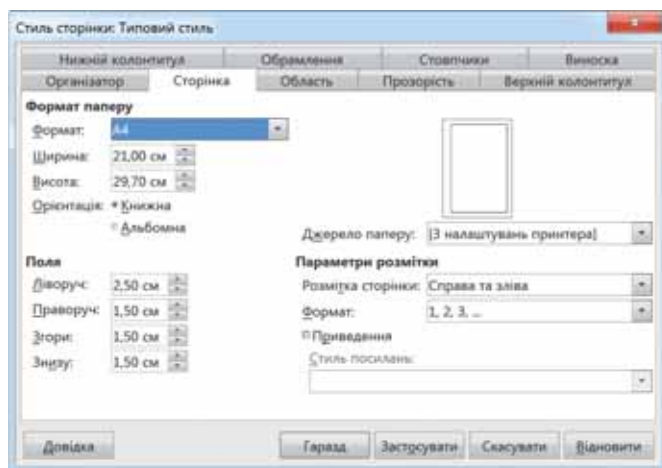


Мал. 5.10



Мал. 5.11

Переглянути та змінити значення параметрів форматування сторінки в *LibreOffice Writer* можна у вікні *Стиль сторінки* (мал. 5.12), яке викликають за допомогою вказівки *Формат/Сторінка*. Розміри полів, орієнтацію сторінки й розмір паперу можна налаштовувати на вкладці *Сторінка* цього вікна.



Мал. 5.12



ДІЄМО

Вправа 2. Створення розділів у документі.

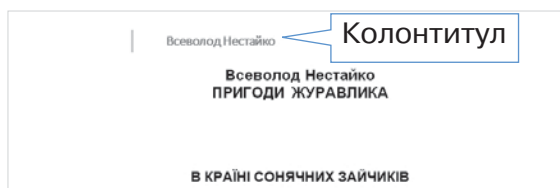
Завдання. Створіть у документі *Кореспонденція*, що зберігається в папці *Тексти* вашої структури папок, розділи так, щоб кожний пункт документа містився в окремому розділі та розпочинався з нової сторінки. Установіть орієнтацію сторінки для першого розділу — альбомну, а для всіх розділів документа значення розмірів полів: ліве — 2,5 см, усі інші — 1,5 см.

1. Відкрийте документ *Кореспонденція*, що зберігається в папці *Тексти* у вашій структурі папок.
2. Установіть параметри полів сторінок для всього документа: ліве — 2,5 см, усі інші — 1 см.
3. Установіть текстовий курсор на початку абзацу, що містить підзаголовки *1. Про листи взагалі*. На вкладці *Розмітка сторінки* в групі *Параметри сторінки* оберіть інструмент *Розриви*. У списку, що розкривається, в області *Розриви розділів* оберіть *Наступна сторінка*.
4. Діючи аналогічно, вставте розділи до кожного пункту документа.
5. Установіть текстовий курсор у межах першого розділу. На вкладці *Розмітка сторінки* в області *Параметри сторінки* оберіть інструмент *Орієнтація* і задайте альбомну орієнтацію сторінки.
6. Перегляньте документ. Зробіть висновки, чи змінилась орієнтація сторінки на альбомну для всіх сторінок документа.
7. Збережіть результати у файлі з тим само ім'ям.

3. Що таке колонтитули та як їх додати до текстового документа?

Ви вмієте додавати до сторінок номер. Номер сторінки — це об'єкт текстового документа, який розташовується у спеціальному розділі — **колонтитулах**.

У колонтитулах можна вказувати відомості про назву документа чи окремого розділу, дату створення, автора, номер сторінки, адреси інтернет-ресурсів, які є джерелом матеріалів, тощо. Колонтитули часто можна побачити в книжках і журналах. Наприклад, у книжках на парних сторінках може бути вказано автора книжки, а на непарних — назву книжки. У журналах в одному з колонтитулів зазвичай вказують назву й номер журналу, а також рік видання. Номери сторінок, які ви вже вмієте додавати до документа, розташовуються саме в області верхнього або нижнього колонтитула документа.





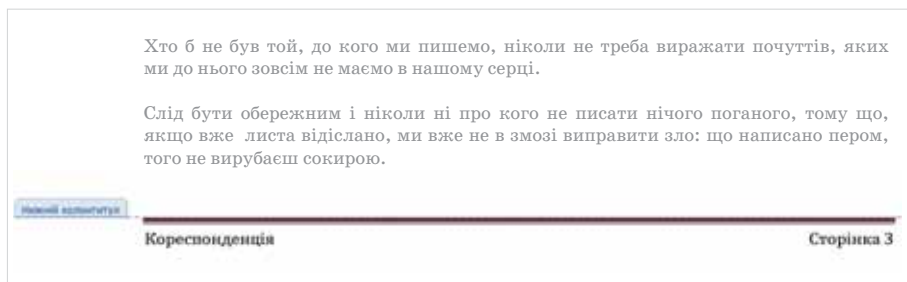
Це цікаво

Назва **колонтитул** утворюється з двох слів: від фр. *colonne* — стовпчик і лат. *titulus* — напис, заголовок.

Колонтитул — об'єкт, що розміщується над текстом (верхній колонтитул) або під текстом (нижній колонтитул) кожної сторінки книжки, газети, журналу, документа.

Колонтитули можуть містити текст і деякі інші об'єкти: малюнки, лінії або інші фігури тощо (мал. 5.13).

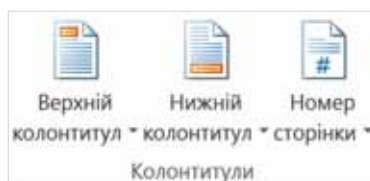
Мал. 5.13



Якщо в документі є кілька розділів, то колонтитули можуть бути однакові для всіх розділів, або кожний розділ може містити різні колонтитули.

4. Які особливості роботи з колонтитулами в середовищі *Microsoft Word*?

У середовищі *Microsoft Word* вводити та редагувати вміст колонтитулів можна в окремих областях у верхній чи нижній частині сторінки. Щоб перейти до області колонтитула, використовують інструменти з вкладки *Вставлення* в групі *Колонтитули* (мал. 5.14).



Мал. 5.14

Інструмент *Номер сторінки* використовують для додавання і налаштування в області колонтитула номерів сторінки. Особливо це зручно, якщо колонтитул не містить інших об'єктів.

Список, що розкривається в разі вибору інструментів *Верхній колонтитул* чи *Нижній колонтитул* (мал. 5.15), містить бібліотеку вбудованих стилів колонтитулів, з яких можна обрати той, що більше підходить за оформленням або наявністю необхідних об'єктів.



Мал. 5.15

Після вибору одного з вбудованих стилів колонтитулів текстовий курсор автоматично встановлюється в крайній лівій частині області колонтитула. Якщо певні об'єкти потрібно розмістити по центру колонтитула або в правій його частині, то швидко перевести текстовий курсор у потрібне положення можна за допомогою клавіші *Tab*. Під час роботи з колонтитулами текст документа відображається неяскравим кольором, і на стрічці в області *Знаряддя для колонтитулів* з'являється вкладка *Конструктор*, що містить інструменти для роботи з колонтитулами (мал. 5.16).

Мал. 5.16



Щоб завершити створення чи редагування колонтитулів і повернутися до вмісту документа, слід на вкладці *Конструктор* натиснути кнопку *Закрити колонтитули* або двічі клацнути мишею за межами колонтитулів.

Якщо документ складається з кількох розділів, за замовчуванням увімкнено режим *Як у попередньому*, і колонтитули всіх розділів будуть однаковими. Якщо необхідно створити різні колонтитули в кожному розділі, то після створення колонтитула першого розділу слід перейти до колонтитула наступного розділу та на вкладці *Конструктор*

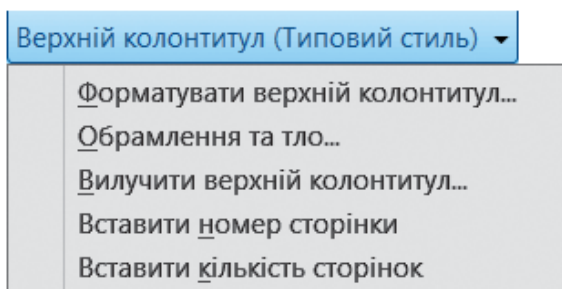
обрати інструмент , щоб вимкнути цей режим. Після цього можна створювати новий колонтитул в обраному розділі.

Якщо документ містить нумерацію сторінок, за замовчуванням усі сторінки нумеруються послідовно. У межах розділів можна налаштувати інші параметри нумерації: вказати, з якого номера починати нумерувати сторінки розділу, долучити до номера сторінки номер розділу, наприклад, 1-1, 1-2, 1-3 та 2-1, 2-2.

5. Які особливості роботи з колонтитулами в середовищі *LibreOffice Writer*?

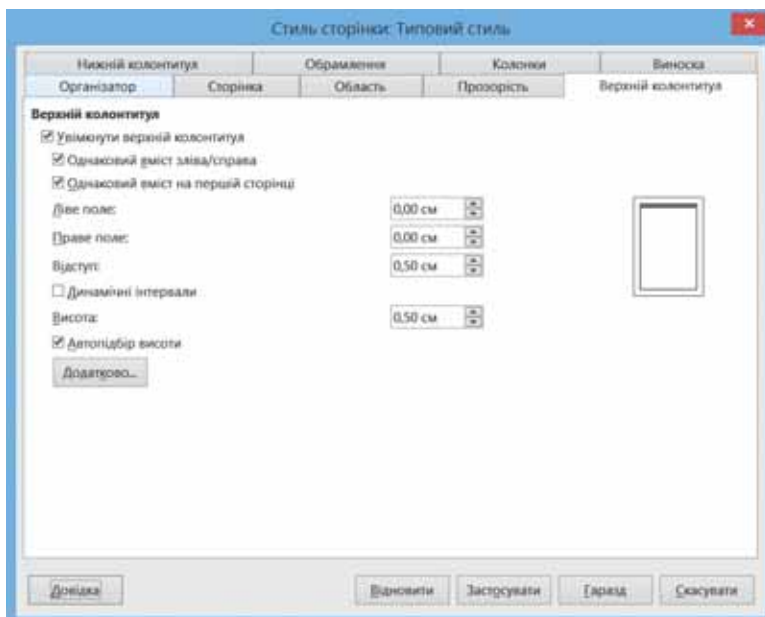
У *LibreOffice Writer* додати колонтитули до документа можна за допомогою вказівок *Верхній колонтитул* і *Нижній колонтитул* з меню *Вставка*.

До тексту в межах колонтитула застосовується типовий стиль колонтитулів (мал. 5.17).



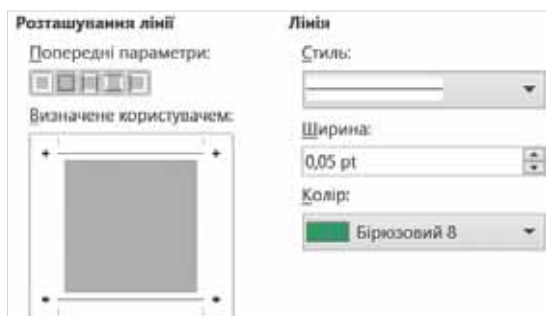
 **Мал. 5.17**

Щоб його змінити, у списку обирають *Форматувати верхній колонтитул...* У вікні, що відкривається, переходять до вкладки *Верхній колонтитул* або *Нижній колонтитул*. Змінюють налаштування колонтитула (мал. 5.18).



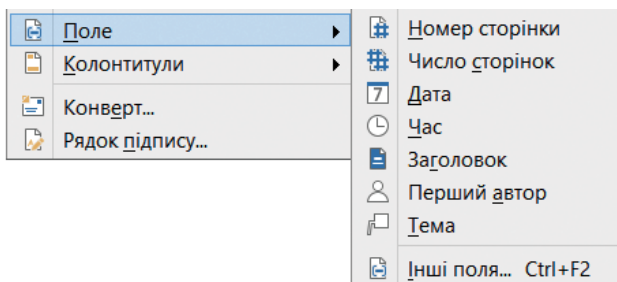
Мал. 5.18

Натиснувши кнопку *Додатково*, отримуємо вікно налаштування колонтитула. Наприклад, вказавши параметри лінії, як на малюнку, отримуємо у верхній частині сторінки поле для колонтитула, яке відділене від верхнього краю сторінки й основного тексту двома горизонтальними лініями зеленого кольору (мал. 5.19).



Мал. 5.19

Щоб до колонтитула додати інші дані про документ, використовують вказівку *Вставка/Поле* й обирають потрібний об'єкт (мал. 5.20).



Мал. 5.20



Діємо

Вправа 3. Створення колонтитула в документі.

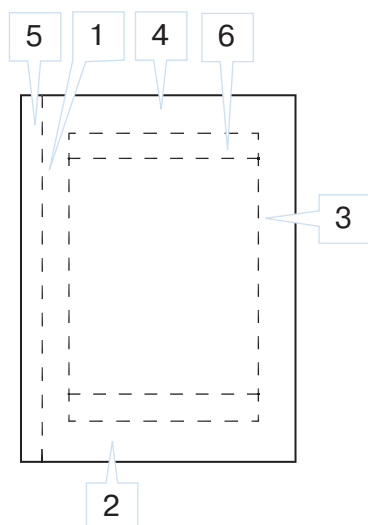
Завдання. У документ *Кореспонденція*, що зберігається в папці *Тексти* вашої структури папок, вставте нижній колонтитул, у якому відображатиметься назва документа, поточна дата й номер сторінки.

1. Відкрийте документ *Кореспонденція*, що зберігається в папці *Тексти* у вашій структурі папок.
2. Додайте нижній колонтитул. У лівій частині нижнього колонтитулу введіть із клавіатури назву документа *Кореспонденція*. Додайте дату в форматі короткої дати, наприклад: 3.11.2021.
3. Натисніть клавішу *Tab*, щоб перейти до правої межі колонтитулу. Додайте номер сторінки.
4. Переконайтеся, що створений колонтитул повторюється на всіх сторінках документа.
5. Збережіть результати у файлі з тим само ім'ям.



Працюємо в парах

1. Зобразіть засобами текстового процесора модель документа, який містить вказані елементи за зразком.



- ☐ верхній колонтитул;
- ☐ нижній колонтитул;
- ☐ ліве поле;
- ☐ праве поле;
- ☐ корінець;
- ☐ верхнє поле;
- ☐ нижнє поле.

Запропонуйте в парі визначити їх розміщення на моделі.



2. Запропонуйте три переваги розбиття великого документа на розділи. Обговоріть свої пропозиції в парі. Сформулюйте спільно дві переваги. Обговоріть свої аргументи з іншою парою.

★ 3. Для чого у друковані видання вставляють колонтитули? Назвіть три переваги використання колонтитулів у друкованих документах. Чи можуть в одному виданні міститися різні колонтитули? Обговоріть у парах.

★★ 4. Попрацюйте в мініпроєкті «Знати історію — легко!». Розподіліть теми шкільного курсу історії 8 класу між собою (за матеріалами сайту <https://cutt.ly/VkUVT39>). Домовтеся про однакові параметри форматування сторінок. Виберіть одну з тем і створіть документ — стислий довідник. Кожну історичну подію обраного періоду історії України подайте у вигляді окремого розділу. Застосуйте потрібні колонтитули для зручного орієнтування в документі. Додайте нумерацію сторінок. Починайте номер сторінки із числа, яке слідує після останнього номера сторінки попередньої теми. Роздрукуйте документ. Зшийте спільний довідник.



Працюємо самостійно

5. Укажіть назви об'єктів сторінки у вправі за посиланням <https://cutt.ly/ck988Mt>.

★★ 6. Відкрийте документ *Пам'ятники Києва*, що зберігається в папці *Текстовий процесор*. Перегляньте, які стилі застосовано до заголовків документа. Установіть такі параметри форматування сторінки: ліве поле — 3 см, праве поле — 1,5 см. Розбийте документ на розділи таким чином, щоб відомості про кожний з пам'ятників починалися з нової сторінки. В одному з розділів відформатуйте текст із використанням колонок. Створіть у документі колонтитули, в яких на непарних сторінках відображатиметься назва *Пам'ятники Києва*, на парних сторінках — назва пам'ятника, що стосується певного розділу. Щоб мати можливість у різних розділах створювати різні колонтитули, відключіть режим *Як в попередньому розділі*. Для цього оберіть відповідний інструмент на вкладці *Конструктор* в області *Знаряддя для колонтитулів*.





Робота з великими документами

ПРИГАДАЙТЕ:

- ▶ параметри розділів текстового документа;
- ▶ як створити новий розділ;
- ▶ призначення гіперпосилання.

ВИ ДІЗНАЄТЕСЬ:

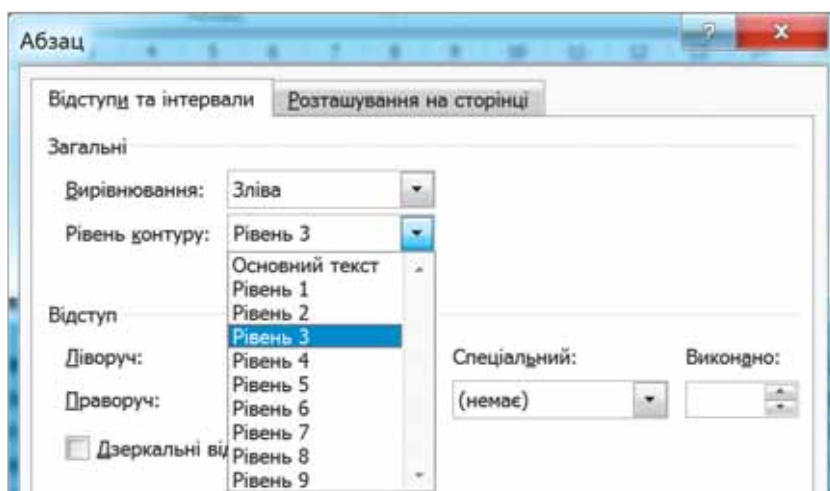
- ▶ як можна переглянути структуру документа;
- ▶ як використовують гіперпосилання в текстових документах;
- ▶ як автоматично створити зміст і покажчик у текстовому документі;
- ▶ як опрацьовувати складні текстові документи.



Вивчаємо

1. Як можна переглянути структуру документа?

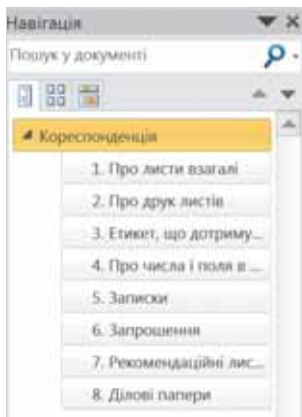
Заголовки розділів і підрозділів документа визначають його структуру. Текстові процесори містять засоби, за допомогою яких можна переглядати структуру документа й використовувати її для швидкого переміщення в потрібне місце документа. Щоб скористатися такими засобами, необхідно, щоб для абзаців, які містять такі заголовки, параметр *Рівень контуру* у вікні *Абзац* мав значення, відмінне від *Основний текст* — *Рівень 1*, *Рівень 2* тощо (мал. 6.1).



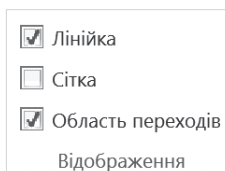
Мал. 6.1

У вбудованих стилях заголовків номер рівня відповідає номеру в назві стилю: *Заголовок 1* має *Рівень 1*, *Заголовок 2* — *Рівень 2* і т. д. Тому для використання засобів роботи зі структурою можна застосувати відповідні стандартні стилі заголовків до заголовків розділів і підрозділів.

Переглянути структуру документа й переміщуватися в місце документа, де розпочинається підрозділ з вибраним заголовком, у *Microsoft Word* можна за допомогою панелі *Навігація* (мал. 6.2), яка відображається на екрані, якщо на вкладці *Вигляд* у групі *Відображення* увімкнути прапорець *Область переходів* (мал. 6.3).



Мал. 6.2



Мал. 6.3

Опрацьовувати великі документи в *Microsoft Word* можна також у *Режимі структури*, в якому можна згорнути документ, відображаючи лише основні заголовки, переміщувати чи копіювати частини тексту за допомогою заголовків тощо. У цьому режимі текст подано без урахування параметрів форматування абзаців. Усі заголовки та звичайний текст відображено з відступами, які представляють рівень цього тексту в загальній структурі документа (мал. 6.4).

Мал. 6.4



Ліворуч від заголовків відображаються позначки **+**, клацання на яких приводить до виділення розділу, що відповідає обраному заголовку, а подвійне клацання — до згортання чи розгортання відображення підзаголовків обраного заголовка.

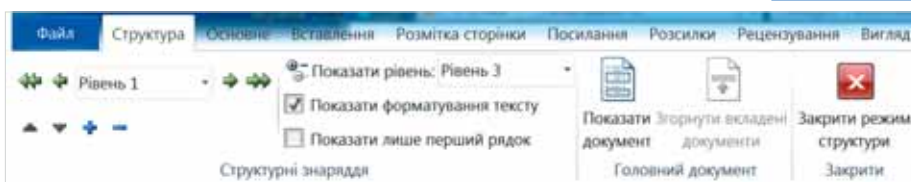
Щоб перейти до режиму структури, потрібно на вкладці

Вигляд у групі *Режими перегляду* документа обрати інструмент  або в правій частині рядка стану натиснути кнопку *Структура*



Під час роботи в режимі структури на стрічці відкривається вкладка *Структура* (мал. 6.5), що в групі *Структурні знаряддя* містить інструменти, за допомогою яких можна вибрати ступінь деталізації, наприклад, відображати всі заголовки чи лише заголовки від першого до третього рівня, підвищити чи понизити рівень для певних заголовків, переміщувати підрозділи в тексті тощо.

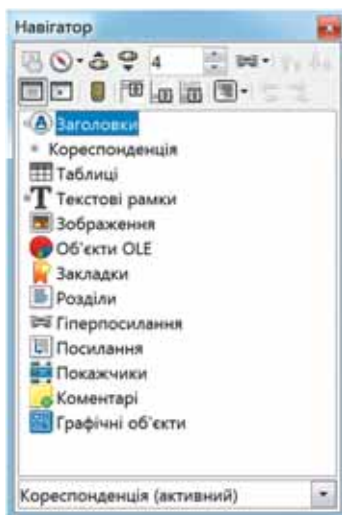
 **Мал. 6.5**



Щоб повернутися в режим розмітки документа, слід на вкладці *Структура* обрати інструмент *Закрити режим структури*.

У *LibreOffice Writer* заголовки й інші об'єкти документа можна переглянути та використати для переміщення в документі на панелі *Навігатор* (мал. 6.6), яку можна відкрити за допомогою вказівки

меню *Перегляд/Навігатор* або кнопки *Навігатор*  на бічній панелі (мал. 6.7), що розташована в правій частині вікна. Заголовки нижчих рівнів можуть бути приховані. Щоб їх відобразити, слід клацнути на позначці зліва від заголовка першого рівня.



 **Мал. 6.6**



 **Мал. 6.7**



ДІЄМО

Вправа 1. Перегляд документа в **Режимі структури**.

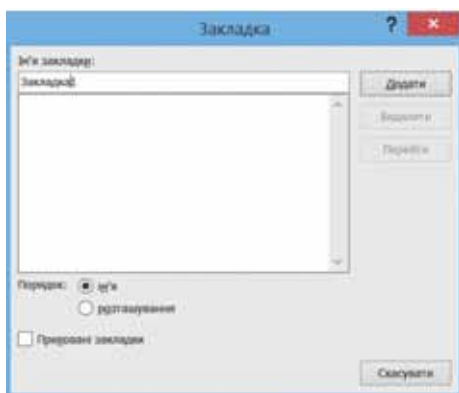
Завдання. Перегляньте документ *Повість-казка* в режимі відображення структури. Не переглядаючи тексту твору, визначте, скільки розділів містить повість.

1. Відкрийте документ *Повість-казка*, що зберігається в папці *Текстовий процесор*.
2. Перейдіть до режиму *Структура документа*.
3. Оберіть другий рівень деталізації структури. Скористайтесь інструментом  **Показати рівень: Рівень 2** на вкладці *Структура* текстового процесора *Microsoft Word* або позначкою  вікна *Навігатор* у *LibreOffice Writer*.
4. Полічіть кількість розділів документа. Дослідіть, який вигляд матиме документ, якщо обрати третій рівень структури.
5. Збережіть зміни у файл *План повісті* в папку *Тексти* своєї структури папок.
6. Відкрийте збережений файл. Перевірте, чи зберігається режим відображення структури після збереження файлу, зробіть висновок. Закрийте вікно текстового процесора.

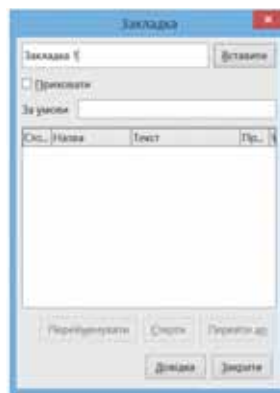
2. Як використовують гіперпосилання в текстових документах?

У текстових документах для переходу по структурі документа можна використовувати закладки. Для того щоб зробити такі посилання, спочатку потрібно встановити закладку в потрібному місці. Створення такої закладки відбувається за алгоритмом:

- 1) виділити слово в документі, на яке буде здійснено перехід;
- 2) обрати вказівку *Закладка* із вкладки *Вставлення* (*Microsoft Word* — мал. 6.8) або *Вставка* (*LibreOffice Writer* — мал. 6.9);



Мал. 6.8

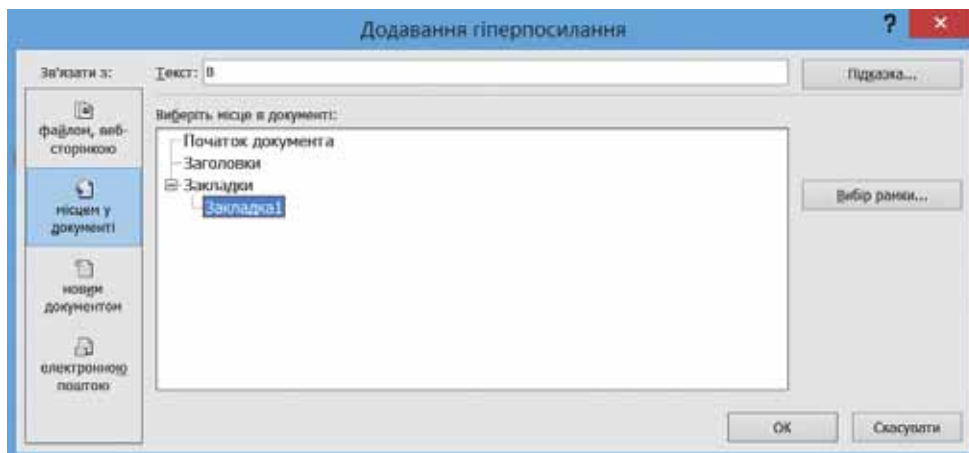


Мал. 6.9

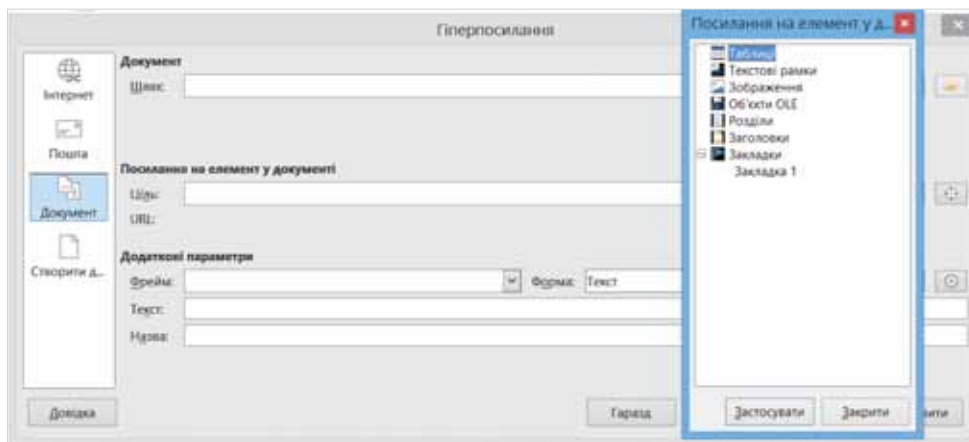
3) додати ім'я закладки й натиснути кнопку *Додати* (*Microsoft Word*) або *Вставити* (*LibreOffice Writer*).

Після створення закладки в документі виділяють об'єкт, з якого планують здійснити перехід на закладку. Обирають вказівку *Вставлення/Гіперпосилання* (*Microsoft Word*) або *Вставка/Гіперпосилання* (*LibreOffice Writer*).

У вікні *Додавання гіперпосилання* (*Microsoft Word* — мал. 6.10) або *Гіперпосилання* (*LibreOffice Writer* — мал. 6.11) вказують об'єкт, з яким слід зв'язати — *місцем у документі* та вибирають ім'я закладки або ціль.



Мал. 6.10



Мал. 6.11

Такі дії приводять до того, що текст або інший об'єкт буде виділено іншим кольором, з підкресленням. При наведенні мишею вказівник змінює свій вигляд на посилання та при натискуванні лівої кнопки миші та клавіші *Ctrl* відбувається перехід на закладку.

Аналогічно вказівками вікна *Додавання гіперпосилання* чи *Гіперпосилання* можна зв'язати основний документ з іншим документом або вебсторінкою зі вказаною адресою.

Таким чином у документі можна організувати щось на зразок навігаційного меню з посилань.



ДІМО

Вправа 2. Навігаційна панель.

Завдання. Перегляньте документ *Повість-казка* в режимі відображення структури. Не переглядаючи тексту твору, визначте, скільки розділів містить повість.

1. Відкрийте документ *Повість-казка*, що зберігається в папці *Текстовий процесор*.
2. На початку документа введіть назву всіх розділів, наприклад, як на малюнку 6.12.

Повість-казка

ДІД МАНОЦІВНИК

ВЕСНЯНКА

НЕЗНАЙОМЕЦЬ

 **Мал. 6.12**

3. Позначте заголовки розділів і заголовок *Повість-казка* як закладки. Здійсніть перехід від навігаційної панелі до створеної закладки.
4. Після кожного розділу поставте позначку ■, за допомогою якої перейдіть до закладки *Повість-казка*.
5. Збережіть документ у своїй структурі папок з іменем *Навігація_Прізвище*. Надішліть його на навчальну скриньку.

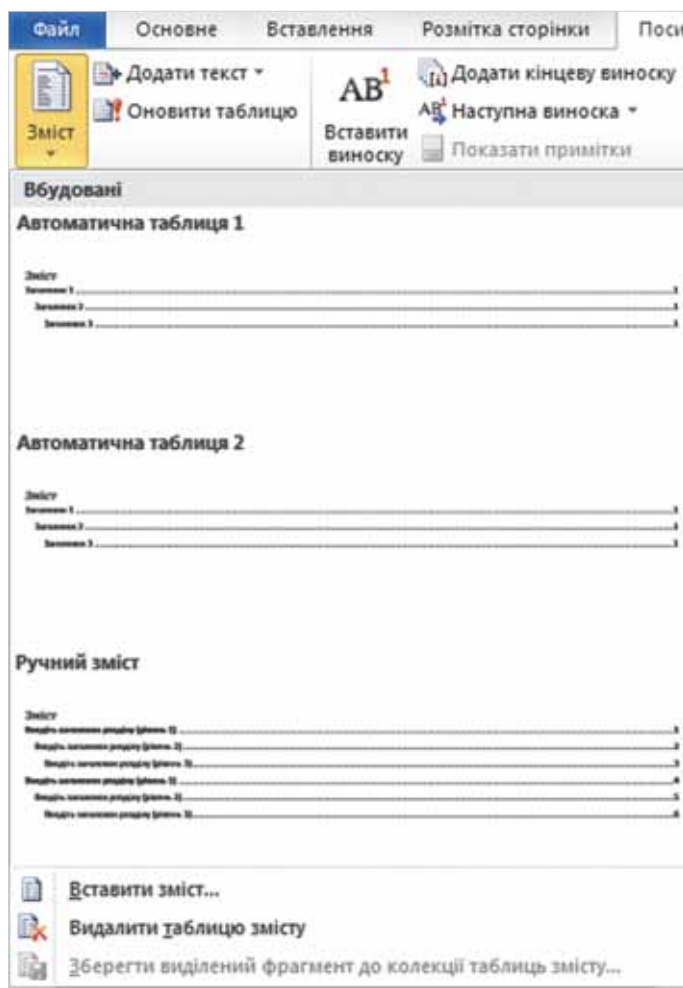
3. Як автоматично створити зміст і покажчик у текстовому документі?

Великі документи, як правило, містять зміст й інші засоби для полегшення орієнтування. **Зміст** являє собою список заголовків документа й номерів сторінок, на яких починаються розділи із цими заголовками, і розміщується, як правило, на початку документа. Наприкінці великих документів, у яких трапляється багато важливих понять, створюють **покажчик** — словник з основних понять документа, які розташовані в алфавітному порядку з вказуванням номера сторінки, де це поняття міститься в документі. Наприклад, зміст і покажчик зазвичай є в підручниках і енциклопедіях.

Текстові процесори дають змогу автоматично створювати зміст і покажчик, які можна також використовувати для переміщення по документу, оскільки кожний елемент такого об'єкта є **посиланням**. Наприклад, якщо, утримуючи натисненою клавішу *Ctrl*, клацнути мишею на деякому заголовку в змісті, текстовий курсор переміститься в місце документа, де розпочинається частина документа із цим заголовком.

Зміст можна створити автоматично, до нього будуть включені абзаци, у яких значення параметра *Рівень контуру* відрізняється від *Основний текст* або до яких застосовано стандартні стилі *Заголовок 1*, *Заголовок 2* тощо. Тому, перш ніж додавати зміст, необхідно сформуванати структуру документа і встановити текстовий курсор на те місце в документі, де його потрібно вставити.

Щоб додати зміст у *Microsoft Office*, слід на вкладці *Посилання* обрати інструмент *Зміст* (мал. 6.13).



Мал. 6.13

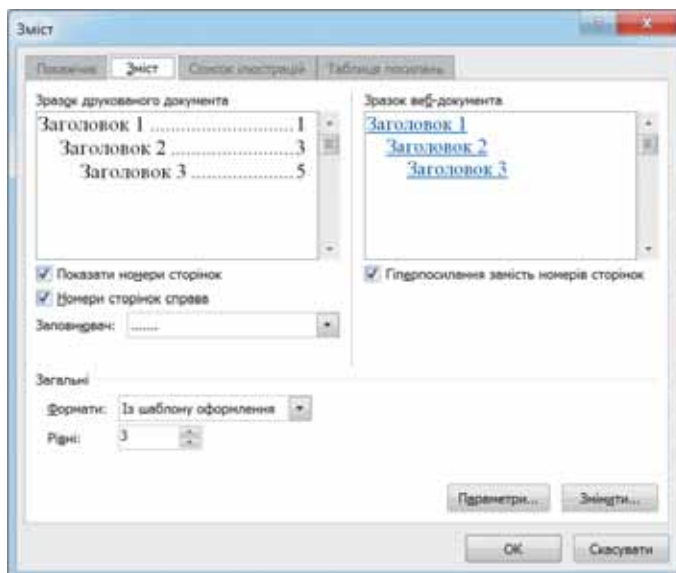
Потрібний стиль можна обрати у списку, що розкривається. При цьому в документі автоматично будуть знайдені всі абзаци, до яких застосовані стилі заголовків, і буде сформовано зміст у документі (мал. 6.14).

Мал. 6.14

Зміст	
КОРЕСПОНДЕНЦІЯ.....	1
1. Про листи взагалі.....	1
2. Про друк листів.....	4
3. Етикет, якого дотримуються в листах.....	5
4. Про числа та поля в листах.....	6
5. Записки.....	6
6. Запрошення.....	7
7. Рекомендаційні листи.....	8
8. Ділові папери.....	8

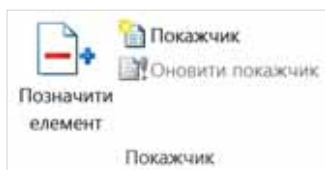
Якщо необхідно задати особливі налаштування змісту, слід обрати вказівку *Вставити зміст*, при цьому відкривається вікно *Зміст* (мал. 6.15), де можна обрати режим додавання номерів сторінок, з яких починається кожний з розділів і підрозділів, обрати формат оформлення, вказати кількість рівнів заголовків, які необхідно включати до змісту тощо.

Мал. 6.15



Для створення покажчика в *Microsoft Office* використовують інструменти з групи *Покажчик* (мал. 6.16) на вкладці *Посилання*.

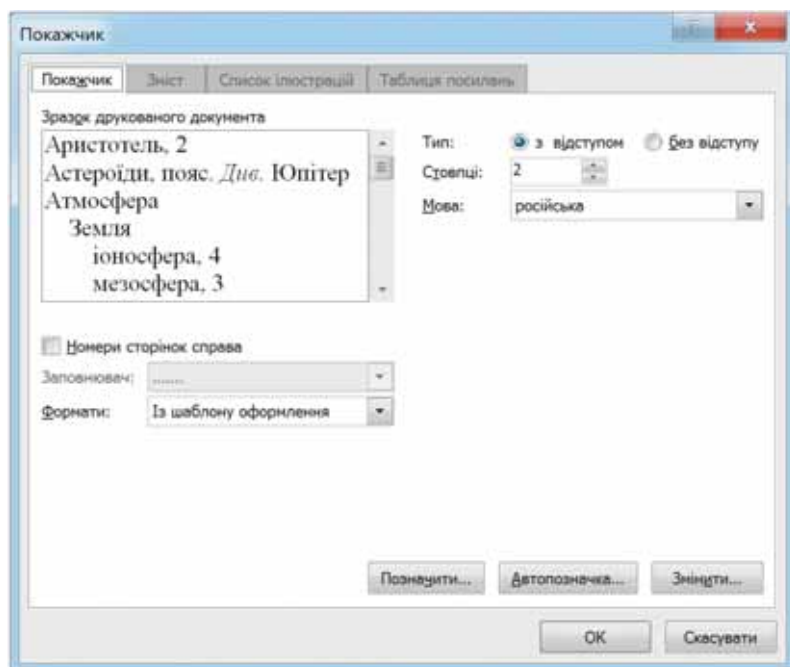
Мал. 6.16



Спочатку необхідно в документі послідовно виділити поняття, які слід включити до покажчика, та обрати для кожного з них інстру-

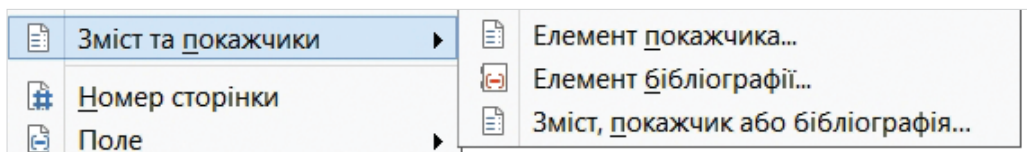


мент **Позначити елемент**. Коли всі елементи позначені, слід встановити текстовий курсор у те місце документа, де має бути створений покажчик, наприклад, наприкінці документа, та обрати інструмент **Покажчик**. У вікні *Покажчик* (мал. 6.17), що відкривається при цьому, можна змінити кількість стовпців (колонок), у які слід розмістити поняття в покажчику, та інші параметри.



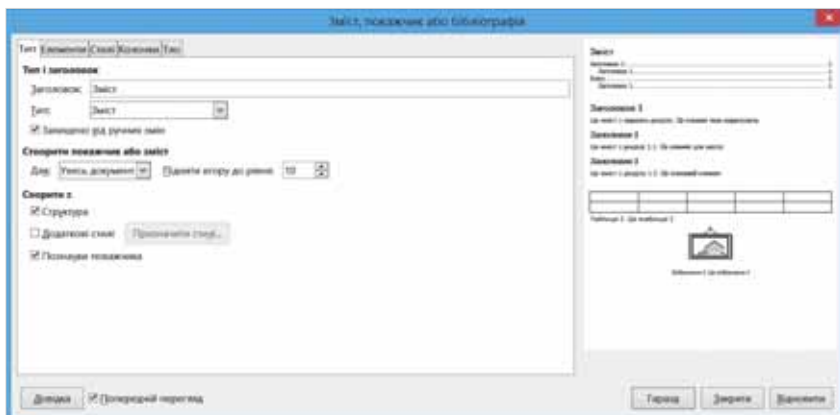
Мал. 6.17

Створення змісту та покажчика в *LibreOffice Writer* відбувається аналогічно. Для цього використовують вказівку меню *Зміст та покажчики* (мал. 6.18).



Мал. 6.18

Налаштування властивостей виконують у вікні *Вставити Покажчик/Зміст* (мал. 6.20).



Мал. 6.20



ДІЄМО

Вправа 3. Створення змісту документа.

Завдання. Для документа *Кореспонденція*, що зберігається в папці *Тексти* вашої структури папок, автоматично створіть зміст.

1. Відкрийте документ *Кореспонденція*, що зберігається в папці *Тексти* вашої структури папок.
У цьому документі до заголовків розділів і підрозділів у праві 1 ви застосували стилі *Заголовок 1* і *Заголовок 3*.
2. На початку документа вставте новий абзац і встановіть до нього текстовий курсор.
3. На вкладці *Посилання* оберіть інструмент *Зміст* і в списку, що розкривається, оберіть один зі стилів оформлення змісту.
4. Створіть новий розділ, щоб основний текст документа розпочинався з наступної сторінки після змісту. Для цього на вкладці *Розмітка сторінки* в групі *Параметри сторінки* оберіть інструмент *Розриви*. У списку, що розкривається, в області *Розриви розділів* оберіть *Наступна сторінка*.
5. Поверніться до змісту. Натисніть на клавіатурі клавішу *Ctrl* і клацніть на назві заголовка *Етикет*, якого дотримуються в листах. Що при цьому відбувається?
6. Збережіть документ із тим само ім'ям.

Вправа 4. Створення алфавітного покажчика.

Завдання. Для документа *Кореспонденція*, що зберігається в папці *Тексти* вашої структури папок, автоматично створіть алфавітний покажчик понять, які використовуються в документі.

1. Відкрийте документ *Кореспонденція*, що зберігається в папці *Тексти* вашої структури папок.

2. Знайдіть у тексті документа такі основні поняття: кореспонденція, листи, звернення, приступ, розповідь, висновок, постскрипtum, конверти, етикет, записки, запрошення, рекомендаційні листи, ділові папери.

3. Послідовно виділяйте кожне з понять й обирайте інструмент



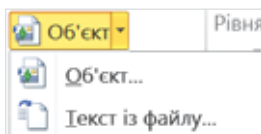
на вкладці *Посилання* в групі *Покажчик*.

4. Установіть текстовий курсор у новому абзаці в кінці документа та оберіть інструмент *Покажчик* на вкладці *Посилання*.
5. У вікні *Покажчик* перегляньте властивості, встановлені за замовчуванням, та натисніть кнопку *ОК*.
6. Клацніть мишею на слові *Етикет* у створеному покажчику. Переконайтеся, що на екрані відобразився фрагмент документа, що містить це поняття. Зробіть висновок про використання елементів покажчика.
7. Збережіть документ із тим само ім'ям.

4. Як опрацювати складні текстові документи?

Ефективне опрацювання складного документа, що складається з кількох розділів та містить велику кількість сторінок, передбачає виконання певних дій:

1. Введення тексту й додавання об'єктів до документа.
2. Якщо необхідні текстові фрагменти збережені в іншому файлі, можна скористатися буфером обміну для їх копіювання і вставлення до документа. Об'єднувати текст кількох файлів можна за допомогою вказівки *Текст із файлу* зі списку інструмента *Об'єкт*, що розташований на вкладці *Вставлення* в групі *Текст*.



Далі у вікні *Вставлення файлу* обирають потрібний файл і натискають кнопку *Вставити*.

3. Форматування тексту й об'єктів документа.
4. Форматування заголовків розділів і пунктів з використанням вбудованих стандартних стилів заголовків або користувацьких параметрів форматування, що визначають рівень абзацу.
5. Створення в документі необхідної кількості розділів і налаштування параметрів сторінки для кожного розділу окремо або для всього документа.
6. Перегляд структури документа за допомогою засобів навігації або в режимі структури, корекція послідовності розташування розділів у разі необхідності.

7. Додавання до документа засобів орієнтування і навігації: змісту, алфавітного покажчика тощо.

Працюємо в парах

1. Обговоріть переваги й недоліки створення змісту документа за допомогою закладок й автозмісту.
2. Обговоріть можливі причини, коли в документі доцільно використовувати зміст, а коли — режим структури.
- ★ 3. Відкрийте документ *Розділ книги*, що зберігається в папці *Текстовий процесор*. Обговоріть, які технології варто застосувати до документа, щоб можна було швидко розрізняти структуру та зміст документа, переходити від однієї частини до іншої і повертатись на початок. Застосуйте ці технології. Продемонструйте свою роботу іншій парі учнів.
- ★★ 4. Попрацюйте в проекті «Поради для тих, хто хоче врятувати світ». Виберіть один з рівнів: суперзірка на дивані, герої побуту, добрі сусіди, зразкові працівники. У текстовому документі створіть гру-квест, що дасть змогу користувачеві ознайомитися з ощадливими та розумними діями, які сприятимуть реалізації цілей сталого розвитку в обраній ролі (<https://cutt.ly/ZhUstyz>). Обговоріть, чи можна для реалізації квесту використати технологію закладок і гіперпосилань у документі.



