



ИНФОРМАТИКА

8



УДК 004(075.3)
ББК 32.97я721
I-74

Автори:

Й.Я. РИВКІНД, Т.І. ЛИСЕНКО, Л.А. ЧЕРНІКОВА, В.В. ШАКОТЬКО

Рекомендовано Міністерством освіти і науки України
(Наказ Міністерства освіти і науки України від 10.05.2016 № 491)

Видаєно за рахунок державних коштів. Продаж заборонено

Експерти, які здійснили експертизу підручника під час проведення конкурсного відбору проектів підручників для учнів 8 класу загальноосвітніх навчальних закладів і зробили висновок про доцільність надання підручнику грифа «Рекомендовано Міністерством освіти і науки України»:

Жук Н.А., методист Гоцанського районного методичного кабінету Рівненської області;

Попова Л.М., учитель інформатики спеціалізованої загальноосвітньої школи I–III ступенів № 7 Світловодської міської ради Кіровоградської області.

Інформатика : підруч. для 8-го кл. загальноосвіт.
I-74 навч. закл. / Й.Я. Ривкінд [та ін.]. — Київ : Генеза,
2016. — 288 с. : іл.
ISBN 978-966-11-0692-4.

Навчальний матеріал підручника поділено згідно з новою програмою на 8 розділів. У підручнику розділи «Текстовий процесор» і «Табличний процесор» подано для пакета **Microsoft Office**.

Кожний пункт підручника містить запитання для актуалізації знань, основний навчальний матеріал відповідно до програми, тренувальні вправи, запитання для самоконтролю та практичні завдання, які розподілено за рівнями навчальних досягнень. Підручник містить шістнадцять практичних робіт.

Для підвищення інтересу до вивчення предмета підручник, крім основного матеріалу, містить рубрики: «Для тих, хто хоче знати більше», «Чи знаєте ви, що...».

УДК 004(075.3)
ББК 32.97я721

ISBN 978-966-11-0692-4


© Ривкінд Й.Я., Лисенко Т.І.,
Чернікова Л.А., Шакоцько В.В., 2016
© Видавництво «Генеза», оригінал-макет, 2016


ШАНОВНІ ВОСЬМИКЛАСНИКИ!


Ви продовжуєте вивчення цікавого та важливого предмета – **інформатики**.

У попередніх класах ви вже навчилися виконувати операції з об'єктами операційної системи комп'ютера, створювати інформаційні моделі, опрацьовувати графічні, текстові, числові та мультимедійні дані, шукати відомості та листуватися в Інтернеті, складати алгоритми для виконавця **Рудий кіт**, розв'язувати компетентнісні задачі та виконувати навчальні проекти.




На уроках інформатики у 8-му класі ви ознайомитеся з кодуванням даних, розглянете апаратне та програмне забезпечення комп'ютера, продовжите опрацьовувати текстові та числові дані, розв'язувати компетентнісні задачі. Ви навчитесь створювати файли з відео, опрацьовувати аудіодані, розробляти та виконувати програмні проекти в середовищі програмування. Уміння застосовувати набуті знання ви відпрацюєте під час виконання навчальних проектів. І в цьому вам допоможе цей підручник.

Навчальний матеріал підручника поділено на розділи. Кожний розділ складається з пунктів, які, у свою чергу, містять підпункти. На початку кожного пункту наведено запитання для повторення вивченого. Відповіді на них полегшать розуміння та засвоєння нового матеріалу. Ці запитання позначено .







Уважно читайте матеріал, викладений у підручнику. Основне зібрано в кінці кожного пункту в окремій рубриці  **«Найважливіше в цьому пункті»**.

Наприкінці кожного пункту розміщено запитання для самоконтролю  **«Дайте відповіді на запитання»**. Рекомендуємо вам після вивчення навчального матеріалу пункту дати відповіді на них. Біля кожного запитання стоїть позначка, яка означає, що правильна відповідь на це запитання відповідає:

- – початковому й середньому рівням навчальних досягнень;
- – достатньому рівню навчальних досягнень;
- * – високому рівню навчальних досягнень.

Так само позначено й рівні практичних завдань у рубриці  **«Виконайте завдання»**, яку наведено після кожного пункту. Завдання, які автори рекомендують для роботи вдома, позначено . Над завданнями з позначкою  доцільно працювати в парах або невеликих групах.

Крім основного матеріалу, пункти підручника містять рубрики:

-  «Для тих, хто хоче знати більше»;
-  «Чи знаєте ви, що...»;
-  «Цікаві факти з історії»;
-  «Для тих, хто працює з ОС Linux»;
-  «Для тих, хто працює з LibreOffice»;
-  «Для тих, хто працює з Microsoft Office 2010»;
- «Тлумачний словник термінів і понять» (на зеленому тлі).

У кінці підручника розміщено **Словничок**.

Файли-заготовки для виконання завдань і практичних робіт розміщено в Інтернеті за адресою <http://allinf.at.ua>

***Бажаємо вам успіхів у вивченні найцікавішої
та найсучаснішої науки – ІНФОРМАТИКИ!***

Автори

Розділ 1. Кодування даних

У цьому розділі ви дізнаєтеся про:

опрацювання даних
як інформаційний
процес

кодування та
декодування
повідомлень

таблиці
кодів символів

двійкове
кодування

одиниці вимірювання
довжини двійкового
коду

1.1. ОПРАЦЮВАННЯ ДАНИХ ЯК ІНФОРМАЦІЙНИЙ ПРОЦЕС. КОДУВАННЯ ТА ДЕКОДУВАННЯ ПОВІДОМЛЕНЬ



1. Які інформаційні процеси ви знаєте? У чому сутність процесу опрацювання повідомлень?
2. Які існують способи подання повідомлень?
3. Що таке дані? Наведіть приклади даних.

ОПРАЦЮВАННЯ ДАНИХ ЯК ІНФОРМАЦІЙНИЙ ПРОЦЕС

У сучасному суспільстві чимало професій пов'язано з опрацюванням певних даних, поданих словами, числами, графічними зображеннями тощо. Бухгалтери здійснюють нарахування заробітної платні, спираючись на дані про обсяги виконаних робіт. Лікарі ставлять діагноз пацієнтам на основі результатів різних аналізів. Синоптики складають прогноз погоди, аналізуючи дані про рух повітряних мас. Астрономи розраховують траєкторії руху космічних об'єктів за даними про їх розташування. Художник створює ілюстрації до книги, ознайомившись з її змістом (мал. 1.1). Виконують опрацювання даних і учні, розв'язуючи задачі, пишучи твори, будуючи графіки функцій тощо.



Мал. 1.1. Опрацювання даних представниками різних професій

Як ви вже знаєте, **опрацювання даних** – це процес отримання нових даних з наявних.

Так, якщо ми розв'язуємо математичну задачу, то з наявних (початкових, вхідних) даних умови задачі ми отримуємо нові (вихідні) дані – розв'язок задачі. Під час визначення переможців легкоатлетичного забігу опрацювання даних полягає в розміщенні прізвищ учасників забігу у списку відповідно до часу подолання ними дистанції від найменшого до найбільшого значення часу. Під час побудови діаграми в табличному процесорі з наявних числових даних утворюються нові дані, подані графічним способом.

Опрацьовувати дані можна з використанням різноманітних пристроїв, у тому числі з використанням комп'ютера. Для цього потрібно подати дані у вигляді, придатному для опрацювання пристроями комп'ютера, та визначити правила, за якими повинно відбуватися опрацювання.

КОДУВАННЯ ПОВІДОМЛЕНЬ

З метою зберігання, передавання, опрацювання, захисту повідомлень часто змінюють спосіб їх подання.



Усні повідомлення записують на папері, замінюючи звуки людської мови літерами алфавіту. При цьому виконується збереження повідомлень. Під час розмови по телефону звуки перетворюються на електричні сигнали. Це робить можливим передавання повідомлень на великі відстані. Розв'язуючи задачу на уроці математики, слова, що позначають числа, записують цифрами, а математичні операції – спеціальними знаками. Це спрощує виконання операцій з опрацювання повідомлень. Для захисту змісту повідомлень від сторонніх осіб проводять шифрування, замінюючи літери в тексті повідомлення іншими літерами, числами або умовними позначеннями.

Процес заміни однієї послідовності сигналів, якою подане повідомлення, іншою послідовністю сигналів називають **кодуванням повідомлення**.

Кодування широко використовується в сучасному житті для подання повідомлень у вигляді, що придатний для опрацювання не лише людиною, а й технічними пристроями. Так, відомості про товари кодують з використанням штрих-кодів (мал. 1.2). Цифрами на штрих-коді позначено відомості про товар, країну та підприємство, що є виробником цього товару. Цей числовий код призначений для опрацювання людиною та використовується для впорядкованого зберігання даних про товари. А товщина ліній штрих-коду та відстань між ними кодують ті самі відомості. Цей графічний код сприймається спеціальними сканерами та опрацьовується далі з використанням комп'ютера.

Для кодування повідомлень визначають не лише набір символів, знаків, сигналів, які будуть використані для заміни складових заданого повідомлення, а й правила, за якими здійснюється ця заміна.