Працюємо самостійно

5. Відкрийте документ *Пам'ятники Києва*, що зберігається в папці *Текстовий процесор*. Визначте, які стилі застосовано до заголовків документа. Створіть зміст на початку документа. Додайте до документа номери сторінок і нижній колонтитул, що містить текст *Пам'ятники Києва*.
- ★ 6. Внесіть зміни в документ *Антонов*, що зберігається в папці *Текстовий процесор*: застосуйте стилі до заголовків, створіть зміст, розбийте документ на розділи, установіть орієнтацію сторінки окремого розділу, додайте колонтитули, що містять посилання на джерело матеріалу та номери сторінок, додайте алфавітний покажчик, що містить 7–9 термінів.
- ★★ 7. За правилами створення складних документів створіть текстовий документ *Твори Лесі Українки*, який об'єднує збережені в папці *Текстовий процесор\Леся Українка* текстові документи, що містять поеми та вірші. Застосуйте до заголовків розділів стиль *Заголовок 2*. Додайте до документа колонтитули. На початку документа створіть автоматичний зміст.



Спільна робота з документом

ПРИГАДАЙТЕ:

- ▶ як організовують персональне навчальне середовище;
- ▶ що таке хмарні технології;
- ▶ як працювати з документами у спільному сховищі.

ВИ ДІЗНАЄТЕСЬ:

- ▶ що передбачає спільна робота з документом;
- ▶ як захистити текстовий документ;
- ▶ у яких режимах спільно працюють з документом;
- ▶ як організувати одночасну спільну роботу над документом кільком авторам.



Вивчаємо

1. Що передбачає спільна робота з документом?

З розвитком технологій та інтернету кардинально змінилося поняття робочого колективу й спільної роботи. Люди, що живуть в різних країнах і працюють у різних організаціях, можуть брати участь у спільній розробці проектів чи опрацюванні тих само документів.

Спільна робота над документами означає, що кілька авторів чи авторок разом працюють над документом або набором документів. Вони можуть або одночасно бути співавторами/співавторками документа, або переглядати специфікацію, що є частиною структурованого робочого циклу.

Доволі часто ті, хто працюють над одним проектом, перебувають у різних будівлях, кабінетах тощо. У таких випадках для організації документообігу між учасниками та учасницями проекту необхідно організувати *віртуальний офіс*, де будуть зберігатися документи для спільного використання. Головна проблема такої організації діяльності полягає в доступі всіх учасників й учасниць до загальних документів і в організації колективного спілкування віддалених одне від одного людей. Для цього можна використати відомі вам можливості — електронну пошту, чати, служби миттєвого обміну повідомленнями, різні сервіси **Веб 2.0**.

Також існує багато спеціально розроблених програм для організації групової роботи й використання спільних документів, наприклад, *Google Документи*, *12Sprints*, *Sharepoint*, *Novell Pulse*, *IPI.Manager*,

Feng Office, Microsoft Office Groove та ін. Такий вид програмного забезпечення поки що не отримав однозначної назви, але найчастіше ці програми називають системами колективної взаємодії. Вони можуть використовуватися як у локальних, так і в глобальних мережах.

Основне призначення систем колективної взаємодії: доступ до спільних ресурсів деякої групи учасників, організація спілкування між учасниками групи, координація і документування ходу робіт.

2. Як захистити текстовий документ?

Текстові документи можна зберегти в різних форматах:

- **RTF** (від англ. *Rich Text Format* — формат збагаченого тексту) — текст із форматуванням і вставленими об'єктами;
- **DOC** — текст із форматуванням і вставленими об'єктами, основний формат *Microsoft Word 97/2000/XP/2003*;
- **DOCX** — текст із форматуванням і вставленими об'єктами, основний формат *Microsoft Word 2007/2010/...*;
- **ODT** — текст із форматуванням і вставленими об'єктами, основний формат *LibreOfficeWriter*;
- **PDF** — формат для перегляду та друкування з незмінним вихідним форматуванням для різних платформ;
- **TXT** — текст із розбиттям на абзаци без форматування;
- **HTML** — (від англ. *HyperText Markup Language* — мова розмітки гіпертекстових документів) — стандартна мова розмітки вебсторінок в інтернеті та відповідне розширення файлу;
- **XML** (від англ. *Extensible Markup Language* — розширювана мова розмітки) — запропонований консорціумом *World Wide Web (W3C)* стандарт побудови мов розмітки ієрархічно структурованих даних для обміну між різними застосунками, зокрема через інтернет, і відповідне розширення файлу.

Вміст файлів останніх трьох типів можна редагувати в середовищі будь-якого текстового процесора. У цьому разі тип вказує лише на програму, якою потрібно відкривати файл.

Усі формати файлів, окрім формату PDF, забезпечують можливість редагування файлу після його збереження. Автор чи авторка документа може обмежити доступ до нього іншим. У меню *Файл* вказівка *Властивості* дає змогу установлювати різні рівні захисту (мал. 7.1).



Це цікаво

Консорціум Всесвітнього павутиння (англ. *World Wide Web Consortium, W3C*) — головна міжнародна організація, що розробляє і впроваджує технологічні стандарти для Всесвітнього павутиння. Заснована 1994 р. Тімом Бернерсом-Лі — винахідником Всесвітнього павутиння, що він його очолює.



Мал. 7.1

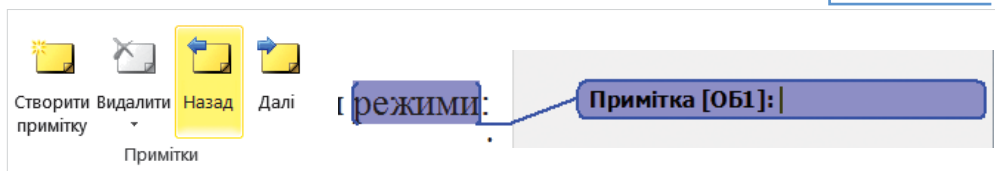
3. У яких режимах спільно працюють з документом?

Під час роботи декількох авторів/авторок з документом можна виділити три режими:

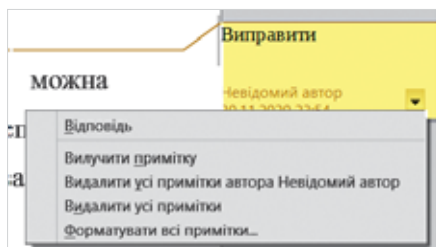
- 1) основний автор чи авторка пише текст, інші автори пропонують зміни в режимі примітки;
- 2) усі автори мають однакові права, внесення змін до текстового документа відбувається по чергові;
- 3) усі автори мають однакові права, зміни в документі відбуваються одночасно.

Перший режим забезпечується додаванням приміток до тексту. Інструменти для роботи з примітками в середовищі *Microsoft Word* розташовані на вкладці *Рецензування*. Щоб додати примітку до текстового об'єкта, його виділяють і вибирають вказівку *Створити примітку* (мал. 7.2). Тоді на полях вводять текст примітки. Зауважте, що примітки кожного автора виділяються різним фоном і вказується ім'я користувача.

Мал. 7.2



У *LibreOffice Writer* примітку можна додати вказівкою *Вставка/Коментар* (мал. 7.3). За цією вказівкою до виділеного об'єкта текстового документа додається виноска, у яку вводять текст. Зазначимо, що кожний автор чи авторка також має свій колір виноска. Його/її ім'я і дата/час написання коментаря додаються автоматично.

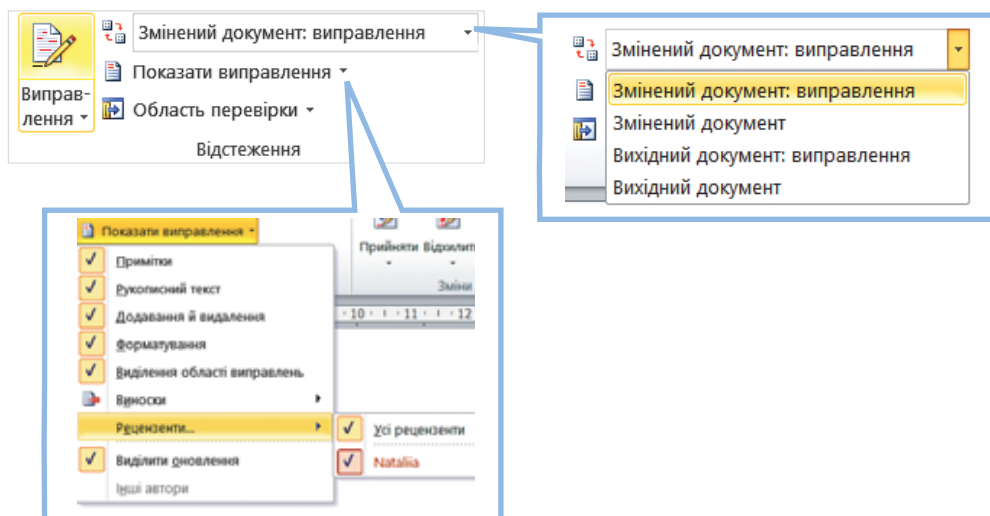


Мал. 7.3

Автори можуть видаляти примітки (виділену або всі примітки в документі).

Для того щоб відстежувати зміни в документі, які були внесені іншим автором чи авторкою, можна використати вказівки групи *Відстеження* (Microsoft Word — мал. 7.4) або *Відстеження змін* меню *Зміни* (LibreOffice Writer — мал. 7.5).

Мал. 7.4

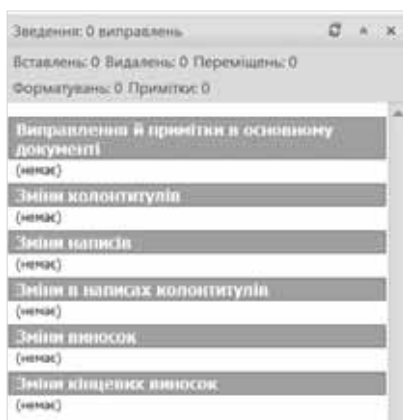


Мал. 7.5

Відстеження змін	
Гіперпосилання	Запис Ctrl+Shift+C
Коментар	Показати
Посилання	Керування...
Поля...	Попередня
Зв'язки з зовнішніми файлами...	Далі
Мапа зображення	Прийняти
Об'єкт	Прийняти всі
Активне джерело даних...	Відхилити
Режим редагування Ctrl+Shift+M	Відхилити всі
	Коментар...
	Захистити...
	Порівняти документ...
	Об'єднати виправлення...

Коли виправлення ввімкнено, видалення позначаються закресленням, а їх додавання позначаються підкресленням. Зміни різних авторів чи авторок позначаються різними кольорами. Коли виправлення вимкнено, текстовий процесор припиняє маркування змін, але кольорові підкреслення і закреслення змін залишаються в документі, доки їх не буде прийнято або відхилено.

Зміни в документі (*Microsoft Word*) можна відстежувати за допомогою вікна *Область перевірки* (мал. 7.6), у якому зазначаються результати процесу реагування текстового документа.



Мал. 7.6



Діємо

Вправа 1. Успішний стартap.

Завдання. У документі *Успішний стартap* виділіть для кожного абзацу головну думку, ідею чи пораду. Запишіть їх у примітки до документа.

1. Відкрийте файл *Успішний стартap* папки *Текстовий процесор*. Прочитайте поради авторки тексту щодо створення успішного стартapу.
2. Виділіть перше слово першого абзацу, додайте до нього примітку. У примітці запишіть головну думку, ідею чи пораду, яку можна отримати із цього абзацу.
3. Так продовжуйте до кінця тексту.
4. Зробіть висновок, чи можна вас вважати співавтором або співаторкою тексту. Де можна використати ваші примітки?
5. Збережіть файл з іменем *Стартap_Прізвище* у своїй структурі папок. Надішліть файл на навчальну скриньку.

4. Як організувати одночасну спільну роботу над документом кільком авторам?

Спільну роботу над документом, у якому одночасно можна вносити зміни декільком користувачам чи користувачкам, можна організува-

ти у хмарному сховищі документів, наприклад, у *Google Диску*. Для цього документ завантажують у *Google Диск* і надають доступ користувачам  **Спільний Доступ** .

Зауважте, що автори документа є його власниками. Автор чи авторка може:

- **змінювати** документи, таблиці та презентації, а також запрошувати інших користувачів для співавторства та читання;
- **видаляти** документи, таблиці та презентації, закриваючи, таким чином, доступ до них для інших користувачів чи користувачок.

Для повного видалення документа, таблиці та презентації або скасування доступу до них необхідно видалити їх, а потім очистити кошик.

Користувач або користувачка, яких ви запрошуєте редагувати ваш документ, стає співавтором або співаторкою. Вони можуть:

- **редагувати** документи, таблиці та презентації;
- **запрошувати** або видаляти інших співавторів і читачів (якщо їм надано дозвіл на ці дії);
- **експортувати** копію документа, таблиці або презентації на свій жорсткий диск.

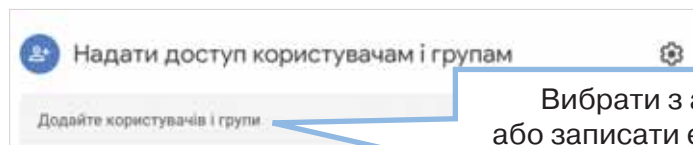
Якщо ви не хочете, щоб ваш документ редагували, але вам необхідно надати комусь можливість перегляду документа, тоді ви запрошуєте його або її як читача або читачку (без права редагування).

Читачі можуть:

- **переглядати** останню версію документа, таблиці або презентації, але не можуть вносити зміни;
- **експортувати** копію документа, таблиці або презентації на свій жорсткий диск.

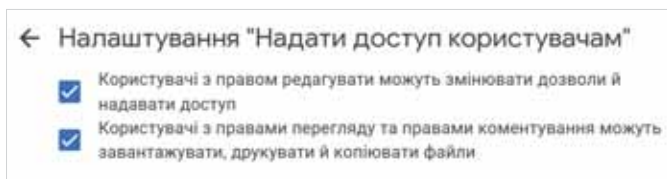
Запросити співавторів чи співавторок до спільного редагування документів можна двома способами. У контекстному меню документа на диску обрати вказівку  Надати доступ (мал. 7.7).

 **Мал. 7.7**



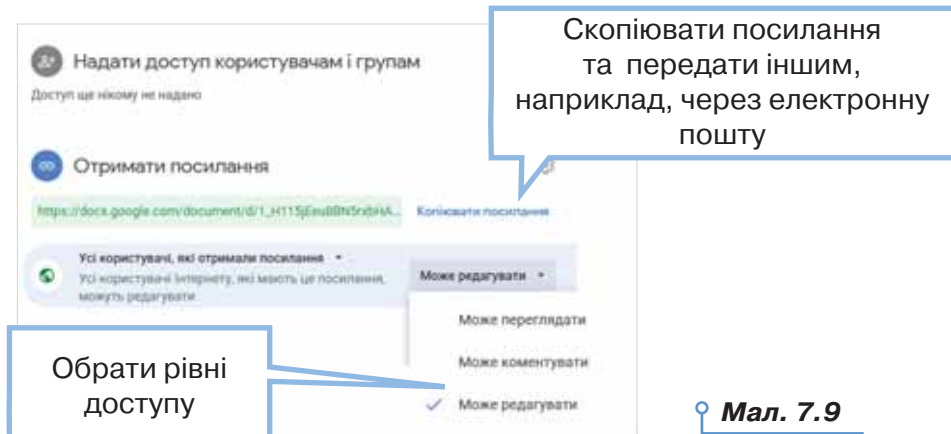
Вибрати з адресної книги
або записати електронні адреси
співавторів/співаторок

Визначити особливі права користування документом можна після натискування кнопки *Налаштування*  у верхньому правому куті вікна, що відкріється (мал. 7.8).



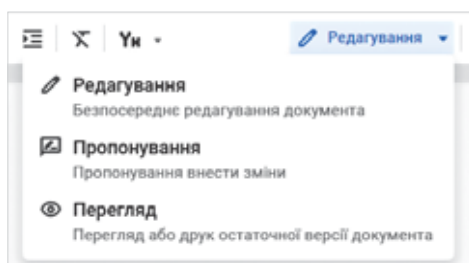
Мал. 7.8

У вікні надання доступу вказують рівні доступу (мал. 7.9).



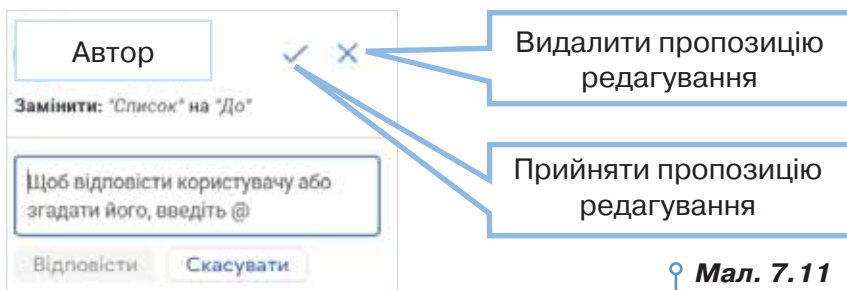
Мал. 7.9

Під час спільної роботи над документом кожен автор чи авторка отримує свій колір, який відповідає кольору обрамлення користувача у списку користувачів. Зміни можна вносити безпосереднім редагуванням тексту або в режимі пропонування (мал. 7.10).



Мал. 7.10

Якщо обрати режим пропонування, то текст в основному тексті буде виділено в рамку **ДоСписок учнів**, і в полі поруч із текстом відображено автора чи авторку й текст пропозиції (мал. 7.11).



Мал. 7.11

На доданий коментар можна відповісти, прийняти правки чи видалити їх.



Діємо

Вправа 2. Технології майбутнього.

Завдання. Створіть у групі з 4 учнів проєкт — опис новітньої технології.

1. Об'єднайтеся в групу з 4 осіб. Оберіть напрям свого дослідження: біологія, дослідження, космос, транспорт, інформаційні технології, їжа, будівництво, робототехніка, медицина й інші.
2. Оберіть лідера чи лідерку групи, що створить документ і надасть усім членам групи доступ до його редагування.
3. Розподіліть обов'язки та знайдіть відомості в інтернеті за обраним напрямом. Обговоріть і сформулюйте пошуковий запит, наприклад, *топ-10 технологій майбутнього*.
4. У спільному текстовому документі додайте опис технології, зображення, посилання на матеріали, які використано в роботі.
5. У коментарях до об'єктів документа вкажіть власний внесок у спільну роботу. Перевірте, чи збігається опис внеску з історією змін у документі.
6. Розмістіть документ у спільному сховищі класу.



Працюємо в парах

1. Обговоріть, чому важливо сьогодні вміти організовувати спільну роботу з документом. Наведіть приклади.
- ★ 2. Обговоріть, як можна відслідковувати зміни в текстовому документі. Визначте переваги й недоліки такої роботи для документів, що редагуються на персональному комп'ютері та в онлайн-овому сховищі.
- ★★ 3. Обговоріть способи створення спільного альманаху з робіт учнівських груп у мініпроєкті «Технології майбутнього». Вирішіть питання однакового форматування, корегування робіт, об'єднання у спільний документ. Додайте до об'єднаного документа зміст, заголовки, авторів, титульний аркуш. Остаточний варіант збережіть у форматі, що не передбачає змін. Роздрукуйте його або розмістіть на сайті школи.



Працюємо самостійно

4. У документі *Доповнена реальність*, що збережений у папці *Текстовий процесор*, ознайомтесь із примітками рецензента. Скористайтесь коментарями та внесіть зміни в документ.

- ★ 5. На *Google* диску створіть текстовий документ про власне захоплення. Додайте до документа рецепт приготування їжі чи інструкцію з виготовлення виробу, опис комп'ютерної гри чи розповідь про спортивне заняття або щось інше. Додайте до документа співавтора чи співаторку, які в класі мають подібні захоплення. Доопрацюйте спільно документ. Візьміть участь у доопрацюванні документа, до якого вам прийшло запрошення для спільної роботи.
- ★★ 6. Розгляньте матеріали про відстеження змін у текстовому документі на сайті підтримки офісного пакету. Знайдіть відповідні довідкові матеріали на офіційних сайтах підтримки офісу. Виділіть основні операції щодо відстеження змін.

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Маю уявлення про принципи та можливості опрацювання текстових даних.
- ☐ Умію знаходити й замінювати символи та групи символів.
- ☐ Розумію і можу пояснити принципи стильового оформлення документа.
- ☐ Умію використовувати стилі для форматування документа.
- ☐ Умію створювати й редагувати колонтитули документа.
- ☐ Розумію, що таке структура документа.
- ☐ Умію структурувати документ і створювати його зміст.
- ☐ Розумію та обґрунтовую необхідність дотримання вимог до стильового оформлення та структурування текстового документа.
- ☐ Умію створювати й використовувати гіперпосилання в текстовому документі.
- ☐ Можу пояснити принципи стильового оформлення та спільної роботи з документом.

Розділ 3

Опрацювання
мультимедійних об'єктів

8

Поняття мультимедіа

ПРИГАДАЙТЕ:

- ▶ склад мультимедійного обладнання;
- ▶ принципи кодування графічних зображень.

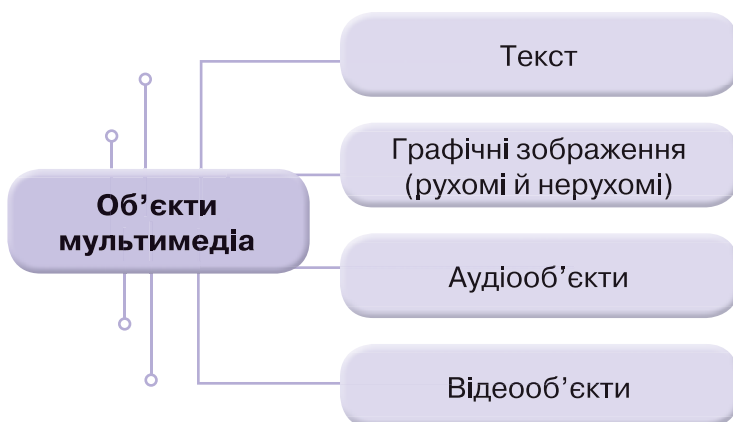
ВИ ДІЗНАЄТЕСЬ:

- ▶ що розуміють під мультимедіа;
- ▶ як кодуються аудіо- та відеодані;
- ▶ які існують формати файлів звукозапису;
- ▶ які формати відеофайлів є найбільш поширеними.

Вивчаємо

1. Що розуміють під мультимедіа?

Мультимедіа — це поєднання різних способів подання повідомлень. Сукупність таких повідомлень є об'єктами мультимедіа (мал. 8.1).



Мал. 8.1

Об'єкти мультимедіа — текст, рухомі й нерухомі графічні зображення, аудіо- та відеооб'єкти.

Опрацювання мультимедійних об'єктів передбачає їх створення і (або) зміну — редагування чи форматування.



Це цікаво

Мультимедіа — від лат. *multum* — багато, *medium* — середовище. Уперше термін мультимедіа виник у 1965 р. для опису театралізованих шоу, наприклад, *Exploding Plastic Inevitable* — шоу, що поєднало живу рок-музику, кіно, експериментальні світлові ефекти.

Об'єкти мультимедіа зберігаються у вигляді файлів, що містять мультимедійні дані.

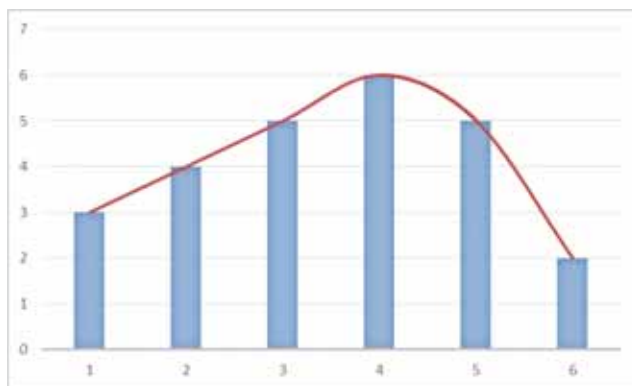
2. Як кодуються аудіо- та відеодані?

Для зберігання та опрацювання аудіо- й відеоданих за допомогою комп'ютера використовують двійкове подання даних. Процес кодування в цьому разі є досить складним, оскільки за допомогою комп'ютера можна зберігати й опрацьовувати лише обмежені обсяги даних, у той час як природні сигнали — носії даних — є неперервними. Тому перетворення аналогових сигналів у цифровий вигляд відбувається шляхом дискретизації.

Дискретизація (від англ. *discretisation*) — процес усунення неперервності (часової і просторової) аналогових сигналів — носіїв даних.

У випадку **просторової дискретизації** дані ділять на невеликі області, в межах яких характеристики (положення, колір, яскравість) вважаються незмінними.

У разі застосування **дискретизації в часі** на невеликі інтервали відповідно ділиться час. У межах цих інтервалів характеристики природних сигналів вважають незмінними (мал. 8.2).



Мал. 8.2

Наочним прикладом часової дискретизації є кодування звуку за допомогою комп'ютера — приведення до цифрової форми (оцифровування). До комп'ютера надходить не сам звук (звукова хвиля), а електричний сигнал, зареєстрований деяким пристроєм, наприклад, мікрофоном, радіо тощо.

Кодування звуку двійковим кодом далеко від стандартизації. Окремі компанії розробили свої корпоративні стандарти. Можна виділити два основні напрями.

Перший напрям базується на методі FM (*Frequency Modulation*), який полягає в тому, що складний звук можна розкласти на послідовність простих гармонійних сигналів різних частот, кожний з яких є правильною синусоїдою, а отже, може бути описаний числовими параметрами, тобто кодом. У природі звукові сигнали мають неперервний спектр, тобто є аналоговими. Їх подання у вигляді дискретних цифрових сигналів виконують спеціальні пристрої — аналого-цифрові перетворювачі. Зворотнє перетворення для відтворення звуку, закодованого числовим кодом, виконують цифро-аналогові перетворювачі. У разі таких перетворень неминучі втрати інформації, пов'язані з методом кодування, тому якість звукозапису зазвичай виходить не цілком задовільною і відповідає якості звучання простих електромузичних інструментів із забарвленням, характерним для електронної музики. Водночас даний метод кодування забезпечує досить компактний код, і тому він знайшов застосування ще в ті роки, коли ресурси засобів обчислювальної техніки були явно недостатні.

Другий напрям використовує метод таблично-хвильового (*Wave-Table*) синтезу та краще відповідає сучасному рівню розвитку техніки. Якщо говорити спрощено, то можна сказати, що десь у наперед підготовлених таблицях зберігаються зразки великої кількості «реальних» звуків. Такі зразки звуків називають **семплами**. Числові коди відображають тип звуку, номер його моделі, висоту тону, тривалість й інтенсивність звуку, динаміку його зміни, деякі параметри середовища, в якому відбувається звучання, а також інші параметри, що характеризують особливості звуку. Оскільки як зразки використовуються «реальні» звуки, то якість звуку, одержаного в результаті синтезу, дуже висока й наближається до якості звучання реальних музичних інструментів.

Для кодування відеоданих використовують дискретизацію рухомого зображення. У цьому разі ілюзія руху створюється шляхом швидкої зміни кадрів. При цьому кадри залишаються статичними. Тоді з кожним кадром пов'язують таблицю кодування, в якій кожний піксель зображення характеризується трьома величинами: розміром кадру по горизонталі, по вертикалі та часом, з яким пов'язаний цей кадр. Зрозуміло, що таких таблиць є дуже багато, тому відеодані займають великий обсяг пам'яті.



ДІЕМО

Вправа 1. Дослідження звуку.

Завдання. Дослідіть, як за допомогою комп'ютера можна візуально закодувати звук у сервісі <https://musiclab.chromeexperiments.com/Spectrogram/>.



1. У браузері перейдіть за посиланням <https://musiclab.chromeexperiments.com/Spectrogram/>.
2. Обирайте по черзі джерела звуку.



3. Розгляньте спектрограму звуку.
Примітка. **Спектрограма** — це візуальне зображення спектра частот сигналу в часі. При застосуванні до звукового сигналу спектрограми іноді називають сонографами, голосовими відбитками або голосограмами. Коли дані подано в тривимірному графіку, вони можуть називатися водоспадами.
4. Висловте припущення про спосіб кодування і подання звукових даних у браузері.

3. Які існують формати файлів звукозапису?

Формати аудіо- й відеофайлів визначають їхні структуру та способи кодування. Способи кодування звукових і відеоданих у різних форматах визначають якість звуку чи відтворення відео та ступінь стиснення даних, що впливає на обсяг файлів.

Існують такі найпоширеніші формати звукозапису (табл. 8.1):

Таблиця 8.1

Аудіо-формат	Повна назва формату	Особливості формату
MIDI	<i>Musical Instrument Digital Interface</i>	На відміну від інших аудіоформатів, являє собою не оцифрований звук, а набори вказівок (інструмент, ноти, що програватимуться, значення параметрів звуку тощо), які можуть відтворюватися по-різному, залежно від пристрою відтворення. Дає змогу обмінюватися даними між музичними інструментами, синтезаторами та комп'ютерами
WAV	<i>Waveform audio format</i> (від англ. <i>wave</i> — хвиля)	Використовується в операційній системі Windows. Аудіоформат без використання стиснення. Точно передає звук, але займає значний обсяг на диску

Продовження таблиці 8.1

Аудіо-формат	Повна назва формату	Особливості формату
MP3	MPEG Layer 3	Запис музики в цьому форматі відбувається зі стисненням обсягу з майже непомітним для слуху погіршенням якості, але обсяг зменшується в 10–12 разів порівняно з оригінальним музичним форматом. Принцип стиснення даних нагадує графічний формат JPEG — стиснення відбувається за рахунок виключення частот, які не сприймаються вухом людини
WMA	Windows Media Audio	Розроблений компанією <i>Microsoft</i> як альтернатива формату MP3. Ступінь стиснення даних і якість звуку майже аналогічні формату MP3. Нові версії формату, починаючи з <i>Windows Media Audio 9.1</i> , передбачають кодування без втрати якості, багатоканальне кодування об'ємного звуку та кодування голосу
AAC	Advanced Audio Coding	Під час кодування значно зменшується обсяг даних, потрібних для передавання високоякісного цифрового аудіо. У цьому форматі відбувається менша втрата якості, ніж у MP3, за однакових обсягів даних



Це цікаво

Кодек (від англ. *coder/decoder* — кодувальник/декодувальник) — пристрій чи програма, що виконує перетворення сигналів і використовується під час цифрового опрацювання відео та звуків для стиснення даних. Стиснення, як правило, відбувається із втратою якості. Кодеки дають змогу кодувати відеозаписи для передавання чи збереження, а також розкодувати — для перегляду. Різні медіаконтейнери можуть підтримувати різні кодеки: DivX, XviD, MJPEG, VC-1 тощо.

4. Які формати відеофайлів є найбільш поширеними?

Формати відео є медіаконтейнерами, тобто можуть містити дані різних типів, стиснені різними кодеками, і дають змогу зберігати аудіо-, відео- та текстові дані (зокрема субтитри) в одному файлі. Медіаконтейнер не лише надає можливість зберігати аудіо- й відеозаписи, а й забезпечує синхронізацію аудіо- та відеопотоків під час відтворення.

Відео може бути збережене на різних носіях. Як правило, відео високої якості має значний обсяг. Наприклад, музична комедія «Сорочинський ярмарок», записана на DVD, має обсяг 6,71 ГБ.

Як і формати звукозапису, різні формати відеофайлів передбачають різні способи кодування даних, що визначають якість відео, ступінь стиснення даних та обсяг файлу. Деякі формати можуть містити потокове відео, що використовується для передавання даних через інтернет у режимі реального часу.

Існують такі поширені формати відео (табл. 8.2):

Таблиця 8.2

Відео-формат	Повна назва формату	Особливості формату
AVI	Audio-Video Interleaved	Може містити потоки чотирьох типів: відео, аудіо, MIDI, текст. Для стиснення аудіо- та відеозаписів можуть використовуватися різні кодеки. Має деякі обмеження, зокрема обсяг файлу не може перевищувати 4 ГБ. На зміну цьому формату створено формат WMV
MPEG	Motion Picture Experts Group	Розроблений експертною групою з питань рухомого зображення (MPEG). Було створено такі алгоритми стиснення даних: MPEG1, MPEG2 та MPEG4
MOV	QuickTime Movie	Один з перших відеоформатів, що набув широкого розповсюдження. Ступінь стиснення є досить високим
ASF	Advanced Systems Format (раніше також Advanced Streaming Format, Active Streaming Format)	Є частиною мультимедійного набору <i>Windows Media</i> для створення і розповсюдження аудіо- та відеофайлів. Формат файлів, що містять потокове аудіо та відео. Може використовуватись як для локального відтворення, так і для передавання і відтворення комп'ютерними мережами, зокрема інтернетом. Особливістю формату є можливість відтворення безпосередньо в момент завантаження по мережі в режимі реального часу, що нагадує телевізійне мовлення, тобто <i>потокового відтворення</i> . Зазвичай використовується розширення файлу <i>asf</i> , крім того, файли, що містять звукові записи, можуть мати розширення <i>wma</i> , а відеофайли — <i>wmv</i>
WMV	Windows Media Video	Є частиною мультимедійного набору <i>Windows Media</i> . Створений на основі формату AVI, але має додаткові можливості, зокрема засоби захисту від несанкціонованого копіювання. Використовується для розповсюдження фільмів і відеокліпів
3GP	3rd Generation (mobile) Phone	Формат для збереження і перегляду відео на мобільних телефонах 3-го покоління. Відеозаписи в цьому форматі мають невеликий обсяг порівняно з іншими форматами відео, але за рахунок погіршення якості

Продовження таблиці 8.2

Відео-формат	Повна назва формату	Особливості формату
RM	<i>Real Media</i>	Розроблений компанією <i>RealNetworks</i> для розповсюдження відео через інтернет. Ступінь стиснення даних і якість відео достатньо високі. Використовується для розповсюдження фільмів і трансляції так званого інтернет-телебачення
VOB	<i>Video Object</i>	Формат файлів, що використовується для збереження DVD-відео. Створений на основі MPEG2, може містити декілька потоків аудіо, відео, субтитри, а також меню фільму. Використовується для розповсюдження фільмів на DVD
FLV	<i>Flash Video</i>	Розроблений компанією <i>Adobe Systems</i> . Використовується для передавання відео через інтернет, зокрема таким сервісом, як <i>YouTube</i> та ін. Файли в цьому форматі можна переглядати в більшості операційних систем, оскільки для цього використовується програвач <i>Adobe Flash Player</i> , який розповсюджується у вигляді <i>плагіна</i> для різних браузерів і різних операційних систем



Це цікаво

Плагін (від англ. *plug-in* — підключати) — незалежний програмний модуль, що підключається до основної програми та призначений для розширення або використання її можливостей. Часто у вигляді плагіна виконується підтримка форматів файлів, наприклад, для звукових і відеопрогравачів, програм опрацювання звуку й графіки тощо. У веббраузерах плагіни використовуються для забезпечення відображення форматів даних, які не мають вбудованої підтримки браузером (наприклад, *Adobe Flash* або *SVG*), для підлаштування можливостей до вимог користувача тощо.



Досліджуємо

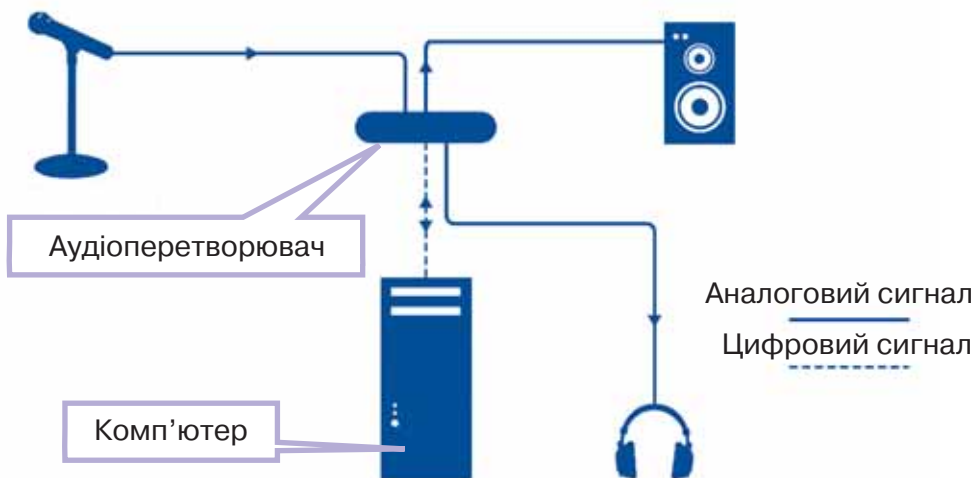
Вправа 2. Формати аудіофайлів.

Завдання. Дослідіть властивості файлів із записами того самого фрагмента мелодії Петра Чайковського до балету *Лускунчик*, які збережені в папці Аудіо папки *Мультимедіа*. Проранжуйте файли за розміром.



Працюємо в парах

1. Користуючись Вікіпедією, знайдіть відомості про особливості:
а) аудіоформатів; б) відеоформатів.
Порівняйте відомості, наведені російською, українською та англійською мовами (за потреби скористайтеся перекладачем). Обговоріть у парах.
2. Обговоріть і поясніть процес, який змодельовано на схемі.



3. Дослідіть візуальний спосіб кодування звуку в середовищі <https://musiclab.chromeexperiments.com/Experiments>. Використайте експерименти *Мелодія*, *Перфолента*, *Виробник пісень*. Порівняйте ці експерименти.



Працюємо самостійно

1. Використовуючи словник або перекладач, перекладіть повні назви аудіо- та відеоформатів українською мовою. За потреби зверніться до Вікіпедії.
2. Приєднайте формати файлів з відповідним поясненням у вправі за посиланням <https://learningapps.org/view1235048>.
3. Відкрийте файл *Експертна група MPEG* в папці *Мультимедіа*. Ознайомтеся із запропонованими матеріалами й дайте відповіді на запитання:
а) як часто збирається експертна група з питань рухомого зображення (MPEG); б) у скільки разів відбувається стиснення відео в разі використання формату MPEG; в) як працює MPEG відео?





Опрацювання об'єктів мультимедіа

9

ПРИГАДАЙТЕ:

- ▶ що захищає авторське право;
- ▶ програми для перегляду зображень.

ВИ ДІЗНАЄТЕСЬ:

- ▶ за допомогою яких програм можна переглядати відео та слухати звукозаписи;
- ▶ за допомогою яких програм захоплюють звук;
- ▶ за допомогою яких програм захоплюють відео;
- ▶ за допомогою яких програм можна перетворити звукозаписи й відеофайли з одного формату на інший.



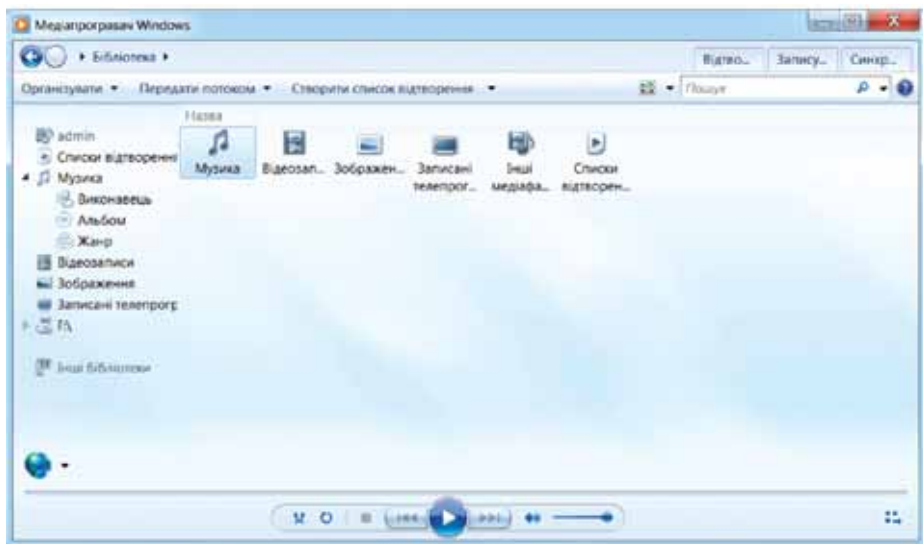
Вивчаємо

1. За допомогою яких програм можна переглядати відео та слухати звукозаписи?

Для відтворювання мультимедійних даних створено спеціальні програми — **мультимедійні програвачі**.

Мультимедійні програвачі входять до складу більшості сучасних операційних систем. Зокрема, в *Apple Mac OS X* — *iTunes* та *QuickTime Player*. В операційних системах на основі *GNU/Linux* може бути встановлено відразу кілька мультимедійних програвачів: *VLC*, *MPlayer*, *Xine* і *Totem*. В *OC Windows* мультимедійним програвачем є *Медіапрогравач Windows*. Для відтворювання звукових файлів і відеозаписів можна використати також інші програми, додатково їх установивши.

Програма *Медіапрогравач Windows* (мал. 9.1) поєднує в собі функції відеомагнітофона, аудіопрогравача й радіо. У нижній частині вікна, що з'являється після запуску програми, розташовано панель, за допомогою якої можна керувати відтворенням звукових файлів і відеозаписів: повторити відтворення обраного з каталогів музики, відео, зображень; зупинити відтворення , відтворити обране , перемотати далі ; змінити гучність .



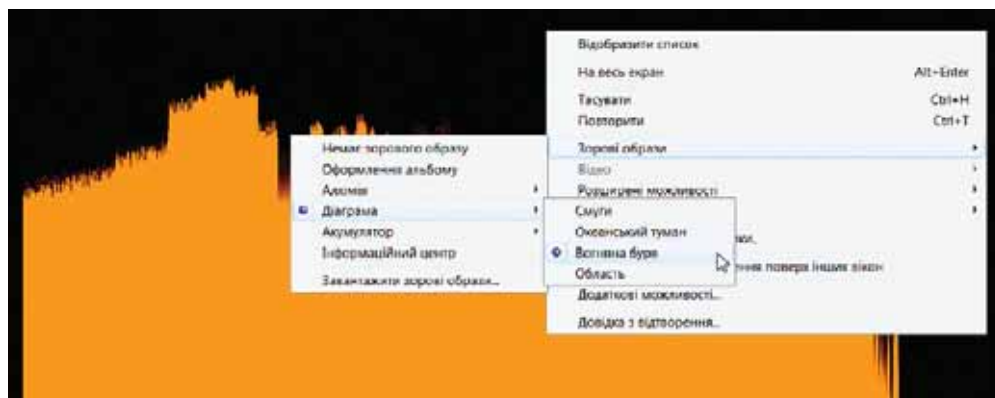
Мал. 9.1

Залежно від того, відтворюється чи ні в даний момент деякий звуковий або відеофайл чи здійснюється перегляд зображення, вигляд і призначення деяких кнопок можуть змінюватися.

Існує два режими роботи програвача: *Бібліотека програвача* та *Поточний список відтворення*. У першому режимі доступно багато функцій програвача — тому він і називається бібліотекою. Перехід у режим *Поточного списку* відтворення здійснюється за допомогою



кнопки у правому нижньому куті вікна програми. При відтворенні звуку у вікні програвача можуть відображатись обрані зорові образи. Наприклад, на малюнку при відтворенні звукового файлу обрано зоровий образ *Діаграма/Вогняна буря*. Його можна змінити, якщо в контекстному меню вікна вибрати пункт *Зорові образи* (мал. 9.2).



Мал. 9.2



ДІЄМО

Мал. 9.3



Вправа 1. Музика.

Завдання. Перегляньте звукозапис із бібліотеки *Музика*, установленної на комп'ютері. Установіть зоровий образ *Яскрава сфера* для мелодії, що відтворюється.

1. Запустіть на виконання програму *Медіапрогравач Windows*. Для цього оберіть *Пуск/Усі програми/Медіапрогравач Windows*.
2. Виберіть папку *Мультимедіа\Аудіо*. У списку звукових файлів виберіть мелодію *Ой, у лузі червона калина.trp3*.
3. Розпочніть прослуховування обраного файлу. На панелі елементів натисніть кнопку *Відтворити*. Регулюйте гучність відтворення звуку на колонках за допомогою *Панелі елементів* програми *Медіапрогравач Windows* або за допомогою регулятора гучності на *Панелі завдань* (мал. 9.3).
4. Призупиніть відтворення файлу, натиснувши на кнопку *Пауза*.
5. Установіть зоровий образ для мелодії. Перейдіть до режиму роботи мультимедійного програвача — *Поточного списку відтворення*. У контекстному меню програвача виберіть *Зорові образи*. У списку *Акумулятор* виберіть зоровий образ — *Яскрава сфера*.
6. Завершіть роботу з програвачем.

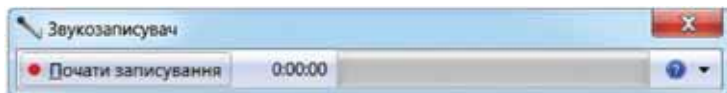
Вправа 2. Прогулянка Києвом.

Завдання. Перегляньте відеофайл *Київ.avi*.

1. Відкрийте папку *Мультимедіа\Відео* й виділіть файл *Київ.avi*.
2. Клацніть правою кнопкою миші на значку файлу й виберіть з контекстного меню вказівку *Відкрити за допомогою*.
3. Визначте, за допомогою яких програвачів мультимедіа, установлених на комп'ютері, можна відтворити цей відеозапис.
4. Зі списку програм у підменю *Відкрити за допомогою* виберіть *Медіапрогравач Windows*. Перегляньте відеозапис.
5. Установіть, які мультимедійні об'єкти вміщує файл *Київ.avi*.