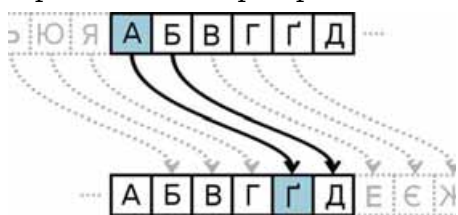
Наприклад, з історії добре відомий шифр, яким користувався для секретного листування зі своїми генералами римський імператор Гай Юлій Цезар (100–44 рр. до н. е.). Правила кодування полягають у тому, що кожна літера в тексті повідомлення замінюється іншою, що міститься в алфавіті на відстані кількох позицій від заданої літери (мал. 1.3).

Так, закодувавши повідомлення «привіт» з використанням шифру Цезаря зі зсувом на 4 літери праворуч, отримаємо повідомлення «уфкелц».

Код (лат. *codex* – книга) – система умовних знаків, сигналів для передавання, опрацювання та збереження повідомлень.



Мал. 1.2. Штрих-код товару



Мал. 1.3. Правила заміни символів за шифром Цезаря зі зсувом на 4 літери праворуч



ДЕКОДУВАННЯ ПОВІДОМЛЕНЬ

Процес отримання початкового повідомлення із закодованого називають **декодуванням** повідомлення.

Декодування повідомлень відбувається, коли ми читаємо вголос надрукований текст, виконуємо музичний твір по нотах, розшифровуємо повідомлення, визначаємо за штрих-кодом країну, в якій виготовлено товар, тощо.

Для декодування повідомлення, закодованого шифром Цезаря зі зсувом на 4 літери праворуч, потрібно кожен літеру закодованого повідомлення замінити іншою, що розміщена в алфавіті на 4 позиції ліворуч від заданої. Таким чином, декодувавши повідомлення «*зтдфкн зисб*», отримаємо «*добрий день*».

Кодування та декодування повідомлень є прикладами процесів опрацювання даних.



Найважливіше в цьому пункті

Опрацювання даних – це процес отримання нових даних з наявних.

Кодування повідомлення – це процес заміни однієї послідовності сигналів, якою подано повідомлення, іншою послідовністю сигналів.

Кодування повідомлень виконується з метою їх зберігання, передавання, опрацювання, захисту.

Для кодування повідомлень визначають набір сигналів, які будуть використані для заміни сигналів заданого повідомлення, та правила, за якими здійснюється ця заміна.

Декодування повідомлення – це процес отримання початкового повідомлення із закодованого.

Кодування та декодування повідомлень є прикладами процесів опрацювання даних.



Дайте відповіді на запитання

- 1°. У чому полягає процес опрацювання даних?
- 2°. Які нові дані можуть бути отримані в результаті опрацювання даних про:
 - а) кількість та ціну проданих у магазині товарів;
 - б) зріст учнів класу;
 - в) відстань між двома населеними пунктами та швидкість руху потягу, що курсує між ними?
- 3°. У чому полягає процес кодування повідомлень?
- 4°. З якою метою кодують повідомлення?
- 5*. З якою метою звуки музики кодують нотами; повідомлення про особливості руху по дорозі кодують дорожніми знаками?
- 6*. Повідомлення про що можна закодувати з використанням кольорів; графічних позначень?
- 7°. Що потрібно попередньо визначити для кодування повідомлення?
- 8*. За якими правилами у грі «Морський бій» кодують положення кораблів? З якою метою?
- 9°. У чому полягає процес декодування повідомлень?



Виконайте завдання

- 1°. Наведіть приклади опрацювання даних.
- 2°. Наведіть приклади кодування повідомлень, які використовуються в музиці, фізиці, хімії, географії та ін.
- 3°. Закодуйте з використанням шифру Цезаря зі зсувом на 4 літери праворуч повідомлення:
 - а) *інформатика*;
 - б) *моя школа*;
 - в) *кодування та декодування повідомлень*.

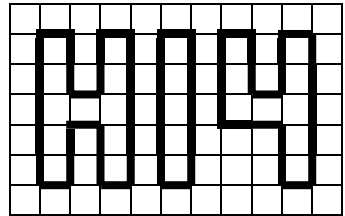


- 4°. Повідомлення закодоване шифром Цезаря зі зсувом на 3 літери праворуч. Декодуйте повідомлення:
 - а) *дзузфзпа*;
 - б) *жсгуг тсесжг*;
 - в) *ргдъгоарйм укн усїтсьгдфв*.

- 5°. Зображення, створене олівцем на аркуші паперу у клітинку, закодоване за таким правилом: стрілка вказує напрямок, у якому переміщується олівець по лінії сітки, а число вказує кількість клітинок для переміщення. Відтворіть закодоване зображення:

- а) $\uparrow 5 \rightarrow 3 \downarrow 5 \leftarrow 1 \uparrow 4 \leftarrow 1 \downarrow 4 \leftarrow 1$;
- б) $\leftarrow 3 \downarrow 5 \rightarrow 3 \uparrow 1 \leftarrow 2 \uparrow 1 \rightarrow 2 \uparrow 1 \leftarrow 2 \uparrow 1 \rightarrow 2 \uparrow 1$.

- 6°. Закодуйте наведені зображення за таким правилом: стрілка вказує напрямок, у якому переміщується олівець по лінії сітки, а число – кількість клітинок для переміщення.



- 7*. Доповніть правила кодування, наведені у попередньому завданні, так, щоб можна було позначати переміщення олівця, при якому не відбувається малювання.

- 8°. Один з найдавніших шифрів розробив Полібій (III ст. до н.е.) – грецький історик, полководець, державний діяч. У шифрі, який назвали «квадрат Полібія», кожна літера алфавіту (або пара літер, що відповідають близьким за вимовою звукам) міститься в таблиці. Під час кодування повідомлення кожна літера замінюється парою цифр – номерами стовпця та рядка таблиці, на перетині яких вона розміщена. Для кодування повідомлень українською мовою може бути використана таблиця, зображена праворуч.

	1	2	3	4	5	6
1	А	Б	В	Г/Ґ	Д	Е
2	Є	Ж	З	И/Й	І/Ї	К
3	Л	М	Н	О	П	Р
4	С	Т	У	Ф	Х	Ц
5	Ч	Ш	Щ	Ь	Ю	Я

Квадрат Полібія

Декодуйте повідомлення, закодовані за розглянутими вище правилами з використанням наведеної таблиці:

- а) 53 43 13 52 21 52 42; б) 62 43 51 34 31 11 33 33 65.



- 9*. Закодуйте повідомлення, використавши правила кодування, наведені в попередньому завданні:
- інформатика;
 - повідомлення.
- 10*. Запропонуйте власні правила кодування символів тексту та закодуйте за цими правилами повідомлення:
- сьогодні гарний день;
 - мені подобається футбол.
- 11*. Знайдіть відомості та підготуйте повідомлення про способи кодування:
- азбуку Морзе;
 - шрифт Брайля;
 - семафорну азбуку;
 - мову квітів «селама».



1.2. КОДУВАННЯ СИМВОЛІВ



- Що таке кодування повідомлень? З якою метою кодують повідомлення?
- Наведіть приклади кодування повідомлень.
- Що таке декодування повідомлень?

КОДУВАННЯ СИМВОЛІВ ТЕКСТУ

Для опрацювання текстових повідомлень з використанням комп'ютера символи повідомлення кодують числами.

Телетайп (лат. *tele* – віддалений, англ. *type* – друкування) – електро-механічна друкуюча машина, яку використовують для передавання текстових повідомлень дротами.

Для кодування тексту використовують **таблиці кодів символів**, у яких для кожного символу, що може бути використаний у текстовому повідомленні, ставиться у відповідність деяке число. У 1963 р. у США було розроблено набір

таких кодів символів для передавання повідомлень телетайпом. Пізніше він став стандартом для використання в комп'ютерній техніці й отримав назву **таблиці кодів символів ASCII** (англ. *American Standard Code for Information Interchange* – американський стандартний код для обміну інформацією).

У таблиці **ASCII** літерам англійського алфавіту, цифрам, розділовим знакам, символам редагування та форматування тексту ставляться у відповідність числа від 0 до 127 (табл. 1.1).

Таблиця 1.1

Групи кодів символів таблиці ASCII

Діапазон кодів	Група символів	Приклад коду	Відповідний символ
Від 0 до 31, 127	Спеціальні символи	10	Символ, що відповідає перенесенню курсора на новий рядок
		13	Символ, що відповідає поверненню курсора на початок рядка
		27	Символ, що відповідає натисканню клавіші Esc



Продовження таблиці 1.1

Діапазон кодів	Група символів	Приклад коду	Відповідний символ
Від 32 до 64, від 91 до 96, від 123 до 126	Розділові знаки та цифри	32	пропуск
		48	цифра 0
		123	{
Від 65 до 90	Великі літери англійського алфавіту	65	A
		66	B
		90	Z
Від 97 до 122	Малі літери англійського алфавіту	97	a
		98	b
		122	z

Текстові символи, що кодуються в таблиці ASCII числами від 32 до 127, наведено у додатку 1, який можна переглянути на сайті **Інформатика для всіх** (allinf.at.ua) у розділі **Матеріали до підручників. 8 клас. Підручник.**

Таблиця кодів символів ASCII містить коди літер лише англійського алфавіту. Для кодування літер інших алфавітів було розроблено інакші таблиці кодів символів. Наприклад, таблиці **KOI8-U** (KOI – код обміну інформацією) і **Windows-1251** містять без змін усі коди таблиці ASCII, а також – коди літер кирилиці. Цим літерам ставляться у відповідність натуральні числа від 128 до 255.

Кирилиця – алфавіт для групи слов'янських мов, до яких належать українська, російська, білоруська, болгарська та інші мови.

Англійська мова та деякі інші західноєвропейські мови мають алфавіт, який називають **латиницею**.

Таблиця кодів символів **Windows-1251** є стандартом для кодування літер кирилиці в операційній системі **Windows**. У ній, наприклад, літері «а» українського алфавіту ставиться у відповідність число 224, літері «і» – число 179, літері «г» – число 180 та ін. Текстові символи, що кодуються у таблиці **Windows-1251** числами від 128 до 255, наведені в додатку 2. Додаток можна знайти на сайті **Інформатика для всіх** (allinf.at.ua) у розділі **Матеріали до підручників. 8 клас. Підручник.**

Цілих чисел від 0 до 255 вистачає, щоб закодувати символи двох алфавітів – латиниці й кирилиці та деякі інші символи. Але для кодування символів інших алфавітів (грецького чи арабського алфавітів, ієрогліфів тощо) потрібно значно більше значень кодів. Для них розроблено таблицю кодів символів **Юнікод** (англ. *Unicode* – уніфіковане кодування).

Таблиця **Юнікод** складається з 17 наборів по 65 536 значень кодів у кожному та дає можливість закодувати 1 114 112 різних символів, тобто майже всі символи писемності всіх світових мов. Як і в інших таблицях кодів, у **Юнікод** незмінними залишаються перші 128 значень кодів, що відповідають таблиці ASCII. Окремий розділ у таблиці **Юнікод** міс-

тить коди літер кирилиці. Наприклад, літері «а» українського алфавіту ставиться у відповідність код 53 424, літері «і» – код 53 654, літері «г» – код 53 905 та ін.

Наразі у новітніх операційних системах використовується таблиця кодів **Юнікод**.

Порівняння структур різних таблиць кодів символів наведено в таблиці 1.2.

Таблиця 1.2

Порівняння структури таблиць кодів символів

Таблиця кодів символів	Кількість кодів символів	Символи з кодами від 0 до 127	Символи з кодами від 128 до 255	Символи з кодами, більшими за 255
ASCII	128	Літери англійського алфавіту, цифри, розділові знаки, спеціальні символи	Немає	Немає
KOI8-U	256	Як у таблиці ASCII	Символи кирилиці та деякі інші символи	Немає
Windows-1251	256	Як у таблиці ASCII	Символи кирилиці та деякі інші символи	Немає
Юнікод	1 114 112	Як у таблиці ASCII	Символи алфавітів різних мов світу та деякі інші символи	



Для тих, хто хоче знати більше

У таблиці **Юнікод** містяться коди не лише літер та цифр, а й символів, які позначають торговельні марки, грошові одиниці, символи транскрипцій, ідеограми тощо. Наприклад, кодом символу української грошової одиниці гривні ₴ є число 8 372, кодом ідеограми *чоловік* ♂ є число 10 080, а ідеограми *жінка* ♀ – число 10 081 тощо.

Для вставлення символів у документ можна застосовувати їх коди з таблиці **Юнікод**. Для цього потрібно натиснути та утримувати клавішу **Alt** і набрати код символу на додатковій цифровій клавіатурі. Так можуть бути вставлені, наприклад, символи *сніговик* ❄️ – код 9 731, *серце у вигляді квітки* 💖 – код 10 087, *нота* 🎵 – код 9 834, *шахова фігура кінь* 🐎 – код 9 822 та інші. Знайти символ та відповідний йому код можна на сайті **Таблиця символів Юнікода** (unicode-table.com). Але якщо вибраний вами символ відсутній серед шрифтів на вашому комп'ютері, то він не буде відображатися в документі.

Ідеограма (давньогрец. *ідея* – ідея, *уріама* – писемний знак, буква) – писемний знак, що передає, на відміну від букви, не звук певної мови, а деяке поняття, ідею.




КОДУВАННЯ СИМВОЛІВ ВЕБ-СТОРИНОК

Особливості кодування тексту з використанням різних таблиць кодів символів можна побачити також під час перегляду веб-сторінок. Іноді під час відкриття веб-сторінки таблиця кодів символів обирається браузером неправильно. У такому випадку текст веб-сторінки непридатний для розуміння (мал. 1.4).



Мал. 1.4. Вигляд веб-сторінки за неправильного вибору таблиці кодів символів

У кожного веб-браузера є інструменти для вибору таблиці кодів символів користувачами. Наприклад, у браузері **Google Chrome** для зміни таблиці кодів символів потрібно:

1. Вибрати кнопку **Налаштування та керування Google Chrome** .
2. Виконати **Інші інструменти** \Rightarrow **Кодування**.
3. Вибрати таблицю кодів символів, при якій текст веб-сторінки буде придатним для розуміння.

Для україномовних веб-сторінок найчастіше застосовують таблицю кодів символів **Юнікод (UTF-8)**, але для окремих веб-сторінок може бути використане кодування **Кирилиця (Windows-1251)** або **Кирилиця (KOI8-U)**.



Для тих, хто хоче знати більше

Таблиця символів в операційній системі Windows

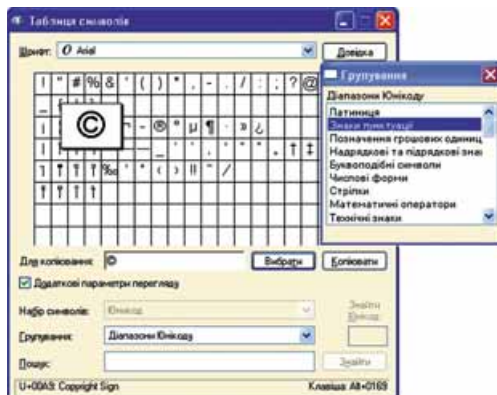
Серед службових програм в операційній системі **Windows** є програма **Таблиця символів**, що призначена для перегляду, пошуку та вставлення в текстові документи різноманітних символів, у тому числі тих, які відсутні на клавіатурі. Для запуску потрібно виконати **Пуск** \Rightarrow **Усі програми** \Rightarrow **Стандартні** \Rightarrow **Службові** \Rightarrow **Таблиця символів**, після чого відкриється вікно програми (мал. 1.5).

Використовуючи цю програму, можна побачити різні групи символів, що призначені для вставлення в документи. Для цього потрібно:

1. Встановити позначку прапорця **Додаткові параметри перегляду** у вікні програми.
2. У полі **Набір символів** у нижній частині вікна вибрати **Юнікод**.
3. У полі **Групування** вибрати **Діапазони Юнікоду**.
4. У списку вікна **Групування**, що відкриється, вибрати діапазон кодів, символи з якого потрібно вставити в документ (мал. 1.6).



Мал. 1.5. Вікно програми **Таблиця символів**



Мал. 1.6. Вікно програми **Таблиця символів** із символами групи **Знаки пунктуації**

Якщо в таблиці вибрати деякий символ, то в рядку статусу вікна програми ліворуч буде відображено код цього символу в таблиці кодів символів **Юнікод** та його опис англійською мовою, а праворуч – сполучення клавіш, яке потрібно натиснути для вставлення цього символу в документ. Наприклад, на малюнку 1.6 вибрано символ © – знак авторського права, у рядку статусу вікна наведено його опис – *U+00A9 Copyright Sign* та сполучення клавіш *Alt+0169* для вставлення цього символу в документ.

Вибрані в таблиці символи можна вставити в документи, що створюються в різних програмах – текстовому процесорі, редакторі презентацій, табличному процесорі та ін. Для вставлення символу в документ потрібно:

1. Вибрати символ у таблиці із символами.
2. Виконати **Вибрати** ⇒ **Копіювати**.
3. Вибрати місце в документі.
4. Вставити скопійований символ з **Буферу обміну**.

Після вибору кнопки **Вибрати** символ виводиться в поле **Для копіювання**. Якщо вибрати ще один символ та кнопку **Вибрати**, то в полі відобразяться обидва символи. Таким способом можна підготувати для копіювання кілька символів. Усі вони разом будуть скопійовані та вставлені в документ.


Найважливіше в цьому пункті

Для опрацювання текстових повідомлень з використанням комп'ютера символи повідомлення кодують числами. Для цього можуть використовуватися різні таблиці кодів символів, наприклад **ASCII**, **KOI8-U**, **Windows-1251**, **Юнікод**.


Таблиця кодів символів **ASCII** містить коди 128 символів, **KOI8-U**, **Windows-1251** – 256 символів, **Юнікод** – 1 114 112 символів. У всіх таблицях кодів символів незмінними залишаються перші 128 кодів, що відповідають таблиці **ASCII**.