2. За допомогою яких програм захоплюють звук?

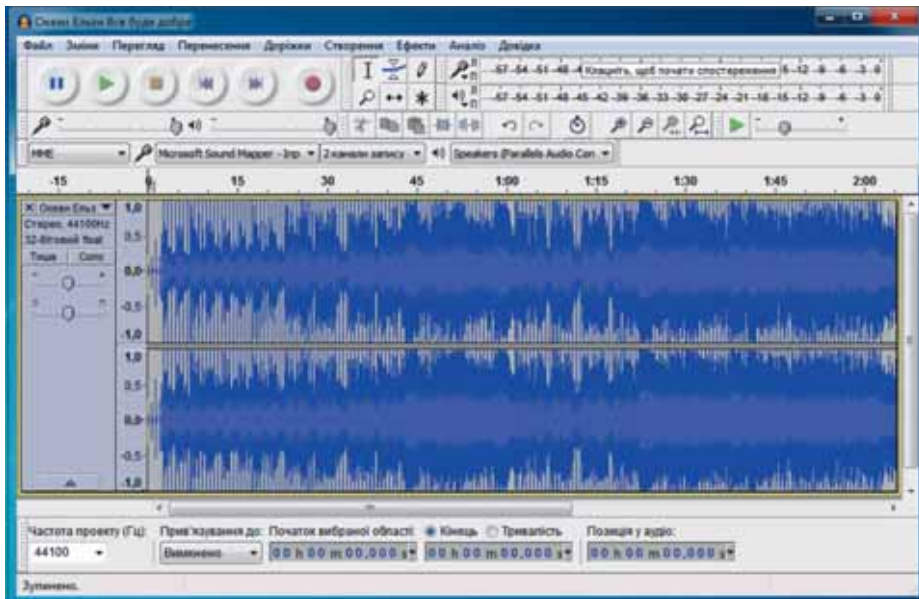
Щоб записати звуковий файл, можна скористатися програмою *Звукозаписувач*, яка входить до стандартних програм *ОС Windows*. Зрозуміло, що при цьому має бути підключений мікрофон (мал. 9.4).



Мал. 9.4

Кнопкою *Почати записування* запускається відлік тривалості звукозапису, а сама кнопка змінює свій вигляд на *Зупинити записування*.

Для роботи з аудіо використовують і більш потужні програми, в яких можна не тільки створювати звукові файли, а й вносити зміни до аудіозаписів. Такі програми мають назву **звукових редакторів**, або **аудіоредакторів** (мал. 9.5).



Мал. 9.5

Аудіоредактори використовують, щоб записати звуковий файл із мікрофона, об'єднати кілька аудіозаписів, вирізати та зберегти фрагмент аудіозапису в окремому файлі, видалити паузи, позбавитися шуму тощо.

Існують як безкоштовні, так і потужніші звукові редактори, які є платними. Прикладами звукових редакторів є *Audacity*, *Acoustica*, *mp3DirectCut*, *Oscenaudio*, *Adobe Audition*, *Audio Edit Deluxe* тощо.

Звукові дані в середовищі аудіоредактора графічно подаються у хвильовій формі. Область із таким графічним зображенням у звуково-му редакторі називають **треком**, або **звуковою доріжкою**.



ДІЄМО

Вправа 3. Аудіоредактор *Audacity*.

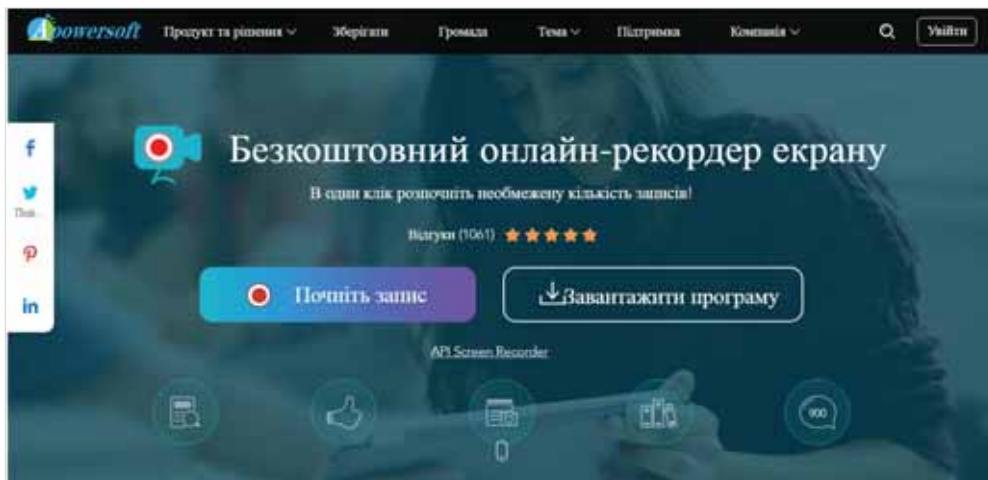
Завдання. Визначте призначення деяких елементів керування аудіоредактора *Audacity*.

1. Завантажте програму *Audacity*.

2. Розгляньте елементи вікна програми.
3. У меню *Файл* виберіть вказівку *Відкрити* та відкрийте файл *Океан Ельзи Все буде добре.mp3*, що збережений у папці *Мультимедіа\Аудіо*.
4. Визначте призначення інструментів.
5. Прослухайте аудіозапис, скориставшись інструментом *Відтворення*. Визначте, коли розпочинається останній приспів.
6. Виконуючи протягування мишею, виділіть фрагмент аудіозапису, що містить останній приспів.
7. У нижній частині вікна перегляньте значення часу для властивостей *Початок вибраної області* та *Кінець*.
8. Ознайомтеся із вказівками, які містять різні пункти меню програми.

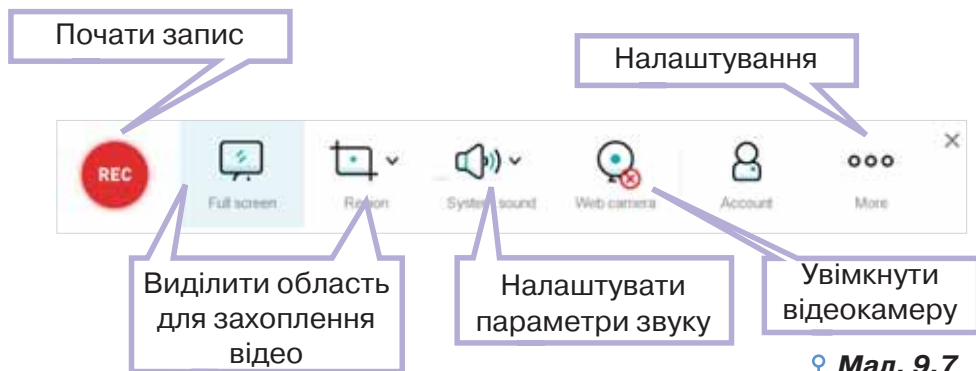
3. За допомогою яких програм захоплюють відео?

Для захоплення відео можна скористатись як онлайн-програмами, так і програмами, які встановлюються на комп'ютер. Наприклад, *Free Online Screen Recorder* (<https://cutt.ly/rkIzzIT>) — сервіс, який може працювати як в онлайн-версії, так і офлайн. Для того щоб використовувати сервіс без встановлення на комп'ютер, потрібно заздалегідь завантажити із сайту і встановити на комп'ютер спеціальну надбудову — **лаунчер**. Після цього програма *Free Online Screen Recorder* стане доступною для роботи онлайн (мал. 9.6).



Мал. 9.6

Після запуску програми на виконання або з екрана браузера запис відео здійснюють вказівкою *Почніть запис*. Користувачам буде доступна панель інструментів (мал. 9.7).



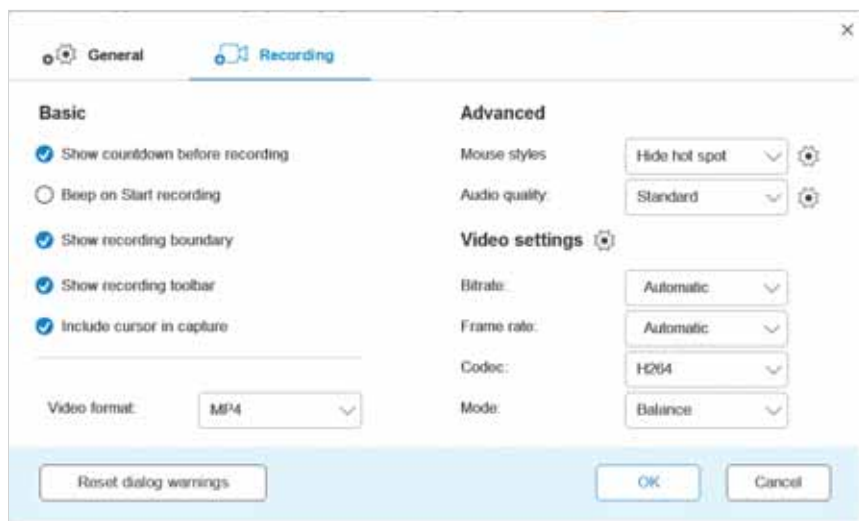
Мал. 9.7

На панелі інструментів можна вказати джерело звуку або зовсім відключити звук:

- системні звуки (*system sound*);
- мікрофон (*microphone*);
- і системні звуки, і мікрофон (*system sound і microphone*).

У налаштуваннях сервісу можна вибирати формат записуваного відео, наприклад, MP4, WMV, AVI, MOV, MPEG та ін. Щоб відкрити налаштування, на панелі обирають *More* → *Settings* → *Recording* (мал. 9.8).

Мал. 9.8



При натисканні *Rec* висвічуються комбінації клавіш, які необхідні для роботи з даною програмою.

Щоб зберегти відео на комп'ютер, можна натиснути на *Stop* і вибрати, куди треба вивантажити файл. Його можна зберегти на комп'ютер. Для цього натисніть *Save as video file*, і документ збережеться за таким шляхом: *Мої документи* → *Apowersoft* → *Apowersoft Online Screen Recorder*. Також отримане відео можна завантажити в *YouTube*, *Vimeo*, *Google Drive*, *Dropbox*.

4. За допомогою яких програм можна перетворити звукозаписи й відеофайли з одного формату на інший?

На практиці часто виникає потреба перетворити аудіо- чи відеофайли в інший формат. Наприклад, відеозаписи, зняті на мобільний телефон у форматі 3GP, для подальшого опрацювання можуть бути перетворені у формат AVI або WMV.

Для перетворення файлів з одного формату в інший використовують спеціальні програми — конвертери, вибір яких залежить від вихідного формату файлу та формату, на який його потрібно перетворити.



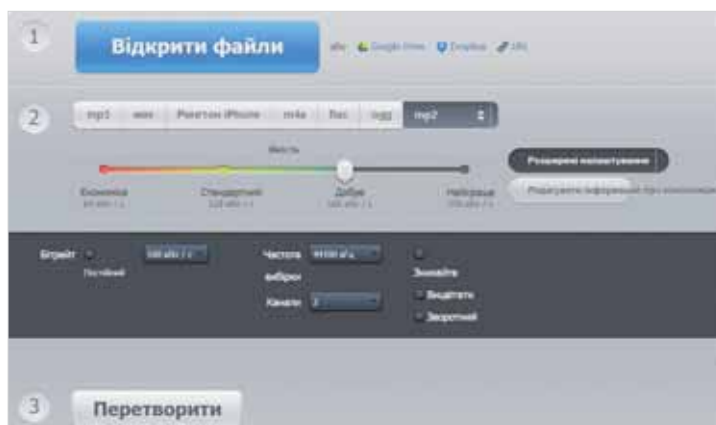
Це цікаво

Конвертер — від лат. *convertere* — перетворювати.

Конвертер — програма, яка перетворює у файлі дані з одного формату в інший. Зміни та втрати даних, які можуть виникнути під час перетворення, залежать від форматів початкового й кінцевого файлів і від застосованої програми перетворення.

Існують різні програми, призначені для перетворення аудіо- чи відеофайлів, деякі з них є безкоштовними. Такі програми можна завантажити з інтернету, або використати програми для конвертації файлів онлайн. Наприклад, простий безкоштовний сервіс для конвертування аудіо <https://online-audio-converter.com/>. Після завантаження сервісу у браузері користувач отримує інструменти з покроковою інструкцією дій (мал. 9.9):

 Мал. 9.9



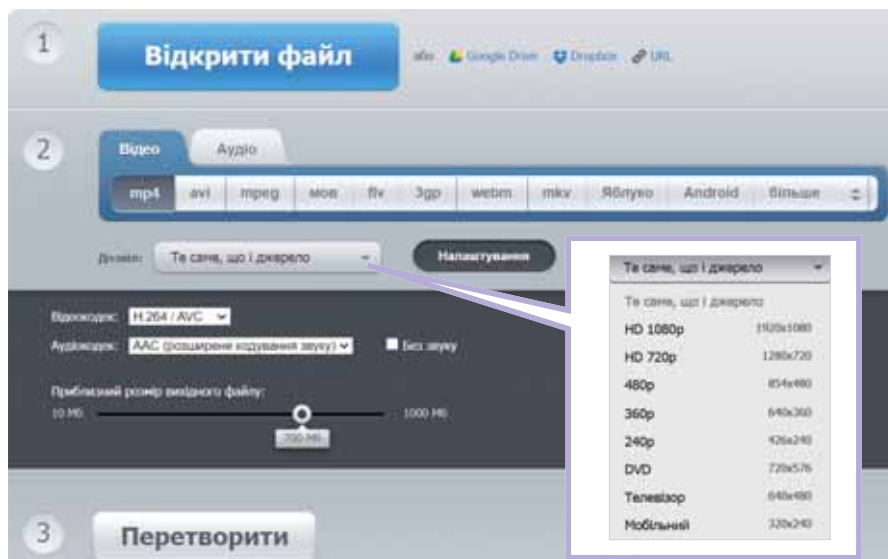
- 1) завантажити файли з комп'ютера або хмарних сховищ;
- 2) обрати формат файлу, якість перетворення і розширені налаштування, відредагувати інформацію про композицію — заголовок,

автор, альбом, рік, жанр. Тривалість процесу конвертування залежить від обраних налаштувань та обсягу файлу для конвертації;

3) виконати конвертацію.

Після конвертування файлу його можна зберегти на комп'ютер або завантажити в хмарне середовище.

За таким само принципом працює відеоконвертер (<https://convert-video-online.com/>). Програма дає змогу налаштувати параметри конвертування файлу, вказати передбачуваний розмір та інші. А в списку *Дозвіл* обирають розміри екрана для відтворення відео (мал. 9.10).



Мал. 9.10



Працюємо в парях

1. Висловіть припущення, як зорові образи в програмі *Медіапрогравач Windows* можуть бути пов'язані з кодуванням звуку. Запропонуйте одне одному демонстрацію своєї версії, вибравши приклади з різних груп зорових образів.
2. Обговоріть, які аудіоредактори ви використовуєте, наприклад, у своєму смартфоні. Чим вони подібні та чим відрізняються від редактора *Audacity*. Зробіть висновок про перевагу одного над іншим.
- ★ 3. Запропонуйте одне одному ситуації, в яких корисно застосувати програму захоплення відео. Як під час захоплення відео дотримуватися Закону про авторське право?
- ★ 4. Дослідіть програми, які зібрані на сайті <https://123apps.com>. Складіть інструкції з використання однієї з програм:





Аудіо різак



Аудіо Столяр



Відеорізак



Відеомагнітофон



Диктофон

Розмістіть інструкції у спільному сховищі. Використайте одну із запропонованих кимось інструкцій і виконайте дії над аудіо- чи відео-файлом, які збережено в папці *Мультимедіа\Файли* для редагування.

- ★ ★ 5. Попрацюйте в мініпроекті «Студія мультимедіа». Оберіть групу, в якій ви будете працювати над створенням: аудіо чи відео. У своїй групі об'єднайтесь у пари, оберіть назву програми чи сервісу онлайн для роботи з обраними даними. Повідомте про свій вибір іншим, щоб не повторюватись. Дослідіть обрану програму чи сервіс. Своріть відеоінструкцію зі встановлення/реєстрації та роботи з обраною програмою. Розмістіть відеоінструкцію у спільному сховищі.



Працюємо самостійно



6. Відновіть алгоритм запису звуку за допомогою програми *Звукозапис* у вправі за посиланням <https://learningapps.org/view8856560>.
- ★ ★ 7. Переконвертуйте файл *Скрипка*, що збережений у папці *Мультимедіа*, у різні формати аудіофайлів. Збережіть нові файли в папку своєї структури папок. Порівняйте розмір і прослухайте файли. Який з форматів файлів найбільше вплинув на якість відтворення звуку? Поміркуйте про причину.
8. Переконвертуйте файл *Колядка*, що збережений у папці *Мультимедіа*, у різні формати аудіофайлів. Збережіть нові файли в папку своєї структури папок. Порівняйте розмір і прослухайте файли. Який з форматів файлів найбільше вплинув на якість відтворення звуку? Поміркуйте про причину.
- ★ ★ 9. Підготуйте фрагмент звукозапису улюбленої української пісні для гри «Відгадай мелодію». Тривалість звукозапису — не більше 10 с. Фрагмент — від початку. Розмістіть звукозапис у спільній папці класу. Назвіть файл своїм прізвищем.
- ★ ★ 10. З відеофайлу *Лісові тварини для дітей*, що збережений у папці *Мультимедіа* (або з каналу *Ладоньки-Ладоньки — мультики українською мовою*) виділіть відеофрагменти, які демонструють процес малювання однієї із тварин: ведмідь, енот, лось, білка, косуля, сова, їжак, олень, заєць, кабан, бобр, лисиця, бурундук, вовк, рись. Розмістіть відео у спільній папці класу. До назви файлу додайте назву тваринки та своє прізвище.



Створення та опрацювання відео

ПРИГАДАЙТЕ:

- ▶ як у презентаціях працюють з мультимедійними даними.

ВИ ДІЗНАЄТЕСЬ:

- ▶ за допомогою яких програм можна створити власний відеокліп;
- ▶ які особливі елементи інтерфейсу має програма *OpenShot Video Editor*;
- ▶ як додати до проєкту мультимедійні об'єкти;
- ▶ як додати відеопереходи між кліпами чи зображеннями й візуальні ефекти;
- ▶ як додати до відеокліпу музичний супровід і налаштувати часові параметри відео й аудіо;
- ▶ як додати до проєкту написи та зберегти створений відеокліп у форматі відео.



Вивчаємо

1. За допомогою яких програм можна створити власний відеокліп?

Для створення відеокліпів використовують відеоредактори.

Відеоредактор — це програма, що містить набір інструментів, за допомогою яких створюють і редагують відеофайли на комп'ютері.

Різні відеоредактори мають різні можливості й набори інструментів для опрацювання відеофайлів.

Прикладом відеоредактора є програма *OpenShot Video Editor*, яку можна безкоштовно завантажити з офіційного сайту (<https://www.openshot.org/uk/features/>). Використовуючи інструменти цього редактора, можна видаляти зайві кадри з відеозапису, розташовувати відеофрагменти в будь-якій послідовності, додавати музичні файли, голосовий супровід, титри тощо.

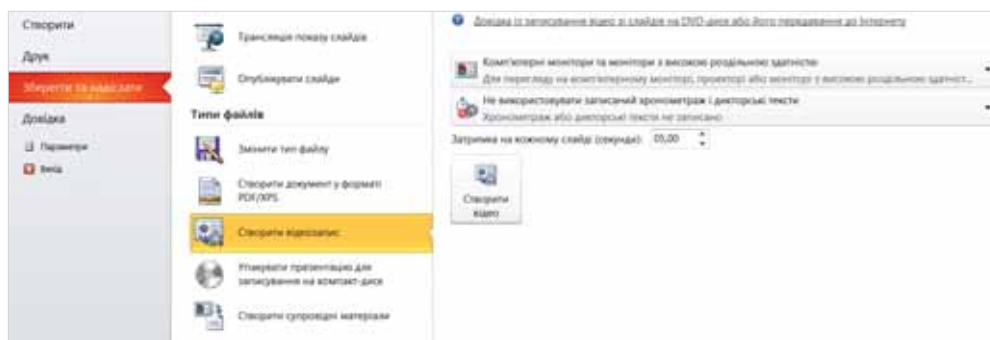
Існують також інші відеоредактори, зокрема *Кіностудія (Windows Movie Maker)*, *VirtualDub*, *Pinnacle Studio*, *MS Producer*, *Adobe After Effects*, *Adobe Premiere*, *Ulead VediaStudio*, *SONY Vegas Pro* тощо.

За допомогою відеоредактора створюється проєкт — файл, що містить відомості про порядок розташування і час відтворення аудіо-

й відеокліпів, відеопереходи, відеоефекти, назви, титри тощо. Після збереження проєкту його файл можна відкрити пізніше в середовищі відеоредактора та внести до нього зміни.

Готовий проєкт може бути збережений як фільм — відеофайл у відповідному форматі, наприклад, WMV або MPEG4. Фільм можна зберегти на комп'ютері чи компакт-диску, відправити електронною поштою чи розмістити в інтернеті.

Окремим способом створення відео є експортування презентації у відео. Для цього після підготовки презентації вибирають вказівку *Файл/Зберегти та надіслати/Створити відеозапис*. Обирають потрібні параметри та натискають кнопку *Створити відео* (мал. 10.1).



Мал. 10.1

2. Які особливі елементи інтерфейсу має програма *OpenShot Video Editor*?

Крім стандартних елементів вікна, зокрема рядка заголовка та стрічки, що містить вкладки з інструментами, вікно відеоредактора *OpenShot Video Editor (OpenShot)* містить також особливі елементи, притаманні відеоредакторам. Це вікно попереднього перегляду, яке використовується для перегляду як окремих кліпів, так і всього проєкту перед збереженням. Також це область, у якій створюються і монтуються проєкти — *Монтажний стіл*, що використовується для перегляду та зміни послідовності об'єктів проєкту — кліпів (мал. 10.2).



Мал. 10.2



ДІЄМО

Вправа 1. Ознайомлення з відеоредактором *OpenShot*.

Завдання. На офіційному сайті <https://www.openshot.org> ознайомтеся з особливостями поточної версії відеоредактора *OpenShot*.

1. Перейдіть за посиланням <https://www.openshot.org>.
2. Ознайомтеся із можливостями користувача програми *OpenShot*. Виділіть три можливості, якими ви хотіли б опанувати.
3. Перегляньте відео про особливості поточної версії. Для цього перейдіть у меню *Можливості/Відео*.
4. Виділіть для себе одну особливість, яка, на ваш погляд, дуже допомагає в роботі з редактором.
5. Розгляньте, де з офіційного сайту можна завантажити поточну версію відеоредактора.

Вправа 2. Робота з відеоредактором *OpenShot*.

Завдання. Перегляньте відеозапис про призначення та основні прийоми роботи з відеоредактором *OpenShot*.

1. Відкрийте папку *Мультимедіа\Відео* та запустіть на відтворення файл *Огляд* програми *OpenShot*.
2. Перегляньте відеозапис.
3. Поясніть призначення елементів вікна відеоредактора *OpenShot* і дії, які можна виконати за допомогою елементів вікна.

3. Як додати до проєкту мультимедійні об'єкти?

Відеокліп, що створюється за допомогою програми *OpenShot Video Editor*, може містити відеофрагменти, аудіозаписи та статичні зображення. Для додавання до проєкту мультимедійних об'єктів, що зберігаються у файлах, слід скористатись інструментом  або просто перетягнути файл у сховище файлів проєкту.

Якщо мультимедійні об'єкти збережені у форматах, що не підтримуються програмою *OpenShot Video Editor*, для використання в проєкті їх слід спочатку конвертувати до одного з підтримуваних форматів.

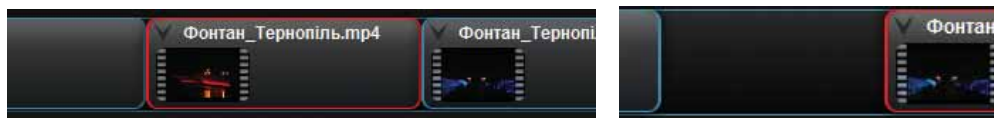
Кожний завантажений об'єкт перетягують на часову шкалу проєкту. Якщо об'єкти містяться на одній доріжці, то у фінальному відео вони будуть відображатись послідовно, а якщо на різних, то одночасно. Якщо об'єкти будуть накладені один на одний, то *OpenShot* автоматично створить плавне затухання між ними, що відображається синіми округлими прямокутниками між кліпами. Зауважте, що кліпи можна переставляти стільки разів, скільки потрібно, просто перетягуючи їх.

Додані до проєкту відеокліпи іноді потребують додаткового редагування — поділу на кілька частин та приховування початку чи кінця кліпу.

Якщо час відтворення кліпу достатньо великий і при цьому він потребує редагування — вилучення кадрів чи додавання відеопереходу всередині кліпу, то такий кліп можна розділити. Для поділу кліпу на

вкладці слід вибрати інструмент *Лезо*  і клацнути індикатором розрізання в потрібному місці об'єкта. І так продовжуємо, поки не будуть «нарізані» потрібні фрагменти. Будь-який фрагмент можна виділити (червона рамка) і видалити (мал. 10.3).

Мал. 10.3



Для того щоб приховати початок або кінець кліпу, перетягують мишею лівий чи правий край.



ДІЄМО

Вправа 3. Створення відеокліпу, що складається з кількох відеозаписів.

Завдання. Створіть проєкт про музичні фонтани, що містить відеокліпи фонтанів в Одесі, Барселоні, Празі, Лас-Вегасі.

1. У власній структурі папок створіть папку *Мультимедіа*.
2. Відкрийте вікно відеоредактора *OpenShot* й оберіть інструмент *Додати мультимедійні об'єкти*.
3. Додайте до проєкту всі відеозаписи, що збережені в папці *Мультимедіа\Відео\Музичні фонтани*.
4. Перемістіть послідовно відеофайли на одну доріжку.
5. Перегляньте створений проєкт. Для цього під вікном перегляду скористайтесь інструментами .

6. Збережіть створений проєкт з іменем *Фонтани_Прізвище* в папці *Мультимедіа* у власній структурі папок. Для цього оберіть вказівку *Файл/Зберегти проєкт*.

Вправа 4. Створення відео з накладанням фото на відео.

Завдання. Створіть відео про фонтани в місті Тернополі, у якому на фоні буде транслюватись відео вечірньої зйомки аераційного фонтана, а по центру екрана демонструватимуться зображення інших фонтанів.

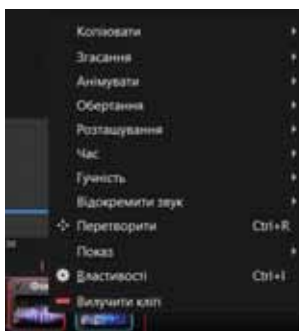
1. Відкрийте вікно відеоредактора *OpenShot* й оберіть інструмент *Додати мультимедійні об'єкти*.
2. Додайте до проєкту всі файли, що збережені в папці *Мультимедіа\Відео\Тернопіль*.

3. Перемістіть відео на доріжку 4, а зображення — на доріжку 5 (мал. 10.4).



Мал. 10.4

4. Відредагуйте властивості відображення одного зображення за допомогою вказівок контекстного меню (мал. 10.5).



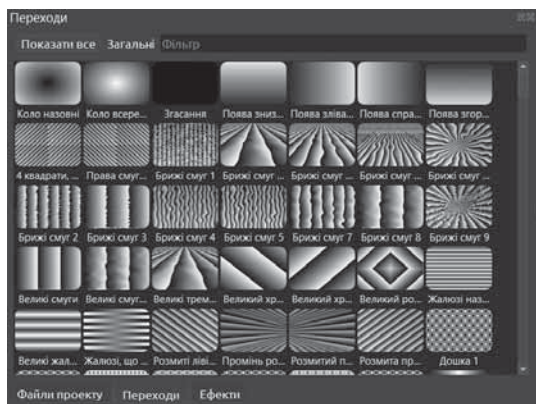
Масштабувати весь кліп з 50 % до 100 %

На екрані змінити розмір, перетягуючи маркери

Мал. 10.5

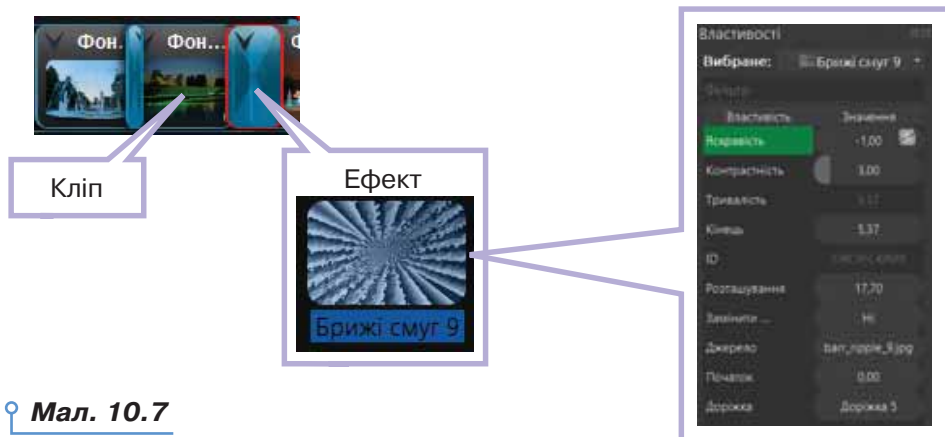
5. Перегляньте змонтоване відео. Внесіть зміни у відображення інших зображень за власним задумом. Поекспериментуйте з іншими вказівками.
 6. Перегляньте остаточне відео. Збережіть проект з іменем *Тернопіль_Прізвище*.
- 4. Як додати відеопереходи між кліпами чи зображеннями й візуальні ефекти?**

Аналогічно до анімаційних ефектів зміни слайдів у комп'ютерних презентаціях, під час створення фільму можна додавати відеопереходи між окремими кліпами. Перелік відеопереходів міститься у вікні *Переходи*, яке відкривають натисненням однойменної кнопки під областю перегляду (мал. 10.6).



Мал. 10.6

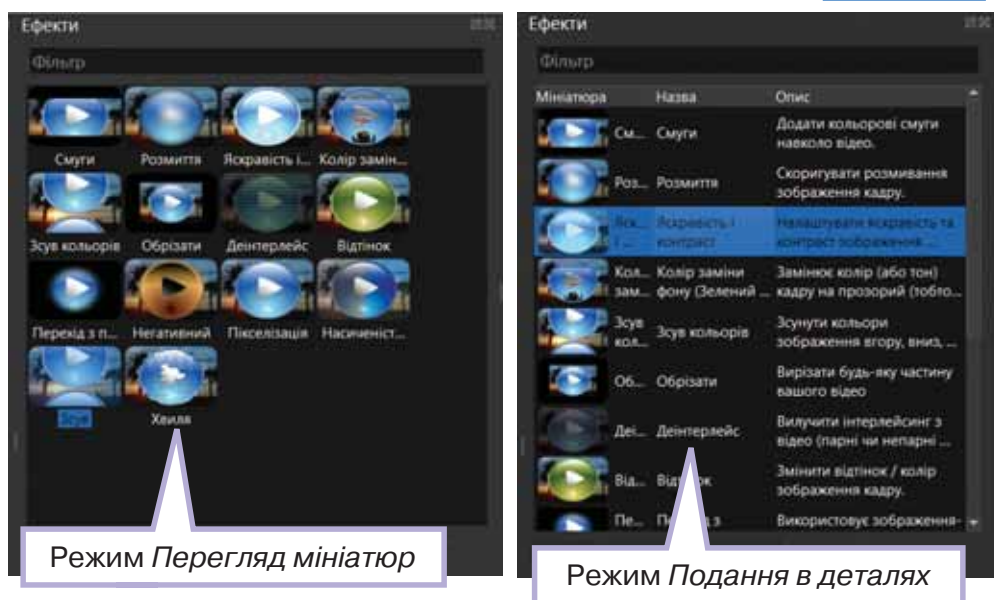
Для додавання відеопереходів до проекту достатньо виділити один з ефектів переходу або панорамування та масштабування і перетягнути його на об'єкти доріжки монтажного столу. Щоб зменшити тривалість дії ефекта, можна перетягнути край, або в контекстному меню обрати *Властивості* та здійснити глибоке налаштування (мал. 10.7).



Мал. 10.7

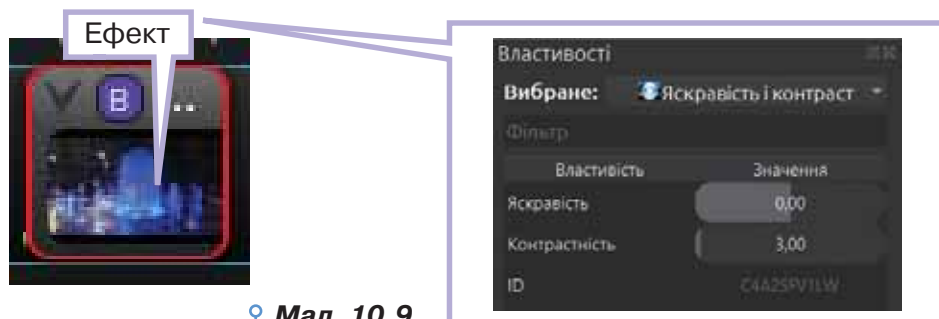
Іноді реалізувати авторський задум допомагають візуальні ефекти, які можна додавати до кліпів. Наприклад, можна застосувати ефекти «старого фільму», ніби записаного на плівці, відтворити кліп у чорно-білому вигляді тощо. Перелік усіх доступних ефектів можна побачити у вікні *Ефекти* (мал. 10.8).

Мал. 10.8



Щоб зрозуміти, який ефект буде накладено на кліп, можна перейти з режиму *Перегляд мініатюр* у режим *Подання в деталях* (вказівка контекстного меню). А тоді приймати рішення щодо застосуван-

ня ефекту. Щоб додати до кліпу відеоефект, слід обрати відповідний ефект і перетягнути його на відеокліп. Дізнатися, що на кліп накладено відеоефекти, можна за позначкою над кліпом (для даного ефекту — позначка В, для іншого — інша). А переглянути властивості ефекту чи змінити його можна у вікні властивостей (мал. 10.9).

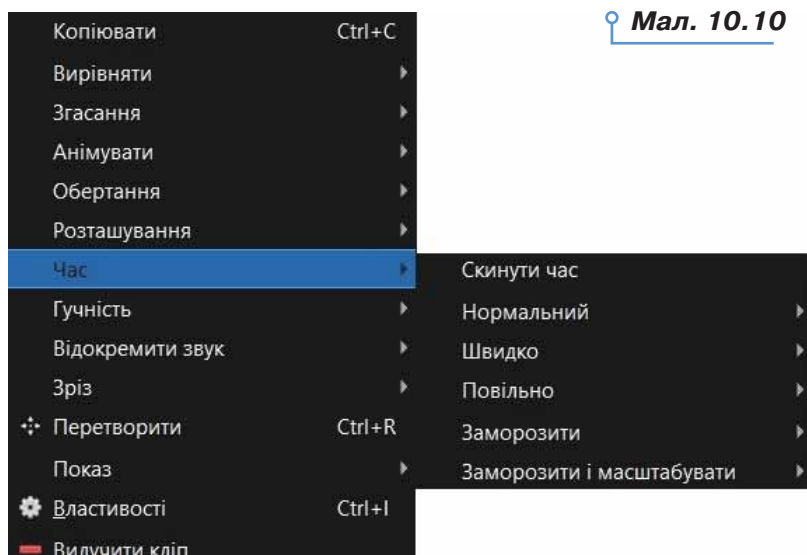


Мал. 10.9

5. Як додати до відеокліпу музичний супровід і налаштувати часові параметри відео й аудіо?

Відеозаписи, записані на відеокамеру чи імпортовані з файлів, можуть містити і звук. Якщо відеозапис не містить звуку або не потрібно відтворювати наявний звук, можна додати до відеокліпу інший музичний супровід чи звуковий коментар.

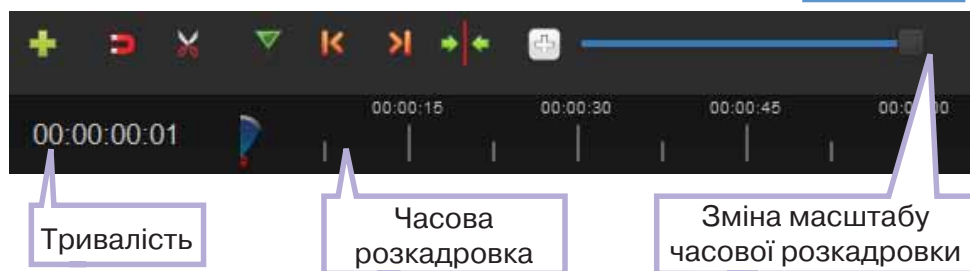
Дуже рідко трапляється ситуація, коли тривалість відео й аудіо збігається, тому виникає необхідність налаштовувати часові параметри. Отримати відеозапис чи аудіокліп потрібної тривалості можна різними способами: приховати початок або кінець певних відео чи аудіокліпів, змінити швидкість відтворення відео, обравши в контекстному меню *Час* та відповідний режим зміни (мал. 10.10).



Мал. 10.10

Для налаштування часу відтворення використовують часову шкалу (мал. 10.11).

Мал. 10.11



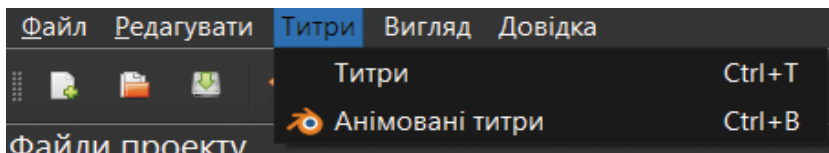
Вправа 5. Додавання до відеокліпу музичного супроводу й налаштування часових параметрів відео й аудіо.

Завдання. До проекту *Ранок*, що збережений у папці *Мультимедіа* власної структури папок, імпортуйте музичний файл *Гріг Ранок.mp3*. Додайте аудіокліп і налаштуйте часові параметри відео й аудіо.

1. Відкрийте вікно відеоредактора *OpenShot* і додайте до проекту відеозапис *Ранок.avi*, збережений у папці *Мультимедіа\Відео*.
2. Додайте до проекту звукозапис *Гріг Ранок.mp3*, що збережений у папці *Мультимедіа\Аудіо*.
3. Розмістіть об'єкти на двох різних доріжках.
4. Щоб збільшити тривалість відтворення відеокліпу *Ранок* у 4 рази (уповільнити його), виділіть цей кліп і в контекстному меню оберіть *Час→Повільно→Вперед→1/4*.
5. Виділіть аудіокліп *Гріг Ранок* у контекстному меню та оберіть *Згасання→Кінець кліпу→Зникнення (швидко)*.
6. Перегляньте створений відеокліп.
7. Збережіть створений проект у папці *Мультимедіа* власної структури папок.

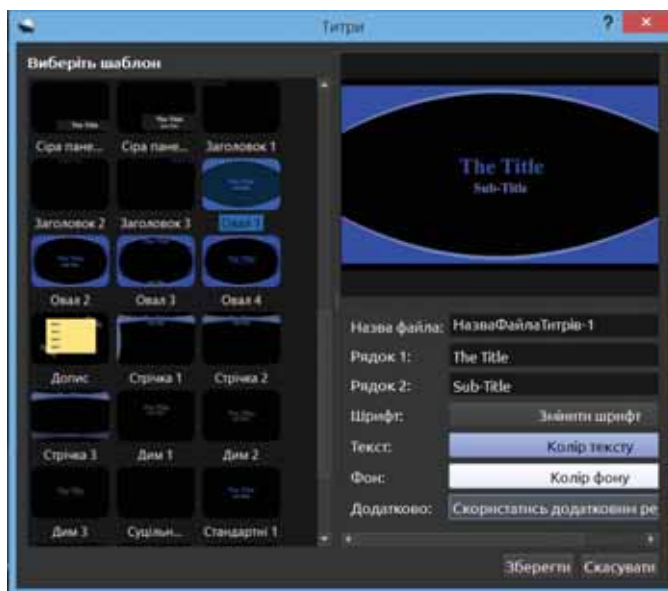
6. Як додати до проєкту написи та зберегти створений відеокліп у форматі відео?

До проєкту можна додати текстові написи — титри. Їхні шаблони викликаються вказівкою меню (мал. 10.12).



Мал. 10.12

У списку шаблонів вибирають потрібний і у вікні властивостей вказують заголовок, підзаголовок, колір тексту і фону, шрифт та інші параметри (мал. 10.13).

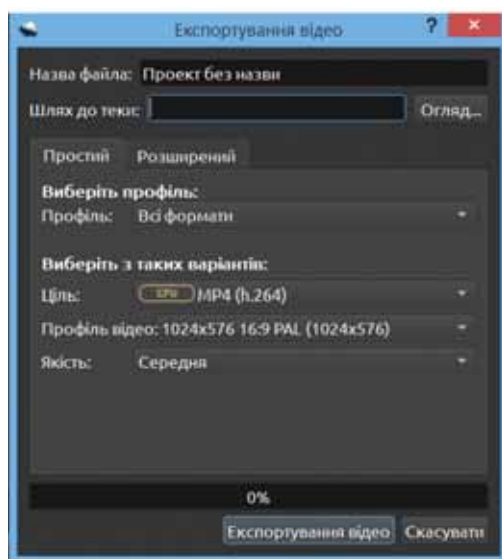


Мал. 10.13

Після завершення роботи над проектом можна зберегти готовий відеокліп як фільм у форматі відео. Для цього слід обрати вказівку



Файл/Експортувати відео або інструмент на панелі інструментів, виконання якої відкриває вікно *Експортування відео*. У вікні властивостей файлу відео вказують назву, місце зберігання, формат файлу та якість (мал. 10.14).



Мал. 10.14



ДІЄМО

Вправа 5. Збереження створеного проєкту відеокліпу у форматі відео.

Завдання. Збережіть проєкт *Фонтани_Прізвище* як фільм.

1. Відкрийте проєкт *Фонтани_Прізвище*, збережений у папці *Мультимедіа* вашої структури папок.
2. У меню *Файл* оберіть вказівку *Експортування відео*.
3. У вікні експортування файлу оберіть папку *Мультимедіа* вашої структури папок, укажіть назву фільму *Музичні фонтани* й натисніть кнопку *Експортування відео*. Зачекайте завершення процесу.



Працюємо в парах

1. Обговоріть, у яких життєвих ситуаціях вам можуть знадобитись навички монтування відео.
2. Розкажіть одне одному, які відеоредактори ви використовуєте, наприклад, на власному смартфоні. Чим вони відрізняються від редактора *OpenShot*?
- ★ 3. Розгляньте вікно програми *Movavi* (мал. 10.15). Наведіть три аргументи на користь того, що програма *Movavi* належить до відеоредакторів. Визначте три ознаки, що підтверджують схожість програм *OpenShot* і *Movavi*. За потреби скористайтесь матеріалами сайту <https://www.movavi.com/video-editor-plus/>.



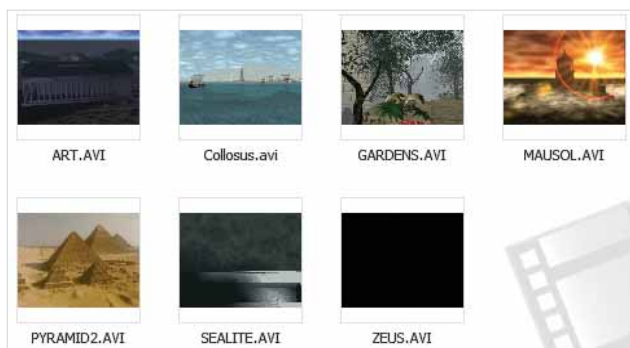
Мал. 10.15

- ★ 4. Обговоріть, які додаткові інструменти для створення проєкту в програмі *OpenShot* пропонують у відео, що розміщене на офіційному сайті. Чи можна за відеорядом зрозуміти настанови майстра/майстрині? Яку англійську лексику потрібно знати для роботи із цим редактором?



Працюємо самостійно

5. Оберіть один зі створених вами проєктів. Відредагуйте його, додавши до кліпів відеопереходи, ефекти. Додайте титри. Збережіть проєкт як фільм у своїй структурі папок під тим само ім'ям.
- ★ 6. Знайдіть в інтернеті відомості про Сім чудес Стародавнього світу. Використовуючи відеозаписи, збережені в папці *Мультимедіа/Сім чудес світу*, створіть відеокліп про Сім чудес світу (мал. 10.16).



Мал. 10.16

Створіть назву фільму та додайте назви до кліпів про кожне із Семи чудес. Додайте відеопереходи між кліпами й відеоефекти.

- ★ 7. Створіть новий проєкт у програмі *OpenShot*. Додайте до проєкту відеозаписи, збережені в папці *Мультимедіа/Ансамбль Вірського*. Створіть відеокліп «Українські танці у виконанні ансамблю Вірського», для цього з кожного танцю виберіть найбільш яскраві логічно завершені фрагменти (використайте поділ кліпу або приховування початку чи/та кінця кліпу). Додайте до кожного фрагмента назву танцю та відеопереходи між фрагментами. Збережіть проєкт з іменем *Танці_Прізвище*.
- ★★ 8. Попрацюйте в мініпроєкті відеогіда «Перлина України», в якому представте відеоекскурсію по куточку України, де ви живете. Передбачте, що до проєкту можна додати авторський голосовий супровід, у фон — мелодію. У відеогід можна вмонтувати відео й зображення, зроблені самостійно або завантажені з інтернет-сторінок місцевих інтернет-сайтів чи соціальних мереж. Додайте до проєкту доцільні відеоефекти та переходи так, щоб привернути увагу тих, хто буде переглядати відео. Збережіть проєкт з іменем *Перлина_Прізвище*. Продемонструйте його у класі.
- ★★ 9. Попрацюйте в мініпроєкті «Старі фотографії», у якому подайте історію свого роду у фото. Використайте відеопереходи, титри й ефекти. Як фон використовуйте фото сімейного помешкання. Додайте звуковий супровід.