У кожного веб-браузера є інструменти для вибору таблиці кодів символів, з використанням якої будуть відображатися символи на веб-сторінці.





 Службова програма **Таблиця символів** операційної системи **Windows** призначена для перегляду, пошуку та вставлення в документи символів з різних шрифтів, поданих з використанням різних таблиць кодів символів.

Дайте відповіді на запитання

- 1°. Як кодують символи текстових повідомлень під час опрацювання їх з використанням комп'ютера?
- 2°. Які таблиці кодів символів ви знаєте?
- 3°. Які символи можуть бути закодовані з використанням таблиць кодів символів **ASCII**, **KOI-8U**, **Windows-1251**, **Юнікод**?
- 4*. Що спільного та чим відрізняються таблиці кодів символів **ASCII**, **KOI-8U**, **Windows-1251**, **Юнікод**?
- 5°. У яких ситуаціях потрібно змінювати таблицю кодів символів під час перегляду веб-сторінок?
- 6°. Як вибрати таблицю кодів символів під час перегляду веб-сторінок у браузері **Google Chrome**?
-  7°. Яке призначення програми **Таблиця символів**?
- 8°. Як запустити на виконання програму **Таблиця символів**?
- 9°. Як уставити до документа символи з використанням програми **Таблиця символів**?

Виконайте завдання

- 1°. Повідомлення закодовано з використанням таблиці кодів символів **ASCII**. Декодуйте його (скористайтеся додатком 1¹):
 - а) 83 117 110;
 - б) 104 111 109 101.
-  2°. Закодуйте повідомлення з використанням таблиці кодів символів **Windows-1251** (скористайтеся додатком 2¹):
 - а) *Підручник «Інформатика»*;
 - б) ваше прізвище та ім'я.
- 3*. Окремі слова в повідомленні закодовані з використанням таблиці кодів символів **Юнікод**. Знайдіть символи за їх кодами в таблиці та декодуйте повідомлення. Запишіть отриманий текст. Для пошуку символів можна використовувати сайт **Таблиця символів Юнікода** (unicode-table.com).
Сьогодні з ранку 9925. Синоптики обіцяють 9928. Збираючись у 127979, не забудьте взяти з собою 9730.
- 4°. Перегляньте веб-сторінку **ІТ в Україні: історії та особистості** (ua.uacomputing.com) з використанням таблиць кодів символів **Кирилиця (Windows-1251)** та **Кирилиця (KOI8-U)**. Установіть, за вибору якої таблиці кодів символів текст веб-сторінки придатний для розуміння.
-  5°. Уставте до текстового документа всі символи, що є позначеннями грошових одиниць, використовуючи програму **Таблиця символів**. Збережіть створений документ у вашій папці у файлі з іменем **завдання 1.2.5** та розширенням імені, що відповідає версії текстового процесора.

¹ Додатки можна переглянути на сайті **Інформатика для всіх** (allinf.at.ua) у розділі **Матеріали до підручників. 8 клас. Підручник**.



- 6*. Уставте до текстового документа буквоподібні символи, використовуючи програму **Таблиця символів**. Збережіть створений документ у вашій папці у файлі з іменем **завдання 1.2.6** та розширенням імені, що відповідає версії текстового процесора.
- 7*. Визначте, які символи будуть уставлені до текстового документа шляхом введення їх коду за натиснутої клавіші **Alt**, для кодів 1234, 3754, 8986, 9708, 9415, 10001, 10239.

1.3. ДВІЙКОВЕ КОДУВАННЯ

- 1. У чому полягають процеси кодування та декодування повідомлень?
- 2. Які таблиці кодів символів використовують для кодування текстових повідомлень?
- 3. Що означають префікси *кіло*, *мега*, *гіга*?

ДВІЙКОВЕ КОДУВАННЯ. БІТ І БАЙТ



Мал. 1.7. Семюель Морзе

Як ви вже знаєте, для кодування даних використовують різні набори сигналів. Для кодування звуків мови використовують букви алфавіту. В українській мові літер 33, в англійській – 26. Набір сигналів для кодування числових значень складається з 10 цифр. Для кодування команд, що регулюють порядок руху на пішохідних переходах, використовують 3 кольори світлофора. Найменшу кількість сигналів для кодування повідомлень має азбука Морзе – лише 2 сигнали: *крапка* і *тире*.

З 1844 року азбуку, запропоновану американським художником Семюелем Морзе (1791–1872) (мал. 1.7), згодом названу на його честь, використовували для передавання повідомлень телеграфом. В азбуці Морзе літери текстових повідомлень кодуються послідовностями крапок і тире. Приклади кодування літер українського алфавіту наведено в таблиці 1.3.

Таблиця 1.3

Приклади кодування літер в азбуці Морзе

Літера	Значення коду в азбуці Морзе
а	· –
о	– – –
п	· – – ·
р	· · ·
т	–



Наприклад, слово *парта* кодується з використанням азбуки Морзе послідовністю сигналів «· — · — · — · — · — · —».

Кодування повідомлень з використанням сигналів лише двох видів називають **двійковим кодуванням**. Повідомлення, отримане в результаті двійкового кодування повідомлення, називають **двійковим кодом** повідомлення.

Двійкове кодування використовується і в сучасних комп'ютерах. У них усі повідомлення кодуються послідовностями сигналів двох видів. Кожний сигнал одного виду умовно позначається цифрою **0**, а другого виду – **1**.

Цифру **0** або **1** у двійковому коді повідомлення називають **біт** (англ. *binary digit* – двійкова цифра).

З таблиці 1.3 видно, що в азбуці Морзе для кодування літери «т» використовується 1 сигнал – тире, для кодування літери «а» – два, а для літери «п» – чотири сигнали. Під час передачі повідомлень телеграфом одну літеру від іншої відокремлюють проміжками в подачі сигналів. Для опрацювання даних з використанням комп'ютера це не зручно. Простіше відокремити одну літеру від іншої, коли на їх кодування буде використано однакову кількість сигналів.

Якщо для кодування повідомлення використовується 1 сигнал (біт) **0** або **1**, то можна закодувати, наприклад:

- висновок про правильність твердження: *хибне* – 0 або *істинне* – 1;
- стать людини: *жіноча* – 0 або *чоловіча* – 1;
- стан вимикача: *вимкнено* – 0 або *увімкнено* – 1

тощо.

З двох бітів можна скласти 4 (2^2) різних коди (00, 01, 10 і 11). Ними можна закодувати, наприклад, чотири сторони горизонту: 00 – північ, 01 – схід, 10 – південь, 11 – захід.

Із трьох бітів можна скласти вже 8 (2^3) різних кодів (000, 001, 010, 011, 100, 101, 110, 111). Ними можна закодувати, наприклад, номери рядків або стовпців шахівниці.

Із чотирьох бітів можна скласти $2^4 = 16$ кодів, з п'яти – $2^5 = 32$ коди і 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. Саме така кількість кодів міститься, наприклад, у таблиці кодів символів **Windows-1251**.

Послідовність із восьми бітів називають **байт**.

1 байт = 8 бітів.

ДОВЖИНА ДВІЙКОВОГО КОДУ ПОВІДОМЛЕНЬ

Якщо символ повідомлення кодується послідовністю з 8 бітів, то довжина двійкового коду цього символу дорівнює 8 бітів, або 1 байт.

Довжина двійкового коду повідомлення – це кількість байтів у двійковому коді цього повідомлення.

Наприклад, повідомлення *Інформатика – цікавий предмет!* містить 30 символів (включаючи символи пропусків, тире, знак оклику). Якщо



кожний символ кодувати двійковим кодом довжиною 1 байт, то довжина двійкового коду такого повідомлення дорівнюватиме 30 байтам.

Як ви знаєте, у таблиці кодів символів **Windows-1251** кожному символу ставиться у відповідність ціле число від 0 до 255. Кожне із цих чисел може бути закодоване одним байтом. Приклади такого кодування наведено в таблиці 1.4.

Таблиця 1.4

Приклади двійкового кодування

Символ	Числовий код символу	Відповідний байт
!	33	00100001
@	64	01000000
W	103	01100111
ю	254	11111110
я	255	11111111

Отже, кожному символу в таблиці кодів символів **Windows-1251** відповідає двійковий код довжиною 1 байт.

Щоб виконати двійкове кодування текстового повідомлення, можна кожний його символ замінити двійковим кодом, що відповідає числу з таблиці **Windows-1251**. Саме так кодуються текстові документи, створені в текстовому редакторі **Блокнот**, під час їх збереження на носії даних, якщо під час збереження використовувати кодування **ANSI** (англ. *American National Standards Institute* – Американський національний інститут стандартів). Довжина двійкового коду повідомлення, що міститься в такому документі, буде дорівнювати в байтах кількості символів у текстовому документі. Треба врахувати, що кожне натискання клавіші **Enter** кодується двома спеціальними символами – символом переходу на новий рядок і символом повернення на початок рядка.

Графічні, звукові, відеоповідомлення для їх опрацювання комп'ютером також кодуються двійковими кодами.

Довжину двійкового коду повідомлень називають **обсягом даних**.

КРАТНІ ОДИНИЦІ ВИМІРЮВАННЯ ДОВЖИНИ ДВІЙКОВОГО КОДУ

Для позначення довжин двійкового коду повідомлень використовують кратні одиниці вимірювання, які утворюються з використанням префіксів *кіло*, *мега*, *гіга*, *тера* та ін. Перелік цих префіксів для позначення кратних одиниць вимірювання визначено у Міжнародній системі одиниць (СИ). Там також наведено відповідні множники. Але історично склалось так, що ці префікси (*кіло*, *мега*, *гіга*, *тера*) в інформатиці трактуються інакше, ніж, наприклад, у фізиці. Із шостого класу вам відомо, що:

- 1 Кбайт (кілобайт) = 2^{10} байтів = 1024 байти;
- 1 Мбайт (мегабайт) = 2^{10} Кбайт = 2^{20} байтів = 1 048 576 байтів;
- 1 Гбайт (гігабайт) = 2^{10} Мбайт = 2^{20} Кбайт = 2^{30} байтів;
- 1 Тбайт (терабайт) = 2^{10} Гбайт = 2^{20} Мбайт = 2^{30} Кбайт = 2^{40} байтів.



Так сталося тому, що в комп'ютері використовуються двійкові коди, для опрацювання яких зручно оперувати степенями числа 2, а не степенями числа 10. І оскільки $2^{10} = 1024$, що приблизно дорівнює 1000, то саме 2^{10} байтів = 1024 байти і стали називати **кілобайт**. Аналогічно, 2^{10} кілобайт стали називати **мегабайт** і т. д.

ПЕРЕВЕДЕННЯ ОДИНИЦЬ ВИМІРЮВАННЯ ДОВЖИНИ ДВІЙКОВОГО КОДУ В ІНШІ

Довжину двійкового коду повідомлень можна вказувати в бітах, байтах та у кратних їм одиницях. Розглянемо кілька задач на переведення одних одиниць вимірювання в інші під час визначення довжини двійкового коду повідомлення.

Задача 1. Довжина двійкового коду повідомлення складає 4,5 Мбайта. Виразити це значення в байтах.

Розв'язання. Для переходу від мегабайтів до байтів спочатку виразимо довжину двійкового коду цього повідомлення в кілобайтах. Враховуємо, що 1 Мбайт = 1024 Кбайт.

$$4,5 \text{ Мбайта} \cdot 1024 = 4608 \text{ Кбайт.}$$

Переведемо кілобайти в байти, враховуючи, що 1 Кбайт = 1024 байти.

$$4608 \text{ Кбайт} \cdot 1024 = 4\,718\,592 \text{ байти.}$$

Відповідь: 4,5 Мбайта = 4 718 592 байти.

Задача 2. Двійковий код повідомлення складається з 4 194 304 000 бітів. Виразити довжину двійкового коду цього повідомлення цілим числом у найбільших можливих одиницях.

Розв'язання. Переведемо біти в байти, враховуючи, що 1 байт = 8 бітів.

$$4\,194\,304\,000 \text{ бітів} : 8 = 524\,288\,000 \text{ байтів.}$$

Отримане число більше за множник 1024, який використовується для переходу від байтів до кілобайтів у визначенні довжини двійкового коду. Тому переведемо байти в кілобайти.

$$524\,288\,000 \text{ байт} : 1024 = 512\,000 \text{ Кбайт.}$$

Аналогічно міркуючи, переведемо кілобайти в мегабайти.

$$512\,000 \text{ Кбайт} : 1024 = 500 \text{ Мбайт.}$$

Отримане значення менше ніж 1024, тому перейти до більшої одиниці зі збереженням цілого результату неможливо.

Відповідь: 4 194 304 000 бітів = 500 Мбайт.

Задача 3. Обчислити наближено довжину двійкового коду тексту підручника з інформатики (не враховуючи форматування та малюнки).

Розв'язання. Уважатимемо, що текст закодовано з використанням таблиці кодів символів **Windows-1251**, тобто довжина двійкового коду кожного символу складає 1 байт.

У кожному рядку тексту міститься близько 60 символів, тобто довжина двійкового коду рядка складає приблизно 60 байтів. На кожній сторінці розміщено близько 50 рядків. Отже, довжина двійкового коду сторінки складає приблизно $60 \cdot 50 = 3000$ (байтів). У підручнику близько 250 сторінок, тобто довжина двійкового коду підручника складає приблизно $3000 \cdot 250 = 750\,000$ (байтів).



Перейдемо до крупнішої одиниці вимірювання довжини двійкового коду.

$$750\ 000 \text{ байтів} : 1024 \approx 732,4 \text{ Кбайта.}$$

Відповідь: довжина двійкового коду тексту підручника з інформатики приблизно 732,4 Кбайта.



Найважливіше в цьому пункті

Кодування повідомлень з використанням сигналів лише двох видів називають **двійковим кодуванням**. Повідомлення, отримане в результаті двійкового кодування початкового повідомлення, називають **двійковим кодом** повідомлення.

Цифру 0 або 1 у двійковому коді повідомлення називають **біт**.

Послідовність із восьми бітів називають **байт**.

$$1 \text{ байт} = 8 \text{ бітів.}$$

Довжина двійкового коду повідомлення – це кількість байт у двійковому коді цього повідомлення.

Кожному символу в таблиці кодів символів **Windows-1251** відповідає двійковий код довжиною 1 байт.

$$1 \text{ Кбайт (кілобайт)} = 2^{10} \text{ байтів} = 1024 \text{ байти;}$$

$$1 \text{ Мбайт (мегабайт)} = 2^{10} \text{ Кбайт} = 2^{20} \text{ байтів} = 1\ 048\ 576 \text{ байтів;}$$

$$1 \text{ Гбайт (гігабайт)} = 2^{10} \text{ Мбайт} = 2^{20} \text{ Кбайт} = 2^{30} \text{ байтів;}$$

$$1 \text{ Тбайт (терабайт)} = 2^{10} \text{ Гбайт} = 2^{20} \text{ Мбайт} = 2^{30} \text{ Кбайт} = 2^{40} \text{ байтів.}$$



Дайте відповіді на запитання

- 1°. У чому полягає двійкове кодування повідомлень?
- 2°. Що таке 1 біт?
- 3°. Чому дорівнює 1 байт?
- 4°. Що розуміють під довжиною двійкового коду повідомлення?
- 5°. Якою є довжина двійкового коду кожного символу в таблиці кодів символів **Windows-1251**?
- 6°. Якою буде довжина двійкового коду повідомлення, створеного в текстовому редакторі **Блокнот** і збереженому в кодуванні **ANSI**, у кодуванні **Юнікод**?
- 7*. На скільки відрізнятиметься довжина двійкового коду повідомлення, у якому зазначене ваше ім'я, якщо його створити в текстовому редакторі **Блокнот** і зберегти в кодуванні **ANSI** у двох варіантах: записаним в один рядок та записаним у стовпчик по одній літері на рядок?
- 8°. Чому дорівнює 1 кілобайт, 1 мегабайт, 1 гігабайт, 1 терабайт?
- 9°. Яку математичну операцію потрібно виконати для переведення довжини двійкового коду повідомлень:
 - а) з бітів у байти;
 - б) з байтів у кілобайти;
 - в) з мегабайтів у кілобайти?

**Виконайте завдання**

- 1°. Наведіть приклади, де використовується двійкове кодування повідомлень.
- 2°. Закодуйте з використанням азбуки Морзе слова *порт, рота, торт*. Порівняйте кількість сигналів у двійковому коді цих слів.
- 3°. Визначте довжину двійкового коду повідомлень за умови, що повідомлення закодовані з використанням таблиці кодів символів **Windows-1251**:

а) Таблиця кодів символів *Windows-1251*;

б) В алфавіті азбуки Морзе два символи – крапка та тире.

- 4°. Порівняйте довжину двійкового коду текстового повідомлення *Людина без друзів – що дерево без коріння*, створеного в текстовому редакторі **Блокнот**, під час збереження його з використанням різних кодових таблиць.

- 5°. Заповніть пропуски:

а) 5 байтів = _____ бітів

б) 4096 байтів = _____ Кбайт

в) 10 Кбайт = _____ бітів

г) 3 Мбайти = _____ байтів



- 6°. Заповніть пропуски:

а) 2 байти = _____ бітів

б) 8192 байти = _____ Кбайт

в) 2 Кбайт = _____ бітів

г) 100 Мбайт = _____ Кбайт

- 7*. В одному рядку деякого тексту розміщується в середньому 60 символів, а на одній сторінці – 40 рядків. Скільки закодованих сторінок цього тексту можна взяти, щоб довжина двійкового коду не перевищила 8 Мбайт, якщо для кодування використати таблицю **Windows-1251**?

ПРАКТИЧНА РОБОТА № 1**«Розв'язування задач на визначення довжини двійкового коду текстових даних»**

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Обчисліть довжину двійкового коду деякого тексту за умови, що символи тексту закодовані з використанням таблиці кодів символів **Windows-1251**. У тексті 32 сторінки, кожна сторінка містить 48 рядків, у кожному рядку в середньому 56 символів. Виразіть довжину двійкового коду цього тексту цілим числом у найбільших можливих одиницях.