Сервіси для роботи з аудіо- й відеоданими та публікування їх в інтернеті

ПРИГАДАЙТЕ:

- ▶ програми для відтворення відео на комп'ютері.

ВИ ДІЗНАЄТЕСЬ:

- ▶ які існують сервіси для розміщення відео- та аудіоматеріалів в інтернеті;
- ▶ як шукати та переглядати відео на *YouTube*;
- ▶ як опублікувати відео на *YouTube*;
- ▶ як можна вносити зміни до відео за допомогою онлайнових сервісів.



Вивчаємо

1. Які існують сервіси для розміщення відео- та аудіоматеріалів в інтернеті?

Сьогодні існують різні сервіси, на яких можна розміщувати, переглядати та прослуховувати відео- та аудіоматеріали. Деякі з них є безкоштовними, інші мають різні тарифні плани. Більшість сервісів для розміщення матеріалів передбачають попередню реєстрацію користувача.

YouTube (<https://www.youtube.com>) — це найбільший у світі відеосервіс в інтернеті, призначений для перегляду та розміщення в глобальній мережі відео, створеного користувачами з усього світу. Переглядати відео на *YouTube* може будь-хто. На порталі є система пошуку, за допомогою якої можна швидко знайти потрібні відеоматеріали.

Відео на сайті *YouTube* користувачі розміщують самостійно, також вони можуть залишати коментарі та оцінки. Правила спільноти й повідомлення на сайті свідчать про те, що користувачі мають володіти авторськими правами або мати дозвіл власників прав на розміщення будь-якого відеоматеріалу. Розміщувати відео, додавати коментарі та оцінки можуть лише зареєстровані користувачі.



Це цікаво

Компанія *YouTube* заснована в лютому 2005 р. Найперше відео на *YouTube* було розміщено 23 квітня 2005 р. У жовтні 2006 р. *Google* придбав *YouTube*, але за угодою за *YouTube* залишилася торгова марка й особливості бренду.

Існують також інші сервіси для розміщення та перегляду відео, зокрема <https://vimeo.com>.

Деякі вебпортали також мають вбудовані сервіси для розміщення аудіо- та відеоматеріалів. Наприклад, на порталі <http://meta.ua> є сервіси для розміщення відео й аудіо — <http://video.meta.ua> та <http://audio.meta.ua>, а також для пошуку та прослуховування радіо онлайн <http://radio.meta.ua>. Прикладами сервісів для розміщення та пошуку аудіозаписів є <http://www.audiomicro.com>, <http://freemusicarchive.org>, <https://soundcloud.com>.

Аудіозаписи, змістом яких є не музика, а радіопередачі, звукові вистави, інтерв'ю, лекції тощо, можуть бути розміщені у вигляді подкастів.

Подкастом називають або окремий аудіофайл, або серію таких файлів, що публікуються за однією адресою в інтернеті та регулярно оновлюються.

Велика кількість сайтів надає можливість прослухати аудіозаписи онлайн. Це можуть бути улюблені пісні, аудіокниги, радіопередачі тощо. У цьому разі на вебсторінці розміщений електронний плеєр з інструментами для початку чи призупинення прослуховування аудіозапису та регулювання гучності.



Це цікаво

Термін подкаст є поєднанням назви портативного програвача музики *iPod* та слова *broadcast* — від англ. трансляція, радіомовлення.

Наприклад, подкаст <https://classical.suspilne.media/list.php> містить збірку класичних мелодій, які можна прослухати просто у браузері (мал. 11.1).



Це цікаво

Одним з перших сайтів, на якому можна завантажувати, передавати, коментувати фото та позначати на них людей, був *Flickr*. Створила його Катерина Фейк.



Мал. 11.1



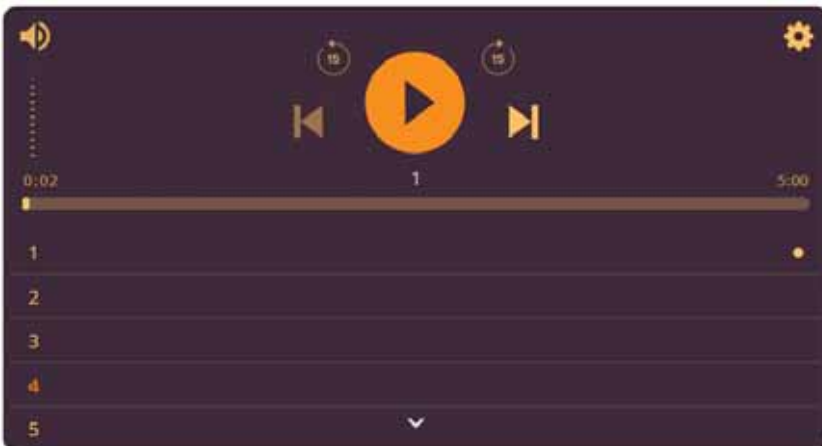
ДІЄМО

Вправа 1. Прослуховування аудіокниги онлайн.

Завдання. Прослухайте фрагмент аудіокниги Євгена Гуцала «Голодомор».

1. Відкрийте вікно браузера й у рядку адреси введіть адресу збірки аудіокнижок <https://4read.org/>.
2. У пошуку за жанрами оберіть **Історична проза**. У списку творів оберіть твір Євгена Гуцала «Голодомор».
3. Визначте тривалість аудіозапису.
4. Визначте, як змінити гучність відтворення, призупинити прослуховування.
5. Дослідіть, як вплине на відтворення аудіозапису перехід на 4 доріжку (мал. 11.2). Прослухайте аудіозапис.

Мал. 11.2



6. Дослідіть, які налаштування заховані в інструменті *Налаштування* .
7. Перегляньте, які твори подано у форматі аудіозапису на сайті улюбленого вами жанру. Які з них ви вже читали чи слухали?

2. Як шукати та переглядати відео на YouTube?

Щоб розпочати роботу з *YouTube*, слід завантажити браузер і в рядку адреси ввести *www.youtube.com*. Головна сторінка сайту, як і пошукових служб, містить рядок введення для ключових слів. У разі його використання пошук відбувається за вказаними ключовими словами серед відеозаписів, розміщених на *YouTube* (мал. 11.3).

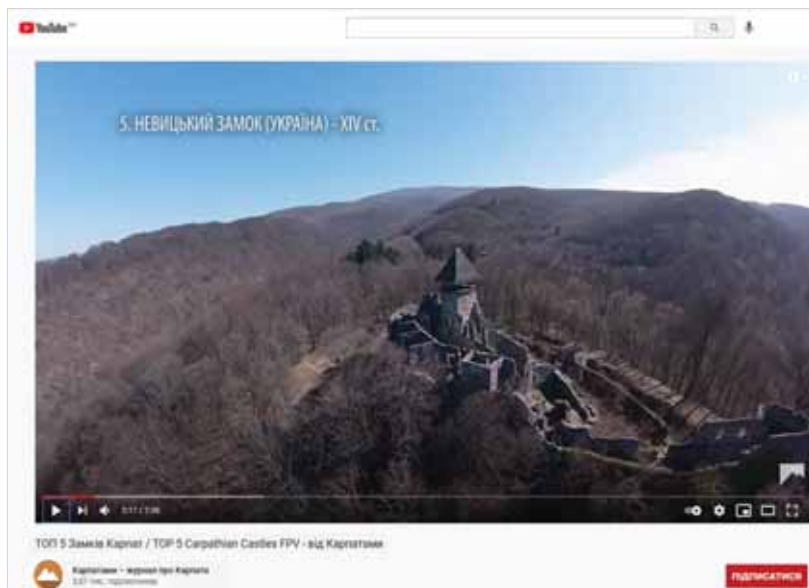
 **Мал. 11.3**



Якщо ви знайшли відео, що вас зацікавило, за його допомогою можна знайти й інші: у правій частині вікна є посилання на інші відео, розміщені цим користувачем, а також схожі відео.

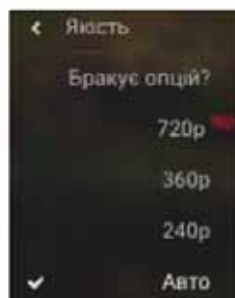
У нижній частині області відтворення відеозапису (мал. 11.4) розміщено елементи керування відтворенням та переглядом: кнопки для початку  або призупинення відтворення , регулювання гучності , переходу до режиму домашнього кінотеатру .

Перейти до перегляду наступного відео обраної тематики можна за допомогою елемента  (мал. 11.4).



 **Мал. 11.4**

Інструмент *Налаштування*  обирають, щоб змінити якість відео залежно від швидкості інтернет-з'єднання (мал. 11.5). Відеозаписи, розміщені на *YouTube*, містять потокове відео, тому переглядати їх можна в режимі реального часу. Якщо швидкість завантаження даних з інтернету недостатня для комфортного перегляду, можна призупинити перегляд за допомогою кнопки *Пауза* та дочекатися, доки буде завантажено хоча б третину чи половину відео.



Мал. 11.5

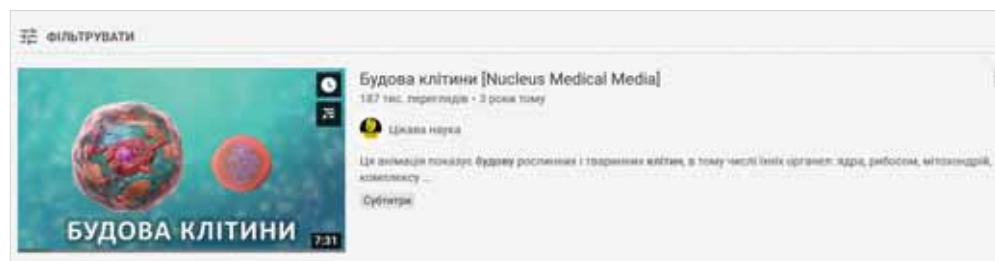


ДІЄМО

Вправа 2. Пошук та перегляд відео на *YouTube*.

Завдання. Знайдіть і перегляньте на *YouTube* відео *Будова клітини* на каналі *Цікава наука*.

1. Відкрийте вікно браузера й у рядку адреси введіть www.youtube.com.
2. На головній сторінці в рядок введення введіть ключові слова *Будова клітини*.
3. Оберіть один зі знайдених результатів пошуку (мал. 11.6).



Мал. 11.6

4. Перегляньте відеозапис. За потреби призупиніть відтворення та дочекайтеся часткового завантаження відео.
5. Серед схожих відео знайдіть інші відеозаписи про будову клітини. Перегляньте знайдені записи. Порівняйте, що ви знаєте про клітину із 6 класу, і що дізнались із відео.

3. Як опублікувати відео на *YouTube*?

Шукати й переглядати відео на *YouTube* можна навіть без реєстрації. Проте для завантаження відео необхідно скористатися власним обліковим записом *Google*. Щоб отримати можливість завантаження відео, слід увійти у свій обліковий запис та обрати вказівку *Завантажити* .

Перш ніж завантажувати відеозаписи, слід створити канал користувача. Це можна зробити у *Творчій студії* або під час першого використання вказівки *Завантажити*: буде запропоновано створити канал користувача, на якому зберігатимуться завантажені відеозаписи.

Відео, збережене у файлі, можна просто перетягнути мишею у вказану область.

Перш ніж опублікувати завантажене відео, необхідно вказати його назву, опис, ключові слова та інші налаштування для відео, за якими інші користувачі зможуть його знайти (мал. 11.7).

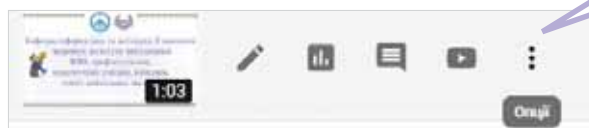
Мал. 11.7

Переглянути список завантажених відео на каналі та внести до них зміни можна за допомогою списку відео в *Меню каналу* (мал. 11.8).

	відео	видимість
 Інформаційна пан...	☐	
 Відео	☐  7:42	 Не для всіх
 Списки відтворен...	☐  11:12	 Не для всіх
 Аналітика	☐	
 Коментарі	☐ 	 Для всіх

Мал. 11.8

Кожне завантажене відео можна редагувати або видалити (мал. 11.9).



Мал. 11.9

4. Як можна вносити зміни до відео за допомогою онлайн-сервісів?

Крім відеоредакторів у вигляді програм, що встановлюються на комп'ютер, існують також онлайн-відеоредактори, наприклад, *Magisto* (<https://www.magisto.com/>). У редакторі *Magisto* після реєстрації користувач може обрати формат використання сервісу (переважно *Особистий*) та за три прості кроки: додавання медіафайлів у проект, додавання тексту та накладання музики — створити своє відео онлайн. Перехід від одного кроку до іншого виконують натисненням на кнопку *Далі*. На основі обраних шаблонів до відео будуть додані візуальні ефекти та переходи.

На *YouTube* є також власний відеоредактор. Перейти до редактора відео на *YouTube* можна зі сторінки каналу за допомогою вказівки



У вікні редактора можна виконати дії з редагування файла, наприклад, додати музику, виконати розмиття фрагмента, додати кінцеву заставку (мал. 11.10).

Мал. 11.10



Досліджуємо

Вправа 3. Фонотека *YouTube*.

Завдання. Дослідіть фонотеку, яка доступна на вашому каналі *YouTube* (мал. 11.11). Знайдіть відповіді на питання:

- чи можна використовувати пропоновану музику без дозволу її авторів;

- які властивості мають музичні фрагменти з фонотеки;
 - як завантажити музику з фонотеки на власний комп'ютер;
 - у якому форматі збережено музичні фрагменти та звуки?
- Запропонуйте, як можна використати музичні фрагменти та звукові ефекти з фонотеки в навчальній діяльності чи ваших проектах.

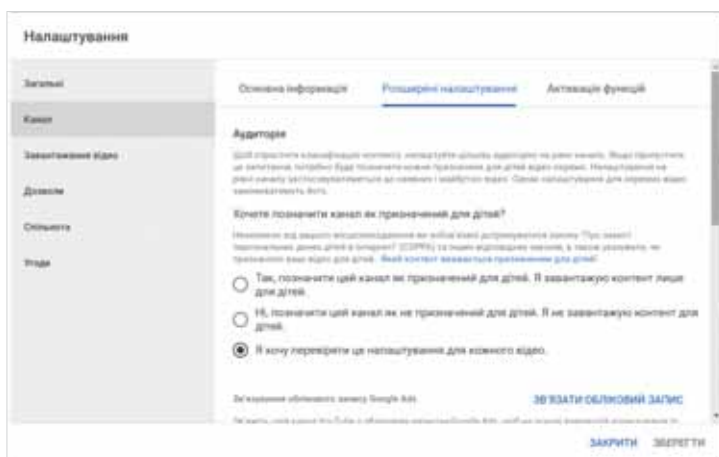


Мал. 11.11

Працюємо в парах



1. Обговоріть матеріали статті <https://podcaster.in.ua/chomu-podcasting/>. Виділіть п'ять найважливіших порад початківцю в користуванні подкастами.
2. Обговоріть переваги та недоліки редагування відеокліпу за допомогою онлайнного відеоредактора в порівнянні з відеоредактором, що встановлений на комп'ютері.
- ★ 3. Ознайомтесь із додатковими налаштуваннями, які можна змінювати для свого каналу YouTube (мал. 11.12). Обговоріть у парах, у яких випадках доцільно змінювати такі налаштування.



Мал. 11.12



Працюємо самостійно

4. Створіть список з посиланнями:

- п'ять найкращих аудіокниг, які ви радите прослухати своїм друзям;
- три подкасти для тих, з ким ви «на одній хвилі».

Розмістіть списки на спільну дошку класу.

5. Розмістіть в інтернеті на своєму відеоканалі одне з відео, створених на заняттях з інформатики. Додайте опис та всі необхідні властивості до відео. Позначте властивість *Не для всіх*. Перешліть посилання на відео на навчальну електронну пошту.

6. Створіть відео про добру справу, яку ви зробили сьогодні. Використайте відеоредактор на власний вибір. Розмістіть відео на своєму каналі. Опишіть відео так, щоб започаткувати челендж із добрих справ в обраній сфері (наприклад, *#добро&люди_з_інвалідністю*, *#добро&тварини*, *#добро&природа* тощо). Поділіться відео з іншими, зробіть його доступним для широкої аудиторії.

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Розумію і можу пояснити принципи оцифровування звуку та відеоряду.
- ☐ Розрізняю формати аудіо- й відеофайлів.
- ☐ Можу навести приклади програмного забезпечення для опрацювання об'єктів мультимедіа та пояснити його призначення.
- ☐ Умію перетворювати формати аудіо- й відеофайлів.
- ☐ Умію використовувати програми для роботи з мультимедіа.
- ☐ Умію побудувати відеоряд.
- ☐ Умію використовувати програми для роботи з мультимедіа.

Розділ 4

Алгоритми та програми

12



Мова та середовище програмування

ПРИГАДАЙТЕ:

- ▶ способи подання алгоритму;
- ▶ складові мови програмування;
- ▶ складові та призначення елементів вікна навчального середовища програмування.

ВИ ДІЗНАЄТЕСЬ:

- ▶ якими бувають мови програмування;
- ▶ які засоби необхідні для виконання створених програм;
- ▶ які середовища програмування використовують для створення програм.



Вивчаємо

1. Якими бувають мови програмування?

Ви вже знаєте, що алгоритми, які орієнтовані для виконання за допомогою комп'ютера, використовують спеціальну мову програмування. Алгоритм, записаний мовою програмування, називають **програмою**.

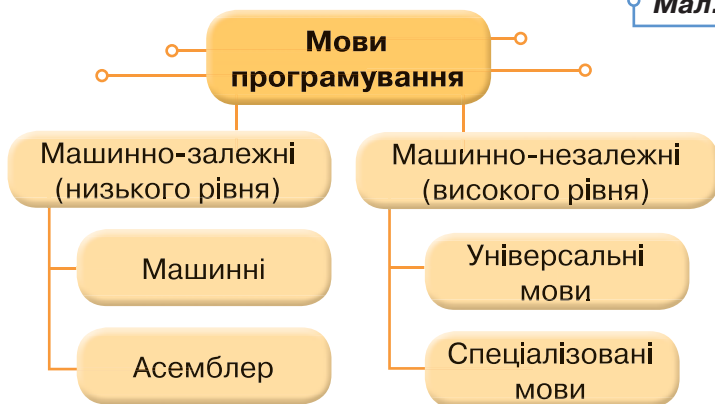
Мова програмування — це система позначень для точного опису алгоритму, який потрібно виконати за допомогою комп'ютера.

Спочатку команди для комп'ютера писали за допомогою звичайних двійкових кодів, і програма виглядала дуже громіздкою, її написання займало багато часу. Із часом такі двійкові коди замінювались певними позначеннями, більш зрозумілими для людини. При цьому створювалися правила, а з часом і програми, які автоматично здійснювали переклад так записаних програм для комп'ютера на мову машинних

кодів. На сьогодні існує понад 3000 різноманітних мов програмування: одні з них уже не використовуються, а інші, навпаки, ще й удосконалюються та дають поштовх розвитку нових мов і їхніх версій.

Мови програмування можна поділити на дві групи: мови **низького рівня** і мови **високого рівня** (мал. 12.1).

Мал. 12.1



Машинна мова — набір команд, які виконуються безпосередньо центральним процесором (мал. 12.2). Працювати людині-програмісту з такою мовою важко через велику кількість команд у двійковій формі. Машинні коди були основним засобом програмування від моменту створення комп'ютерів.

Мал. 12.2

Машинний код	Асемблер
0005 B4 09	7 mov AH, 09h
0007 BA0000r	8 mov DX, offset msg
00A CD 21	9 int 21 h

До мов низького рівня належить мова **асемблера** (від англ. *assemble* — складати, komponувати). У мові асемблера використовуються символічні позначення команд, які легше зрозуміти й запам'ятати.

Програма, записана мовою програмування високого рівня, містить команди, що схожі на звичайні слова, записані англійською мовою. Наприклад, програма, записана мовою програмування *Python*, містить команди *print* — друкувати, *while* — поки, *if* — якщо тощо.

Програми, описані мовами високого рівня, дають змогу формулювати завдання для виконання на комп'ютері у звичному та зрозумілому для людини вигляді, а самі мови зробили використання комп'ютерів доступним широкому колу людей, які не є фахівцями в галузі програмування.

Розрізняють універсальні та спеціалізовані мови програмування високого рівня. Універсальні мови використовують для розв'язування різних завдань. До них відносять *C++*, *C#*, *Pascal*, *Delphi*, *Java*, *C*, *Objective-C*, *Basic*, *Swift*, *Python*, *Cobol*, *D*, *Ada*. Спеціалізовані мови призначені для вирішення завдань одного, максимум кількох видів завдань. Наприклад, роботи з базами даних, вебпрограмування або написання скриптів для адміністрування операційних систем. Прикладами спеціалізованих мов програмування є *Visual FoxPro*, *PHP*, *Perl*, *VBScript*, *JavaScript*, *VBA в Microsoft Office* та інші.

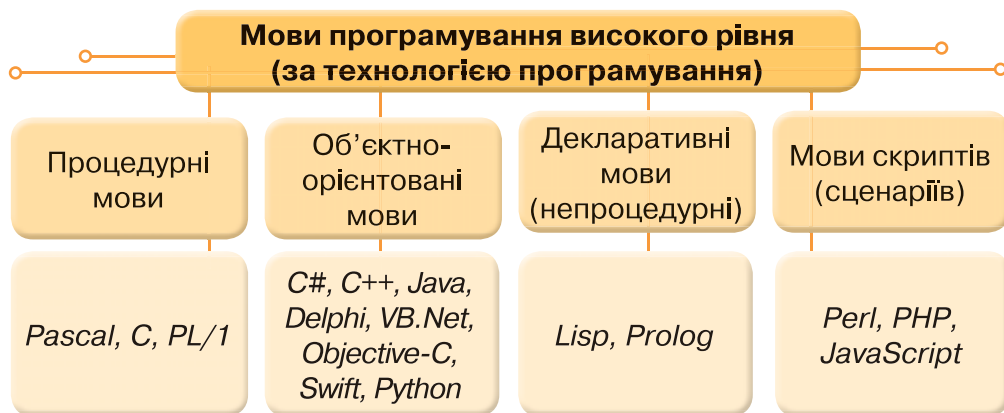


Це цікаво

Скриптова мова (англ. *scripting language*) — мова програмування, розроблена для запису «сценаріїв», послідовностей операцій, які користувач може виконувати на комп'ютері. У навчальному середовищі створення та виконання алгоритмів *Скретч* програми створюють в області, яка має назву *Скрипти*, бо вони власне реалізують сценарій подій, що мають відбутися з об'єктом-виконавцем.

За технологією програмування розрізняють (мал. 12.3):

- **процедурні** мови — програма складається з набору підзадач (процедур), які реалізують завдання;
- **об'єктно-орієнтовані** — головними елементами алгоритму є **клас** — новий тип даних, що розширює мову, та **об'єкт**, що має властивості й методи — дії, які виконуються об'єктом;
- **декларативні** — мови створення програм зі штучним інтелектом;
- **мова скриптів** (мова сценаріїв) — мова, розроблена для запису «сценаріїв», послідовностей команд, які користувач може виконувати на комп'ютері.



Мал. 12.3



Вправа 1. Рейтинг мов програмування.

Завдання. Визначте трійку найкращих мов програмування, які б ви, за порадами спеціалістів, визначили для своєї самоосвіти.

1. Відкрийте статтю про рейтинг мов програмування за версією сайту <https://dou.ua/>.
2. Проранжуйте критерії за ступенем вагомості для вас особисто:
 - комерційне використання;
 - мова написання програми;
 - мова програмування особистого вподобання;
 - мова написання робочих проектів.
3. Визначте, яку мову програмування ви вважаєте найбільш перспективною для вивчення.
4. Напишіть умотивований висновок та надішліть його електронним листом на навчальну скриньку. Яке місце серед таких мов займає мова програмування *Python*?



2. Які засоби необхідні для виконання створених програм?

Для перетворення команд мови програмування, яка відрізняється від машинної мови, використовують спеціальні програми — **транслятори**.

Транслятор (від англ. *translation* — переклад) — програма, яка записує команди програми, описаної деякою мовою програмування, на машинну мову.

Розрізняють два способи трансляції: компіляція та інтерпретація.

Компілятор зчитує одразу всю програму й переписує її машинним кодом або мовою асемблера. Процес трансляції, який за такого підходу називається **компіляцією**, відбувається один раз, і результат перекладу зберігається в окремому файлі. Якщо код програми змінюється, її необхідно буде перекомпілювати. Скомпільована програма прив'язується до операційної системи й набору команд процесора, тому не завжди може бути перенесена та виконана на іншому комп'ютері. З іншого боку, вона є «готовою до використання» і може бути швидко виконана на тому самому або аналогічному комп'ютері: з точки зору користувача — просто клацнути на ім'я виконуваного файлу й запустити на виконання, з точки зору комп'ютера — просто «прочитати» та виконати набір команд.

Інтерпретатор зчитує вихідний код програми по одній команді й одразу намагається їх «перекладати» та виконувати. Це дає змогу

програмісту швидше перевіряти правильність виконання програми та знаходити помилки в коді. Виконання програми за застосування інтерпретатора займає трохи більше часу, оскільки кожного разу під час запуску програми на виконання відбувається аналіз коду та його перетворення із самого початку програми.

Тому для підвищення швидкодії більшість сучасних інтерпретаторів насправді працює за змішаною схемою, спочатку транслюючи вихідний код програми в деяку проміжну форму — так званий байт-код. Це дає змогу, за відсутності змін в оригінальній програмі, не перечислювати її повністю, а використовувати байт-код як «напівфабрикат» для роботи. Виконання байт-коду все одно повільніше від виконання машинного коду, але такий підхід є компромісом, що намагається поєднати переваги інтерпретації та компіляції. Такий спосіб інтерпретації використовується в мові програмування *Python*. Підхід із проміжною компіляцією в байт-код використовують також *Java* та *C#*.



Це цікаво

Розробку мови *Python* (вимовляють «пайтон») було розпочато наприкінці 1980-х років співробітником голландського інституту CWI Гвідо ван Россумом.

Першовідкривачкою в галузі оптимізації компіляторів та першою жінкою, яка отримала у 2006 р. найпрестижнішу в галузі інформатики премію Тюрінга, є Френсіс Аллен.

Для створення програм, редагування, пошуку помилок та їх виправлення, а також виконання програм, написаних мовою програмування, використовують середовище програмування.

Середовище програмування — це комплекс програм, що має засоби автоматизації процесів підготовки та виконання програм користувача, а саме:

- редактор текстів програм — у ньому можна створювати та редагувати текст програми;
- довідково-інформаційну систему про мову програмування та середовище;
- бібліотеки, у яких зберігаються найбільш уживані фрагменти програм або цілі програми;
- компілятор чи інтерпретатор, використання якого дає змогу швидко знайти помилку в програмі та виправити її;
- покроковий «виконавець» програми.

3. Які середовища програмування використовують для створення програм?

Розрізняють різні види середовищ програмування (мал. 12.4).

Середовища програмування



Мал. 12.4

Для тієї само мови програмування може існувати декілька середовищ програмування, що підтримують різні технології програмування. Вони можуть бути встановлені на комп'ютер або реалізовані у вигляді онлайнового сервісу.

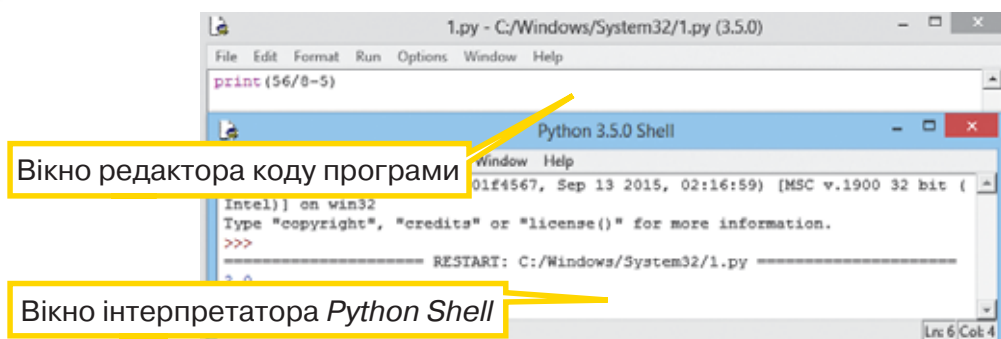
Ви вже використовували навчальне середовище *Thonny* для складання та виконання програм мовою *Python*.

Під час встановлення на комп'ютер мови програмування *Python* одночасно також встановлюється інтегроване середовище програмуван-



ня *IDLE*, яке можна відкрити з *Головного меню*. Особливістю цього середовища є те, що під час роботи з ним використовується два вікна, в одному з яких записується та редагується текст програми, а в іншому відображається результат її виконання. Після запуску програми *IDLE* за замовчуванням відкривається вікно інтерактивного середовища інтерпретатора *Python Shell*, у якому відображається результат виконання програми та повідомлення про помилки.

Якщо в цьому вікні в меню *Файл (File)* обрати вказівку *Новий файл (New File)*, то відкриється вікно редактора коду програми, в якому можна створювати, зберігати, переглядати, редагувати, запускати програму, написану мовою програмування *Python*. Наприклад, щоб відобразити результат обчислення значення виразу $56:8-5$, достатньо у вікні редактора коду програми використати команду `print (`, де в дужках записати вираз із використанням знаків арифметичних операцій: `+` (додавання), `-` (віднімання), `*` (множення) та `/` (ділення). Результат виконання такого фрагмента програми після його збереження та запуску за допомогою вказівки *Run* буде відображено у вікні *Python Shell* (мал. 12.5).

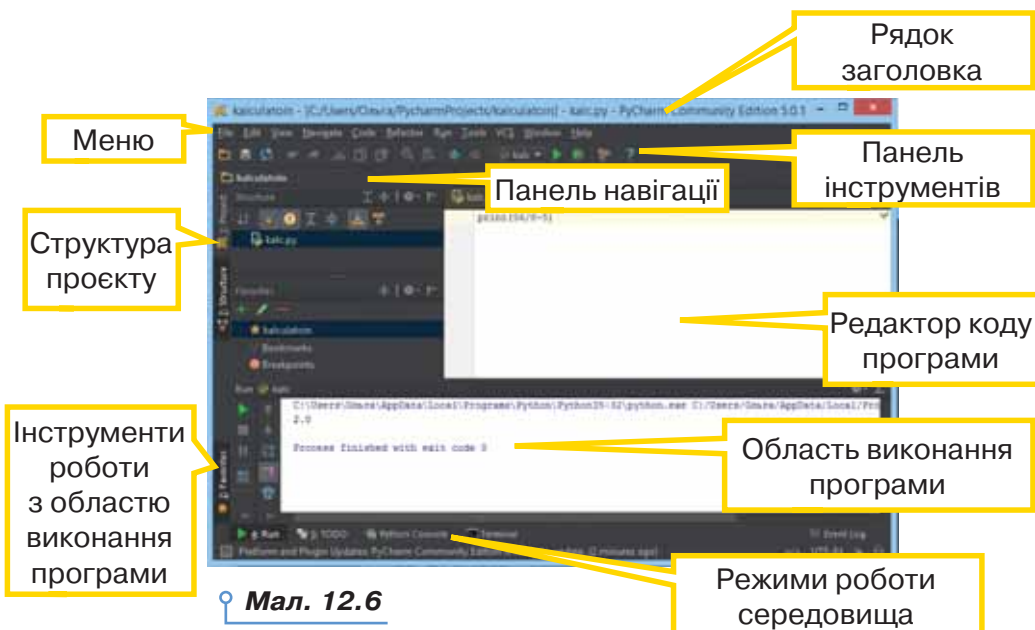


Мал. 12.5

Для створення та виконання програм, написаних мовою програмування *Python*, можна ще використовувати онлайнове середовище програмування.

Зручним для використання інтегрованим середовищем програмування, що підтримує сучасні складові мови *Python*, є *PyCharm*. Це середовище поєднує редактор для зручного введення і редагування програми, транслятор і налагоджувач помилок. Користувач має змогу використовувати статичний аналіз коду, підсвічування синтаксису та помилок, навігацію серед проектів і коду програми, відображення файлової структури проекту, швидкий перехід між файлами, класами, методами та багато іншого.

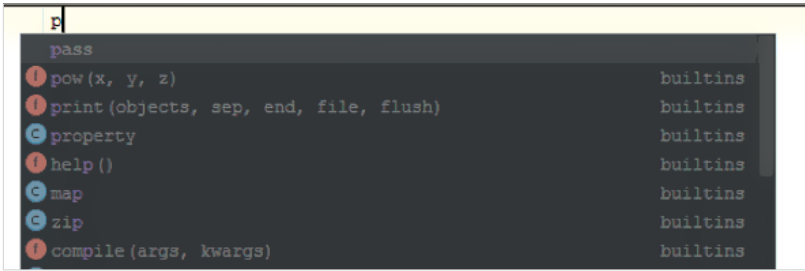
Вікно середовища складається з рядка заголовка, меню, панелі інструментів, панелі навігації, структури проекту, редактора коду програми, області виконання програми, інструментів роботи з областю виконання програми, режимів роботи середовища (мал. 12.6).



Мал. 12.6

Інтегроване середовище програмування *PyCharm* містить набір засобів допомоги та підтримки користувача. Після завантаження середовища з'являється інформаційне вікно, у якому можна дізнатись про корисні послуги при роботі з програмою та комбінації клавіш, якими вони викликаються. Окрім цього, під час введення команди в області введення програми одразу після введення з клавіатури першої літери команди стає доступним список службових команд із підказками про їхній синтаксис (мал. 12.7).

Мал. 12.7



Це цікаво

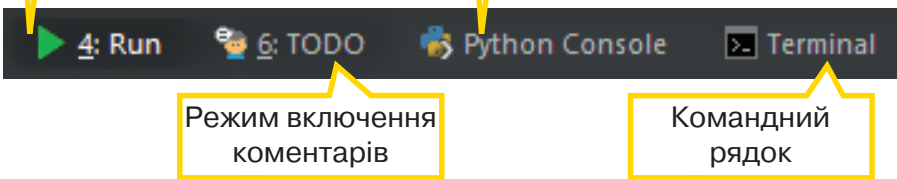
Першою у світі програмісткою вважають Аду Лавлейс, яка заклала наукові основи програмування на обчислювальних машинах за сторіччя до того, як почала розвиватися ця наукова дисципліна.

Область виконання програми підтримує декілька режимів, які можна змінювати за допомогою відповідних інструментів: звичайне виконання, режим включення коментарів, консольний режим, який нагадує середовище *IDLE*, командний рядок (мал. 12.8).

Звичайне виконання

Консольний режим

Мал. 12.8



Для спрощення розробки графічного інтерфейсу користувача використовують середовища для візуального програмування, які дають змогу конструювати програми шляхом оперування графічними об'єктами. Для мови *Python* розроблені спеціальні бібліотеки, використання яких дає змогу візуалізувати процес розробки проекту. Візуальні компоненти графічного інтерфейсу проекту можна також створювати вручну, використовуючи в програмі відповідні команди.



Досліджуємо

Вправа 2. Онлайнові середовища.

Завдання. Дослідіть онлайнові середовища, які підтримують мову програмування *Python*. Визначте їхні переваги та недоліки. Заповніть таблицю порівняння. Зробіть висновок.

1. Завантажте файл *Онлайнові середовища*, який збережено в папці *Програмування* на спільному навчальному диску.
2. Відкривайте середовища за запропонованим списком. Проводьте дослідження за запропонованими критеріями.



Працюємо в парах

1. Знайдіть в інтернеті відомості про історію виникнення різних мов програмування. Обговоріть і сплануйте, як можна було б подати знайдені відомості на лінії часу та в якій програмі зручно було б створити таку лінію часу. Створіть її.
- ★ 2. Обговоріть інфографіку «Застосування мов програмування», яка збережена у файлі *Застосування_МП* папки *Програмування*. Знайдіть відомості про мову програмування, яка використовується в більшості застосунків.
- ★★ 3. Обговоріть, у яких випадках ви порадили б своїм однокласникам й однокласницям для вивчення мови *Python* використовувати середовище програмування *IDLE*, онлайнове середовище, інтегроване середовище програмування *PyCharm*. Оберіть хоча б три аргументи для підтвердження своїх порад.



Працюємо самостійно

4. Відкрийте в середовищі програмування програму *Guess.py*, що збережена в папці *Програмування*. Запустіть її на виконання. Введіть дані, що відповідають запитам програми. Визначте, яке завдання реалізовано в цій програмі.
- ★★ 5. Попрацюйте в мініпроекті «Середовище програмування». Дослідіть одне із середовищ програмування, яке підтримує мову *Python*. Складіть інструкцію з використання цього середовища програмування. Виділіть в інструкції розділи: введення коду, робота з файлом програми та отримання результатів виконання програми. Збережіть інструкцію з іменем *Інструкція_Прізвище* у своїй структурі папок. Надішліть інструкцію у спільне сховище класу. Перегляньте інструкції інших учнів. Прокоментуйте їх. Ознайомтесь із коментарем до своєї інструкції. Внесіть зміни до інструкції за доцільними порадами.



Структура програми мовою *Python*

ПРИГАДАЙТЕ:

- ▶ поняття величини, види та типи величин;
- ▶ команду присвоювання, введення та виведення даних мовою програмування *Python*.

ВИ ДІЗНАЄТЕСЬ:

- ▶ що описують на початку програми;
- ▶ що може містити тіло програми;
- ▶ з якими функціями працюють у програмі мовою програмування *Python*.



Вивчаємо

1. Що описують на початку програми?

Програма, записана мовою програмування, має свою структуру. Різні мови програмування висувають свої вимоги до такої структури. Але є і загальні. Перше, що зазвичай наявне в коді програми, — це оголошення додаткових модулів — бібліотек, команди яких будуть використані у програмі. Ми вже використовували модулі `math`, `random`, `turtle`. Пригадайте, що підключення модуля здійснюється так:

```
import назва_модуля
```

Щоб використати команду модуля, наприклад, квадратного кореня із числа x , використовують конструкцію:

```
math.sqrt(x)
```

Якщо у програмі буде використана тільки одна конкретна функція, нехай той же $\sqrt{}$, тоді її оголошення здійснюють так:

```
from math import sqrt
```

А якщо буде використано більше функцій, то:

```
from math import *
```

Тоді в програмі виклик функції відбувається без оголошення назви модуля.

Другою складовою початку програми є оголошення змінних. *Python* є досить гнучким і дає змогу створювати змінні будь-де в коді. Але гарний стиль оформлення програм передбачає, що змінні, за можливості, оголошуються в одному місці — це, як мінімум, зручніше для програміста, який одразу бачить, які змінні є і яких вони типів. Тут же змінним задаються початкові значення. Якщо конкретні значення

не відомі або не потрібні на початку роботи, можна присвоїти спеціальне значення `None` — «нічого». Тобто не нуль, не одиниця, не порожній рядок, а взагалі відсутність значення.

Усі змінні бажано називати змістовними іменами, щоб, побачивши їх далі в коді, можна було одразу зрозуміти їхнє призначення. За домовленістю змінні іменуються малими літерами (зверніть увагу: для мови *Python* регістр важливий, тобто з точки зору інтерпретатора, наприклад, *a* і *A* — різні змінні). Якщо одного слова для змістовної назви недостатньо, слова в імені змінної розділюються підкресленням. Отже, *string_length*, *variables_number*, *counter* і навіть *my_den_narodzhennia* — хороші назви для змінних. *FirstMonth*, *counter-of-loops* — погані. *My_variable*, *asd1* — дуже погані.

Вибрати змінній ім'я, яке повністю відображає її суть, можливо не завжди. Для цього використовують коментарі.

Коментар — це будь-який текст для пояснення роботи програми, який не виконується інтерпретатором.

Коментарі можуть бути короткими: відокремлюються від основного коду знаком `#` і все, що міститься в коді програми після `#` і до кінця рядка, інтерпретатором ігнорується. Або довгими, в декілька рядків: перед початком багаторядкового коментаря ставиться трое подвійних лапок, і ще трое в кінці — все, що міститься між ними, інтерпретатор також не виконає (мал. 13.1).

```
number = 10 # кількість змінних
""" Це початок програми. Фрагмент цього тексту після
запуску програми на виконання буде проігноровано """
number = number + 1
```

 **Мал. 13.1**

Крім тлумачення змінних, коментарі можуть застосовуватися будь-де в коді програми для пояснення, що там відбувається. Часто буває, що, повертаючись через деякий час (тижня чи двох достатньо), автор уже не пам'ятає, для чого виконувалася та чи інша дія. А якщо код програми мають читати сторонні люди (наприклад, програма розробляється кількома програмістами), коментарі тим паче не будуть зайвими.

2. Що може містити тіло програми?

Змінні опрацьовують в тілі програми. Ми вже опрацьовували три типи даних:

- **цілі числа** (тип `int`) — додатні та від'ємні числа, а також 0 (наприклад, 4, 687, -45, 0);

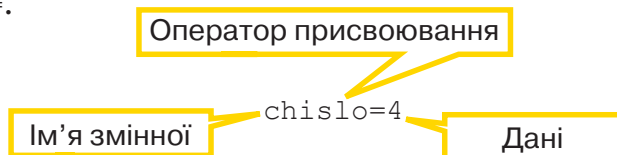
- **числа з плаваючою крапкою** (тип `float`) — дробові, або дійсні, числа (наприклад, 1.45, -3.789654, 0.00453). Нагадаємо: для розділення цілої і дробової частин використовують крапку, а не кому, як у математиці;

- **рядкові** (тип `str`) — набір символів, обмежених лапками (наприклад, "ball", "Як тебе звати?", 'dkfjUUv', '2020'). Нагадаємо: лапки в *Python* можуть бути одинарними чи подвійними; один символ, взятий у лапки, — це теж рядковий тип.

Над даними виконують **операції**. Тоді ці дані називають **операндами**. Саму дію називають **оператором**.



Для надавання значення змінним використовують **оператор присвоювання**. Нагадаємо, що в мові програмування *Python*, як і в більшості інших мов програмування, оператор присвоювання позначається знаком `=`.



Діємо

Вправа 1. Типи величин.

Завдання. У вправі за посиланням <https://cutt.ly/0k1BYtk> згрупуйте дані за типами.

Вправа 2. Оператори.

Завдання. У вправі за посиланням <https://cutt.ly/Vk1BPZ1> з'єднайте назву операції та відповідний запис мовою програмування *Python*.

3. З якими функціями працюють у програмі мовою програмування *Python*?

У тілі програми, записаній мовою програмування *Python*, використовують **вбудовані функції** та **функції користувача**.

Функція — це фрагмент програми, який може використовуватись багато разів.



Ми вже використовували функції для роботи з даними різних типів.

Таблиця 13.1

№	Призначення функції	Опис	Приклад
1	Виведення даних	<code>print()</code>	<pre>print('Моя перша програма') >>> %Run 'Зразок1.py'</pre> Моя перша програма
2	Введення даних	<code>input()</code>	<pre>a=input('a=') print(a+a) >>> %Run 'Зразок2.py'</pre> a=5 55
3	Перетворення тексту в ціле число	<code>int()</code>	<pre>a=int(input('a=')) print(a+a) >>> %Run 'Зразок3.py'</pre> a=5 10
4	Перетворення тексту в дійсне число	<code>float()</code>	<pre>a=float(input('a=')) print(a+a) >>> %Run 'Зразок4.py'</pre> a=5 10.0
5	Визначення кількості елементів	<code>len()</code>	<pre>a=input('Введіть слово') print(len(a)) >>> %Run 'Зразок5.py'</pre> Введіть словоUkraine 7
6	Округлення дійсного числа	<code>round()</code>	<pre>a=float(input('a=')) print(round(a,1)) >>> %Run 'Зразок6.py'</pre> a=5.55 5.5

Для опису власної функції використовують службове слово `def`, а команда `return` указує, яке значення потрібно повернути після виклику функції.

Наприклад, після запуску програми на виконання для значень змінних $c=5$, $b=6$ отримаємо $c=11$ (мал. 13.2).

Аргументи функції

Мал. 13.2

```
1 def add(x, y):
2     return x + y
3 a=int(input('a='))
4 b=int(input('b='))
5 c=add(a, b)
6 print('c=',c)
```

Опис функції додавання значень двох змінних

Виклик функції для конкретних значень аргументів

Опис функції подають у заголовку програми. А виклик функції здійснюють у тілі програми. Погодьтеся, що рядки 1, 2 програми на малюнку — це заголовок програми, а рядки 3–6 — тіло програми.

В описі функції після імені функції в дужках записують аргументи. Функція може мати довільну кількість аргументів, або не мати їх зовсім. Деякі аргументи можуть бути обов'язковими, а інші — необов'язковими. Наприклад, у функції

```
def func(a,b,c=2):
    return a+b+c
```

є три аргументи: a , b , c — необов'язковий аргумент. Його значення за замовчуванням дорівнює 2.



Досліджуємо

Вправа 3. Сума трьох.