2. Визначте, використовуючи текстовий редактор **Блокнот**, довжину двійкового коду повідомлення *Практична робота № 1* за умови, що повідомлення закодоване з використанням таблиці кодів символів:

- а) **Windows-1251**;
- б) **Юнікод**.

Збережіть текстові документи з повідомленням у файлах з іменами **практична 1 Windows.txt** та **практична 1 Юнікод.txt**.

3. Заповніть у табличному процесорі таблицю:

<i>Бітів</i>	<i>Байтів</i>	<i>Кілобайт</i>	<i>Мегабайт</i>
614 400			
	524 288		
		256	
			4

Збережіть результат у файлі з іменем **практична 1.xlsx**.

Розділ 2. Апаратно-програмне забезпечення комп'ютера

У цьому розділі ви дізнаєтеся про:

архітектуру
комп'ютера

історію засобів
опрацювання
повідомлень

класифікацію та
загальну характеристику
програмного
забезпечення

системне та службове
програмне забезпечення

класифікацію та основні
функції операційних
систем

архівування та
стиснення даних

пристрої комп'ютера, їх
призначення та основні
характеристики:

процесори

зовнішні та внутрішні
запам'ятовуючі пристрої

монітори

принтери

мультимедійне
обладнання



2.1. АРХІТЕКТУРА КОМП'ЮТЕРА



1. Які пристрої входять до складу комп'ютера? Для чого вони призначені?
2. Який пристрій комп'ютера виконує опрацювання даних? Де, зазвичай, він розміщується?
3. Назвіть інформаційні процеси. Які пристрої комп'ютера забезпечують збереження даних?

АРХІТЕКТУРА КОМП'ЮТЕРА

У п'ятому класі ви вже ознайомилися зі складовими персонального комп'ютера. Розглянемо трохи детальніше схему реалізації інформаційних процесів у комп'ютері з використанням його пристроїв (мал. 2.1). Дані потрапляють до внутрішніх запам'ятовуючих пристроїв (внутрішньої пам'яті) комп'ютера через пристрої введення даних (клавіатура, мі-



Мал. 2.1. Схема реалізації інформаційних процесів у персональному комп'ютері.

крофон, миша, сканер тощо) або із зовнішніх запам'ятовуючих пристроїв (зовнішньої пам'яті) – пристроїв для роботи з жорсткими магнітними та оптичними дисками, флеш-накопичувачів, флеш-карт тощо. З внутрішньої пам'яті дані потрапляють до процесора, де опрацьовуються і результати опрацювання знову передаються до внутрішньої пам'яті, а звідти – до пристроїв виведення даних (монітор, принтер, звукові колонки тощо) або до пристроїв зовнішньої пам'яті. Усі операції в комп'ютері пов'язані

з реалізацією інформаційних процесів можливі тільки під керуванням відповідних програм.

Архітектура (грец. *αρχιτεκτονήμα*) – будівля, задум, затія.



Схема, зображена на малюнку 2.1, описує типову архітектуру персонального комп'ютера. **Архітектура комп'ютера** – це модель, що описує взаємодію пристроїв та програм для забезпечення інформаційних процесів у комп'ютері.

ПРОЦЕСОР, ЙОГО ПРИЗНАЧЕННЯ

Ви вже знаєте, що опрацювання даних у комп'ютері виконується процесором (мал. 2.2). Він є пристроєм, що забезпечує виконання комп'ютерних програм. У процесорів, як у будь-якого виконавця алгоритмів, є свій набір команд.



Мал. 2.2. Процесор Helio X20 з 10-ма ядрами

Процесор комп'ютера є його основною складовою. Він містить **пристрій керування**, який забезпечує виконання команд комп'ютерної програми, та **арифметично-логічний пристрій**, який і здійснює операції над даними. До складу сучасних процесорів входить ще й складова внутрішньої пам'яті – так звана **кеш-пам'ять**. Її призначення ми розглянемо трохи згодом.

Основною властивістю процесора є **швидкість опрацювання даних**, яка залежить від значення кількох інших властивостей: **тактової частоти роботи процесора, кількості ядер, розрядності, обсягів кеш-пам'яті** тощо (табл. 2.1).

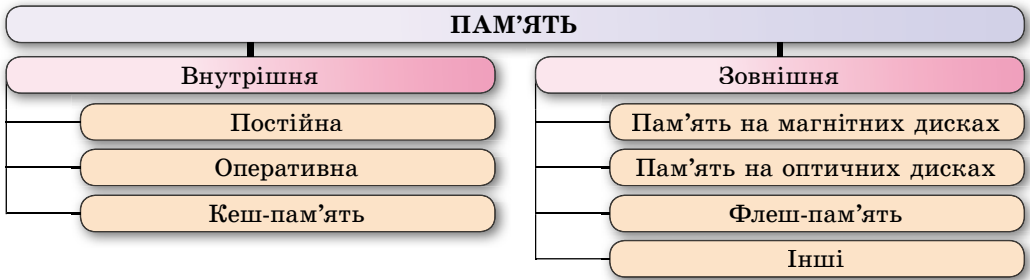
Таблиця 2.1

Приклади значень властивостей процесора для настільного і планшетного комп'ютерів

Властивість	Що характеризує	Значення властивостей процесора	
		для настільного комп'ютера	для планшетного комп'ютера
Модель	Ім'я, що надається фірмою-виробником	Intel Core i5-6600	Qualcomm Snapdragon MSM8939
Тактова частота	Частота керуючих сигналів, які узгоджують роботу пристроїв процесора	3,3 ГГц	1,8 ГГц
Кількість ядер	Кількість однакових за структурою процесорів, що об'єднані в один пристрій	4	8
Розрядність	Кількість двійкових розрядів, що можуть одночасно опрацьовуватися процесором	64 біти	64 біти

ПАМ'ЯТЬ КОМП'ЮТЕРА. ВНУТРІШНЯ ПАМ'ЯТЬ

Пам'ять комп'ютера призначена для збереження даних. Її поділяють на **внутрішню** та **зовнішню** (мал. 2.3). Основною властивістю всіх видів пам'яті є її **ємність**, що вимірюється в байтах, кілобайтах, мегабайтах, гігабайтах, терабайтах тощо.

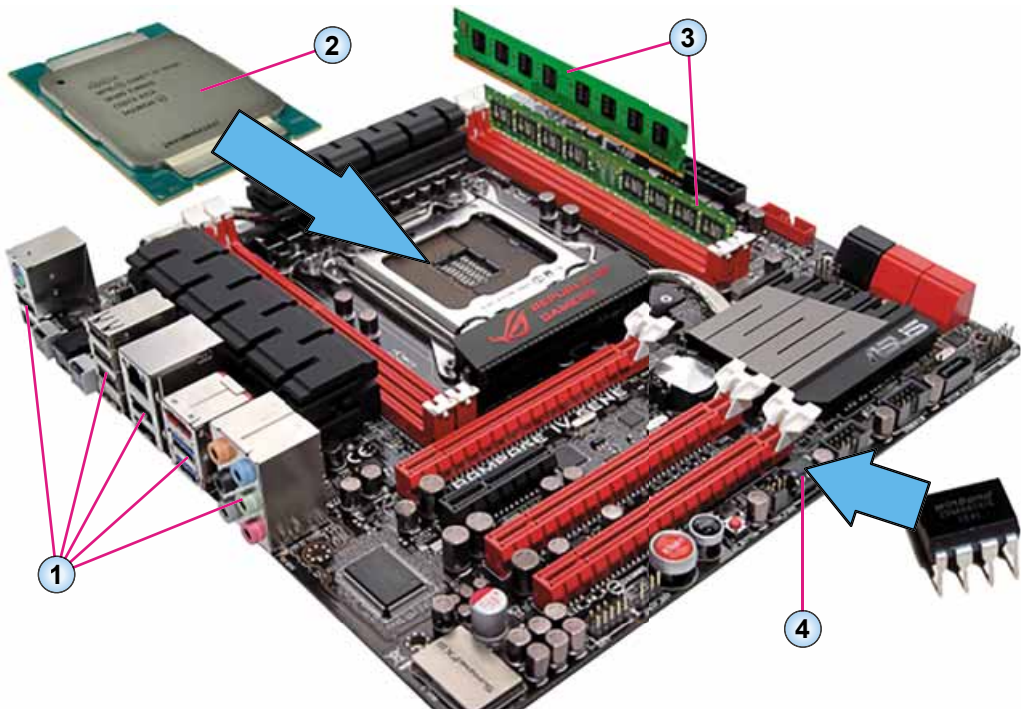


Мал. 2.3. Види пам'яті

Материнська або головна плата (англ. *motherboard* або *main-board*) – основна складова сучасних персональних комп'ютерів, яка забезпечує передачу даних між пристроями комп'ютера.

Внутрішня пам'ять служить для забезпечення роботи процесора. Пристрої внутрішньої пам'яті, як і процесор, розміщуються на материнській платі комп'ютера (мал. 2.4).