Завдання. Дослідіть, як впливає надання значення аргументам функції на результат у програмі за зразком (мал. 13.3).

```
1 def func(a,b,c=2): #c - необов'язковий аргумент
2     return a+b+c
3 rez1=func(3,5) #a=3, b=5, c=2
4 rez2=func(3,5,8) #a=3, b=5, c=8
5 rez3=func(a=5, b=8) #a=5, b=8, c=2
6 rez4=func(a=5, c=8) #a=5, c=8
7 print('rez1=', rez1)
8 print('rez2=', rez2)
9 print('rez3=', rez3)
10 print('rez4=', rez4)
```

Мал. 13.3

1. Відкрийте середовище програмування. Створіть новий файл мовою *Python*.
2. Введіть програму, яку подано на малюнку 13.3.
3. Запустіть програму на виконання. Розгляньте область виконання програми. Зробіть висновки про правильність команди в рядку 6. Введіть потрібні зміни, щоб змінна *rez4* набула значення 20.



Працюємо в парах

1. Обговоріть, чому важливо подавати програму в структурованому вигляді. Одне одному наведіть свої аргументи.
2. Які типи величин опрацьовують у програмі мовою *Python*? Називайте одне одному значення, інший або інша має назвати відповідний тип.
3. Які функції використовують у програмах мовою *Python*? Чи може аргументом функції бути інша функція? Наведіть одне одному приклади.
- ★ 4. Наведіть одне одному свої міркування щодо доцільності створення власних функцій для обчислення значень виразів:

$$1) y = \frac{\sqrt{|a+b|}}{\sqrt{|a-b|}} \cdot \sqrt{|ab|} ; \quad 2) y = a^2 + b^2 - 2(ab)^2 ; \quad 3) y = \frac{1}{a + \frac{1}{b}}.$$

За потреби опишіть функції та складіть відповідні програми в середовищі програмування. Знайдіть значення змінної при значенні змінних $a = 16$ і $b = 9$.



- ★ 5. Візьміть участь у мініпроекті «Програмування на допомогу учням». Оберіть навчальний предмет: фізика, хімія, математика. Виділіть в обраній науці задачі, під час розв'язування яких застосовують формули та обчислення. Складіть програму, яка здійснюватиме обчислення за виділеними формулами. Використовуйте коментарі для пояснення свого коду. Продемонструйте свій проект іншим групам.



Працюємо самостійно

6. Дано дві змінні $x=0.5$, $y=0.1$. Визначте, які значення та який тип матимуть ці змінні після виконання команд присвоювання:

1) $x=2.5$ $x=x+2*y$ $y=y/2$ $x=x+y$ $y=-y$	2) $x=x+y$ $y=y*10$ $y=y-x$ $x=x**2+y$ $x=x/3$
---	--

7. Дано три цілочисельні змінні з поточним значенням $a = 3$, $b = 5$, $c = 7$. Які значення будуть мати ці змінні в результаті послідовних операторів присвоювання? Чи змінюють ці змінні свій тип? Складіть відповідну програму та перевірте свої здогадки.

1) $c = a * b + 2$

$b = b ** 2$

$a = c / b$

2) $a = (a + b) ** 2$

$b = a - c$

$c = a * b + c$

3) $b = 25 / b$

$a = b ** 2 + c$

$c = c + \text{round}(a)$

8. З комп'ютерного диска (діаметр 120 мм) восьмикласники виготовили квадратну заготовку для сувеніру. Складіть програму, яка визначатиме сторону подарункової коробки для цього сувеніру.

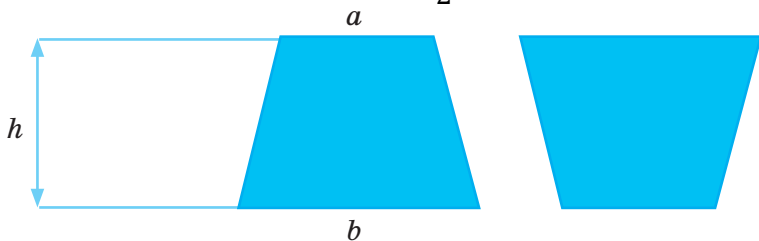
9. Складіть програму для знаходження кількості теплоти, яка потрібна для нагрівання тіла масою m , з початковою і кінцевою температурами тіла t_1 і t_2 , питомою теплоємністю речовини c , яка обчислюється за формулою:

$$Q = cm(t_2 - t_1).$$

Для обчислення використайте такі дані: $m = 5$ кг, $t_1 = 20$ °C; $t_2 = 100$ °C, $c = 2000$ Дж/°C. Якого типу буде змінна Q ?

10. Доріжку у парку вирішили встелити плиткою, яка має форму трапеції. Складіть програму, яка визначає найменшу кількість плиток для заданої площі P , якщо відомо, що ширина доріжки кратна висоті h трапеції. Врахуйте, що площу трапеції знаходять за формулою:

$$S = \frac{(a+b)}{2} h$$



11. Відомо, що відстань між двома точками, які задані на координатній площині своїми координатами (x_1, y_1) та (x_2, y_2) , визначається за формулою:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Складіть програму, за допомогою якої знаходять:

- периметр трикутника, заданого координатами своїх трьох вершин;
- площу прямокутника, побудованого на сторонах AB та AC , для яких задано координати вершин A , B , C .



Поняття об'єкта в мові програмування, його властивостей і методів

ПРИГАДАЙТЕ:

- ▶ що таке об'єкт і чим різняться об'єкти;
- ▶ які особливості має графічний інтерфейс програми;
- ▶ які об'єкти графічного інтерфейсу використовують в операційній системі *Windows*;
- ▶ як створюють проекти в середовищі *Скретч*.

ВИ ДІЗНАЄТЕСЬ:

- ▶ що розуміють під об'єктом у мові програмування;
- ▶ які особливості мають програми з графічним інтерфейсом;
- ▶ які є способи створення об'єктів графічного інтерфейсу;
- ▶ як в об'єкті *вікно* розмістити напис.



Вивчаємо

1. Що розуміють під об'єктом у мові програмування?

Постановка серйозної задачі практично ніколи не містить опису змінних, які треба використовувати. Навпаки, більшість реальних задач формуються як опис поведінки об'єктів, наприклад, під час комп'ютерного моделювання складних систем.

Наприкінці 80-х років XX ст. програмування опинилося в досить серйозній кризі. Головна проблема полягала в тому, що складність програм стала перевищувати можливості людського інтелекту. Одній людині стало не під силу охопити розумом усі аспекти складної програми.

Виходом із кризової ситуації став підхід, заснований на понятті об'єкта. Нова ідея полягала в тому, щоб розбити складну систему на частини — об'єкти, і спробувати зрозуміти й удосконалити кожну частину окремо, незалежно від інших. При цьому розробку кожної частини, пов'язаної з окремим об'єктом, можна доручити окремій людині та робити окремо. Такий підхід називають **об'єктно-орієнтованим програмуванням**.

Об'єкт (від лат. *objectum* — предмет, явище) — це те, на що спрямована певна діяльність (на противагу суб'єкту, який здійснює таку діяльність).

Ми живемо в оточенні величезної кількості різноманітних об'єктів. Будь-яка інформація, яку ми отримуємо, пов'язана з певними об'єктами. Об'єктами є не лише предмети чи явища матеріального світу, а й поняття, які створюють і використовують люди під час спілкування. Наприклад, «освіта», «держава» тощо. Кожний об'єкт має свою назву: «м'яч», «годинник», «школа», «книжка». Крім назви, кожний об'єкт має низку *властивостей* (параметрів), які описують об'єкт. Значення властивостей ми спостерігаємо чи визначаємо і, можливо, змінюємо. Наприклад, властивості об'єкта *людина*: стать, зріст, вага, колір очей, характер, зачіска і т.п. Властивості об'єкта *автомобіль* — це його марка, колір, об'єм двигуна, рік випуску. При цьому кожна властивість може мати різні значення.

Властивості поділяють на:

- **кількісні** — такі властивості можна подати числом у певних одиницях виміру (метрах, секундах, грамах);
- **якісні** — такі властивості можна лише назвати. Наприклад, стать людини може бути або жіноча, або чоловіча.

Значення властивостей об'єкта в певний момент часу визначають його стан. Вони можуть змінюватися, але їхня сукупність залишається сталою. Якщо така сукупність зазнає змін, то ми маємо справу вже з іншим об'єктом. Наприклад, значення властивості зріст об'єкта *учениця Ірина* може збільшуватися. Але якщо в неї зникла властивість успішність, то Ірина завершила навчання.

Подія — зміна властивостей об'єкта, взаємодія між об'єктами, утворення нового об'єкта або знищення наявного об'єкта.

Дії можуть виконувати власне об'єкти або їх виконують над об'єктами. Наприклад, об'єкт *учень* може виконувати такі дії: читати, грати у волейбол, розв'язувати задачу з фізики. У свою чергу, над цим об'єктом можна виконувати певні дії: вимірювати зріст, проводити виховну бесіду тощо.

Наприклад, об'єкт *прямокутник* характеризується такими даними: 1) довжина сторони *a*; 2) довжина сторони *b*. Також для прямокутника можна визначити такі дії: 1) знаходження периметра; 2) знаходження площі; 3) визначення рівності двох прямокутників, що задані довжинами своїх сторін; 4) масштабування (збільшення або зменшення сторін прямокутника в певну кількість разів). Отже, прямокутник можна описати логічною структурою (мал. 14.1).

Прямокутник

Мал. 14.1

Сторона a
Сторона b } Дані

Дія знаходження периметру
Дія знаходження площі
Дія порівняння (з іншим прямокутником)
Дія масштабування

} Функціональні можливості

З об'єктом пов'язане також поняття середовища — місця, в якому об'єкт може перебувати або де з ним можна виконувати деякі дії.

Середовище об'єкта — середовище, у якому перебуває об'єкт, незалежно від того, які дії ним чи над ним (крім знищення об'єкта) виконано.

У середовищі об'єкт може виконувати деякі команди — їх називають методами.

Методи — це команди, які об'єкт розуміє і може виконувати.

Наприклад, автомобіль реагує на натиснення педалей, поворот руля, перемикання швидкостей.

Ми нічого не знаємо про те, що містить об'єкт усередині. Для нас це, як говорять кібернетики, «чорний ящик». Але найголовніше — для того, щоб працювати з об'єктом, нам і не потрібно знати його внутрішній устрій. Достатньо, що ми можемо визначити й змінити його властивості, а також застосовувати доступні методи управління.

Властивості й методи є **інтерфейсом** об'єкта, тобто способом його спілкування із зовнішнім світом.

Як пов'язаний об'єкт із властивостями, подіями та методами в середовищі *Скретч*, показано на малюнку 14.2.



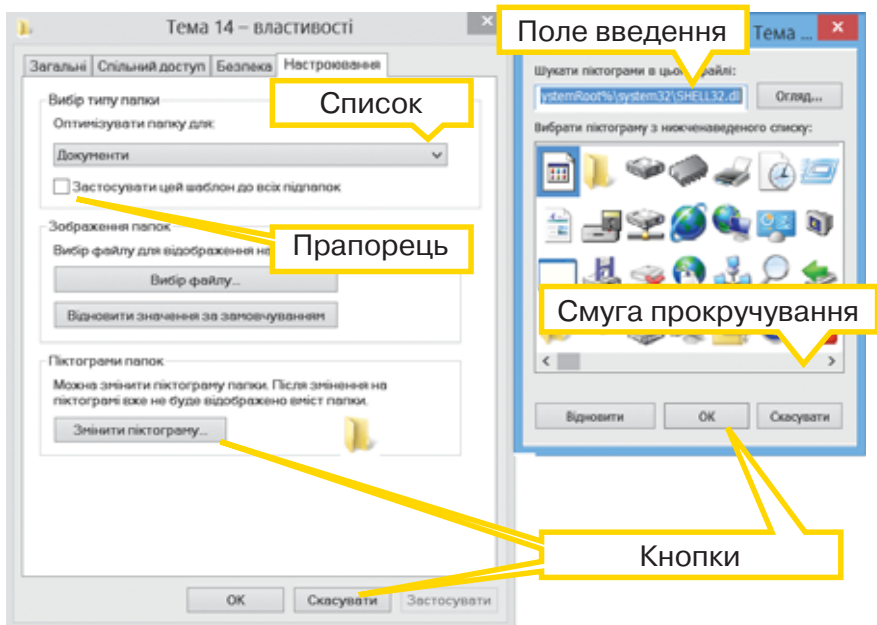
Мал. 14.2

2. Які особливості мають програми з графічним інтерфейсом?

Вам уже доводилося працювати з різними програмами, що мають графічний інтерфейс: редакторами, тренажерами, програмами для виконання обчислень, іграми тощо. Вони розроблені за допомогою певних середовищ програмування та мають спільні ознаки:

- програма відкривається у вікні, розмір якого, як правило, можна змінювати;
- графічний інтерфейс програми, який містить зображення значків, меню, кнопки, текстові поля тощо, дає змогу користувачеві за допомогою миші виконувати потрібні команди, змінювати значення властивостей об'єктів і вводити за допомогою клавіатури текстові та числові дані (мал. 14.3);
- усі об'єкти, які використано у програмі, мають властивості, що їх можна змінювати;
- події, які відбуваються у програмі, пов'язані з певними об'єктами. Наприклад, з об'єктом *меню програми* пов'язана подія, яка викликається натисненням лівої кнопки миші на обраному розділі — розгорнути список вказівок, що об'єднані в цьому розділі меню.

У програмному коді, що орієнтований на опрацювання подій, програміст повинен вказати, як слід реагувати на різні події (чи дії користувача). Це можуть бути, наприклад, такі події: вибір вказівки, клацання кнопкою миші, переміщення миші тощо.



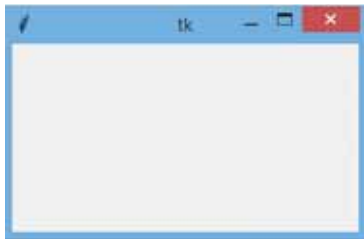
Мал. 14.3

3. Які є способи створення об'єктів графічного інтерфейсу?

Для того щоб створити об'єкти графічного інтерфейсу мовою програмування *Python*, можна використати модуль підключення команд для створення графічного інтерфейсу *tkinter* або *easygui*.

Створення об'єктів програми, їх розташування та запуск опрацювання подій програми задаються командами мови програмування.

Наприклад, найпростіша програма з графічним інтерфейсом користувача — виведення вікна (мал. 14.4), яка створена в середовищі програмування *Python* з використанням модуля *tkinter*, складається з таких команд:



Мал. 14.4

```
# підключення модуля віконного графічного інтерфейсу
користувача
import tkinter
# створення вікна програми
window = tkinter.Tk()
window.mainloop()
```

Для запуску циклу опрацювання подій у модулі *tkinter* використовують метод *mainloop()*.

Виклик метода здійснюють конструкцією:

Ім'я_об'єкта.Ім'я_методу

Об'єкт *вікно* має свої властивості: назва, розмір, розміщення. Їх можна змінити у програмі вбудованими методами. Наприклад,

- *title* — заголовок вікна;
- *geometry* — встановлює геометрію вікна у форматі ширина×висота + *x* + *y*.

Наприклад, *geometry ("600x400 + 40 + 80")* означає таке: помістити вікно в точку з координатами (40,80) і встановити розмір (600, 400). Розмір або координати можна не вказувати;

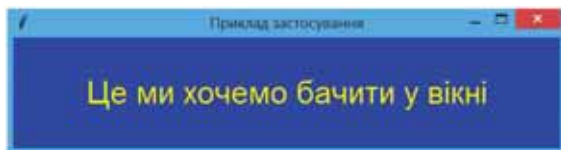
- *geometry("600x400")* — лише змінити розмір;
- *geometry("+ 40 + 80")* — лише перемістити вікно.

4. Як в об'єкті *вікно* розмістити напис?

Для того щоб у вікні розмістити напис, використовують об'єкт *Label*.

```
labell = tkinter.Label (window, text='Це ми хочемо ба-
чити у вікні', width=30, height=3, bg='blue', fg='yellow',
font='windows 24')
```

Отримаємо (мал. 14.5):



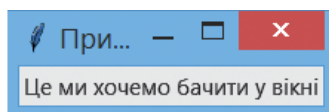
Мал. 14.5

Розберемо цей код. Для створення напису використано змінну `label1(напис1)`. Змінній присвоюють `label1` значення, яке викликається з модуля `tkinter` методом `Label` — напис. Об'єкт *напис* має властивості (можна використовувати не всі):

- `window` — назва вікна розташування;
- `text` — текст напису;
- `width, height` — ширина й довжина (у знакомісцях);
- `bg` — колір (скорочено від *background*);
- `fg` — колір напису (скорочено від *foreground*);
- `font` — шрифт і його розмір.

Для того щоб розмістити об'єкт у вікні, використовують метод `pack()`.

Наприклад, для вікна на малюнку 14.6 програма матиме вигляд, як на малюнку 14.7.



Мал. 14.6

```
1 # вікно з написом
2 import tkinter
3 # створення вікна програми
4 window = tkinter.Tk()
5 window.title('Приклад застосування')
6 # створення напису
7 label1 = tkinter.Label(window, text='Це ми хочемо бачити у вікні')
8 # розміщення напису
9 label1.pack()
10 # запуск події
11 window.mainloop()
```

Мал. 14.7



Діємо

Вправа 1. Повідомлення.

Завдання. Створіть програму, за допомогою якої у вікні повідомлення буде виводитися цитата:

Людина, яка в ХХІ столітті не вмітиме користуватись комп'ютером, буде подібна людині ХХ століття, що не вміє ні писати, ні читати.

Академік В. М. Глушков

1. Запустіть середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Текст* у папці *Навчальні проекти* своєї структури папок.
2. Підключіть до файлу модуль віконного графічного інтерфейсу користувача, створіть об'єкти: вікно, два текстові надписи, що відображають зміст висловлювання та його автора.
3. Розмістіть напис та команду запуску опрацювання події програми на головній формі. Наприклад, для виведення у вікні повідомлення тексту цитати потрібно ввести такий програмний код:

```
label1 = tkinter.Label(text="Людина, яка в ХХІ столітті не вмітиме користуватись комп'ютером, буде подібна людині ХХ століття, що не вміє ні писати, ні читати.")
label1.pack()
```

4. Запишіть команди створення напису та запуску обробки події для виведення у вікні автора висловлювання.
5. Запустіть файл програми на виконання. Переконайтеся, що на екрані ви отримали потрібне вікно із цитатою.
6. Завершіть роботу із середовищем програмування. Закрийте всі вікна програм.



Досліджуємо

Вправа 2. Напис.

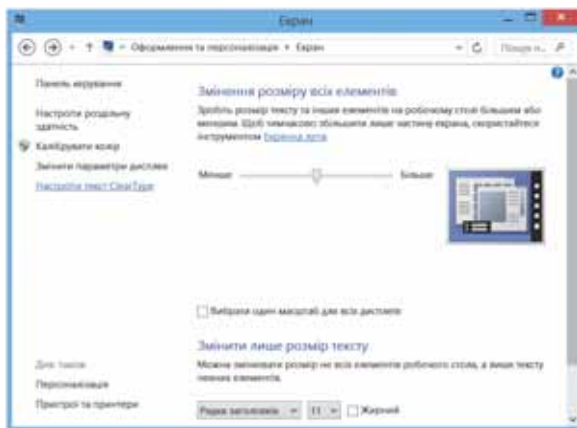
Завдання. Дослідіть, як впливають значення властивостей напису на відображенні тексту у вікні, створеному мовою *Python*.

1. У середовищі програмування введіть текст програми з малюнка 14.7.
2. Збережіть програму з іменем *Прізвище_напис*.
3. По черзі додавайте значення властивостей напису, які відповідають малюнку 14.5. Запускайте програму на виконання. Як змінюється вигляд вікна результату?
4. Дослідіть, що відбуватиметься, якщо ширину напису дати такою, що дорівнює 10.



Працюємо в парах

1. Визначте три аргументи, які підтверджують, що навчальне середовище програмування *Скретч* підтримує об'єктно-орієнтоване програмування.
2. Визначте, які елементи графічного інтерфейсу використано у вікні налаштування екрана (мал. 14.8). Хтось називає елемент, а інший чи інша — його властивості та події, що з ним пов'язані.



Мал. 14.8

3. Обговоріть, чим для вас може бути корисним посилання <https://www.thinkful.com/learn/intro-to-python-tutorial/>.
4. Попрацюйте в мініпроекті «Фрази, що надихають». Оберіть тему, яка буде корисна учням вашого класу. Наприклад, навчання, дружба, здоров'я тощо. Доберіть фрази, цитати, прислів'я до теми та створіть для кожної вікно з написом. Доберіть властивості написів. Поекспериментуйте з кольорами, розмірами та шрифтами. Передбачте, що щоразу вікно з написом має з'являтися в іншому місці екрана.

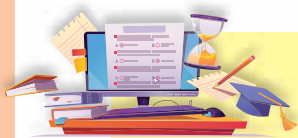


Працюємо самостійно

5. Ваш молодший брат любить довго засиджуватися за комп'ютером. Створіть програму, після запуску якої буде з'являтися вікно з повідомленням про необхідність дотримання часу роботи за комп'ютером. Доберіть властивості напису, які привернуть увагу брата.
6. Ви хочете застосувати свій комп'ютер для планування діяльності своєї родини. Сплануйте та створіть прикладну програму, після запуску якої у вікні з'являтиметься повідомлення, актуальне для вашої родини, наприклад, нагадування про день народження, важливу подію, завдання, яке слід виконати тощо.
7. Створіть програму, після запуску якої на екранній формі буде відображено календар на поточний місяць. Скористайтесь підказкою: в середовищі програмування в контекстному меню команд можна використовувати команди для роботи з фрагментами коду (мал. 14.9).

Скасувати	Ctrl+Z
Повторити	Ctrl+Y
Вирізати	Ctrl+X
Скопіювати	Ctrl+C
Вставити	Ctrl+V
Виділити все	Ctrl+A

Мал. 14.9



Обробники подій, пов'язаних з елементами керування

ПРИГАДАЙТЕ:

- ▶ які події можна описати командами групи *Керувати* в середовищі *Скретч*.

ВИ ДІЗНАЄТЕСЬ:

- ▶ як пов'язані елементи керування та події;
- ▶ як у програмі описати елемент керування *кнопка*;
- ▶ що таке обробник події;
- ▶ як отримати вікно повідомлення.



Вивчаємо

1. Як пов'язані елементи керування та події?

Окрім напису, який є статичним об'єктом вікна графічної форми результату виконання програми мовою *Python*, можна використовувати об'єкти — елементи керування, які можуть «реагувати» на деякі *події*.

Елементи керування — це графічні об'єкти, розташовані на вікні програми для показу або введення даних, виконання дій або полегшення роботи: текстові поля, списки, перемикачі, кнопки, прапорці тощо.

Якщо властивості характеризують стан елемента керування, методи — дії, які він може виконати, то події — зовнішній вплив на елемент керування, на який цей елемент може реагувати.

Тому елемент керування умовно можна описати формулою:

Елемент керування = *властивості (стан)* + *методи (дії)* + *події (зворотні зв'язки)*

Виділяють такі три основні типи подій:

- взаємодія з мишею;
- натискання клавіш на клавіатурі;
- зміна стану об'єктів, створених у результаті виконання програми.

Приклади подій взаємодії з мишею:

- <Button-1> — клацання лівою кнопкою миші;
- <Button-2> — клацання середньою кнопкою миші;

- <Button-3> — клацання правою кнопкою миші;
- <Double-Button-1> — подвійне клацання лівою кнопкою миші;
- <Motion> — рух миші.

При описі події натискання клавіш клавіатури літерні клавіші можна записувати без кутових дужок (наприклад, 'L'). Для неалфавітних клавіш існують спеціальні зарезервовані слова:

- <Return> — натискання клавіші *Enter*;
- <Space> — пропуск;
- <Control> — натискання клавіші *Ctrl*;
- <Shift> — натискання клавіші *Shift*;
- <z> — натискання клавіші *z*.

В останньому випадку замість *z* можна використовувати будь-який символ клавіатури, що містить літеру. Назву події натискання комбінації клавіш вказують через дефіс. Наприклад: <Control-a> або <Control-Shift>.

2. Як у програмі описати елемент керування кнопка?

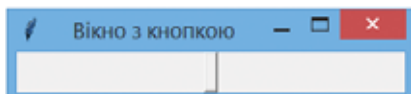
Для додавання елемента керування *кнопка* до вікна програми з графічними елементами, яке створене мовою *Python* з використанням модуля *tkinter*, використовують команду `Button()`.

Якщо у програмі використати код, як на малюнку 15.1,

```
1 # вікно з кнопкою
2 import tkinter
3 # створення вікна програми
4 window = tkinter.Tk()
5 window.title('Вікно з кнопкою')
6 button1 = tkinter.Button() # створення кнопки button1
7 button1.pack()             # розташування кнопки button1 на головному вікні
8 window.mainloop()
```

Мал. 15.1

то на головному вікні буде розташована кнопка, яка матиме всі властивості вікна (мал. 15.2):



Мал. 15.2

Так само, як і напис, об'єкт *кнопка* може мати свої властивості, які описують службовими словами:

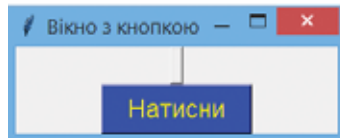
- `text` — напис на кнопці;
- `width`, `height` — ширина й довжина (у знаках);
- `bg` — колір (скорочено від *background*);
- `fg` — колір напису (скорочено від *foreground*);
- `font` — шрифт і його розмір;
- `padx` — відступ від межі до об'єкта по горизонталі;
- `pady` — відступ від межі до об'єкта по вертикалі.

Змінити властивість об'єкта під час виконання програми можна за допомогою методу `configure`.

Якщо програму, код якої подано на малюнку 15.1, доповнити командами додавання кнопки 2 (мал. 15.3), то отримаємо вікно з двома кнопками, які за замовчуванням розміщуються по центру верхнього краю (мал. 15.4).

```
button2 = tkinter.Button(window, text='Натисни',
                          width=10, height=1,
                          bg='blue', fg='yellow', font='Arial 14')
button2.pack()
```

Мал. 15.3



Мал. 15.4

Розміщенням кнопок, як і іншими елементами керування на вікні, управляють за допомогою властивостей методу `pack()`:

- `expand`: якщо дорівнює `True`, то елемент управління заповнює все вікно;
- `fill`: визначає, чи буде елемент управління розтягуватися, щоб заповнити вільне поле. Цей параметр може набувати значень: `NONE` (за замовчуванням, елемент не розтягується), `X` (елемент розтягується тільки за горизонталлю), `Y` (елемент розтягується тільки за вертикаллю) и `BOTH` (елемент розтягується і за горизонталлю, і за вертикаллю);
- `side`: вирівнює елемент управління щодо сторін вікна. Може набувати значень: `TOP` (за замовчуванням, вирівнюється зверху, по центру), `BOTTOM` (вирівнюється по центру, по нижній основі), `LEFT` (вирівнюється за лівою межею), `RIGHT` (вирівнюється за правою межею).

Ці властивості заздалегідь потрібно імпортувати з модуля `tkinter`:

```
from tkinter import TOP, BOTTOM, RIGHT, LEFT.
```



ДІЄМО

Вправа 1. Сторони горизонту.

Завдання. Складіть програму, яка за допомогою кнопок ілюструє сторони горизонту.

1. Відкрийте середовище програмування. Створіть файл *Сторони_горизонту_Прізвище* в папці *Навчальні проекти* вашої структури папок.
2. Імпортуйте потрібні модулі та їхні складові.
3. Доповніть текст програми (мал. 15.1) так, щоб додати кнопку з написом *Південь* і розмістити її по центру внизу.

```
button2 = tkinter.Button(window, text="Південь")
button2.pack(side=BOTTOM)
```

4. Додайте інші кнопки. Перевірте, чи отримали ви потрібний результат. За потреби виправте помилки.

3. Що таке обробник подій?

Щоб описати дії, які мають виконуватися за настання деякої події для об'єкта, необхідно написати програмний код — обробник події.

Обробник події — це частина програми, що виконує певні дії у відповідь на подію (спричинену, наприклад, дією користувача).

Описують подію за допомогою функції користувача. Наприклад, подія «натиснуто кнопку» `button_click()` викликатиме команду `print`, за якою у вікні середовища отримаємо результат — *Привіт* (мал. 15.5).

```
def button_click():
    print('Привіт!')
```

```
>>> %Run 'Кнопка.py'
```

 Мал. 15.5

Привіт!

Зверніть увагу, що мовою програмування *Python* під час запису наступної команди в новому рядку важливою є відстань зліва, що формується за допомогою відповідної кількості пропусків: команди, які мають однаковий рівень за вертикаллю, незалежні одна від одної і виконуються послідовно, а для команд, що розпочинаються правіше — кожний наступний рівень вказує, що команди цього рівня виконуються в межах команди вищого рівня.

Для опрацювання натиснення на кнопку в конструкторі кнопки потрібно встановити параметр `command`, надавши йому посилання на функцію, яка діятиме при натисненні. Повний текст програми подано на малюнку 15.6.

```
1 import tkinter
2 import tkinter.messagebox
3 # опрацювання події натиснення на кнопку
4 def button_click():
5     print('Привіт!')
6 window = tkinter.Tk()
7 # створення текстового напису та його розміщення на головній формі
8 label = tkinter.Label(text="Hello World!")
9 label.pack()
10 # створення кнопки та її розміщення на головній формі
11 button = tkinter.Button(window, text="Привіт!", command=button_click)
12 button.pack()
13 window.mainloop()
```

 Мал. 15.6

Зв'язувати об'єкт керування, подію та дію можна й за допомогою методу `bind`. У функцій-обробників, які викликають через `bind`, а не через властивість `command`, має обов'язково бути параметр `event`, через який передається подія.

Тоді виклик цієї функції задається конструкцією:

Об'єкт `bind('подія', назва_функції)`



ДІЄМО

Вправа 2. Перетворення кольору.

Завдання. Створіть програму, у якій вікно змінюватиме колір фону на зелений після клацання лівою кнопкою миші й на червоний — після клацання правою.

1. Запустіть середовище програмування. Створіть новий файл *Перетворення_Прізвище* у своїй структурі папок.

2. Імпортуйте потрібні модулі та їхні складові, наприклад,
`import tkinter`

3. Створіть вікно.

```
window=tkinter.Tk()
window.mainloop()
```

4. Створіть функцію з іменем `left_mouse`, на місці аргументу вкажіть `event`, що означає, що функція пов'язана з подією.

```
def left_mouse (event):
    window['bg'] = 'green'
```

5. Створіть функцію з іменем `right_mouse`

```
def right_mouse (event):
    window['bg'] = 'red'
```

6. Для вікна створіть події і прив'яжіть їх до створених функцій.

```
window=tkinter.Tk()
window.bind('<Button-1>', left_mouse)
window.bind('<Button-3>', right_mouse)
```

7. Запустіть програму на виконання. Переконайтесь, що вона виконує поставлене завдання (мал. 15.7).

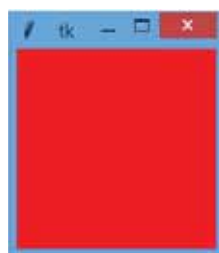
 **Мал. 15.7**



Після запуску



Після клацання
лівою кнопкою миші



Після клацання
правою кнопкою миші



Досліджуємо

Вправа 3. Годинник.

Завдання. Дослідіть, що отримують у результаті виконання програми (мал. 15.8). Як змінюється вікно програми після натиснення кнопки? Яке призначення модуля `time`?

```
1 import tkinter
2 import time
3 def button_clicked():
4     button['text'] = time.strftime('%H:%M:%S')
5 window = tkinter.Tk()
6 button = tkinter.Button(window)
7 button.configure(text=time.strftime('%H:%M:%S'), command=button_clicked)
8 button.pack()
9 window.mainloop()
```

Мал. 15.8

4. Як отримати вікно повідомлення?

Окрім головного вікна програми, за допомогою додаткових модулів `tkinter` можна отримати вікна повідомлень, вибору «так-ні», відкривання файлів тощо. Одним з модулів є модуль `messagebox`.

Щоб викликати цей модуль, на початку програми записують команду: `import tkinter.messagebox`

Тоді вікно повідомлення із заголовком та текстом формується за допомогою команди:

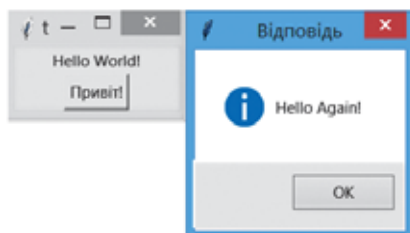
```
tkinter.messagebox.showinfo("Заголовок", "Команда")
```



Діємо

Вправа 4. Вітання.

Завдання. Створіть вікно з написом *Hello World!* і кнопкою *Привіт*. Після натискування кнопки на екрані має з'явитись вікно повідомлення із заголовком *Відповідь* і текстом *Hello Again!* (мал. 15.9).



Мал. 15.9

1. Відкрийте середовище програмування. Створіть файл *Вітання_Прізвище*.
2. Змініть текст програми так, щоб після натискування кнопки на екран виводилось вікно повідомлення. Використайте фрагмент коду:

```
def button_click():
    tkinter.messagebox.showinfo("Відповідь", "Hello Again!")
```

3. Запустіть програму на виконання. Переконайтеся, чи отримали ви бажаний результат. Чи потрібно вносити інші зміни у програму?



Досліджуємо

Вправа 5. Інформаційні вікна.

Завдання. Дослідіть, як зміниться вигляд вікна повідомлення у програмі *Вітання*, якщо замість команди методу `showinfo` використати метод `showerror` або `showwarning`.



Працюємо в парах

1. Обговоріть, як описують події в навчальному середовищі складання та виконання алгоритмів *Скретч* і мовою програмування *Python*. Чим відрізняються події?
2. Обговоріть, як у програмах із графічним інтерфейсом використовують елементи управління кнопки. Які події пов'язані з ними? Хтось називає приклад, а хтось — подію.
- ★ 3. Пограйте у гру «Властивість–призначення». Хтось називає властивість елемента керування кнопка, інший чи інша — призначення. Якщо відповідь правильна, то поміняйтесь ролями. Виграє той чи та, чиє запитання буде останнім.
- ★ 4. Придумайте ідею, сплануйте та створіть власний проект із використанням кнопок, вікон повідомлення, модуля часу. Запропонуйте його для виконання іншим учням. Порівняйте отримані результати.



Працюємо самостійно

5. Складіть програму, після запуску якої на екрані з'являється вікно, де послідовно розміщено 4 різнокольорові кнопки з назвами кольорів та різними розмірами кнопок за шириною.
- ★ 6. Складіть програму, після запуску якої на екрані буде розміщено напис «Автори Гімну України» та дві кнопки. Натискування першої «Слова» виводить відповідь у середовище програмування, а другої «Музика» — в інформаційне вікно.
- ★ 7. Складіть опитувальник з вашого улюбленого предмета. У вікні опитувальника під запитанням розмістіть кнопки «так» і «ні». Якщо користувач натисне правильну кнопку, то у відповідь має отримати вікно з повідомленням, яке пояснює відповідь на запитання. А якщо неправильну — тоді вікно із цим поясненням буде мати вигляд вікна `showerror`.



16

Опрацювання величин у програмах з елементами керування

ПРИГАДАЙТЕ:

- ▶ які події можна описати командами групи *Керувати* в середовищі *Скретч*.

ВИ ДІЗНАЄТЕСЬ:

- ▶ як використовують текстове поле для роботи з величинами;
- ▶ які операції виконують над текстовими величинами;
- ▶ які функції використовують для опрацювання текстових величин.



Вивчаємо

1. Як використовують текстове поле для роботи з величинами?

Надавати значення величинам можна різними способами:

- під час написання програмного коду за допомогою оператора присвоєння;
- під час виконання програми, що містить команду введення *Input*;
- за допомогою елементів управління.

Entry — це елемент керування для введення *одного рядка тексту*.

Має такі додаткові (порівняно з *Button* і *Label*) властивості:

- *borderwidth* (або скорочено *bd*) — ширина межі;
- *show* — відображувані символи.

Наприклад, змінна *text1* отримуватиме значення з однорядкового текстового поля висотою 7, шириною 8, яке написано шрифтом за замовчуванням 14 розміром (мал. 16.1).

Виклик методу: текстове поле

Властивості

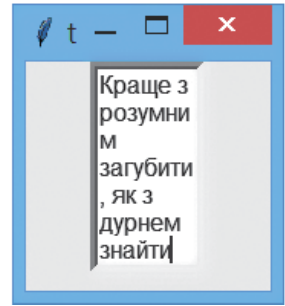
```
text1 = tkinter.Entry(window, height=7, width=8, font='14', bd=5)
```

Змінна

Елемент, на якому розміщено поле

Мал. 16.1

Text — це елемент керування для введення довільного багаторядкового тексту. Має (додаткову) властивість *wrap*, що відповідає за перенесення на новий рядок. Щоб перенести по словам, потрібно надати їй значення WORD, яке додається з бібліотеки додатково. Наприклад, текст у текстовому полі (мал. 16.2) вводили без натискування клавіші *Enter* у вікні з елементом управління багаторядкове поле за допомогою такого коду:



Мал. 16.2

```
import tkinter
from tkinter import WORD
window = tkinter.Tk()
text1 = tkinter.Text(window, height=7, width=8, font='14',
wrap=WORD, bd=5)
text1.pack()
window.mainloop()
```

Методи *insert*, *delete* і *get*, відповідно, додають, видаляють або витягають текст. Перший аргумент — місце вставлення у вигляді 'x.y', де *x* — це рядок, а *y* — стовпчик.

```
text1.insert('1.0', 'Додати цей текст \n \ в початок першого рядка')
text1.delete('1.0', END) # Видалити все
text1.get('1.0', END) # Витягти все
```

Наприклад, змінній *s* надають значення командою

```
s = tkinter.StringVar()
s.set('Значення')
```

А отримати значення можна так:

```
s.get()
```

Для того щоб отримати значення з текстового напису, використовують об'єкти (внутрішні змінні): *BooleanVar()*, *StringVar()*, *IntVar()*, *DoubleVar()*. Тоді серед параметрів елемента керування має бути *textvariable*.

Наприклад, змінній текстового типу *str_var* буде надано значення, яке отримано з текстового поля *edit*:

```
str_var = tkinter.StringVar()
edit = tkinter.Entry(window, textvariable = str_var)
```



ДІЄМО

Вправа 1. Нагадування.

Завдання. У середовищі програмування розробіть проєкт, у якому користувач буде вводити день тижня, натискати кнопку *Гаразд* й отримувати

вати у вікні із заголовком *Увага!* таке повідомлення: *Сьогодні – <день тижня, який було введено>*.

1. Запустіть середовище програмування.
2. Створіть новий файл програми мовою *Python* з іменем *Нагадування* в папці *Навчальні проекти* своєї структури папок.
3. У вікні редактора коду запишіть команди для імпортування потрібних модулів і методів. Створіть вікно проекту.

```
window = tkinter.Tk()
```

4. Створіть об'єкт для отримання значення з текстового поля.

```
str_var = tkinter.StringVar()
```

5. Додайте функцію опрацювання події натиснення на кнопку.

```
def button_click():
    tkinter.messagebox.showinfo("Увага!", "Сьогодні "
    + str_var.get())
```

6. Створіть текстовий напис та розмістіть його у вікні.

```
label = tkinter.Label(text="Введіть день тижня")
label.pack()
```

7. Створіть текстове поле та розмістіть його у вікні.

```
edit = tkinter.Entry(window, textvariable = str_var)
edit.pack()
```

8. Створіть кнопку та розмістіть її у вікні.

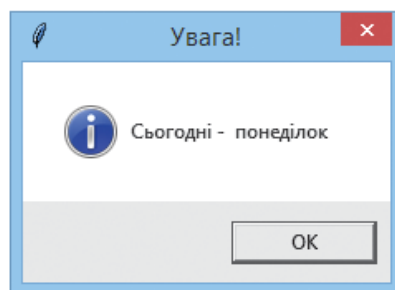
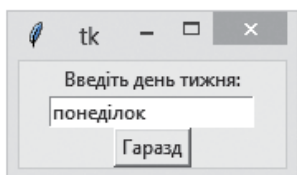
```
button = tkinter.Button(window, text="Гаразд",
    command=button_click)

button.pack()
```

9. Завершіть програму командою запуску опрацювання подій програми.

10. Запустіть проект на виконання. Перевірте, чи результат для введеного дня тижня *понеділок* відповідає малюнку 16.3.

 **Мал. 16.3**



11. Завершіть роботу з проектом і середовищем.

Вправа 2. Робота з текстом.

Завдання. Напишіть програму графічного інтерфейсу *Python* для створення текстового поля за допомогою модуля *tkinter*. Вставте рядок *В чужім краю не цвірінькають..* на початку, а потім вставте рядок *і солов'ї* в поточний текст так, щоб отримати приказку *В чужім краю і солов'ї не цвірінькають*. Видаліть перший і два останні символи тексту. Використайте текст програми, після виконання якої на екрані отримаємо таке зображення (мал. 16. 4):



Мал. 16.4

```
import tkinter
window = tkinter.Tk()
mytext = tkinter.Text(window)

# Вставте рядок на початку
mytext.insert('1.0', "_Моя Батьківщина - ")

# Вставте рядок в кінці
mytext.insert('1.19', ' Україна!! ')

# Видаліть перший і два останні знаки
mytext.delete('1.0')
mytext.delete('end - 2 chars')
mytext.pack()
window.mainloop()
```

2. Які операції виконують над текстовими величинами?

Над текстовими величинами виконують операцію **склеювання** — поєднання кількох рядкових величин, яка позначається символом «+».

Наприклад, у результаті виконання фрагменту програми

```
F=' алго'
L=' ритм'
R=F+L
```

отримаємо значення *R*, що дорівнює 'алгоритм'.

Текстові величини є незмінюваними, це означає, що не можна змінити частину рядка, не створивши нового.

Ви вже знаєте, що в програмі мовою програмування *Python* під час роботи з текстовими величинами можна використовувати операцію **зрізу**, за допомогою якої копіюється послідовність або її частина.

Наприклад, для змінної

```
s = 'Hello world'
```

операції зрізу подано в таблиці 16.1.

Таблиця 16.1

Опис	Приклад команди	Результат — значення змінної <i>s1</i>
Зріз від початку рядка до символу з номером <i>n</i>	<code>s1 = s[:6] + 'Python!'</code>	<code>'Hello Python!'</code>
Зріз від символу з номером <i>n</i> рядка включно до символу з номером <i>m</i>	<code>s1=s[3:5]</code>	<code>'lo'</code>
Зріз від символу з номером <i>n</i> рядка включно до кінця рядка	<code>s1=s[6:]</code>	<code>'world'</code>

Щоб записати символи в рядку в зворотному порядку, використовують операцію:

```
s3=s[::-1].
```

Тоді змінна *s3* матиме значення `'dlrow olleH'`.

У мові програмування *Python* можна вставляти в рядок дані в необхідному форматі. Для цього використовують оператор `%`, першим операндом якого є рядок-шаблон, а другим — **тьюпл** із даними для вставки.

Рядок-шаблон містить «заглушки», що позначають, у які місця мають вставлятися дані та в якому форматі. Наприклад:

```
s = "My name is %s, I'm %s years old." % ('Vasia', 14)
```

аналогічне запису

```
s = "My name is ' + 'Vasia' + ', I'm ' + str(14) + ' years old."
```

Дані вставляються в рядок у тому порядку, в якому вони перебувають у тьюплі. «Заклушка» `%s` позначає вставлення даних із приведенням їх до типу рядок — `str()`.

3. Які функції використовують для опрацювання текстових величин?

Ви вже використовували функції для перетворення текстової величини у величину числового типу та навпаки.

У мові програмування *Python* для опрацювання текстових величин використовують функції, наведені в таблиці 16.2.

Таблиця 16.2

Опис функції мовою програмування	Призначення
<code>len(S)</code>	Визначення кількості символів у рядку <code>S</code>
<code>s.upper()</code>	Змінює регістр усіх символів рядка на верхній
<code>s.lower()</code>	Змінює регістр усіх символів рядка на нижній
<code>s.replace(s_old, s_new)</code>	Замінює всі входження фрагмента <code>s_old</code> у рядку на <code>s_new</code>
<code>s.replace(s_old, s_new, count)</code>	Замінює перші <code>count</code> входжень фрагмента <code>s_old</code> у рядку на <code>s_new</code>
<code>s.find(s1)</code>	Повертає позицію входження (індекс першого символу) фрагмента <code>s1</code> у рядку, або <code>-1</code> , якщо фрагмент не знайдено
<code>s.find(s1, start_pos)</code>	Повертає позицію входження (індекс першого символу) фрагмента <code>s1</code> у рядку, починаючи з позиції <code>start_pos</code> , або <code>-1</code> , якщо фрагмент не знайдено

У мові програмування *Python* є функції для роботи з кодами текстових величин (табл. 16.3).

Таблиця 16.3

Опис функції мовою програмування	Тип аргументу	Тип результату	Призначення
<code>chr(x)</code>	цілий	символьний	Визначає символ з кодом <code>x</code>
<code>ord(c)</code>	символьний	цілий	Визначає код символу <code>c</code>

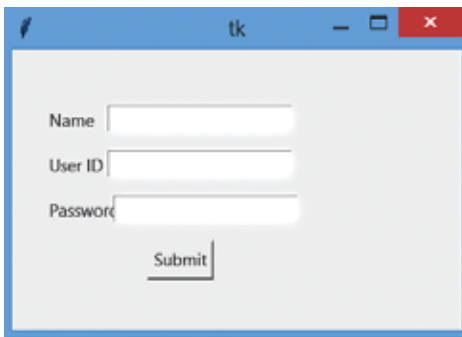


Досліджуємо

Вправа 3. Доступ.