Оперативна пам'ять є основною в комп'ютері, у ній розміщуються програми і дані, які в подальшому опрацьовуються процесором. Він може отримувати дані й команди тільки з оперативної пам'яті. У неї ж записуються



- | | |
|--|--|
| 1. Роз'єми для підключення пристроїв уведення та виведення даних | 3. Блоки мікросхем оперативної пам'яті |
| 2. Процесор | 4. Мікросхема постійної пам'яті |

Мал. 2.4. Материнська плата комп'ютера



і результати опрацювання даних. Для того щоб дані були опрацьовані процесором, вони попередньо повинні бути передані («завантажені») з пристроїв уведення даних або запам'ятовуючих пристроїв до оперативної пам'яті.

Після вимкнення живлення комп'ютера всі дані з оперативної пам'яті зникають.

Оперативна пам'ять виготовляється у вигляді блоку мікросхем, який встановлюється у спеціальні роз'єми (слоти) на материнській платі (мал. 2.4, 3). Як правило на материнській платі можна встановити кілька блоків мікросхем оперативної пам'яті.

Для забезпечення початку роботи комп'ютера основні (базові) програми зберігаються в **постійній пам'яті**. Після ввімкнення комп'ютера дані з по-

Слот (англ. *slot*) – щілина, паз, отвір.

Сокет (англ. *socket*) – заглиблення, гніздо, розетка.

Кеш (англ. *cache*) – схованка, запас харчів.



а)



б)

Мал. 2.5. Мікросхеми постійної пам'яті на материнській платі

стійної пам'яті копіюються («завантажуються») в оперативну пам'ять, і запускаються програми підготовки комп'ютера до роботи.

Постійна пам'ять виготовляється у вигляді однієї або двох мікросхем, що вставляються у спеціальні гнізда – «сокети» (мал. 2.5, а) або впаюються на материнській платі (мал. 2.5, б).

Як уже зазначалося, до складу сучасних процесорів включають **кеш-пам'ять**. Вона призначена для прискорення обміну даними між основним масивом оперативної пам'яті і процесором. Розрізняють кеш-пам'ять першого, другого і третього рівнів. Ємність кеш-пам'яті першого рівня в більшості процесорів – 128 Кбайт. Ємності кеш-пам'яті другого і третього рівнів суттєво відрізняються в різних процесорах. На початок 2016 року кеш-пам'ять другого рівня у процесорах для настільних персональних комп'ютерів становить 1–8 Мбайт, а третього – 2–20 Мбайт. Збільшення ємності кеш-пам'яті зазвичай збільшує швидкість процесора.

ЗОВНІШНІ ЗАПАМ'ЯТОВУЮЧІ ПРИСТРОЇ

Якщо пристрої внутрішньої пам'яті розміщені на материнській платі, то пристрої зовнішньої пам'яті приєднуються до неї з використанням різних з'єднань. Пристрої зовнішньої пам'яті, на відміну від пристроїв внутрішньої пам'яті (за винятком постійної пам'яті), призначені для довго-

тривалого зберігання даних. Після вимкнення живлення дані з носіїв зовнішньої пам'яті не зникають.

До пристроїв зовнішньої пам'яті належать накопичувачі на жорстких магнітних дисках, пристрої для роботи з оптичними дисками, пристрої, що використовують флеш-пам'ять тощо.



1. Поверхня одного з жорстких дисків
2. Електромагнітна головка

Мал. 2.6. Будова накопичувача на жорстких магнітних дисках

Носієм даних у **накопичувачі на жорстких магнітних дисках (НЖМД)** є один або кілька металевих дисків, покритих шаром магнітної речовини (мал. 2.6). Запис даних виконується шляхом намагнічування ділянок поверхні диска з використанням електромагнітної головки, яка виконує і зчитування даних.

Властивості накопичувачів на жорстких магнітних дисках та приклади їх значення в сучасних пристроях подано в таблиці 2.2.

Таблиця 2.2

Значення властивостей сучасних НЖМД

Властивість	Значення властивості НЖМД	
	для настільних комп'ютерів	для ноутбуків
Ємність	6 Тбайт	2000 Гбайт
Діаметр жорстких дисків	3,5 дюйма	2,5 дюйма
Швидкість обертання жорстких дисків	5400–7200 об/хв	5400 об/хв

Пристрої для роботи з **оптичними дисками** використовують промінь лазера для зчитування (запису) даних з оптичних дисків різних типів, які відрізняються ємністю та можливістю виконувати запис і перезапис даних. Основні типи оптичних дисків, їх властивості та значення властивостей подано в таблиці 2.3.

Таблиця 2.3

Основні типи оптичних дисків

Позначення дисків, призначених для			Ємність дисків
тільки зчитування даних	зчитування і запису даних	зчитування, запису і перезапису даних	
CD-ROM	CD-R	CD-RW	640–800 Мбайт
DVD-ROM	DVD-R, DVD+R	DVD-RW DVD+RW	9,4 Гбайта
BD-ROM	BD-R	BD-RE	100 Гбайт



Пристрої флеш-пам'яті останнім часом набули широкого розповсюдження і використовуються не тільки як пристрої для перенесення даних між комп'ютерами, але і як основний пристрій зовнішньої пам'яті



Мал. 2.7. Флеш-карти різних типів



Мал. 2.8. SSD-диск

в переносних комп'ютерах – нетбуках, планшетних комп'ютерах, не кажучи вже про мобільні пристрої, а також фото- і відеокамери (мал. 2.7). Флеш-пам'ять значно зменшила використання оптичних дисків.

Зі збільшенням ємності пристроїв флеш-пам'яті вони почали активно використовуватися і як замітники накопичувачів на жорстких магнітних дисках. Такі пристрої отримали назву **твердотілих накопичувачів** або **SSD-дисків**.

SSD (англ. *Solid-State Drive*) – твердотілий (суцільний) пристрій.

Ємність SSD-дисків (мал. 2.8) не набагато менша від ємності накопичувачів на жорстких магнітних дисках і складає більше ніж 2 Тбайт, однак вони можуть забезпечувати більшу швидкість запису та зчитування даних.

Найважливіше в цьому пункті

Архітектура комп'ютера – це модель, що описує взаємодію пристроїв та програм для забезпечення інформаційних процесів у комп'ютері.

Процесор – це пристрій, який забезпечує опрацювання даних. Він містить **пристрій керування**, який забезпечує виконання команд комп'ютерної програми, та **арифметично-логічний пристрій**, який і здійснює операції над даними. Основною властивістю процесора є **швидкість опрацювання даних**, яка залежить від значення кількох інших властивостей: **тактової частоти, кількості ядер, розрядності, обсягів кеш-пам'яті** тощо.

Пам'ять комп'ютера призначена для збереження даних, її поділяють на **внутрішню** та **зовнішню**.

Пристрої **внутрішньої пам'яті** (оперативна, постійна, кеш-пам'ять) розміщуються на материнській платі. Дані з оперативної пам'яті та кеш-пам'яті після вимкнення живлення зникають, а з постійної – ні.

Пристрої **зовнішньої пам'яті** (накопичувачі на жорстких магнітних дисках, пристрої для роботи з оптичними дисками, пристрої, що використовують флеш-пам'ять) призначені для довготривалого зберігання даних. Після вимкнення живлення дані з носіїв зовнішньої пам'яті не зникають.

Основною властивістю всіх видів пам'яті є її ємність, що вимірюється в байтах, кілобайтах, мегабайтах, гігабайтах, терабайтах тощо.



Дайте відповіді на запитання

- 1°. Що таке архітектура комп'ютера?
- 2°. Опишіть схему реалізації інформаційних процесів у комп'ютері за малюнком 2.1.
- 3°. Для чого призначений процесор? Назвіть пристрої, що входять до складу процесора.
- 4°. Який пристрій пам'яті включено до складу сучасних процесорів?
- 5°. Назвіть властивості процесора та наведіть приклади їх значень.
- 6°. Наведіть класифікацію пам'яті комп'ютера. Яка основна властивість пам'яті?
- 7°. Назвіть одиниці вимірювання ємності пам'яті.
- 8°. Для чого призначена оперативна пам'ять? Де вона розміщується?
- 9°. Для чого призначена постійна пам'ять комп'ютера? У чому її відмінність від оперативної?
- 10°. Для чого призначена кеш-пам'ять?
- 11°. Назвіть пристрої зовнішньої пам'яті. Наведіть приклади значень властивостей кожного з них.
- 12°. Чим внутрішня пам'ять відрізняється від зовнішньої?
- 13*. Що спільного і в чому відмінності між пристроями зовнішньої пам'яті?



Виконайте завдання

- 1°. Використовуючи дані одного із сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад **hotline.ua**, визначте значення властивостей трьох різних процесорів, що надійшли останнім часом у продаж у магазини України. Заповніть таблицю:

<i>Властивості</i>	<i>Значення властивостей процесора</i>		
Модель			
Тактова частота			
Кількість ядер			
Розрядність			



- 2*. Дослідіть, використовуючи дані сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад **hotline.ua**, значення властивостей оперативної пам'яті з максимальною ємністю, що пропонуються для ноутбуків. Заповніть таблицю:

<i>Властивості</i>	<i>Значення властивостей</i>		
Модель			
Виробник			
Ємність			
Вартість			

- 3°. Розташуйте носії даних, які ви знаєте, у порядку зменшення їх ємності.