Завдання. Розгляньте команди та їхні параметри, що використані у програмі *Доступ*, після запуску якої отримують вікно, як на малюнку 16.5. Дослідіть призначення методу `place`. Зверніть увагу, що для того щоб скоротити назву бібліотеки `tkinter`, призначили ім'я `tk` (`as tk`).

```
#Доступ
import tkinter as tk
window = tk.Tk()
window.geometry("400x250")
name = tk.Label(window, text = "Name").place(x = 30, y = 50)
email = tk.Label(window, text = "User ID").place(x = 30,
y = 90)
password = tk.Label(window, text = "Password").place(x = 30,
y = 130)
sbmitbtn = tk.Button(window, text = "Submit").place(x = 120,
y = 170)
entry1 = tk.Entry(window).place(x = 85, y = 50)
entry2 = tk.Entry(window).place(x = 85, y = 90)
entry3 = tk.Entry(window).place(x = 90, y = 130)
window.mainloop()
```



Мал. 16.5



Працюємо в парах

1. Обговоріть, чим подібне та чим відрізняється опрацювання величин у стандартних програмах мовою програмування *Python* і програмах із графічним інтерфейсом.

2. Пограйте у гру «Рядок програми–пояснення». Хтось називає рядок програми, а хтось — пояснює її призначення. За першої помилки поміняйтеся ролями. Виграє той чи та, хто останнім назве правильну відповідь. Використайте для цього текст програми, яку створили в цій темі.



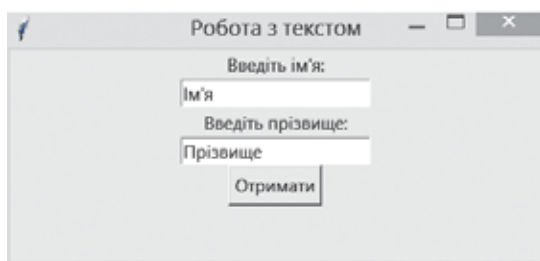
3. Попрацюйте у проєкті. Поміркуйте, що буде в результаті виконання програми. Запропонуйте коментарі-пояснення після символів #. Обговоріть, як змінити чи доповнити код, щоб отримати програму *Калькулятор*, у вікні якої на чотирьох кнопках будуть написані знаки арифметичних дій, а натискування кнопки передаватиме відповідний результат у вікно повідомлення.

```
import tkinter
import tkinter.messagebox
window = tkinter.Tk() #
str_var1 = tkinter.StringVar()
str_var2 = tkinter.StringVar() #
def button1_click():
    tkinter.messagebox.showinfo("Результат додаван-
ня", int(str_var1.get()) + int(str_var2.get()))
def button2_click():
    tkinter.messagebox.showinfo("Результат
віднімання", int(str_var1.get())-int(str_var2.
get())) #
label = tkinter.Label(window, text="Введіть числа")
label.pack() #
edit1 = tkinter.Entry(window, textvariable = str_
var1)
edit1.pack()
edit2 = tkinter.Entry(window, textvariable = str_
var2)
edit2.pack() #
button1 = tkinter.Button(window, text="+",
command=button1_click)
button1.pack()
button2 = tkinter.Button(window, text="-",
command=button2_click)
button2.pack() #
window.mainloop()
```

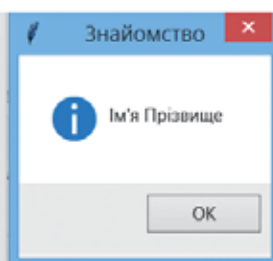


Працюємо самостійно

4. Розробіть програму, після запуску якої на екрані в текстове поле вводять текст. Натискування кнопки *Перетворити* викликає на екран інформаційне вікно з перетвореним текстом:
 - а) у тексті забрано першу та останню літери;
 - б) текст написано двічі;
 - в) усі літери тексту записані у верхньому регістрі;
 - г) виведено кількість символів у тексті.
- ★ 5. Розробіть програму, після запуску якої на екрані отримують вікно (мал. 16.6).



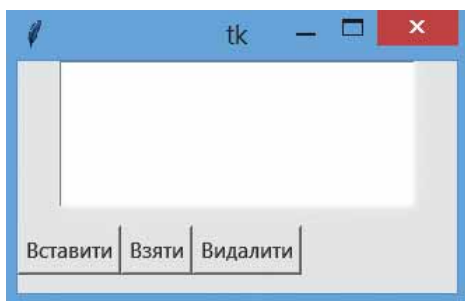
Мал. 16.6



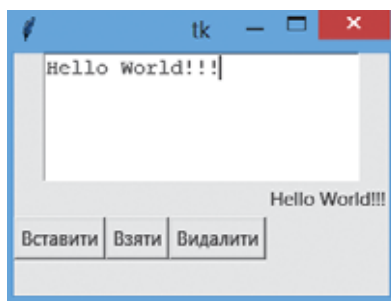
Мал. 16.7

А після введення імені та прізвища — вікно повідомлення (мал. 16.7).

- ★ 6. Розробіть програму, після запуску якої на вікні форми розміщено три кнопки та текстове поле (мал. 16.8). Кнопка *Вставити* на початок тексту до введеного тексту додає *Hello World*, кнопка *Взяти* переміщує текст у напис, вирівняний по правому краю, кнопка *Видалити* очищає текстове поле (мал. 16.9).



Мал. 16.8



Мал. 16.9

- ★★ 7. Придумайте ідею та складіть програмний код для програми, яку можна використати на уроках математики, фізики, хімії, географії. У вікні програми потрібно розмістити принаймні два текстові вікна, дві кнопки, два написи та використати одне вікно повідомлення.



Розгалуження мовою програмування *Python*

ПРИГАДАЙТЕ:

- ▶ як порівнюють значення величин у програмах;
- ▶ які операції виконують над логічними величинами;
- ▶ як описують алгоритмічну структуру розгалуження;
- ▶ якими бувають розгалуження.

ВИ ДІЗНАЄТЕСЬ:

- ▶ які команди використовують для опису розгалужень мовою програмування *Python*;
- ▶ як за допомогою елементів управління задати логічне значення.



Вивчаємо

1. Які команди використовують для опису розгалужень мовою програмування *Python*?

Ви вже знаєте, що для опису алгоритмів з розгалуженням мовою програмування *Python* використовують команди неповного розгалуження, повного розгалуження та вибору (табл. 17.1).

Таблиця 17.1

Назва	Опис	Пояснення
Команда неповного розгалуження	<pre>if <логічний вираз>: блок команд</pre>	Команда або блок команд у команді неповного розгалуження буде виконуватись тільки в разі, коли логічний вираз набуває значення <i>True</i>

Продовження таблиці 17.1

Назва	Опис	Пояснення
Команда повного розгалуження	<pre> if <логічний вираз>: блок команд 1 else: блок команд 2 </pre>	Якщо логічний вираз набуває значення <i>True</i> , то буде виконуватись команда або блок команд 1, якщо логічний вираз набуває значення <i>False</i> , то буде виконуватись команда або блок команд 2
Команда вибору	<pre> if <логічний вираз1>: блок команд 1 elif <логічний вираз2>: блок команд 2 ... elif <логічний вираз n>: блок команд n else: блок команд інакше </pre>	Якщо значення логічного виразу 1 <i>True</i> , то виконується команда 1 чи блок команд 1. Якщо значення логічного виразу 1 — <i>False</i> і значення логічного виразу 2 — <i>True</i> , то виконується команда 2 чи блок команд 2 і т.д. Інакше виконується команда інакше або блок команд інакше

Для складання логічних виразів використовують логічні величини та операції над ними.

Окрім операцій порівняння над логічними величинами (>, <, >=, <=, ==, <>), в усіх мовах програмування для запису складених логічних виразів використовують логічні операції:

- `not` (*не*) — заперечення;
- `and` (*і*) — кон'юнкція;
- `or` (*або*) — диз'юнкція (нестрога);
- `xor` (*виключне або*) — диз'юнкція (строга).

Щоб визначити значення складеного логічного виразу, який містить кілька логічних операцій, використовують таблицю істинності та враховують пріоритет виконання операцій: у першу чергу виконується операція `not`, далі — `and`, в останню чергу — `or`. Як і для числових виразів, для зміни порядку виконання логічних операцій використовують дужки.



Діємо

Вправа 1. Таблиця істинності для логічного виразу.

Завдання. Визначте, якого значення може набувати логічний вираз $(\text{not } A) \text{ or } (B \text{ and } A)$ залежно від значень, яких набуватимуть логічні змінні A та B .

1. Обчислимо кількість можливих наборів значень логічних змінних за формулою

$$N = 2^n,$$

де N — кількість наборів значень; n — кількість змінних. У нашому випадку $N = 2^2 = 4$. Це число визначає кількість рядків у таблиці істинності.

2. Визначимо кількість і порядок логічних операцій у виразі.

$$\begin{array}{ccc} 1 & 3 & 2 \\ (\text{not } A) & \text{or} & (B \text{ and } A) \end{array}$$

Ця кількість і кількість логічних змінних визначає кількість стовпців у таблиці істинності. Для нашого випадку: $3 + 2 = 5$.

3. Побудуємо таблицю істинності. Для зручності запису приймемо $\text{True} = 1$, $\text{False} = 0$. Для визначення значень кожної логічної операції використаємо таблицю істинності 17.2.

Таблиця 17.2

A	B	1	2	3
		not A	B and A	1 or 2
1	1	0	1	1
1	0	0	0	0
0	1	1	0	1
0	0	1	0	1

4. Зробимо висновок: логічний вираз набуде значення *False* тільки тоді, коли змінна A набуде значення *True*, а змінна B — значення *False*. В усіх інших випадках логічний вираз матиме значення *True*.

Вправа 2. Слово чи декілька.

Завдання. Складіть у середовищі програмування програму, за допомогою якої можна визначити, чи введений текст складається з одного слова, чи з декількох. Використайте в програмі візуальні елементи.

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Слова_Прізвище* в папці *Навчальні проекти* власної структури папок.
2. Поясніть, які команди із запропонованих ви будете використовувати у програмному коді:

```

import tkinter
import tkinter.messagebox
window = tkinter.Tk()
str_var = tkinter.StringVar()
def button_click():
    s = str_var.get()
    if ' ' in s:
        tkinter.messagebox.showinfo("Результат", "Відповідь1")
    else:
        tkinter.messagebox.showinfo("Результат", "Відповідь2")
label = tkinter.Label(window, text="Введіть текст")
label.pack()
edit = tkinter.Entry(window, textvariable = str_var)
edit.pack()
button = tkinter.Button(window, text="Перевірити",
                        command=button_click)
button.pack()
window.mainloop()

```

3. Використайте потрібні команди та змініть за необхідності їхні параметри. Запишіть програмний код у середовище.
4. Запустіть програму на виконання. Отримайте результат для введеного одного слова та декількох. Завершіть роботу із середовищем.

Вправа 3. Більше із чотирьох.

Завдання. Дано чотири різних цілих числа. Складіть у середовищі програмування програму, за допомогою якої натискуванням кнопки *Визначити* перевіряють, у яке текстове поле введено найбільше число, і зафарбовують це поле в червоний колір.

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Перевірка* в папці *Навчальні проєкти* своєї структури папок.
2. Опишіть команду завантаження модуля *tkinter*. Опишіть змінні *window* — вікно, *var1* — текст із текстового поля 1, ..., *var4* — текст із текстового поля 4, *edit1* — текстове поле 1, ..., *edit4* — текстове поле 4, *button* — кнопка з надписом *Визначити*. Використайте наведені у зразку команди:

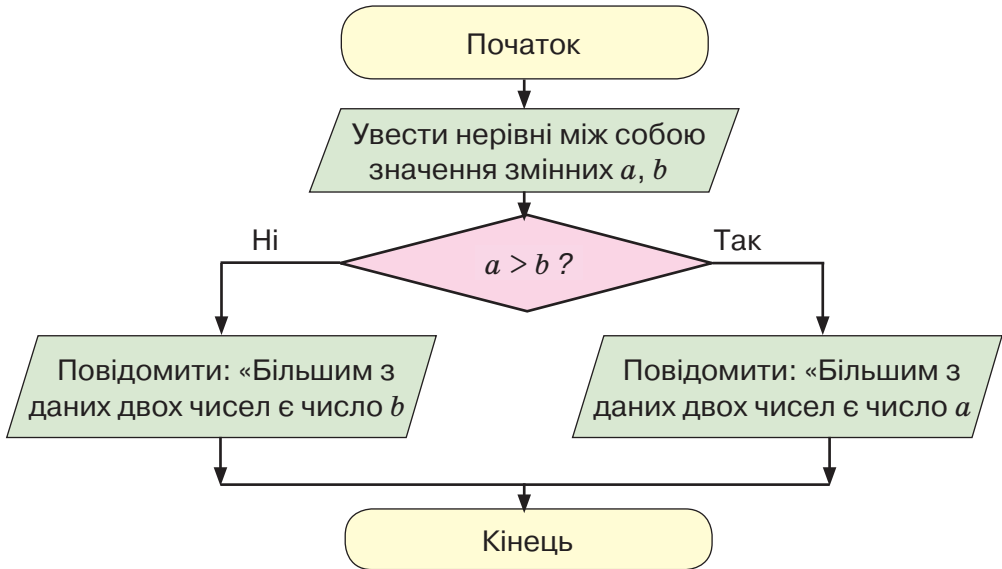

```

window = tkinter.Tk()
var1 = tkinter.StringVar()
edit1 = tkinter.Entry(window, textvariable = var1)
button = tkinter.Button(window, text="Визначити",
                        command=m)

```
3. Розмістіть створені об'єкти у вікні. Використайте наведені у зразку команди:

```
edit1.pack()
button.pack()
```

4. Пригадайте алгоритм порівняння двох чисел. Використайте блок-схему.



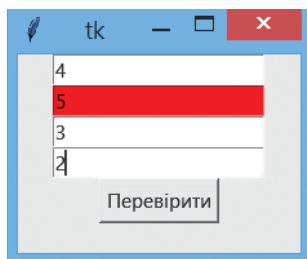
5. Опишіть функцію перевірки m . Використайте для цього ідею порівняння чисел за допомогою числового променя: серед множини цілих чисел найбільшим буде те, яке на числовому промені правіше. Перевірте, чи використана ця ідея в програмі, та поясніть інші команди:

```
def m():
    v1=int(var1.get())
    v2=int(var2.get())
    v3=int(var3.get())
    v4=int(var4.get())
    if v1 > v2 > v3 > v4:
        edit1['bg'] = 'red'
    elif v2 > v1 > v3 > v4:
        edit2['bg'] = 'red'
    elif v3 > v1 > v2 > v4:
        edit3['bg'] = 'red'
    else:
        edit4['bg'] = 'red'
```

6. Перевірте, чи зміниться результат роботи програми, якщо для визначення найбільшого використати принцип зважування: визначити більші в парах, а потім порівняти більші з пар. Для цього в умові використати логічні елементи, наприклад:

```
if (v1 > v2) and (v3 > v4) and (v1>v3):
    edit1['bg'] = 'red'
```

7. Завершіть програму командою запуску подій у вікні `window.mainloop()`
8. Запустіть програму на виконання. Введіть у текстові поля значення, наприклад, як на малюнку 17.1:



Мал. 17.1



Досліджуємо

Вправа 4. Менше з трьох.

Завдання. Дослідіть, які зміни потрібно внести в програму *Перевірка*, щоб у результаті у вікні повідомлення отримати повідомлення про те, яке з трьох введених у текстові поля чисел найменше.

1. Відкрийте середовище програмування. Відкрийте файл *Перевірка* з папки *Навчальні проекти* своєї структури папок. Збережіть його там само з новим іменем *Менше з трьох*.
2. Вносьте правки в програму згідно з умовою. Врахуйте, що в команді розгалуження потрібно отримати деяке *min_chislo*, яке після всіх перевірок слід вивести у вікні повідомлення.
3. Визначте, чи можна використати описані команди розгалуження в тілі функції:

```
if v1 < v2:
    min_chislo=v1
else:
    min_chislo=v2
if v3 < min_chislo:
    min_chislo=v3
```

Поясніть, як відбувається визначення найменшого.

4. Доберіть самостійно дані, які потрібно внести в текстові поля, щоб перевірити складений алгоритм.
5. Запустіть програму на виконання. За потреби внесіть правки.
2. **Як за допомогою елементів управління задати логічне значення?**

У програмах з графічним інтерфейсом для роботи з логічними змінними використовують елементи управління прапорці і перемикач.

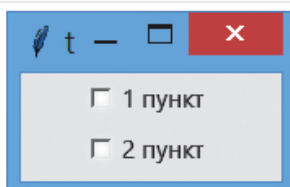
Checkbutton, або прапорець, — елемент списку з місцем для мітки. Значення кожної мітки (прапорця) пов'язане зі своєю змінною. Значення цієї змінної визначається властивістю `onvalue` (ввімкнено) і `offvalue` (вимкнено). При використанні кількох елементів списку потрібно для кожного використати свою змінну. Наприклад, у програмі (мал. 17.2) у вікні отримуємо два прапорці з підписами *1 пункт*, *2 пункт* (мал. 17.3).

```

1 import tkinter
2 window = tkinter.Tk ()
3 var1 = tkinter.IntVar ()
4 var2 = tkinter.IntVar ()
5 check1 = tkinter.Checkbutton (window,
6                               text = '1 пункт ',
7                               variable = var1,
8                               onvalue = 1, offvalue = 0)
9 check2 = tkinter.Checkbutton (window,
10                              text = '2 пункт ',
11                              variable = var2,
12                              onvalue = 1, offvalue = 0)
13 check1.pack ()
14 check2.pack ()
15 window.mainloop ()

```

Мал. 17.2



Мал. 17.3

В описі прапорця `tkinter.Checkbutton` використано параметри:

- *window*: ім'я елемента керування, на якому розташовано прапорець;
- *onvalue*: значення, яке пов'язане з прапорцем змінною `IntVar`, в позначеному стані за замовчуванням дорівнює 1;
- *offvalue*: значення, яке пов'язане з прапорцем змінною `IntVar`, в непозначеному стані за замовчуванням дорівнює 0;
- *variable*: посилання на змінну, як правило, типу `IntVar`, яка зберігає стан прапорця;
- *text*: текст елемента.

Окрім вказаних параметрів можуть бути й ті, які описані для інших елементів керування: висота, ширина, колір тексту, фону, межі, відступи та інші.

Radiobutton, або перемикач, — елемент списку з місцем для мітки за умови можливості виставити лише одну мітку. Перемикач завжди використовують у групі, причому увімкненим може бути лише один перемикач. Реалізація цього елемента керування відрізняється від реалізації для `Checkbutton` (мал. 17.4).

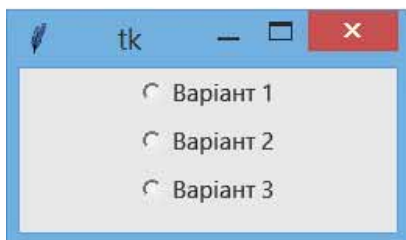
```

1 import tkinter
2 window = tkinter.Tk ()
3 v = tkinter.IntVar ()
4 rbutton1 = tkinter.Radiobutton (window, text='Варіант 1',
5                                 variable=v, value=1)
6 rbutton2 = tkinter.Radiobutton (window, text='Варіант 2',
7                                 variable=v, value=2)
8 rbutton3 = tkinter.Radiobutton (window, text='Варіант 3',
9                                 variable=v, value=3)
10 rbutton1.pack ()
11 rbutton2.pack ()
12 rbutton3.pack ()
13 window.mainloop ()

```

Мал. 17.4

У результаті виконання програми на екрані отримуємо вікно із трьома перемикачами: *Варіант 1*, *Варіант 2*, *Варіант 3* (мал. 17.5).



Мал. 17.5

Тут:

- *variable* — властивість, що описує прикріплення до елемента керування змінної;
- *onvalue*, *offvalue* — властивості, які надають прикріплений до елемента керування змінній значення залежно від стану:
 - *onvalue* — при вибраному пункті;
 - *offvalue* — при невибраному пункті.

У цьому прикладі використано лише одну змінну. Залежно від того, який пункт обрано, вона змінює своє значення. Найцікавіше, що якщо надати цій змінній якесь значення, то зміниться і сам елемент керування.

Якщо потрібно отримати дані про те, яка кнопка увімкнута, використовують метод `get()`, щоб надати значення — метод `set()`.

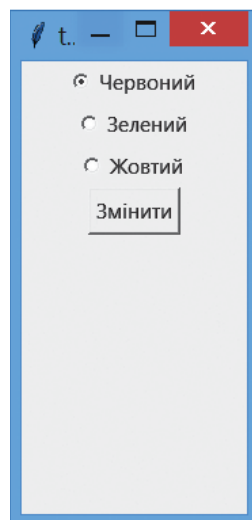


Діємо

Вправа 3. Світлофор.

Завдання. Складіть у середовищі програмування програму, за допомогою якої обирають позначку з підписом кольору, що супроводжується зафарбовуванням деякої області цього вікна у вибраний колір. Використайте в програмі візуальні елементи — позначки, напис, кнопку (мал. 17.6).

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Світлофор_Прізвище* в папці *Навчальні проекти* своєї структури папок.
2. Поясніть команди у програмному коді (мал. 17.7). Зверніть увагу, що якщо у програму завантажено всю бібліотеку модуля (`from tkinter import *`), то перед методами не потрібно вказувати назву бібліотеки `tkinter`. Окрім цього, якщо у програмі використовується одне вікно, то назву вікна в параметрах елементів керування теж можна не вказувати — вона обирається автоматично.
3. Введіть запропоновані команди. Додайте команди розміщення елементів керування у вікні за зразком (мал. 17.6).
4. Запустіть програму на виконання. Переконайтеся, що вона виконує завдання.



Мал. 17.6

```

1 from tkinter import *
2 def change():
3     if var.get() == 0:
4         label['bg'] = 'red'
5     elif var.get() == 1:
6         label['bg'] = 'green'
7     elif var.get() == 2:
8         label['bg'] = 'yellow'
9 window = Tk()
10 var = IntVar()
11 var.set(0)
12 red = Radiobutton(text="Червоний",
13                  variable=var, value=0)
14 green = Radiobutton(text="Зелений",
15                    variable=var, value=1)
16 yellow = Radiobutton(text="Жовтий",
17                      variable=var, value=2)
18 button = Button(text="Змінити",
19                 command=change)
20 label = Label(width=20, height=10)

```

Мал. 17.7



Працюємо в парях

1. Перевірте одне в одного знання правил опису команд неповного й повного розгалуження та вкладених розгалужень. Використайте для цього вирази типу:
 $y = x + 10$, якщо $x > 1$

$$y = \begin{cases} x^2, & \text{якщо } x < 0 \\ \sqrt{x}, & \text{якщо } x \geq 0 \end{cases}$$

$$\text{кут} = \begin{cases} \text{гострий, якщо градусна міра} < 90 \\ \text{прямий, якщо градусна міра} = 90 \\ \text{тупий, якщо градусна міра} > 90 \end{cases}$$

- ★ 2. Обговоріть способи визначення найбільшого та найменшого серед декількох чисел. Який з них ви порекомендували б застосовувати в кожному з випадків: більше/менше з трьох чисел, чотирьох і п'яти чисел? Складіть відповідні програми та порівняйте їх результати й тексти.
3. Обговоріть, чим відрізняється в мові програмування *Python* створення елемента керування прапорець від елемента керування перемикач. Наведіть два міркування.
- ★ 4. Попрацюйте в мініпроєкті. Обговоріть ідею створення програми з використанням елементів керування прапорець або кнопки, напису, текстового поля, яка допомагатиме у вивченні мови, літератури, історії (на вибір). Створіть програму та продемонструйте її у класі.



Працюємо самостійно

5. Складіть програму *Округлення*, що у вікно повідомлення виводить округлене число до знака, яке обирається за допомогою елемента керування на вікні (оберіть доцільний елемент): десяті, соті, тисячні. Для числа, яке введено в текстове поле, результат виводиться у вікно повідомлення.
6. Складіть програму *Улюблений колір*, у якій серед п'яти кольорів, описаних прапорцями, користувач обирає улюблені. На екрані це відображається прямокутниками відповідного кольору.
- ★ 7. Складіть програму *Відмінювання іменників*, що за обраним відмінком у елементі керування на вікні виводить у надпис запитання, на які відповідають іменники.
- ★ 8. Складіть програму *Коло*, яка за введеним у текстове поле значенням радіусу у вікно повідомлення виводить результат: довжина кола або площа круга. Значення, які потрібно обчислити, обираються за допомогою елементів управління.
- ★ 9. Складіть програму *Калькулятор теореми Піфагора*. Відомо, що квадрат гіпотенузи (сторона, протилежна прямому куту), дорівнює сумі квадратів двох інших сторін (катетів). Користувач обиратиме у вікні програми, яку сторону він буде обчислювати, вводитиме у два текстові поля значення інших сторін, а у вікні повідомлення отримуватиме результат.



Цикли мовою програмування *Python*

ПРИГАДАЙТЕ:

- ▶ що таке алгоритмічна структура повторення;
- ▶ як реалізують алгоритми повторення мовою програмування *Python*.

ВИ ДІЗНАЄТЕСЬ:

- ▶ які команди використовують для опису циклів мовою програмування *Python*;
- ▶ що таке вкладені цикли;
- ▶ як створити шкалу вибору.



Вивчаємо

1. Які команди використовують для опису циклів мовою програмування *Python*?

Ми вже складали найпростіші програми для реалізації алгоритмів з розгалуженням мовою програмування *Python*. Для цього використовували такі команди (табл. 18.1).

Таблиця 18.1

Опис		Приклад
<pre>while <логічний вираз>: <команда></pre>	Використовують в алгоритмах, коли кількість повторень не відома, а визначається логічним виразом	<pre>while a<5: a+=1</pre>
<pre>while <логічний вираз>: <команда1> <команда2> . . . <команда n> else: <команда інакше></pre>		<pre>while a<5: s=s+a a+=1 else: print('Сума не змінюється')</pre>

Продовження таблиці 18.1

Опис	Приклад
<code>for i in <діапазон>: <команда></code>	Використовують, коли кількість повторень наперед визначена <code>for i in range(5): a+=1</code>
<code>break</code>	Використовують, коли потрібно перервати виконання циклу <code>for i in 'hello world': if i == 'o': break print(i * 2, end='')</code>
<code>continue</code>	Використовують, коли потрібно продовжити виконання циклу <code>for i in 'hello world': if i == 'o': continue print(i * 2, end='')</code>

Наприклад, у програмі, яка використовує модуль малювання, побудовано спіраль червоного кольору з товщиною 2 пікселі (мал. 18.1).

```
1 from turtle import *
2 color('red')
3 width(2)
4 for x in range(1,100,4):
5     forward(x)
6     left(90)
```



Мал. 18.1

Команду `range()` використовують, щоб сформувати діапазон. За параметр вона приймає ціле число, і повертає послідовність цілих чисел від 0 до вказаного в аргументі числа, але не включаючи його. Якщо використати команду `range(n,m,k)`, то отримаємо діапазон чисел від n до включно $(m - 1)$ з кроком k .

Команди циклу використовують і у програмах, складених мовою програмування *Python*, які використовують графічний інтерфейс.



Діємо

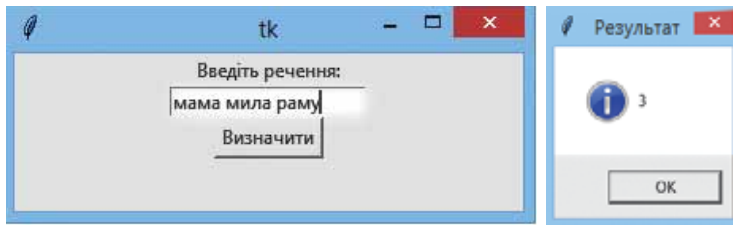
Вправа 1. Кількість слів у реченні.

Завдання. Розробіть у середовищі програмування проєкт, за допомогою якого можна визначити, скільки слів уведено в текстове поле, якщо відомо, що між словами міститься лише один пропуск.

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Кількість_слів* у папці *Навчальні проєкти* своєї структури папок.

- Імпортуйте об'єкти, які потрібні для опису графічного інтерфейсу програми (мал. 18.2).

Мал. 18.2



- Оголосіть імена величин.

```
window = tkinter.Tk() #головне вікно
str_var = tkinter.StringVar() #текст, який вводять у
текстове поле
```

- Запишіть програмний код опрацювання події натиснення на кнопку *Визначити*.

```
def button_click():
    s = 0
    z = ' '
    st = str_var.get()
    k = len(st)
    for i in range(0, k):
        if z == st[i]:
            s = s + 1
    messagebox.showinfo("Результат", str(s+1))
```

- Запишіть у програмному коді команди для створення об'єктів у вікні *window* та отримання значень змінних.
- Запишіть команду запуску подій на формі
- Запустіть проект на виконання. Перевірте, чи результат для введеного речення відповідає малюнку 18.2.
- Завершіть роботу з проектом і середовищем.

2. Що таке вкладені цикли?

У програмах, складених мовою програмування *Python*, можна використовувати алгоритмічні конструкції, вкладені одні в одні. Наприклад, у командах розгалуження можна використовувати деякі циклічні розрахунки, або навпаки, під час виконання циклу виконувати блоки команд, які викликаються в командах розгалуження. Можуть бути також цикли, які виконуються в тілі інших циклів, або так званих **вкладених циклів**.

Розглянемо, наприклад, задачу про покупку з обмеженим бюджетом. Є 1000 грн. Скільки упаковок цукерок, вафель і печива можна купити на всі ці гроші, якщо вартість упаковки цукерок — 100 грн, вафель — 50 грн, печива — 10 грн. Обов'язково потрібно купити щонайменше по одній одиниці товару.

Фрагмент розв'язку матиме такий вигляд.

```

1. x=1
2. while x<=9:
3.     y=1
4.     while y<=19:
5.         z=1
6.         while z<=99:
7.             cost=100*x+50*y+10*z
8.             if cost==1000:
9.                 print("x=", x, "; y=", y, "z=", z)
10.                z=z+1
11.            y=y+1
12.        x=x+1

```

Цикл у рядках 6–10 містить обчислення суми, перевірку умови та цикл підрахунку кількості упаковок печива. Цикл у рядках 4–11, у свою чергу, містить цикл підрахунку кількості печива та вафель. Цикл 2–12 — містить цикли підрахунку кількості цукерок, вафель та печива.

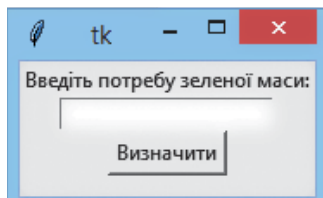


Діємо

Вправа 2. Урожай.

Завдання. Фермер вирощує новий сорт рослин для відгодівлі тварин. Він дослідив закономірність, що після кожного зрізу, який здійснюється через тиждень не більше 5 разів, кількість зеленої маси збільшується за формулою: $s=s+(i-1)*i$, де i — номер тижня, s — кількість зеленої маси. Розробіть проєкт у середовищі програмування, за яким буде визначено, чи встигне фермер зібрати n одиниць зеленої маси.

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Урожай* у папці *Навчальні проєкти* своєї структури папок.
2. Імпортуйте об'єкти, які потрібні для опису графічного інтерфейсу програми (мал. 18.3).



Мал. 18.3

```
import tkinter
import tkinter.messagebox
```

- 3.** Опишіть об'єкти, які ви будете використовувати у проєкті.

```
window = tkinter.Tk()
str_var = tkinter.StringVar()
```

- 4.** Запишіть програмний код опрацювання події натиснення кнопки *Визначити*.

```
def button_click():
    s = 1
    i=1

    st = str_var.get()
    k = int(st)
    while i <= 10:
        s += (i-1) * i
        i+=2

    if k <= s:
        tkinter.messagebox.showinfo("Результат",
            'Встигне отримати ' + str(s))
    else:
        tkinter.messagebox.showinfo("Результат", 'Не
            встигне отримати ')
```

- 5.** Запишіть програмний код розміщення об'єктів на головній формі.

```
label = tkinter.Label(window, text='Введіть потре-
бу зеленої маси:')
label.pack()
edit = tkinter.Entry(window, textvariable=str_var)
edit.pack()
```

```
button = tkinter.Button(window, text="Визначити",
command=button_click)
button.pack()
```

- 6.** Запишіть програмний код запуску подій головної форми.

```
window.mainloop()
```

- 7.** Запустіть проєкт на виконання. Перевірте, чи для значення 50 ви отримаєте повідомлення «Встигне отримати 121».

- 8.** Завершіть роботу з проєктом і середовищем.



Досліджуємо

Вправа 3. Фігури.

Завдання. Дослідіть, як зміниться зображення на екрані, якщо у програмі побудови спіралі:

- 1) замість команди `left(90)` обрати `left(92)`;
 - 2) замість команди `forward(x)` обрати `circle()`;
 - 3) виконати заміну 1) та 2);
 - 4) замінити команду `for` на відповідну команду `while`.
1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Фігури_Прізвище* в папці *Навчальні проекти* своєї структури папок.
 2. Введіть програму побудови спіралі (мал. 18.1). Запустіть її на виконання.
 3. По черзі вносьте запропоновані зміни 1), 2) і 3) та переглядайте результати виконання програми. Зробіть висновки.
 4. Поміркуйте, що важливо додати до тіла циклу `while`, щоб отримати потрібні зображення.
 5. Запустіть програму на виконання та зробіть висновок.

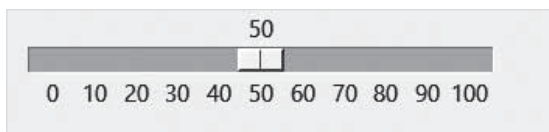
3. Як створити шкалу вибору?

У вікні проекту можна використати елемент керування — шкала, за допомогою якої користувач може обрати значення з певного діапазону.

Шкала — це горизонтальна чи вертикальна смужка з розміткою, по якій можна рухати повзунком, здійснюючи вибір.

Наприклад, горизонтальна та вертикальна шкала задається командами:

```
scale = tkinter.Scale(window, orient=HORIZONTAL,
length=300, from_=0, to=100, tickinterval=10, resolution=5)
```



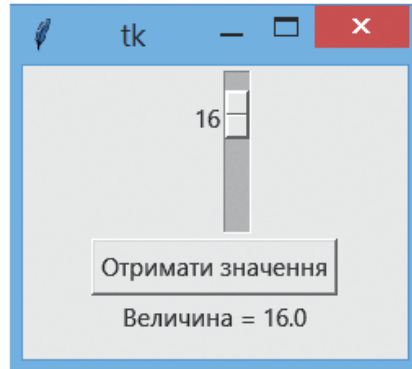
```
scale = tkinter.Scale(window,
                        orient=VERTICAL,
                        length=400,
                        from_=1, to=2,
                        tickinterval=0.1,
                        resolution=0.1)
```



Властивості:

- `orient` — визначає напрям;
- `length` — довжина шкали в пікселях;
- `from_`, `to_` — з якого значення шкала починається і яким закінчується (так званий діапазон значень);
- `tickinterval` — інтервал, через який відображаються мітки на шкалі;
- `anchor` — розташування на екрані;
- `resolution` — мінімальна довжина відрізка, на яку можна пересунути повзунок;
- `variable` — змінна, яка містить або надає значення.

Щоб отримати значення зі шкали, використовують метод `get()`. Наприклад, нехай у вікні, що містить вертикальну шкалу за замовчуванням (від 0 до 100, з кроком 1), кнопку *Отримати значення*, напис *Величина*, після натиснення кнопки буде додано значення зі шкали (мал. 18.4).



Мал. 18.4

Тоді програма матиме вигляд (мал. 18.5).

```

1 from tkinter import *
2
3 def sel():
4     selection = "Величина = " + str(var.get())
5     label.config(text = selection)
6
7 window = Tk()
8 var = DoubleVar()
9 scala = Scale( window, variable = var )
10 scala.pack()
11
12 button = Button(window, text="Отримати значення", command=sel)
13 button.pack()
14
15 label = Label(window)
16 label.pack()
17
18 window.mainloop()

```

Мал. 18.5

Працюємо в парах

1. Обговоріть, чому в мові *Python* використовують різні команди циклу. У чому їх відмінність?
2. Поясніть одне одному ситуації, коли при розв'язуванні задач потрібно використовувати вкладені алгоритмічні конструкції. Наведіть приклади.
- ★ 3. Обговоріть у парі ідею розв'язування задачі та які графічні елементи можна використати на головному вікні проєкту. Розробіть проєкт.
 - а) Проєкт *Дії*, у якому у введеному виразі, що містить арифметичні операції без дужок та математичних функцій, у повідомленні буде виводитись, які дії потрібно виконати.
 - ★ б) Проєкт *Найменший спільний дільник*, за допомогою якого можна визначити найменший спільний дільник двох цілих чисел.
 - ★ в) Проєкт *Просте число*, за допомогою якого буде перевірятись, чи є простим число введене в текстове поле ціле значення. Відомо, що число називають простим, якщо воно ділиться без остачі тільки на одиницю і на себе.
 - ★★ г) Проєкт *Щасливий квиток*, за допомогою якого буде визначатись, чи відповідає введений номер квитка, що містить латинські літери та цифри, критерію «щасливий». Квиток вважається «щасливим», якщо число номера ділиться на суму його цифр.
 - ★★ д) Гра *Відгадай число*, яка відбувається між комп'ютером і користувачем. Користувач не більше ніж за 6 спроб має відгадати «задумане» число з діапазону від 1 до 20. Якщо користувач відгадає, то на екран виводиться повідомлення про кількість спроб, якщо ні — «задумане число».

Працюємо самостійно

4. Розробіть проєкт у середовищі програмування з використанням елементів керування, у якому знаходять:
 - а) суму послідовних чисел від 0 до числа, обраного на шкалі (розміри шкали за замовчуванням);
 - ★ б) добуток парних чисел від 2 до числа, обраного на шкалі (максимальне значення на шкалі — 10);
 - ★ в) суму чисел, які кратні 5, від 1 до числа, обраного на шкалі (розміри шкали за замовчуванням);
 - ★★ г) кількість чисел, які діляться на 3 від введеного в текстове поле числа до числа, обраного на шкалі (розміри шкали за замовчуванням). Якщо обране на шкалі число менше від введеного початкового значення, то отримати повідомлення про помилку.



Графічне відображення даних мовами програмування

ПРИГАДАЙТЕ:

- ▶ які інструменти малювання використовують у середовищі графічного редактора;
- ▶ як створюють і змінюють формат малюнків у середовищі текстового процесора;
- ▶ чим відрізняються малюнки, створені у векторному та растровому графічному редакторах;
- ▶ засоби для малювання в навчальному середовищі створення та виконання алгоритмів *Скретч*.

ВИ ДІЗНАЄТЕСЬ:

- ▶ як у мові програмування *Python* створюють полотно для малювання;
- ▶ які геометричні примітиви можна створювати за допомогою модуля *Canvas*;
- ▶ як додавати малюнки та підписи до малюнків.



Вивчаємо

1. Як у мові програмування *Python* створюють полотно для малювання?

У програмах часто використовують малюнки: ілюстрації, рухомі зображення, фони тощо. Деякі середовища програмування мають засоби, які забезпечують додавання готових зображень до програмного коду або створення і форматування малюнків у самій програмі. Наприклад, у навчальному середовищі створення та виконання алгоритмів *Скретч* ви використовували такі засоби для роботи з графікою:

- змінювали образи об'єктів, завантажуючи їх із бібліотеки;
- малювали об'єкти у вбудованому графічному редакторі;
- виконували побудову зображень виконавцем, вказуючи у програмному коді команди групи *Олівець*.

У мові програмування *Python* ви вже використовували модуль побудови зображень *turtle*. У вікні побудови можна було виконувати різні зображення у вигляді піксельної графіки.

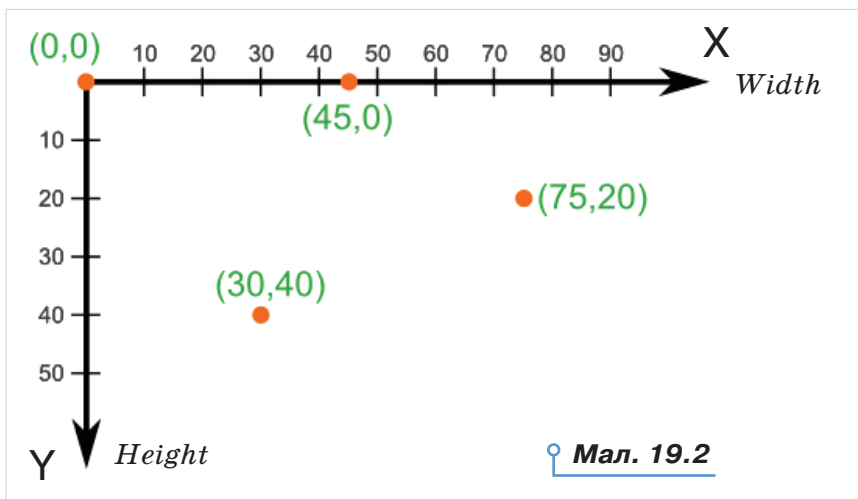
Для побудови графічних об'єктів, що складаються з геометричних примітивів і готових малюнків, викликають відповідний метод модуля *tkinter*, який імпортують до проекту (мал. 19.1).

Створення полотна

Мал. 19.1

`tkinter.Canvas(<назва форми>, width=<значення>, height=<значення>)`

Розмір полотна визначається значеннями властивостей `Height` — кількість точок за вертикаллю, та `Width` — за горизонталлю. Полотно складається з окремих точок — пікселів, координати яких задаються значеннями x та y (мал. 19.2).



Мал. 19.2

2. Які геометричні примітиви можна створювати за допомогою модуля *Canvas*?

Створення зображень на полотні викликається методом:

`<Ім'я_об'єкту_полотно>.create_<об'єкт>.`

Можна побудувати такі об'єкти:

- лінія — `line(x1, y1, x2, y2)`
- прямокутник — `rectangle(x1, y1, x2, y2)`
- багатокутник — `polygon(x1, y1, x2, y2, x3, y3, x4, y4)`
- ламана — `polyline(x1, y1, x2, y2, x3, y3)`
- еліпс — `oval(x1, y1, x2, y2)`
- дуга — `arc(x1, y1, x2, y2, x3, y3, x4, y4).`

Об'єкти можуть мати параметри: `fill` — колір заливки, `dash` — тип заливки, `width` — ширина лінії та інші.



ДІЄМО

Вправа 1. Стрілка.

Завдання. Створіть малюнок стрілки за зразком (мал. 19.3).

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Стрілка_Прізвище* в папці *Навчальні проекти* своєї структури папок.



Мал. 19.3

2. Введіть програму побудови стрілки (мал. 19.4).

```

1 from tkinter import *
2 window = Tk()
3 c = Canvas(window, width=100, height=100, bg='white')
4 c.pack()
5 c.create_line( 10,10,  90,10)          #малювання горизонтальної лінії
6 c.create_line((50,20),(50,90),        #малювання вертикальної лінії
7             fill='green',              #колір лінії
8             width=3,                   #товщина лінії
9             arrow=FIRST,               #напрям стрілки: FIRST - вгору, LAST - донизу
10            dash=(10,2),               #тип штриховки
11            activefill='red',          #колір при наведенні
12            arrowshape="10 25 10")     #вигляд стрілки
13 window.mainloop()

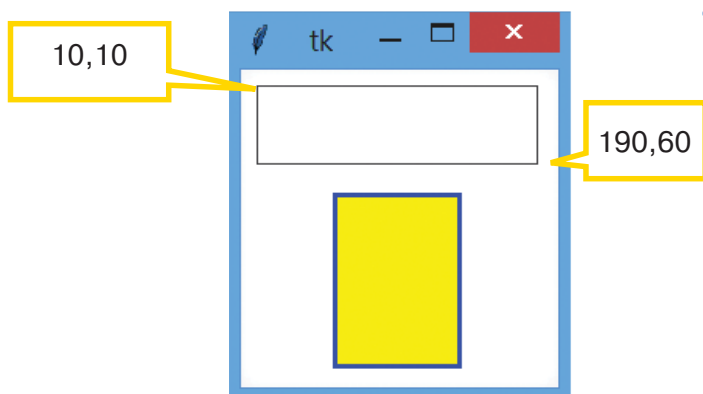
```

Мал. 19.4

3. Запустіть програму на виконання. Перевірте, чи відповідає результат зразку. За потреби виправте помилки.
4. Проекспериментуйте з описаними в коментарях властивостями. Як їх зміна впливає на отриманий результат?

Вправа 2. Прямокутники.

Завдання. Створіть малюнок двох прямокутників за зразком (мал. 19.5).



Мал. 19.5

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Прмокутники_Прізвище* в папці *Навчальні проекти* своєї структури папок.

2. Введіть програму побудови жовтого прямокутника (мал. 19.6).

```

1 from tkinter import *
2 window = Tk()
3 c = Canvas(window, width=200, height=200, bg='white')
4 c.pack()
5
6 c.create_rectangle(60, 80, 140, 190,      # координати діагоналі
7                   fill='yellow',           # колір заповнення
8                   outline='blue',          # колір межі
9                   width=3)                 # ширина межі
10
11 window.mainloop()

```

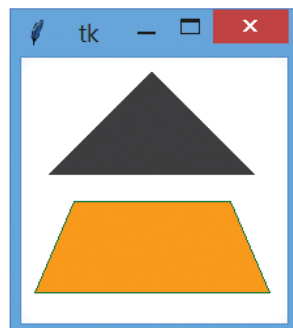
 Мал. 19.6

3. Доповніть програму командою побудови верхнього прямокутника з координатами діагоналі побудови (10, 10, 190, 60).
4. Запустіть програму на виконання. Перевірте, чи відповідає результат зразку. За потреби виправте помилки.

Вправа 3. Багатокутники.

Завдання. Створіть малюнок геометричних фігур за зразком (мал. 19.7).

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Багатокутники_Прізвище* в папці *Навчальні проекти* своєї структури папок.
2. Введіть програму побудови вікна, розміщення об'єктів. Доповніть її командами побудови багатокутників (мал. 19.8).



 Мал. 19.7

```

5 c.create_polygon(100, 10, 20, 90, 180, 90)
6
7 c.create_polygon(40, 110, 160, 110,
8                 190, 180, 10, 180,
9                 fill='orange', outline='green')

```

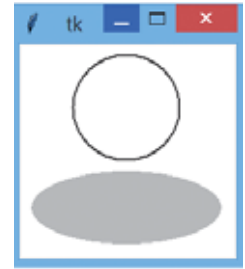
 Мал. 19.8

3. Запустіть програму на виконання. Перевірте, чи відповідає результат зразку. За потреби виправте помилки.
4. Внесіть зміни у програму так, щоб усі багатокутники були оранжевого кольору із зеленим контуром.

Вправа 4. Тінь.

Завдання. Створіть малюнок за зразком (мал. 19.9).

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Тінь_Прізвище* в папці *Навчальні проекти* своєї структури папок.
2. Введіть програму побудови вікна, розміщення об'єктів. Доповніть її командами побудови овалів (мал. 19.10). Врахуйте, що овал — це фігура, вписана в уявний прямокутник, що визначається діагоналлю: ліва верхня точка (50, 10), нижня права — (150, 100).



Мал. 19.9

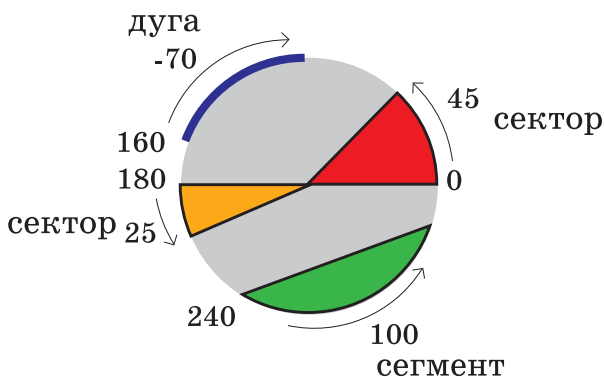
Мал. 19.10

```
c.create_oval(50, 10, 150, 110, width=2)
c.create_oval(10, 120, 190, 190,
              fill='grey80', outline='white')
```

Насиченість заповнення фігури сірим кольором: від звичайного сірого (за замовчуванням 0), до практично білого (100)

Вправа 5. Елементи круга.

Завдання. Використовуючи модуль *Canvas*, створіть зображення за зразком (мал. 19.11).



Мал. 19.11

1. Відкрийте середовище програмування. Створіть новий файл програми мовою *Python* з іменем *Елементи круга_Прізвище* в папці *Навчальні проекти* своєї структури папок.
2. Введіть програму побудови вікна, розміщення об'єктів. Доповніть її командами побудови овалів.
3. Введіть програму побудови вікна, розміщення об'єктів.
4. Доповніть її кодом побудови овалів і дуг. За вказаними значеннями властивостей визначте, які складові круга малює кожний фрагмент побудови (мал. 19.12).

```

c.create_oval(10, 10, 190, 190,
              fill='lightgrey',
              outline='white')
c.create_arc(10, 10, 190, 190,
              start=0, extent=45,
              fill='red')
c.create_arc(10, 10, 190, 190,
              start=180, extent=25,
              fill='orange')
c.create_arc(10, 10, 190, 190,
              start=240, extent=100,
              style=CHORD, fill='green')
c.create_arc(10, 10, 190, 190,
              start=160, extent=-70,
              style=ARC, outline='darkblue',
              width=5)

```

Мал. 19.12

3. Як додавати малюнки та підписи до малюнків?

Зображення на полотні можна доповнювати текстом і готовими малюнками. Для цього використовують об'єкти:

- `text` — текст (<властивості>);
- `window` — вікно (<властивості>);
- `image` — зображення (<властивості>).

Зображення має бути збережене в ту само папку, що і файл програми.

Ці об'єкти мають спільну властивість `anchor`, яку використовують для того, щоб вказати, чим є точка з вказаними координатами для створюваного об'єкта:

- `N` — центр верхнього краю;
- `S` — центр нижнього краю;
- `E` — центр правого краю;
- `W` — центр лівого краю;
- `NE` — верхній правий кут;
- `NW` — верхній лівий кут;
- `SE` — нижній правий кут;
- `SW` — нижній лівий кут;
- `CENTER` — центр.

Наприклад, після виконання програми (мал. 19.13) на екрані відображається вікно з малюнком (мал. 19.14), файл якого збережений у папці разом з файлом програми.

```

1 from tkinter import *
2 window = Tk()
3 c = Canvas(window, width=500, height=300, bg='white')
4 c.pack()
5 c.create_text(100, 200, text="Мій малюнок",
6               justify=CENTER, font="Calibri 24", fill="green")
7
8 f = PhotoImage(file = "1.png")
9 image = c.create_image(100, 150, anchor=SW, image=f)
10 window.mainloop()

```

Мал. 19.13



Мал. 19.14



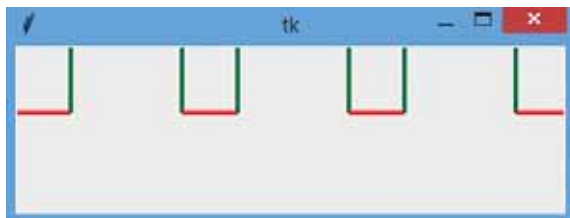
Працюємо в парах

1. Спрогнозуйте, що буде результатом виконання програми (мал. 19.15). Перевірте свій прогноз на практиці. Поясніть, як у програмі обираються кольори. Обговоріть можливі зміни в програмі, які дадуть подібне, але змінене зображення. Реалізуйте таку програму.

```
import random
from tkinter import *
window = Tk()
c= Canvas(window, width=500, height=500)
kolory=["black", "red", "green", "blue", "cyan",
"yellow", "magenta", "azure", "brown", "coral", "grey",
"khaki", "olive", "orange", "lavender", "ivory",
"navy", "orchid", "plum"]
for x in range(0,500,10):
    k=random.choice(kolory)
    c.create_line(x,0,x,500, width=2, fill=k)
    c.pack()
window.mainloop()
```

Мал. 19.15

2. Обговоріть, якими командами слід доповнити програму (мал. 19.17), щоб отримати зображення (мал. 19.16). Створіть відповідний проект і реалізуйте його в середовищі програмування. Продемонструйте свою програму іншій групі. Чи може програмний код відрізнятись?



Мал. 19.16

```
canvas=Canvas(tk,width=500,height=500)
```

```
x=0
```

```
y=60
```

```
a=50
```

```
b=100
```

```
while x<500:
```

```
    canvas.create_line(x,y,x+a,y, width=4, fill='red')
```

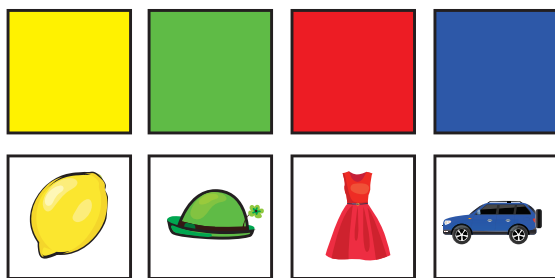
```
    canvas.create_line(x+a,y,x+a,y-b)
```

```
    canvas.create_line(x+a,y-b,x+a+b,y-b)
```

```
    canvas.create_line(x+a+b,y-b,x+a+b,y,width=4)
```

Мал. 19.17

3. Розробіть проєкт *Фігура*, у якому в текстові поля вводять пари координат чисел, інші параметри побудови, обирають одну з геометричних фігур, натискають кнопку *Побудувати* й отримують обрану фігуру, побудовану за вказаними значеннями. Розподіліть між собою обов'язки: хтось проєктує графічний інтерфейс і добирає команди щодо його програмування, хтось — команди для побудови геометричних фігур.
4. Обговоріть ідею проєкту, у якому можна продемонструвати побудову ліній, прямокутників, еліпсів, ламаних, дуг та секторів. Намалюйте ескіз зображення, який ви плануєте реалізувати. Проведіть розрахунок координат. Реалізуйте цей проєкт у середовищі програмування.
5. Обговоріть, які засоби мови програмування *Python* можна використати для реалізації проєкту *Об'єкти та їхні кольори*. На екрані користувачам пропонуються чотири квадрати деякого кольору, під ними — об'єкти, основний колір яких відповідає кольору геометричної фігури (мал. 19.18). При натисканні на квадрат під зображенням з'являється надпис англійською мовою з назвою об'єкта.



Мал. 19.18

Реалізуйте проєкт у середовищі програмування.



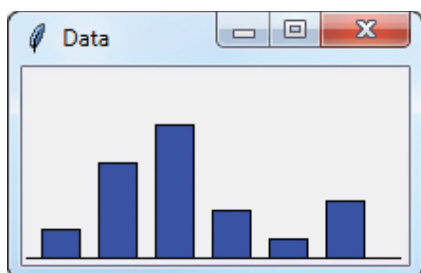
Працюємо самостійно

6. Складіть програму побудови малюнків (мал. 19.19) та реалізуйте її в середовищі програмування.

Мал. 19.19



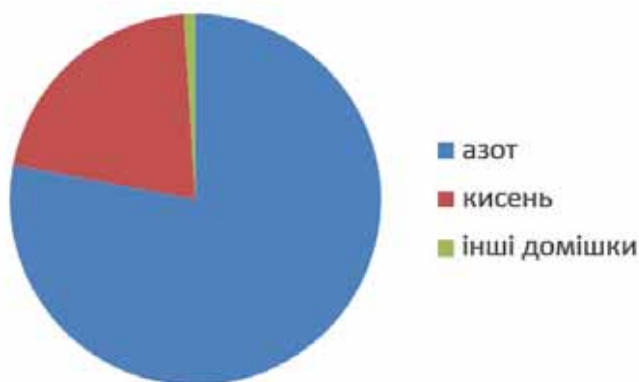
- ★ 7. У середовищі програмування розробіть проект, у якому у вікні головної форми буде побудована стовпчаста діаграма, яка демонструє співвідношення між числами 15, 50, 70, 25, 10, 30 (мал. 19.20).



Мал. 19.20

- ★ 8. У середовищі програмування розробіть проект, у якому у вікні головної форми буде побудована кругова діаграма, яка демонструє склад повітря (мал. 19.21).

Мал. 19.21



- ★★ 9. Створіть проект *Полюси магнітів*, який містить два малюнки магнітів, розташовані один навпроти іншого. Потрібні зображення збережені в папці *Програмування* у файлах *Магніт1*, *Магніт2*. Коли клацнути мишею на лівому малюнку, вони обидва наближаються один до одного, а на правому — віддаляються.
- ★★ 10. Створіть проект *Площа фігур*, вікно якого містить зображення трикутника, прямокутника й кола та три кнопки. Після натиснення кнопки на екрані з'являється формула для обчислення площі фігури.

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Розумію призначення мови програмування та основних її елементів.
- ☐ Можу навести приклади сучасних мов програмування.
- ☐ Умію порівнювати особливості різних середовищ програмування.
- ☐ Знаю відмінність між змінними та константами.
- ☐ Можу пояснити структуру програми.
- ☐ Умію планувати процес розв'язування задачі з використанням програмування.
- ☐ Умію обґрунтувати вибір типів даних для розв'язування задачі.
- ☐ Умію використовувати в програмах вирази, коректно добирати типи даних.
- ☐ Умію створювати й налагоджувати програми.
- ☐ Можу обґрунтувати вибір типів даних для розв'язування задачі.
- ☐ Розпізнаю задачі, для розв'язання яких доцільно використовувати засоби програмування.
- ☐ Розумію та можу пояснити поняття об'єкта в мові програмування, його властивостей і методів.
- ☐ Можу пояснити функції елементів графічного інтерфейсу та користуватися ними.
- ☐ Умію створювати й налагоджувати програми, зокрема подійно-й об'єктно-орієнтовані.
- ☐ Умію розв'язувати задачі з використанням усіх базових алгоритмічних структур, змінних та констант.
- ☐ Умію оцінити відповідність результатів виконання програми поставленій задачі.
- ☐ Можу обґрунтувати вибір типів даних для розв'язування задачі.

Розділ 5

Створення та публікація
вебресурсів

20



Автоматизовані засоби для
створення та публікації вебресурсів.

ПРИГАДАЙТЕ:

- ▶ призначення вебсайту;
- ▶ різновид вебсторінок сайту;
- ▶ складові вебсторінки.

ВИ ДІЗНАЄТЕСЬ:

- ▶ яку структуру можуть мати сайти;
- ▶ які існують вимоги до структури сайту;
- ▶ які відмінності можуть мати вебсторінки сайту;
- ▶ у чому полягає процес створення вебсайту;
- ▶ що передбачає вебдизайн;
- ▶ як створити вебсторінку в мережі.



Вивчаємо

1. Яку структуру можуть мати сайти?

Сукупність вебсторінок певної тематики із системою навігації, що дає можливість переміщуватись між ними за допомогою гіперпосилань, які збережені на одному сервері, утворює сайт, або вебсайт. Під час створення структури сайту слід дотримуватися певних принципів, зокрема принципу простоти та принципу цільової сегментації.



Це цікаво

Вебсайт — від англ. *Website: web* — «павутиння», мережа та *site* — «місце») — дослівно означає «місце в мережі».

Оскільки структура сайту складається з різних його розділів, то для зручності його відвідування та швидкого знаходження потрібних відомостей доцільно робити структуру простою, без додаткових підрозділів.

«Ідеальна» структура сайту має мати один вхід й один вихід. Як це зробити, якщо сайт має безліч сторінок? У такому разі сайт матиме кілька входів і лише один вихід. Для зручності орієнтування головна сторінка сайту має містити меню з повним його описом, яке дасть змогу користувачеві зорієнтуватися у вмісті сайту та швидко перейти до потрібного розділу. Така структура сайту будується за принципом «воронки». Особливо ефективною зазначена структура є при створенні інтернет-магазину, корпоративного сайту та бізнес-сайту, де виходом є сторінка замовлення товару.

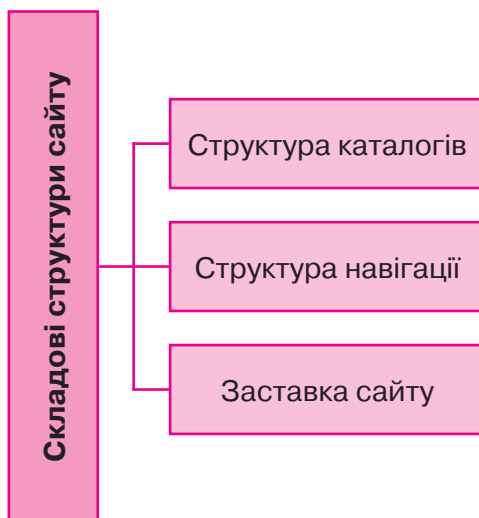
Принцип цільової сегментації базується на тому, що користувач спочатку виділяє основні групи цільової аудиторії і на основі потреб кожної групи створюється структура сайту. Такий принцип побудови є ефективним для сайтів, на яких виділяється 2–3 основні групи користувачів.

Під час планування структури сайту необхідно продумати кілька основних складових: структуру каталогів, структуру навігації, необхідність заставки сайту (мал. 20.1). Важливо визначитися з необхідною кількістю сторінок і встановити зв'язки між ними.

Розглядають **внутрішню** та **зовнішню** структуру сайту. Продумана внутрішня структура сайту дає змогу всі необхідні матеріали розташувати зручно й доступно в розділах і підрозділах. Навіть якщо сайт містить великий обсяг відомостей, слід розробити таку внутрішню структуру, у якій можна легко зорієнтуватися потенційним клієнтам.

Розрізняють лінійну, ієрархічну та довільну внутрішню структуру сайту (мал. 20.1).

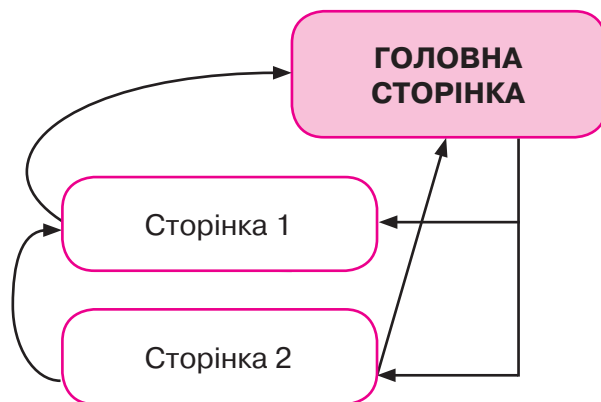
Мал. 20.1



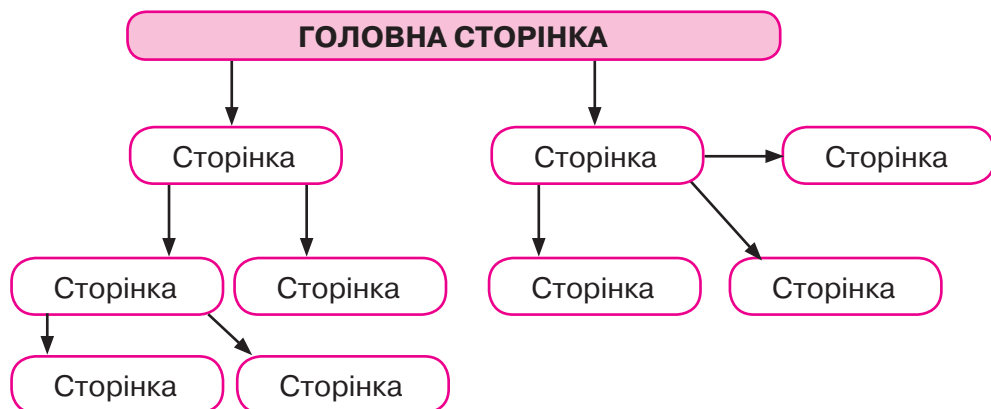
Лінійну (послідовну) структуру вебсайту (мал. 20.2) доцільно використовувати в разі послідовного подання відомостей, наприклад, про товари та послуги або матеріали навчального посібника. Перегляд

таких сайтів здійснюється послідовно: від початкової (головної) до останньої сторінки. Кожна сторінка має посилання тільки на одну, наступну сторінку сайту. Інколи, для зручності навігації по сайту, до сторінки також додається посилання на попередню сторінку.

Мал. 20.2



За ієрархічної, або деревоподібної, структури (мал. 20.3) створюється одна головна сторінка, яка не має попередніх, решта сторінок має лише одну попередню сторінку.

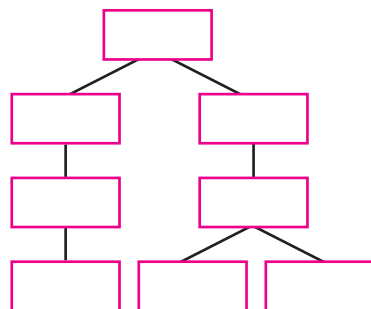


Мал. 20.3

За довільної структури кожна сторінка може містити посилання на довільну кількість сторінок сайту. Така структура найкраще підходить для сайтів, що містять різні за тематикою матеріали: каталогів, зібрань статей з різних тем або добірок посилань. Прикладом такої структури є сайт *Бібліотека української літератури* (www.ukrplib.com.ua).

На практиці майже завжди використовується поєднання деревоподібної та послідовної структур, тобто **гібридна** структура (мал. 20.4).

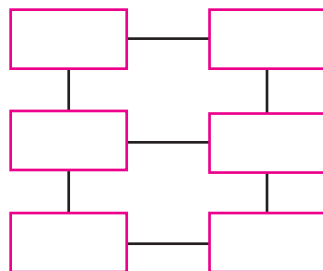
Мал. 20.4



Наприклад, зміст електронної книги або каталог статей починається з ієрархічно розташованих сторінок, але кожний окремий розділ чи статтю доцільно розбити на кілька частин, які розташовуються послідовно одна за одною.

Інколи розглядають **мережеву** структуру (мал. 20.5), яка базується на побудові системи такої навігації сайту, коли між вертикальними й горизонтальними елементами (сторінками) існує взаємний зв'язок і можливість швидкого переходу з однієї сторінки на іншу без додаткової необхідності відвідування проміжних сторінок. Така структура призводить до зайвого збільшення кількості гіперпосилань, а її застосування є обмеженим для сайтів з великою кількістю сторінок.

Мал. 20.5



Зовнішня структура визначає спосіб розміщення відомостей на сторінці в разі вибору кожного пункту меню. Також можна вказати конкретне місце для розміщення банера, картинок, додаткових посилань на сайті. Зовнішня структура має гармонійно відповідати дизайну сайту.



Це цікаво

Банер — це невеличке рекламне зображення, що розміщується на одній сторінці й посилається на іншу.

2. Які існують вимоги до структури сайту?

Найважливішою вимогою до структури сайту є його відповідність **ергономічним вимогам**:

- комфорт під час використання, який полягає у зменшенні фізичної та психологічної втоми;
- безпека, яка передбачає вибір відповідних рішень для захисту користувача.

Ергономічний сайт — сайт, створений з урахуванням і на основі наукових знань про будову й роботу людського ока, що переглядає та збирає (для подальшого аналізу) інформацію із сайту. Ергономічний сайт забезпечує необхідні зручності відвідувачеві, зберігає його сили, здоров'я та працездатність. А це, зрештою, підвищує ефективність сайту й приносить прибуток його власників.

Створення сайту з урахуванням ергономіки може бути визначено як здатність ефективно реагувати на потреби користувачів і забезпечувати їм комфорт під час перегляду сторінки.

Головною перешкодою у створенні сайту з урахуванням ергономіки є різні профілі відвідувачів. Визначальні аспекти, які слід врахувати для побудови ергономічного сайту, такі:

- очікування користувачів: не всі відвідувачі шукають однакову інформацію і не обов'язково мають ті само вимоги в плані графіки;
- звички: тобто набута поведінка;
- вік: загалом характеризує здатність до адаптації та швидкість користувача під час перегляду;
- обладнання: одна з головних перешкод. Відображення сайту може відрізнятися від одного комп'ютера до іншого, зокрема залежно від браузера й роздільної здатності екрана;
- рівень знань: не всі відвідувачі є фахівцями в інформаційних технологіях.

Ергономіка сайту має бути розрахована на найменш досвідченого користувача.

Навігація по сайту має бути простою та зрозумілою: користувач у будь-який час має знати відповіді на такі запитання: *Де він перебуває? Куди можна піти? Як туди дістатися? Як повернутися назад?* Розв'язувати ці задачі можна по-різному. Наприклад, для легкого визначення місця розташування на сайті можна виділити ту сторінку в меню, на якій перебуває користувач, а саме меню зробити доступним на всіх сторінках сайту. Також можна зробити так звані навігаційні ключі, тобто у верхній частині кожної сторінки вказати повний шлях до цієї сторінки за схемою:

Головна сторінка — назва розділу — назва сторінки.

Розрізняють текстову систему навігації, навігацію за допомогою кнопок та використання навігаційних карт.

Інколи замість головної сторінки сайту з'являється заставка. Це велике фото зі звуковим супроводом або відеоролик. Такі сторінки створюють, щоб привернути увагу користувача.

Головна сторінка сайту має задовольняти правило «3 по 3», тобто три абзаци по три речення про тематику та зміст, авторів і призначення сайту. Крім того, вона має відповідати таким вимогам:

- відображати тематику сайту;
- бути цікавою, щоб користувач мав бажання переглянути інші сторінки сайту;
- містити інструмент реалізації пошуку по сайту;
- містити розділ з актуальними даними, що постійно оновлюються, наприклад, новини, акції, поради дня тощо;
- містити відомості про спосіб зворотного зв'язку.

Зовнішній вигляд решти сторінок сайту має бути витриманим у єдиному стилі. На них має бути вказано назву ресурсу й посилання на головну сторінку.

Залежно від засобу створення, сайти можуть мати однакові елементи структури. До них належить логотип власника сайту чи графічний декор, назва сайту, фірми чи сторінки, гіперпосилання чи панель навігації сайту, дані, які наповнюють вміст сторінки сайту, контактні дані тощо.

Більшість сайтів побудовано з використанням **адаптивного дизайну** компоновки вебсторінок, тобто такого, що налаштовується до зміни розмірів екрана. У сучасних сайтах може застосовуватися складне компонування розділів, що поєднує наявність незмінних областей та частин, розмір яких змінюється відповідно до змін розміру екрана. Рідше складові вебсторінки мають чітко встановлені розміри, які відображають сайт саме так, як було задумано дизайнером.



Це цікаво

Вебдизайн — це процес розробки й підтримки роботи вебсайтів.

3. Які відмінності можуть мати вебсторінки сайту?

Вебсторінки, як правило, створюються мовою HTML чи XHTML і можуть містити гіперпосилання для швидкого переходу на інші сторінки. Відомості на вебсторінці можуть подаватися в різних формах:

- текст;
- статичні та анімовані графічні зображення;
- аудіо;
- відео тощо.

Інформаційний вміст вебсторінки, як правило, називають **контентом**.



Це цікаво

Мова HTML (від англ. *Hyper Text Markup Language* — мова розмітки гіпертексту) забезпечує форматування тексту, виділення абзаців, відступів, дає змогу створювати гіпертекстові посилання та розміщувати на сторінці зображення, звукозапис чи інші мультимедійні елементи.

Вебсторінки можуть бути **статичними** та **інтерактивними**.

Статичні вебсторінки створюють за допомогою мови розмітки HTML. Створення нових статичних сторінок або редагування наявних автор виконує на комп'ютері в середовищі HTML-редактора, а потім завантажує на вебсайт. Це трудомісткий процес. Переважно статичні вебсторінки використовують для сайтів з постійною структурою та зовнішнім виглядом сторінки, при цьому кожна сторінка може мати свій шаблон оформлення.

Інтерактивна вебсторінка надає можливість користувачеві вводити деякі дані, а потім забезпечує, відповідно до введених даних, їхнє опрацювання на сервері.

Під час створення інтерактивних вебсторінок використовують спеціальні скрипти мовами сценаріїв *JavaScript* і *VBScript*, які можуть виконуватися за наявності певних дій користувача або під час завантаження вебсторінки.

Сайти, що містять інтерактивні вебсторінки, називають **динамічними**. Для створення та супроводження динамічних сайтів використовують CMS (від. англ. *Content Management System*) — систему управління сайтом, яку називають **двигунцем сайту**.

Засоби розробки сайтів забезпечують відокремлення змістової частини (контенту) від дизайну (шаблону вебсторінки), що дає змогу змінювати вміст вебсторінки, не змінюючи дизайн, і змінювати шаблон сайту, не змінюючи вміст його вебсторінок. Такі двигунці мають багато різних модулів: форуми, гостьові книги, поштові розсилки, контакти, опитувальники, форми реєстрації, форми пошуку, систему обміну повідомленнями між користувачами та інші компоненти, які перетворюють сайт з інформаційного засобу на комунікаційний.

Вебсторінки також мають свої специфічні елементи, що відрізняє їх між собою. Розрізняють такі типи вебсторінок: сторінки розділу вебсайту, форми, домашні сторінки, сторінки форуму та чату, вебкаталоги тощо.

Сторінка розділу вебсайту — це статична вебсторінка, зміст якої користувач лише переглядає і не виконує жодних додаткових дій.

Вебсторінка у вигляді форми по своїй суті є аналогом паперових опитувальників. До форми користувач може записувати свої дані та надсилати їх на сервер, де розміщується вебсторінка й де дані будуть опрацьовані.

Форми зазвичай містять поля для заповнення, списки чи перемикачі для вибору одного чи декількох елементів. Форми застосовують для отримання відомостей про користувачів сайту, їхніх думок щодо питання, винесеного для обговорення, виявлення споживчих переваг тощо. До найуживаніших форм належать: форма пошуку відомостей на сайті, форма для ідентифікації користувача (введення логіна та пароля), зворотний зв'язок, каталог посилань, гостьова книга тощо.

Поняття **домашня сторінка** має декілька значень:

- головна вебсторінка деякого сайту, на яку користувач потрапляє, коли вводить своє доменне ім'я у вікні браузера. Така сторінка містить елементи для переходу на інші сторінки сайту;

- вебсайт, що належить окремій людині. Спочатку цей термін (від англ. *Home Page*) означав дім людини в інтернеті, місце, де вона зберігає власні дані. Тепер такі сторінки містять, окрім відомостей про власника, і корисні дані з певної тематики, зображення, повідомлення чи цікаві факти;

- вебсторінка (стартова), яка першою завантажується в браузері. Стартова сторінка — сторінка, яка завантажується до вікна браузера за замовчуванням при кожному запуску браузера на певному комп'ютері або при кожному натисненні кнопки *Додому*. Як правило, за стартову обирають одну зі спеціалізованих сторінок, які містять посилання, що використовуються конкретним користувачем найчастіше, зокрема: вебпошту, новини, каталоги ресурсів, а також популярні сервіси, такі як пошук в інтернеті, перевірка на віруси, онлайн-переклад тощо.

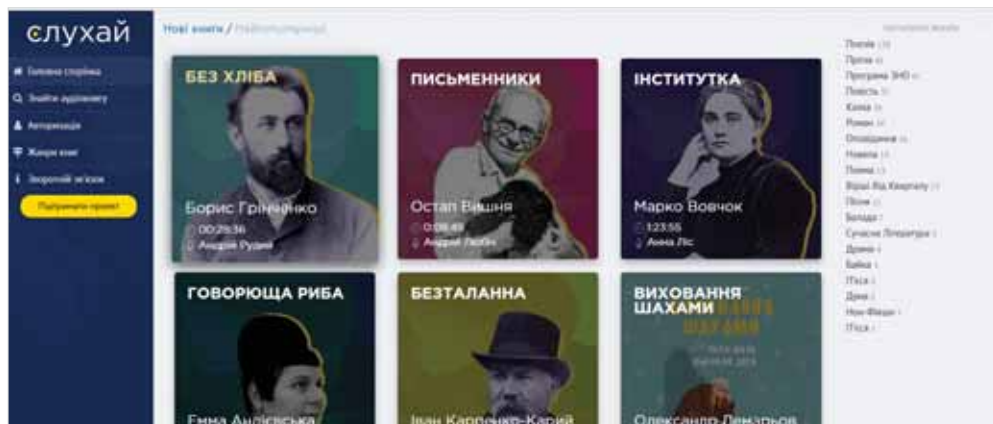
Сторінка **форуму** або **чату** схожа на таблицю, у рядках якої користувач записує своє повідомлення; при цьому відображається його аватар (маленьке зображення, що ідентифікує користувача), його мережеве ім'я (нік) та інструменти для коментування повідомлень.

Під час створення **вебкаталогу** застосовують мінімум засобів оформлення, розміщуючи посилання на ресурси в таблиці розділів або лінійному списку. Наприклад, перша сторінка бібліотеки аудіокниг (мал. 20.6 (<https://sluhay.com.ua/>)).



Це цікаво

Скрипт (сценарій) — це програма, що автоматизує деяку задачу, яку без сценарію користувач виконував би вручну, використовуючи інтерфейс програми.



Мал. 20.6



Це цікаво

Зараз популярними системами управління є *Drupal*, *Joomla* і *WordPress*. На основі цих систем можна створювати функціональні сайти, якими легко управляти. Двигунці для *Drupal*, *Joomla* та *WordPress* є безкоштовними.

4. У чому полягає процес створення вебсайту?

Процес створення вебсайту складається з таких етапів:

- визначення мети створення сайту, розробка концепції вебсайту та його структури;
- проведення дослідження з обраної теми;
- вибір способу подання даних і стилю оформлення вебсторінок;
- розмітка та програмування сайту;
- наповнення сайту даними;
- художнє оформлення сторінок;
- розміщення вебсторінок на сервері;
- популяризація сайту та підтримка його роботи.

Перший етап передбачає визначення мети створення сайту, його цільової аудиторії, з'ясування, якими будуть інформаційні джерела сайту, з яких розділів він складатиметься тощо. Перш ніж приступати до розробки, необхідно проаналізувати тему, вивчити сайти потенційних конкурентів — це допоможе створити власну концепцію. Далі слід визначити вид і тип сайту та відповідно до цього визначити його структуру, зазначити функціональні можливості (пошук, каталог, стрічка новин тощо), вказати, які сторінки мають бути на сайті та які блоки слід розміщувати на сторінках і в якому вигляді.

Як тільки концепцію і загальну структуру сайту буде визначено, можна обирати стиль його оформлення, проєктування структури

окремих вебсторінок, добирати найбільш вдалі способи зображення тих чи інших даних. Ці два етапи майже не піддаються формалізації і вимагають від їх виконавця не стільки володіння технологіями вебпрограмування, скільки спеціалізованих знань у тій галузі, якій буде присвячено сайт, художнього смаку й досвіду користування інтернетом.

Розмітка сайту визначає параметри форматування тексту та інших об'єктів на сторінках. Програмування сайту передбачає створення різних вебкомпонентів, завдяки яким опрацьовуються дані, що вводяться на сайті його відвідувачами.

Вміст сайту, який зазвичай займає 70–80 % його площі, безпосередньо впливає на дизайн майбутнього сайту. Тому спочатку добирають контент сайту, тобто всі матеріали та дані, які потрібно буде розмістити: текст, таблиці, малюнки, графіки тощо, і лише потім обирають його художнє оформлення. Перед розміщенням в інтернеті сайт тестується як розробником, так і замовником, після чого виправляють знайдені помилки та некоректності.

Публікація сайту в інтернеті передбачає його розміщення за обраним доменним ім'ям, яке реєструється у великих пошукових машинах і каталогах. Для того щоб сайт відвідувала значна кількість користувачів, потрібно здійснювати заходи щодо його популяризації — «розкрутки» сайту. Для цього використовується контекстна й банерна реклама, SEO, SMO та інші методи.



Це цікаво

SEO (від англ. *Search Engine Optimization* — пошукова оптимізація) — комплекс заходів для підняття позицій сайту в результатах пошуку пошукових систем за певними запитами користувачів. Як правило, що вищою є позиція сайту в результатах пошуку, то більше зацікавлених відвідувачів переходить на нього з пошукових систем.

SMO (від англ. *Social Media Optimization* — оптимізація сайту під соціальні мережі) — комплекс заходів, спрямованих на залучення на сайт відвідувачів із соціальних медіа: блогів, соціальних мереж тощо.

Пошукова система враховує такі параметри сайту при визначенні його релевантності (ступені відповідності запиту, що вводиться при пошуку):

- щільність ключових слів;
- індекс цитування сайту — залежить від кількості та авторитетності вебресурсів, які посилаються на цей сайт; багато пошукових систем не враховують посилання одна на одну; важливо, щоб посилання були зі схожих за тематикою сайтів.

До факторів, які знижують рейтинг сайту, належать: технології, які пошукові системи розглядають як спам, занадто довгі URL-адреси, надлишкова кількість посилань, графічна навігація, переадресація URL тощо.

5. Що передбачає вебдизайн?

Дизайн вебсторінок — це кропітка робота, яка спирається на знання основних понять **дизайну** (тобто понять точки, лінії, текстури, фактури, форми, кольору, пропорції, композиції, симетрії, асиметрії, рівноваги, динаміки, ритму, гармонії, контрасту, колориту, перспективи, центру мас), використання правил **ергономіки** (науки, яка вивчає робочі процеси з метою створення оптимальних умов праці, що сприяють підвищенню її продуктивності, а також зберігають сили, здоров'я, працездатність людини) та врахування таких основних складових **вебдизайну**:

- форма подання та стиль наповнення сайту, його зміст;
- оформлення — графічні елементи, навігація, шрифти, кольорова гама;
- технології — засоби створення, форматування сторінок і надання їм динамізму;
- подання — швидкість і надійність відображення сайту в мережі;
- мета — причина створення сайту та очікувані результати.

Вебдизайнеру, який вперше приступає до створення сайту, слід враховувати такі поради:

- недоцільно використовувати великі анімовані зображення на головній сторінці, рядки, що «розбігаються» в різні боки, «миготливі» елементи — це заважає зосередитись на вмісті сторінки;
- не слід використовувати широкі зображення — вони уповільнюють процес завантаження сторінки;
- не варто розміщувати на головній сторінці велику кількість кнопок, емблем каталогів, посилань — це перевантажує графіку сайту;
- недоцільно розміщувати на сторінці «довгі» тексти. Користувач не має прокручувати екран більш ніж на 2,5–3 екрани;
- не слід робити кольорову гаму занадто яскравою. Треба прагнути до створення єдиної атмосфери на сайті, і фон не має переважати;
- необхідно перевіряти сайт на наявність помилок — вони псують враження про важливість сайту;
- потрібно використовувати правило «п'ятірки»: не більш ніж п'ять шрифтів (як за типом, так і за розміром) на одній сторінці;
- не слід використовувати напис *Сторінка в розробці* — краще прибрати посилання на таку сторінку.

Під час вибору відповідного дизайну для сайту важливо пам'ятати, що форма має завжди підкреслювати зміст, не бути «агресивною» при перегляді, гармонійно поєднувати елементи декору, стилю, кольору, бути зручною для перегляду.

6. Як створити вебсторінку в мережі?

Для створення вебсторінки безпосередньо в інтернеті можна скористатись одним із сайтів, які надають відповідні засоби та інструменти для створення сайту та послугу безкоштовного хостингу — розміщення сайту на власному сервері. Для цього потрібно спочатку пройти процедуру реєстрації на обраному сайті, а потім слідувати відповідним вказівкам.

Розглянемо процес створення сайту в інтернеті на прикладі сервісу *Google Сайти (Google Sites)*. Якщо в користувача вже існує свій акаунт *Google*, то він може скористатись ним, якщо ні — спершу потрібно налаштувати свій обліковий запис.



ДІЄМО

Вправа 1. Створення структури сайту.

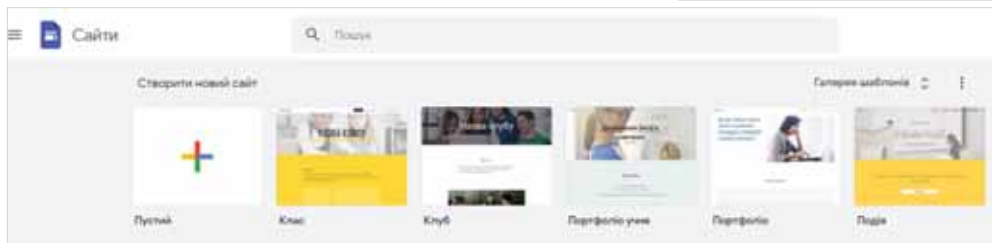
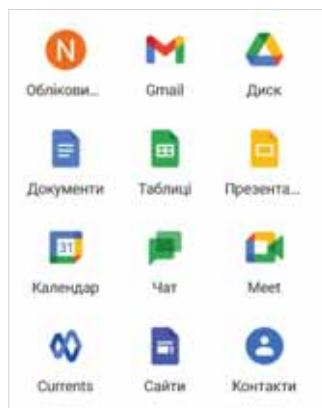
Завдання. Створіть сайт про дивовижні споруди світу, який складатиметься з двох сторінок — опис сайту та опис однієї зі споруд. Розмістіть на першій сторінці сайту зображення, що ілюструє дивовижні споруди світу.

1. У вікні браузера відкрийте головну сторінку пошукової системи *Google*. Увійдіть у власний акаунт, скориставшись кнопкою

Увійти

. Оберіть інструмент *Додатки Google* й у списку додатків оберіть послугу *Сайти* (мал. 20.7).

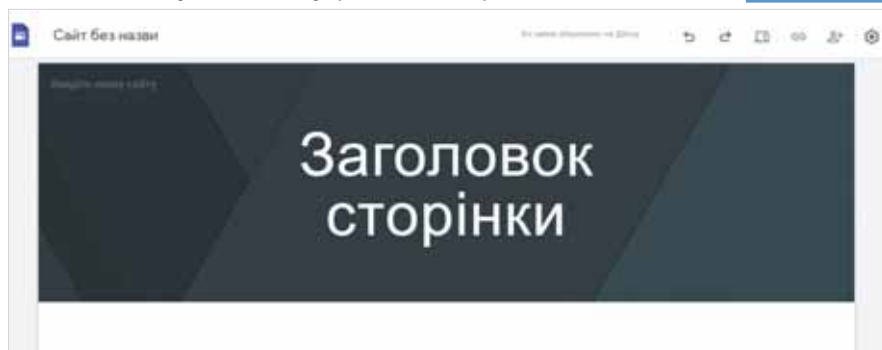
2. Підтвердіть намір створення сайту, обравши шаблон для створення. Оберіть *Порожній* для створення сайту.



Мал. 20.7

3. Введіть назву вебсайту (мал. 20. 8).

Мал. 20.8

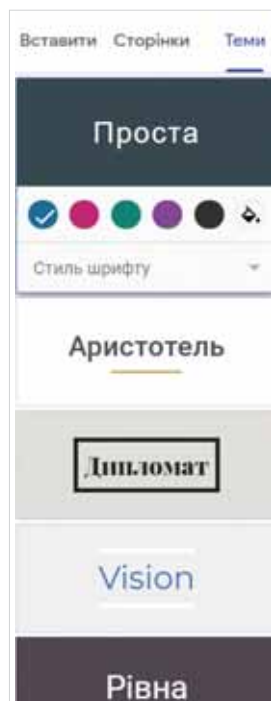


4. Натисніть клавішу **Опублікувати**, після чого введіть URL-адрес (мал. 20.9).

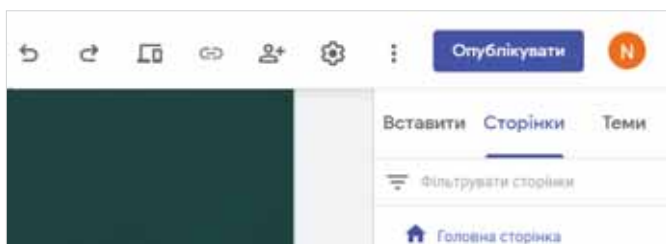
Мал. 20.9



5. Розгорніть список тем, які можна застосувати для стильового оформлення сторінок сайту (мал. 20.10).
6. Налаштуйте додаткові параметри: опис сайту, який потрібен для розміщення його в певній категорії, параметри доступу чи тип контенту (обмеження за віком).
7. Завершіть процес створення макету сайту, натиснувши кнопку *Створити сайт*.
8. Доповніть головну сторінку сайту сторінкою з описом однієї зі споруд. Для цього оберіть інструмент *Сторінки* у верхній частині вікна (мал. 20.11).



Мал. 20.10



Мал. 20.11

9. Оберіть зі списку запропонованих шаблонів сторінки (вебсторінка, оголошення, файловий менеджер, список) шаблон *вебсторінка*. Укажіть назву сторінки *Варто побачити*. Укажіть, що створена сторінка буде розміщуватись на головній сторінці (мал. 20.12).



Мал. 20.12

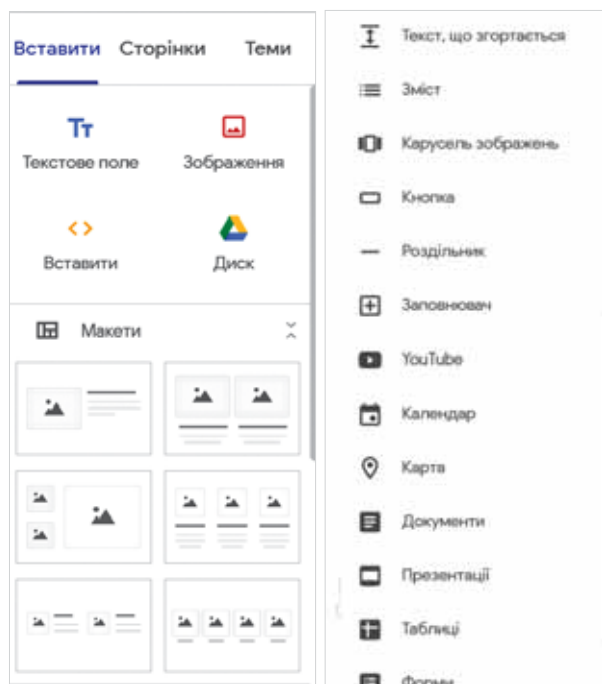
Вправа 2. Створення вмісту вебсайту.

Завдання. Додайте текст і зображення до створеного макету сайту *Дивовижні споруди*.

1. Перейдіть за адресою навчального сайту <https://sites.google.com/site/divoviznisporudi/>, ознайомтесь із вмістом його головної сторінки, перейдіть на сторінку *Варто побачити*! Скористайтесь матеріалами як зразком для наповнення вмісту сайту.
2. Скориставшись своїм обліковим записом, перейдіть до *Google Сайти*. Відкрийте сайт *Дивовижні споруди*, створений у попередній вправі. Перейдіть у режим редагування головної сторінки —

оберіть інструмент *Редагувати* . Змініть заголовок *Домашня сторінка*, вказавши свою назву титульної сторінки сайту (наприклад, *Дивовижні споруди світу*).

3. Заповніть текстове поле сторінки вступом про дивовижні споруди світу (можна скористатися матеріалами з файлу *Дивовижні споруди* з папки *Вебдизайн\Дивовижні споруди* або підготувати власний вступ).
4. Завантажте зображення *Дивовижні споруди* з папки *Вебдизайн\Дивовижні споруди*, скориставшись послугою *Вставити* з меню вбудованого редактора вебсторінки (мал. 20.13).
5. Перейдіть на сторінку *Варто побачити*. Додайте до неї опис однієї зі споруд на ваш розсуд. Додайте зображення обраної споруди. Для вибору споруди скористайтесь посиланнями у файлі *Дивовижні споруди* з папки *Вебдизайн\Дивовижні споруди* або пошуком в інтернеті.
6. Збережіть зміни й перегляньте створений сайт. Переконайтеся, що за допомогою засобів навігації можна переходити на домашню сторінку та сторінку *Варто побачити*.



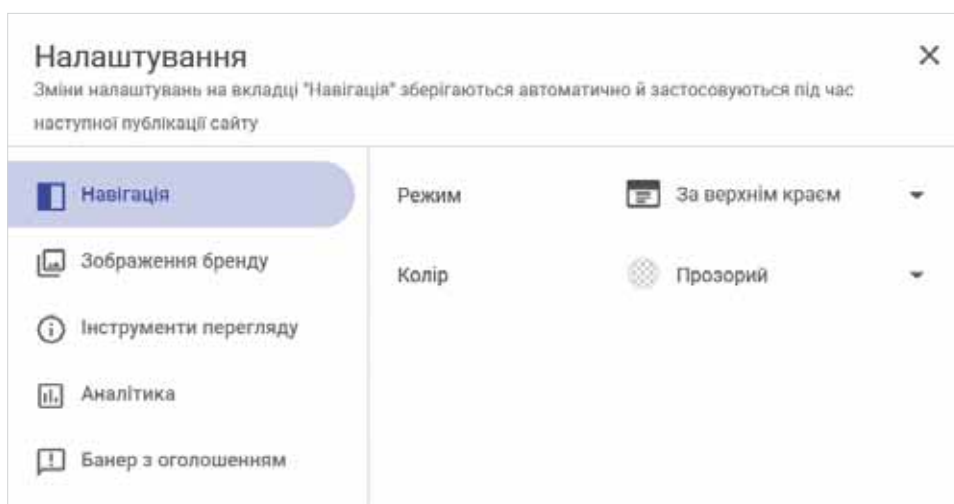
Мал. 20.13



Досліджуємо

Вправа 3. Налаштування.

Завдання. Дослідіть, які дії можна виконувати після створення сайту засобами сервісу *Google Сайти*. Для цього оберіть інструмент *Налаштування* у верхній частині вікна. Дослідіть, які параметри можна змінити (мал. 20.14).



Мал. 20.14

Працюємо в парах

1. Сформулюйте п'ять ознак професійних і любительських сайтів. Обговоріть їх у парі. Виділіть з них три ознаки, які ви вважаєте найсуттєвішими для проведення такої класифікації.
2. Чи можна назвати використання готових шаблонів та елементів для створення сайтів порушенням авторських прав? Обговоріть у парах.
3. Розгляньте приклади дизайну сайтів у презентації *Дизайн сайтів* із папки *Вебдизайн*. Обмінюйтесь думками, для яких прикладів дизайн дібрано доцільно. Що б ви запропонували змінити для покращення сприйняття матеріалу тих сайтів, дизайн яких, на вашу думку, дібрано невдало?
- ★ 4. Чи можна стверджувати, що в користувача, який володіє прийомами роботи із сайтом, наприклад, www.site.pryklad.com, буде достатньо навичок для роботи із сайтом www.karta.pryklad.com? Аргументуйте відповідь. Обговоріть у парах.
- ★ 5. Чому різні сайти відвідує різна кількість користувачів? Чи завжди популярний сайт є корисним для користувачів? Сплануйте, які відомості ви б хотіли розмістити на персональному сайті. Обговоріть у парі, чи користуватиметься такий сайт популярністю та на яку аудиторію він буде розрахований.
- ★★ 6. Чи можна створити якісний сайт без спеціальної освіти з вебпрограмування? Якими уміннями мають володіти майбутні автори сайтів?

Працюємо самостійно

7. Розгляньте приклади сайтів навчальних закладів, які розташовані поблизу вашої школи (їхні адреси можна знайти в інтернеті). Відшукайте, чи є на цих сайтах зразки банерів. Засобами графічного редактора створіть банер сайту свого навчального закладу.
- ★ 8. Знайдіть в інтернеті приклади сайтів інтернет-магазинів, які займаються онлайнним продажем комп'ютерної техніки. Проаналізуйте їх вміст і структуру. Сплануйте макет сайту інтернет-магазину, який матиме спеціалізацію з продажу принтерів. Створіть схему макета засобами графічного редактора або текстового процесора. Обґрунтуйте вибір програмного середовища для створення макета. Засобами безкоштовної служби *Google* створіть статичну вебсторінку згідно з розробленим вами макетом. Надішліть електронною поштою запрошення відвідати створену вами вебсторінку вчителю-ві або вчительці та сусіду чи сусідці по парті.

-  **9.** Доповніть власний сайт *Дивовижні споруди світу*. На головній сторінці сайту додайте гіперпосилання на вебсторінку, що містить добірку цікавих для вас споруд. Додайте до сайту ще три вебсторінки, на яких розмістіть матеріали про інші дивовижні споруди, обрані на власний розсуд. Проілюструйте матеріали зображеннями або додайте посилання на відео, що розміщені на *YouTube*. Матеріали про споруди знайдіть в інтернеті, зображення, які ви плануєте розмістити на вебсторінках, збережіть у папці *Сайти* своєї структури папок.
-   **10.** Засобами сервісу *Google Сайти* створіть сайт про власні захоплення. Матеріали, які вам знадобляться для додавання на вебсторінки, збережіть у папці *Сайти* своєї структури папок. На головній сторінці розмістіть вступ та додайте зображення. Створіть інші три вебсторінки, на яких детальніше опишіть свої три захоплення, проілюструйте їх зображеннями чи додайте посилання на відео, що розміщені на *YouTube*.
-   **11.** Засобами сервісу *Google Сайти* створіть сайт про рідкісних тварин або рослини. Матеріали, які вам знадобляться для додавання на вебсторінки, збережіть у папці *Сайти* своєї структури папок. На головній сторінці розмістіть вступ і додайте зображення. Створіть інші три вебсторінки, на яких подайте відомості про трьох тварин/три рослини, проілюструйте їх зображеннями чи додайте посилання на відео, що розміщені на *YouTube*.
-   **12.** Засобами сервісу *Google Сайти* створіть сайт про проєкт, який реалізовано учнями вашого класу на уроках чи в позаурочний час. Матеріали, які вам знадобляться для додавання на вебсторінки, збережіть у папці *Сайти* своєї структури папок. На головній сторінці сайту опишіть обраний проєкт, укажіть його мету та завдання, додайте відгуки учнів про проведений проєкт і пропозицію його підтримки іншими учнями шкіл. Створіть інші три вебсторінки, на яких розмістіть відомості про реалізацію проєкту, список використаних ресурсів, отримані результати, проілюструйте їх зображеннями чи додайте посилання на відео, що розміщені на *YouTube*.
-   **13.** Засобами сервісу *Google Сайти* створіть сайт про безпечну поведінку в інтернеті. Матеріали, які вам знадобляться для додавання на вебсторінки, збережіть у папці *Сайти* своєї структури папок. На головній сторінці розмістіть вступ та додайте зображення. Створіть інші три вебсторінки, на яких подайте відомості про правила безпечної поведінки в інтернеті, основні види загроз та способи їх уникнення, збірку корисних посилань з питань безпеки.



Створення вебсторінки за допомогою мови гіпертекстової розмітки

ПРИГАДАЙТЕ:

- ▶ особливості вебсторінки;
- ▶ призначення текстового редактора *Блокнот*;
- ▶ розширення файлу вебсторінки;
- ▶ що називають кодом.

ВИ ДІЗНАЄТЕСЬ:

- ▶ який вигляд має код розмітки вебсторінки мовою HTML;
- ▶ як створити простий HTML-документ;
- ▶ як формувати текст у HTML-документі;
- ▶ як працювати зі списками в HTML-документі;
- ▶ як працювати із зображеннями в HTML-документі;
- ▶ як побудувати гіпертекстові зв'язки в HTML-документі.



Вивчаємо

1. Який вигляд має код розмітки вебсторінки мовою HTML?

Ви вже пробували розробляти сайти за допомогою автоматизованих засобів. Вони самі конвертували сайт у файл із деяким HTML-кодом.

HTML (мова гіпертекстової розмітки) — це код, який використовується для структурування і відображення веб-сторінки та її контенту.

Створювати вебсторінки мовою HTML можна двома способами: за допомогою спеціального редактора HTML, у якому є заготовки та фрагменти готового коду, або писати безпосередньо на HTML. Другий спосіб часто буває легшим, ніж вивчати HTML-редактор або конвертер, які часто обмежені у своїх функціях, містять помилки або проводять поганий HTML-код, який не працює на різних платформах.

Мова HTML існує в декількох варіантах і продовжує розвиватися, але її конструкції, найімовірніше, будуть використовуватися і надалі. Вивчаючи й пізнаючи HTML детальніше, створюючи документ на початку вивчення HTML і розширюючи його, наскільки це можливо, ми маємо змогу створювати документи, які можуть бути переглянуті багатьма браузерами, — як зараз, так і в майбутньому.

Типовий простий вебдокумент складається з текстових блоків (абзаців, списків, таблиць), кількох малюнків невеликих розмірів, горизонтальних розмежувальних ліній та гіперпосилань. Складніші вебдокументи містять **фрейми** (структуровані сторінки), елементи управління (кнопки, перемикачі, поля діалогу), динамічні ефекти та графічні рухомі об'єкти тощо.

Основою мови HTML є **дескриптори** — команди, що вказують на правила, за якими форматуються документи. Команди мови HTML називають **тегами**. Отже, крім текстових даних, вебсторінка містить невидимий для користувача HTML-код, тобто набір тегів. Під час відкриття вебсторінки браузер використовує цей набір для відображення її елементів у заданому порядку.

Назва кожного тегу — це одна або декілька літер, взятих у дужки «<» та «>», яким відповідають певні англійські слова, що пояснюють їх призначення. Наприклад, назва тегу <BODY> виникла від англійського слова *body* — тіло, головна частина, а сам тег використовується для створення вмісту вебсторінки.

Тег записують так: <Тег> текст </Тег>. Слово <Тег> (так званий відкриваючий тег) позначає місце початку дії тегу на сторінці, текст — фрагмент тексту, до якого буде застосовано форматування, </Тег> (закриваючий тег) визначає місце закінчення дії тегу на сторінці. Пару тегів (відкриваючий і закриваючий) називають **контейнером**. Закриваючий тег відрізняється від відкриваючого лише тим, що в коді перед його ім'ям ставиться символ «/». Більшість тегів є парними, однак є і одинарні теги.

Один тег може мати кілька параметрів — **атрибутів**. Наприклад, <Тег параметр_1= значення_1 параметр_2=значення_2 >текст, що відображається на сторінці </Тег>.

Закриваючий тег не містить жодних параметрів. На відміну від нього, у відкриваючому тегу вказуються всі необхідні параметри.

Правила запису параметрів: після назви тегу після пропуску зазначається ім'я параметра (наприклад, ALIGN — ім'я параметра, який вказує на спосіб вирівнювання тексту), далі вказується знак «=» й у лапках записується значення параметра (наприклад, ALIGN="center"). За необхідності далі вказують інші параметри через пропуск. Після останнього параметра записується символ >.

Теги можуть бути вкладеними, тобто один контейнер може перебувати всередині іншого, у такому разі загальний запис виглядатиме так:

```
<Тег1>текст<Тег2>текст </Тег2>текст </Тег1>
```

2. Як створити простий HTML-документ?

Для підготовки найпростішого HTML-файлу можна використати текстовий редактор *Блокнот*, після написання коду в якому слід

задати файлу ім'я та вказати розширення — *htm*. Програмування тут суто символічне.

Програма мовою HTML має таку структуру:

<HTML> — заголовок файлу;
 <HEAD> — заголовок документа;
 <TITLE> — назва сторінки

Текст, який буде відображатися на рядку заголовка браузера

</TITLE>

</HEAD>

<BODY> — тіло;

Вміст вебсторінки

</BODY>

</HTML>

Основними параметрами тегу <BODY> є:

- BACKGROUND="1.jpg" — задає картинку для фону за шляхом *1.jpg*;
- BGCOLOR="white" — задає колір фону — білий;
- TEXT="black" — задає колір тексту — чорний.

Для зручного сприймання всі теги набирають великими літерами, хоча браузер однаково сприймає теги, написані й малими літерами. Не є обов'язковим також і написання тегів з нового рядка, проте код, записаний в одному рядку, читається гірше, особливо тими, хто цей код не створював. У багатьох випадках для складних сторінок між окремими блоками коду вставляють порожні рядки, поділяючи його на фрагменти, щоб полегшити читання коду.



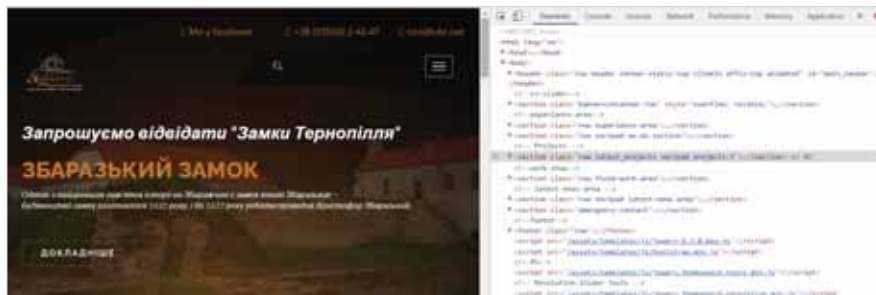
ДІЄМО

Вправа 1. Перегляд коду вебсторінки.

Завдання. Перегляньте HTML-код вебсторінки національного заповідника «Замки Тернопілля». Знайдіть заголовок, титульну назву, малюнок, звичайний текст, відформатований текст, посилання на іншу сторінку, список, таблицю.

1. Відкрийте вебсторінку національного заповідника «Замки Тернопілля» за посиланням <https://zamky.te.ua/> у браузері та знайдіть заголовок, титульну назву, малюнок, звичайний текст, відформатований текст, посилання на іншу сторінку.

Не закриваючи вікно браузера, відкрийте цю сторінку для перегляду. Використайте для цього вказівку контекстного меню *Переглянути джерело сторінки* (якщо ви працюєте в браузері *Chrome*) (мал. 21.1).



Мал. 21.1

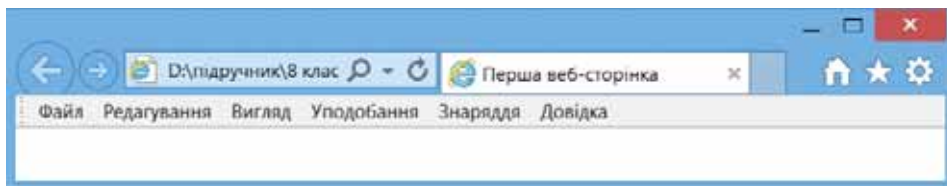
2. Проаналізуйте структуру HTML-документа та знайдіть відповідні елементи в браузері й у тексті коду.
3. У HTML-документі знайдіть подвійні теги, одинарні теги, визначте призначення окремих елементів, відшукуючи відповідні англomовні слова, які допоможуть пояснити кожний елемент.

Вправа 2. Перегляд коду вебсторінки.

Завдання. Складіть простий документ, який міститиме лише заголовок вікна за зразком (мал. 21.2).

Мал. 21.2

```
<HTML>
  <HEAD>
    <TITLE>Домашня сторінка школи</TITLE>
  </HEAD>
</HTML>
```



Мал. 21.3

1. Відкрийте текстовий редактор *Блокнот*.
2. Введіть у ньому структуру HTML-документа (мал. 21.2).
3. Збережіть файл з ім'ям *page1.htm*.
4. Відкрийте створений файл, проаналізуйте зовнішній вигляд створеного файлу в браузері (мал. 21.3).
5. Поверніться до збереженого файлу в *Блокноті*.
6. Внесіть до нього такі зміни: укажіть у назві сторінки: прізвище, ім'я, школу, клас, захоплення. До елемента `<TITLE>` введіть:
Домашня сторінка (Ваше ім'я і прізвище)
7. Збержіть файл як *page2.htm*.
8. Перегляньте результат у браузері.

3. Як формувати текст у HTML-документі?

Процес структурування сторінки можна розглядати з декількох позицій: заголовки, вирівнювання, новий рядок, абзац, примусове розміщення, лінія.

Для формування **заголовків** використовують теги `<Hn>...</Hn>`, де $n = 1 \dots 6$. При $n = 1$ отримуємо заголовок найвищого рівня, відповідно при $n = 6$ — заголовок найнижчого рівня.

Об'єкти сторінки можуть бути вирівняні за допомогою атрибута **ALIGN**, який має чотири значення, що визначають спосіб вирівнювання тексту:

- `center` — по центру;
- `left` — ліворуч;
- `right` — праворуч;
- `justify` — за шириною.

На відображення документа в браузері не впливають символи кінця рядка, які отримують натискуванням клавіші *Enter* або *Shift+ Enter*, а також пропуски. Для **перенесення** тексту на новий рядок використовують одинарний тег `<BR>`.

Для створення **абзаців** використовують парний тег `<P>...</P>`.

При використанні тегу `<P>` спочатку вставляється порожній рядок, далі текст виводиться з початку нового рядка. Якщо вжито тег `<BR>`, то порожній рядок не вставляється.

Для **виділення** тексту, який має певне розміщення символів (наприклад, шаблон документа), використовують тег `<PRE> текст із вимушеними пропусками та перенесенням у рядки </PRE>`.

Щоб відділити одну ділянку тексту від іншої, зазвичай використовують підкреслення **горизонтальною лінією**. Тег `<HR>` зображує на екрані горизонтальну лінію. Наприклад,

`<HR ALIGN="center" WIDTH="50%" SIZE="3">` — лінія по центру, розміром 3 пункти (позначається «пт»), яка займає 50 % ширини сторінки;

`<HR ALIGN="center" WIDTH="400" SIZE="5" COLOR="red">` — лінія по центру червоного кольору, довжиною 400 пікселів, розміром 5 пт.

Як видно з прикладів, графічний об'єкт може мати такі **атрибути**:

- розмір — у пунктах;
- ширину **WIDTH** — у відсотках або пікселях;
- колір — назвами кольорів англійською мовою або їхніх кодів (наприклад, `#0000FF` — бірюзовий).

Для роботи із символами використовують такі **теги форматування**:

- тип шрифту;
- колір шрифту;
- розмір шрифту;
- накреслення.

Тег використовують для роботи із символами. Гарнітура шрифту задається за допомогою атрибута FACE. Його можливі значення — назви шрифтів (наприклад, *Arial*) або ключові слова, які визначають зовнішній вигляд шрифту, наприклад:

- *Serif* — із насічками;
- *San_serif* — без насічок;
- *Monospace* — непропорційний;
- *Cursive* — рукописний.

Якщо вказується ключове слово, то зі списку шрифтів на комп'ютері користувача браузер вибирає для вебсторінки найбільш відповідний.

Якщо вказувати конкретну гарнітуру, то такого шрифту на комп'ютері користувача може не виявитися, і він в такому разі не зможе переглянути текст. Тому можна вказувати декілька шрифтів:

```
<FONT FACE="Verdana, Arial">.
```

Опрацьовуючи такий запис, браузер спочатку намагатиметься знайти шрифт *Verdana*, а потім, якщо такого шрифту не виявиться, — *Arial*.

Якщо жодного зі шрифтів не знайдено, то встановлюється будь-який шрифт без насічок.

Колір шрифту задається за допомогою атрибута COLOR за назвою кольорів:

- *aqua* — бірюзовий;
- *black* — чорний;
- *blue* — світло-синій;
- *fuchsia* — бузковий;
- *gray* — сірий;
- *green* — зелений;
- *lime* — салатний;
- *maroon* — бордовий;
- *navy* — синій;
- *olive* — оливковий;
- *purple* — фіолетовий;
- *red* — червоний;
- *silver* — сріблястий;
- *teal* — сіро-зелений;
- *white* — білий;
- *yellow* — жовтий.

Розмір шрифту задається за допомогою атрибута SIZE, що може набувати таких значень:

- цілі числа від 1 до 7, де числу 1 відповідає розмір шрифту 8 пт, числу 2 — 10 пт, 3 — 12 пт, 4 — 14 пт, 5 — 18 пт, 6 — 24 пт, 7 — 36 пт;
- числа від 1 до 6 та від -6 до -1, на скільки одиниць шрифт має відрізнятися від встановленого за замовчуванням (у більшості за замовчуванням розмір шрифту 3 — 12 пт).

Розміри шрифту можна змінювати за допомогою тегів <BIG> і <SMALL>. Перший з них збільшує розмір шрифту щодо прийнятого за замовчуванням на одиницю, другий — зменшує його, теж на одиницю.

Тег <BASEFONT> дає змогу встановити базові параметри шрифту. Так, запис <BASEFONT SIZE=4> означає, що для всього коду до закриваючого тегу </BASEFONT> значення шрифту становитиме 14 пт, крім тих фрагментів, де це значення змінюється тегом .

Для встановлення типу накреслення використовуються такі контейнери:

- ... — напівжирний;
- <I>...</I> — курсив;
- <U>...</U> — підкреслений;
- _{...} — нижній індекс;
- ^{...} — верхній індекс.

4. Як працювати зі списками в HTML-документі?

До HTML-документа можна вставляти маркований і нумерований списки. Для цього використовуються відповідні теги:

Маркований список

```
<UL>
<LI> Елемент списку</LI>
<LI> Елемент списку</LI>
<LI> Елемент списку</LI>
<LI> Елемент списку</LI>
.....
<LI> Елемент списку</LI>
</UL>
```

Нумерований список

```
<OL>
<LI> Елемент списку</LI>
<LI> Елемент списку</LI>
<LI> Елемент списку</LI>
<LI> Елемент списку</LI>
.....
<LI> Елемент списку</LI>
</OL>
```

5. Як працювати із зображеннями в HTML -документі?

Зображення можна використовувати в HTML-документах так:

- для зміни фону сторінки;
- як ілюстрацію на сторінці;
- як посилання на інший об'єкт;
- як маркер у списку.

У HTML-документі можна використовувати зображення різного формату. Розмір файлу зображення визначає швидкість появи зображення на екрані. Графіка, яка повільно завантажується, може «відлякати» відвідувачів сторінок. Для зменшення розміру файлу необхідно використовувати зображення формату JPG або GIF. Допустимими зображеннями є такі, розмір яких не перевищує 40–50 кБ.

На вебсторінках графічні зображення можна розміщувати різними способами. **Вбудовані зображення** — це графічні зображення, які завжди залишаються в одному й тому самому місці сторінки й не обтікаються текстом. Плаваючі зображення — це зображення, що не прив'язуються до одного рядка тексту й ніби «плавають» уздовж одного з полів та обтікаються текстом.

Для вставляння до документа малюнка використовується одинарний тег `<IMG>` (від англ. *image* — зображення) з такими параметрами: `SRC=ім'я_файла` (від англ. *source* — джерело).

Наприклад, за командою `<IMG SRC=pr.gif>` на екрані буде відображатися вміст графічного файлу *pr.gif*, що зберігається в поточному каталозі.

Параметр `ALT="текст_напису"` використовується в тому разі, коли браузер не знаходить зображення у вказаному місці на диску, тоді замість нього на екрані відображається маленький прямокутник з відповідним надписом у його середині, який задано параметром `ALT`. Цей параметр рекомендується використовувати завжди.

Параметри `WIDTH=n` і `HEIGHT=m` задають ширину й висоту (у пікселях) прямокутника, у який виводиться картинка. Якщо параметри не задано, то зображення відображається в її розмірах. Коли розміри прямокутника не збігаються з розмірами зображення, браузер масштабує його, враховуючи значення, вказані параметрами `WIDTH` і `HEIGHT`.

Параметр `ALIGN` дає змогу визначити положення ілюстрації щодо сусідніх об'єктів, цей параметр використовується для створення плаваючих зображень. Для плаваючих зображень значення цього атрибута можуть бути *left* або *right*. Для вбудованих зображень цей параметр не використовується. При використанні вирівнювання зображень до лівої та правої меж часто намагаються зробити так, щоб текст виводився під зображенням, а не збоку від нього. Для цього використовується вказівка

`<BR=CLEAR LEFT>` або `<BR=CLEAR RIGHT>`.

Вправа 4. Вставляння зображень до вебсторінки.

Завдання. Змініть файл *Відпочинок* з папки *Сайти*, що міститься в спільному сховищі за вказаними параметрами (мал. 21.4).

1. Додайте малюнок *Rafting.jpg*.
2. Уведіть текст, який має з'являтися на екрані, якщо при перегляді файла браузером графічний файл не знайдено.

3. Змініть тип вирівнювання малюнка.
4. Задайте заголовок так, щоб він був розташований праворуч від малюнка.
5. Змініть шрифт тексту в списку.



Мал. 21.4

6. Як побудувати гіпертекстові зв'язки в HTML-документі?

Гіперпосилання може бути подано різними способами: крім тексту, ще й малюнком, картою (розміченим малюнком), списком-меню, екранною кнопкою тощо.

Існують два основні типи гіперпосилань: внутрішні та зовнішні.

Внутрішні гіперпосилання — це посилання на об'єкти в межах одного документа. За їхньою допомогою можна переміщуватися всередині однієї вебсторінки. Такі посилання доцільно використовувати на довгих сторінках, щоб мати можливість швидко переміщуватись між розділами.

Зовнішні гіперпосилання — це посилання на віддалені HTML-файли або на будь-які файли, які належать до HTML-документів.

Для завдання гіпертекстового переходу до іншого документа використовується парний тег `<A>` (від англ. *anchor* — якір).

Формат вказівки для зовнішнього посилання такий:

`<A HREF=ім'я файла> текст посилання </A>`

Після виконання такої вказівки на екран виводиться посилання: *текст посилання*. При зверненні лівою кнопкою миші до посилання з текстом *текст посилання* (для чого досить установити курсор миші на виділений текст і натиснути ліву клавішу миші) здійснюватиметься перехід до файлу, у якому зберігається ця сторінка. Ім'я такого файлу вказується після атрибута `HREF`.

Якщо в тегу `<A>` вказано ім'я файлу, то при перегляді такого документа з відповідним посиланням браузер шукає його в тому ж

каталозі, у якому розташований файл із посиланням. Коли необхідно здійснити посилання на файл, розташований в іншому каталозі, слід зазначити в тегу `<A>`:

- або повний шлях до файлу, починаючи з імені дисководу (тобто його абсолютну адресу);
- або шлях щодо файлу з посиланням — відносну адресу.

Приклади опису посилань:

`HREF="file.html"` — файл *file.html* міститься в активному каталозі;

`HREF="files/file.html"` — файл *file.html* міститься в каталозі з іменем *files*, що міститься в поточному каталозі;

`HREF="files/morefiles/file.html"` — файл *file.html* міститься в каталозі з іменем *morefiles* каталогу *files*, що міститься в активному каталозі;

`HREF="../file.html"` — файл *file.html* міститься на один рівень нижче від активного каталогу;

`HREF="../../file.html"` — файл *file.html* міститься на два рівні нижче від активного каталогу.

Для здійснення гіпертекстового переходу всередині документа використовують два теги `<A>`. Перший тег з параметром `HREF` є джерелом переходу, тобто посиланням, яке вказує, куди слід перейти; другий з параметром `NAME` — приймачем, тобто це мітка, яка фіксує місце переходу.

Елемент `<A HREF=#назва_мітки>текст посилання </A>` задає перехід на мітку, яка має ім'я *назва_мітки* (для цього прикладу) і міститься в документі. Тут `#` — ключовий символ, який вказує, що використовується мітка, а не файл.

Назва мітки — ім'я для мітки, яке вибирає користувач.

Текст посилання — фрагмент, який буде відображатися на екрані як посилання. Для того щоб посилання спрацювало, необхідно, щоб в цьому самому документі існувала вказівка:

```
<A NAME=назва_мітки> текст</A>.
```

Назва мітки — це довільний текст, який не з'являється на екрані й однозначно визначає місце переходу. Якщо на сторінці є кілька міток, то всі вони повинні мати різні назви.

Рекомендації до використання переходів усередині документа.

1. Вибрати ім'я для мітки. Ім'я має бути унікальним у HTML-документі, це означає, що інших міток з таким ім'ям у документі бути не повинно.

2. Запрограмувати перехід за міткою, тобто в місці переходу потрібно написати вказівку виду `<A HREF=#мітка> текст </A>`.

3. Установити тег з міткою в потрібному місці HTML-документа, тобто перед фрагментом HTML-документа, на який повинен виконуватися перехід, записати тег-мітку:

```
<A NAME=мітка> </A>.
```

При цьому ім'я мітки повинно бути таким само, як і в команді, що задає перехід.

Для того щоб використати картинку як посилання, достатньо записати елемент посилання в основному документі:

```
<A HREF=ім'я файлу для переходу > <IMG SRC= ім'я гра-  
фічного файлу> </A>.
```

Тобто до звичайного посилання замість тексту вставляється елемент, за яким викликається на екран графічний файл. При цьому у вказівках виклику графічних файлів можна використовувати всі традиційні параметри.

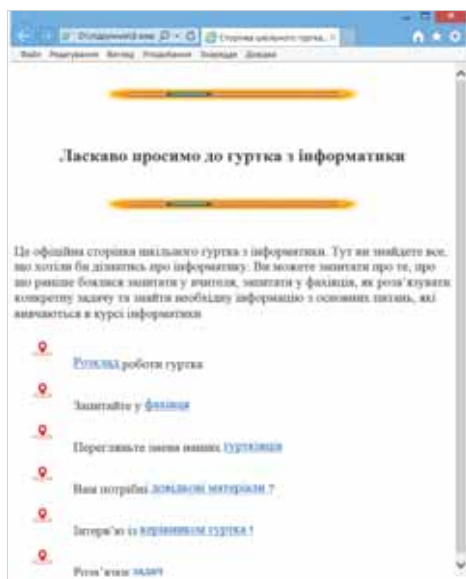


ДІЄМО

Вправа 5. Сторінка з гіперпосиланнями.

Завдання. Змініть файл *Гурток* з папки *Сайти*, що міститься в спільному сховищі, так, щоб у браузері отримати сторінку (мал. 21.5).

1. Відкрийте файл *Гурток* з папки *Сайти*, що міститься в спільному сховищі, у браузері. Перегляньте, чи відображається його вміст так, як на малюнку.



Мал. 21.5

2. Збережіть файл у папці *Сайти* своєї структури папок. Додайте до своєї папки також усі потрібні графічні файли.
3. Внесіть правки до параметрів тегу `<IMG SRC="point.gif" ALIGN=BOTTOM>`, додавши висоту (`HEIGHT=100`) та ширину (`WIDTH=100`) малюнка.

- Змініть фрагмент тексту списку так, щоб додати до виділеного слова гіперпосилання. Наприклад, `<A HREF="notthere.htm">` Розклад `</A>`.
- Вносьте зміни за зразком. Збережіть остаточний файл.



Працюємо в парах

- Обговоріть, які відомості, подані на сайті за адресою <https://goo.gl/bC3WD4>, доповнюють матеріал підручника.
- Обговоріть, як можна застосувати онлайнний підручник (https://www.w3schools.com/html/html_intro.asp) та онлайнний тренажер (https://www.w3schools.com/html/html_examples.asp) з мови HTML у своєму навчанні. Наведіть одне одному свої пропозиції. У чому відмінність цих ресурсів?
- Обговоріть, як можна використати приклад форматування вебсторінки (мал. 21.6) на сайті <https://htmlg.com/html-editor/> для створення власних вебсторінок. Хтось висловлює припущення, а хтось — перевіряє його. Потім міняєтесь ролями.



Мал. 21.6



- Попрацюйте в груповому проєкті «Професії майбутнього». Оберіть, хто для якої професії створюватиме вебсторінку. Створіть вебсторінки, у яких зазначено опис та основні вимоги до працівників в обраній професії. Оформіть їх відповідним чином і збережіть кожну окрему сторінку під відповідним іменем до папки *Професії*. Створіть спільну сторінку зі списком усіх професій, з якої можна перейти на кожну створену сторінку.
- Попрацюйте в груповому проєкті «Відомі люди нашого краю». Обговоріть список відомих людей. Оберіть, хто для якої людини створюватиме вебсторінку. Створіть вебсторінки, у яких вказано біографічні дані та досягнення відомої людини, її фото та посилання на профіль в інтернеті (наприклад, у Вікіпедії). Оформіть сторінки

відповідним чином і збережіть кожну окрему сторінку під відповідним іменем до папки *Люди*. Створіть спільну сторінку зі списком усіх персоналій, з якої можна перейти на кожну створену сторінку.



Працюємо самостійно

6. Створіть вебсторінку про вашу родину: склад, традиції, професії та інше.

★ 7. Створіть вебсторінки, які містять такі відомості (за потреби скористайтесь пошуком в інтернеті):

- 1) області України: кількість населення, площа;
- 2) держави-сусіди України: їхні назва, площа, адміністративний устрій, столиця;
- 3) назви міст України, у яких нараховується населення понад 1 млн жителів, зображення-візитівка міста;
- 4) список великих річок, що протікають через Україну.

★★ 8. Створіть вебсторінку вашого класу. Як приклад, вона може мати вигляд, поданий у файлі *Структура вебсторінки* в папці *Сайти* на спільному ресурсі.

Оцінюємо себе

Оцініть результати свого навчання за шкалою: від дуже погано до дуже добре.



Сплануйте кроки задля підвищення своїх результатів до найвищої позначки шкали. Скористайтесь порадами вчителя або вчительки чи допомогою друзів.

- ☐ Можу навести приклади засобів автоматизованого створення вебсторінок.
- ☐ Умію створювати вебсторінки за допомогою автоматизованих засобів та публікувати їх в інтернеті.
- ☐ Дотримуюсь правил ергономічного розміщення матеріалів на вебсторінці.
- ☐ Розумію поняття мови гіпертекстової розмітки.
- ☐ Використовую гіпертекстові, графічні й мультимедійні елементи на вебсторінках.
- ☐ Усвідомлю важливість участі в діяльності глобальної інтернет-спільноти.

ЗМІСТ

Розділ 1. Кодування даних та апаратне забезпечення

1. Кодування даних	4
1. Що означає «опрацювання даних»?	4
2. Що таке кодування повідомлень та для чого воно потрібне?	6
3. Які особливості має двійкове кодування?	8
4. Як кодуються в комп'ютері текстові повідомлення?	9
5. Як визначити довжину двійкового коду повідомлення?	12
6. Як кодують графічні дані?	14
2. Апаратне забезпечення комп'ютера	18
1. Як технічний прогрес вплинув на розвиток комп'ютерної техніки?	18
2. Якими бувають сучасні комп'ютери?	21
3. Яку архітектуру мають сучасні комп'ютери?	23
3. Призначення і характеристики складових комп'ютера	26
1. Чи може комп'ютер працювати без процесора?	26
2. Які пристрої комп'ютера розміщуються всередині системного блока?	28
3. Що і як можна зберігати в пам'яті комп'ютера?	29
4. Які особливості мають пристрої введення-виведення даних?	33
5. Які пристрої входять до складу мультимедійного обладнання?	36
6. Як обрати найкращий комп'ютер для роботи?	37

Розділ 2. Опрацювання текстових даних

4. Редагування і форматування текстового документа	41
1. Як виконувати заміну фрагментів тексту в <i>Microsoft Word</i> ?	41
2. Які особливості заміни фрагментів тексту в <i>LibreOffice Writer</i> ?	43
3. Як до тексту вставити символи, яких немає на клавіатурі?	44
4. Як застосувати стильове форматування для оформлення документа?	46
5. Як додати формули до текстового документа?	49
5. Структурування текстового документа	53
1. Як розмістити текст у кілька колонок?	53
2. Як у документі створити кілька розділів і відформатувати їх?	55
3. Що таке колонтитули та як їх додати до текстового документа?	59
4. Які особливості роботи з колонтитулами в середовищі <i>Microsoft Word</i> ?	60
5. Які особливості роботи з колонтитулами в середовищі <i>LibreOffice Writer</i> ?	62
6. Робота з великими документами	66
1. Як можна переглянути структуру документа?	66
2. Як використовують гіперпосилання в текстових документах?	69
3. Як автоматично створити зміст і покажчик у текстовому документі?	71
4. Як опрацьовувати складні текстові документи?	76
7. Спільна робота з документом	78
1. Що передбачає спільна робота з документом?	78
2. Як захистити текстовий документ?	79
3. У яких режимах спільно працюють з документом?	80
4. Як організувати одночасну спільну роботу над документом кільком авторам?	82

Розділ 3. Опрацювання мультимедійних об'єктів

8. Поняття мультимедіа	87
1. Що розуміють під мультимедіа?	87
2. Як кодуються аудіо- та відеодані?	88
3. Які існують формати файлів звукозапису?	90
4. Які формати відеофайлів є найбільш поширеними?	91
9. Опрацювання об'єктів мультимедіа	95
1. За допомогою яких програм можна переглядати відео та слухати звукозаписи?	95
2. За допомогою яких програм захоплюють звук?	97
3. За допомогою яких програм захоплюють відео?	99
4. За допомогою яких програм можна перетворити звукозаписи й відеофайли з одного формату на інший?	101
10. Створення та опрацювання відео	104
1. За допомогою яких програм можна створити власний відеокліп?	104
2. Які особливі елементи інтерфейсу має програма <i>OpenShot Video Editor</i> ?	105
3. Як додати до проєкту мультимедійні об'єкти?	106
4. Як додати відеопереходи між кліпами чи зображеннями й візуальні ефекти?	108
5. Як додати до відеокліпу музичний супровід і налаштувати часові параметри відео й аудіо?	110
6. Як додати до проєкту написи та зберегти створений відео кліп у форматі відео?	111
11. Сервіси для роботи з аудіо- й відеоданими та публікування їх в інтернеті	115
1. Які існують сервіси для розміщення відео та аудіоматеріалів в інтернеті?	115
2. Як шукати та переглядати відео на <i>YouTube</i> ?	118
3. Як опублікувати відео на <i>YouTube</i> ?	119
4. Як можна вносити зміни до відео за допомогою онлайнових сервісів?	121

Розділ 4. Алгоритми та програми

12. Мова та середовище програмування	124
1. Якими бувають мови програмування?	124
2. Які засоби необхідні для виконання створених програм?	127
3. Які середовища програмування використовують для створення програм?	128
13. Структура програми мовою <i>Python</i>	133
1. Що описують на початку програми?	133
2. Що може містити тіло програми?	134
3. З якими функціями працюють у програмі мовою програмування <i>Python</i> ?	135
14. Поняття об'єкта в мові програмування, його властивостей і методів	140
1. Що розуміють під об'єктом у мові програмування?	140

2. Які особливості мають програми з графічним інтерфейсом?	143
3. Які є способи створення об'єктів графічного інтерфейсу?	144
4. Як в об'єкті <i>вікно</i> розмістити напис?	144
15. Обробники подій, пов'язаних з елементами керування	148
1. Як пов'язані елементи керування та події?	148
2. Як у програмі описати елемент керування <i>кнопка</i> ?	149
3. Що таке обробник події?	151
4. Як отримати вікно повідомлення?	153
16. Опрацювання величин у програмах з елементами керування	155
1. Як використовують текстове поле для роботи з величинами?	155
2. Які операції виконують над текстовими величинами?	159
3. Які функції використовують для опрацювання текстових величин?	160
17. Розгалуження мовою програмування <i>Python</i>	164
1. Які команди використовують для опису розгалужень мовою програмування <i>Python</i> ?	164
2. Як за допомогою елементів управління задати логічне значення?	169
18. Цикли мовою програмування <i>Python</i>	174
1. Які команди використовують для опису циклів мовою програмування <i>Python</i> ?	174
2. Що таке вкладені цикли?	176
3. Як створити шкалу вибору?	179
19. Графічне відображення даних мовами програмування	182
1. Як у мові програмування <i>Python</i> створюють полотно для малювання?	182
2. Які геометричні примітиви можна створювати за допомогою модуля <i>Canvas</i> ?	183
3. Як додавати малюнки та підписи до малюнків?	187

Розділ 5. Створення та публікація вебресурсів

20. Автоматизовані засоби для створення та публікації вебресурсів	192
1. Яку структуру можуть мати сайти?	192
2. Які існують вимоги до структури сайту?	196
3. Які відмінності можуть мати вебсторінки сайту?	197
4. У чому полягає процес створення вебсайту?	200
5. Що передбачає вебдизайн?	202
6. Як створити вебсторінку в мережі?	203
21. Створення вебсторінки за допомогою мови гіпертекстової розмітки	209
1. Який вигляд має код розмітки вебсторінки мовою HTML?	209
2. Як створити простий HTML-документ?	210
3. Як формувати текст у HTML-документі?	213
4. Як працювати зі списками в HTML-документі?	215
5. Як працювати із зображеннями в HTML-документі?	215
6. Як побудувати гіпертекстові зв'язки в HTML-документі?	217

Глосарій

Нетбук — пристрій, який схожий на ноутбук, тільки менший за розміром та з низькою продуктивністю, тому використовується для виконання обмеженого набору завдань.

Неттоп — вид стаціонарного ПК.

Ноутбук — це комп'ютер, у корпусі якого поєднуються стандартні комплектуючі портативного комп'ютера (дисплей, клавіатура, тачпад, батарея). Може виконувати всі функції настільного комп'ютера.

Об'єкт (від лат. *objectum* — предмет, явище) — це те, на що спрямована певна діяльність (на противагу **суб'єкту**, який здійснює таку діяльність).

Обробник події — це частина програми, що виконує певні дії у відповідь на подію (спричинену, наприклад, дією користувача).

Оперативний запам'ятовуючий пристрій — ОЗП (RAM — від англ. *Random Access Memory* — пам'ять з довільним доступом) — швидка й енергозалежна пам'ять.

Опрацювання даних — це інформаційний процес отримання нових даних з наявних.

Плагін (від англ. *plug-in* — підключати) — незалежний програмний модуль, що підключається до основної програми та призначений для розширення або використання її можливостей.

Планшет — портативний комп'ютер із сенсорним екраном.

Подія — зміна властивостей об'єкта, взаємодія між об'єктами, утворення нового об'єкта або знищення наявного об'єкта.

Подкаст — аудіофайл, або серія таких файлів, що публікуються за однією адресою в інтернеті й регулярно оновлюються.

Покоління ЕОМ (*computer generation*) — один із класів у класифікації обчислювальних систем за ступенем розвитку апаратних і програмних засобів.

Постійний запам'ятовуючий пристрій — ПЗП (ROM — від англ. *Read Only Memory* — пам'ять тільки для читання) — швидка та енергонезалежна пам'ять.

Портативні комп'ютери — це комп'ютери, у яких є батарея, що дає змогу певний час працювати пристроєм без підключення до електромережі. Вони менші за розміром і вагою, тому їх можна легко переносити із собою.

Процесор — це мікросхема, яка створюється на напівпровідниковому кристалі (або кількох кристалах) шляхом застосування складної мікроелектронної технології.

Розділ — частина документа, яка має задані параметри форматування сторінки чи колонок.

Середовище програмування — це комплекс програм, що має засоби автоматизації процесів підготовки та виконання програм.

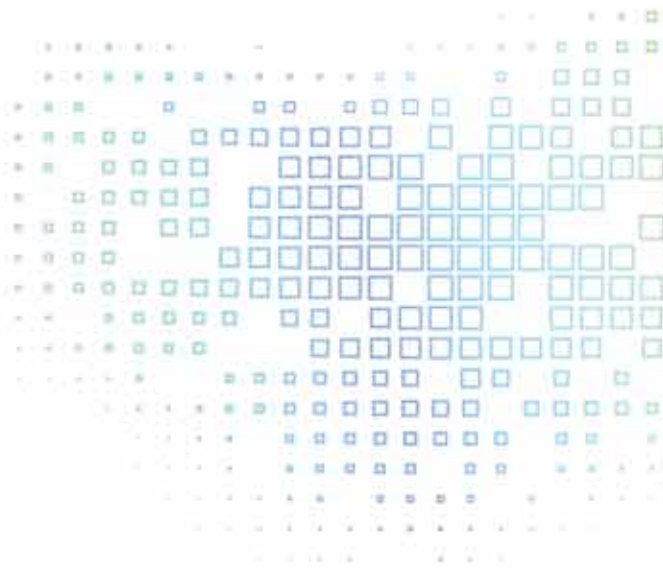
Середовище об'єкта — середовище, у якому перебуває об'єкт, незалежно від того, які дії ним чи над ним (крім знищення об'єкта) виконано.

Системна плата — мікросхема, що розміщується всередині комп'ютера, яку ще називають материнською. На ній встановлюються процесор, внутрішня пам'ять комп'ютера та інші пристрої.

Стиль — сукупність параметрів форматування, що позначається унікальним іменем і може застосовуватися для форматування текстових фрагментів.

Транслятор — (від англ. *translation* — переклад) — програма, яка записує команди програми, описаної деякою мовою програмування, на машинну мову.

Функція — це фрагмент програми, який може використовуватись багато разів.



ISBN 978-966-991-119-3



9 789669 911193

Право для безоплатного розміщення підручника в мережі Інтернет має
Міністерство освіти і науки України <http://mon.gov.ua/> та Інститут модернізації змісту освіти <https://imzo.gov.ua>