- 4*. За відомостями з Інтернету або каталогами комп'ютерних магазинів визначте значення властивостей процесорів, які доступні на ринку комп'ютерів вашого регіону.
- 5*. За відомостями з Інтернету або каталогами комп'ютерних магазинів порівняйте ємність кеш-пам'яті другого рівня процесорів з однаковою тактовою частотою і кількістю ядер різних виробників.
- 6*. Підготуйте повідомлення про можливість і способи зміни даних у постійній пам'яті.



2.2. ПРИСТРОЇ ВВЕДЕННЯ ТА ВИВЕДЕННЯ ДАНИХ



1. Які пристрої введення даних входять до складу комп'ютера? Для введення яких даних вони використовуються?
2. Які пристрої виведення даних входять до складу комп'ютера? Для виведення яких даних вони використовуються?
3. Які пристрої використовуються для введення і виведення мультимедійних даних? Наведіть приклади таких пристроїв.

ПРИСТРОЇ ВВЕДЕННЯ ДАНИХ

Як ви вже знаєте, до складу комп'ютерів входять пристрої введення та виведення даних.



Мал. 2.9. Геймпед



Мал. 2.10. Кейпед



Мал. 2.11. Танцювальний майданчик

Пристрої введення даних можна розділити за типом даних, з якими вони працюють. Так, можна виділити пристрої введення текстових (**клавіатура**), графічних (**сканер, фотокамера, графічний планшет**), звукових (**мікрофон**), відеоданих (**відеокамера, веб-камера, ТВ-тюнер**). Ще однією групою пристроїв введення даних є пристрої для забезпечення управління в різноманітних програмах. До них належать миша, тачпед, мультимедійна (електронна) дошка, сенсорний екран, джойстик, геймпед (мал. 2.9), кейпед (мал. 2.10), руль, педалі, танцювальний майданчик (денспед) (мал. 2.11) тощо.

Геймпед (англ. *gamepad*) – ігрова площадка (майданчик, підкладка).

Кейпед (англ. *keypad*) – площадка для клавіш.

Денспед (англ. *dancepad*) – майданчик для танців.

У комп'ютерах, які використовуються для навчальних та наукових цілей, можуть застосовуватися додаткові пристрої для введення даних. Наприклад, датчики для визначення значення температури, вологості повітря, наявності певних домішок у речовинах, швидкості руху різноманітних об'єктів тощо.

Розглянемо властивості (технічні характеристики) і класифікацію пристроїв введення даних.

Клавіатура призначена для введення символічних даних і команд. Незважаючи на те що клавіатури за останні кілька десятиліть дуже мало змінювались, їх можна поділити на кілька груп залежно від значення певних властивостей:

- за призначенням: *стандартні* (для настільних ПК), *компактні*, *мультимедійні*, *ігрові* тощо;
- за типом підключення до системного блоку: *дротові* або *бездротові* (радіо, *Bluetooth*, *Wi-Fi* тощо).

Для переносних комп'ютерів можуть бути використані проекційні клавіатури (мал. 2.12).

Маніпулятори **миша** поділяються залежно від значень таких властивостей:

- за призначенням: *для настільних ПК*, *для ноутбуків* (компактні), *ігрові* тощо;
- за типом підключення до системного блоку: *дротові* або *бездротові* (радіо, *Bluetooth*, *Wi-Fi* тощо);
- за типом датчиків руху: *оптичні*, *лазерні*;
- за кількістю кнопок: *2*, *3–5*, *6–9*, *10* і *більше* тощо;
- за типом корпусу: *симетрична стандартна*, *симетрична для шувльги*, *ергономічна* тощо.

На малюнку 2.13 зображено ігрову дротову лазерну ергономічну мишу, що має 19 кнопок і одне коліщатко.

Тачпеди (англ. *touch* – дотик, *pad* – площадка, майданчик, подушечка) є складовими практично всіх типів ноутбуків і нетбуків, проте можуть використовуватись і під час роботи з настільними ПК (мал. 2.14). Переміщення пальця по поверхні тачпеда приводить до переміщення вказівника на екрані монітора. Одночасове короткочасне натиснення на площадку



Мал. 2.12. Проекційна клавіатура



Мал. 2.13. Миша



Мал. 2.14. Зовнішній тачпед



тачпада аналогічне одноразовому клацанню лівою кнопкою миші, а подвійне натиснення – подвійному клацанню лівою кнопкою миші. Ліва і права кнопки тачпада виконують ті самі функції, що й відповідні кнопки миші.

Мультимедійні (електронні) дошки використовують здебільшого в закладах освіти, а також під час проведення різноманітних презентацій. Розрізняють дошки залежно від їх розмірів (довжина діагоналі 70–79, 80–89 чи більше 90 дюймів) та принципу визначення місця дотику на дошці.

Сенсорні екрани використовують здебільшого у планшетних комп'ютерах, смартфонах, а також у різноманітних довідкових (мал. 2.15) та презентаційних системах. Уведення даних здійснюється дотиком пальця або стилуса до певних ділянок поверхні екрана. Екран «відчуває», у якому місці відбувся дотик, і передає відповідний сигнал комп'ютеру. Основні властивості – розмір екрана (довжина діагоналі може бути від 3 до 70 і більше дюймів) і принцип відслідковування місця дотику.

Близький за принципом дії до сенсорного екрану **графічний планшет**. Він використовується для створення малюнків та введення інших графічних даних. Користувач створює малюнок, дотикаючися стилусом до поверхні графічного планшета. В окремих видів графічних планшетів робоча поверхня не відображає зображення, воно з'являється на екрані комп'ютера, до якого приєднано графічний планшет. В інших створене зображення відображається на робочій поверхні графічного планшета (мал. 2.16).

Розрізняють графічні планшети за розміром робочої поверхні, роздільною здатністю (від 2000 до більше ніж 4000 точок на дюйм), а також кількістю варіантів реагування на силу натискання стилусом на робочу поверхню (від 512 до 2048).

Сканери також використовують для введення графічних даних. Серед них за принципом дії виділяють *планшетні, з протяжним механізмом, ручні та скан-камери* (мал. 2.17). Основними властивостями сканерів є розмір області сканування, роздільна здатність (від 600 до більш ніж 6400 точок на дюйм) і швидкість сканування (5–20 с).

З використанням фото- і відеокамер, веб-камер, мікрофонів та інших мультимедійних

Сенсор (англ. *sensor* від лат. *sen-sus* – відчуття) – датчик, пристрій, що перетворює сигнал у форму, зручну для опрацювання.
Стилус (лат. *stilus*) – паличка для письма.



Мал. 2.15. Пункт інформаційно-довідкової системи залізниці



Мал. 2.16. Графічний планшет

пристроїв ви ознайомилися в курсі інформатики 6-го класу. Зазначимо, що основними властивостями фото- і відеокамер, веб-камер є їх роздільна здатність, яка визначає якість зображення. Для **відеокамер** якість зображення визначається стандартами **SD** (англ. *Standard Definition* – стандартна чіткість, 720×576 точок), **HD** (англ. *High Definition* – висока чіткість, 1280×720 точок), **Full HD** (англ. *Full High Definition* – повна висока чіткість, 1920×1080 точок), **Ultra HD** (англ. *Ultra High Definition* – надвисока чіткість, 3840×2160 точок).



планшетний

з протяжним механізмом

скан-камера

Мал. 2.17. Типи сканерів

У **фотокамер** якість зображення залежить від оптичних властивостей об'єктива та кількості точок (пікселей) (від 10 до понад 24 мегапікселей), які може відтворити пристрій. **Веб-камери** мають роздільну здатність від 320×240 пікселей до 2560×2048 пікселей і більше.

ТВ-тюнер – це пристрій, що забезпечує перегляд телевізійних передач на комп'ютері.

ПРИСТРОЇ ВИВЕДЕННЯ ДАНИХ

Пристрої виведення, як і пристрої введення даних, можна поділити за типом даних, з якими вони працюють. Так, можна виділити пристрої виведення текстових і графічних даних (**монітор, принтер, плотер**), звукових (**навушники, звукові колонки**), відеоданих (**мультимедійні проектори, екранні панелі**). Певні пристрої можуть виводити як графічні, так і відеодані (монітор, мультимедійні проектори, екранні панелі тощо).

Монітор є основним пристроєм для виведення даних у персональних комп'ютерах. Їх поділяють за системою створення зображення на **монітори на рідких кристалах**, або **LCD-монітори** (англ. *Liquid Crystal Display* – рідкокристалічний дисплей), **плазмові, OLED** (англ. *Organic Light Emitting Diode* – органічний світлодіод), на **електронних чорнилах** – *e-ink* (англ. *Electronic ink* – електронні чорнила) тощо. У більшості сучасних моніторів використовується рідкокристалічна технологія.

LCD-монітори поділяють за принципом дії (*TN + film, PLS*, різні версії *IPS* та *VA*), за довжиною діагоналі (від 19 до понад 30 дюймів), роздільною здатністю (від 1280×1024 до 5120×2880 пікселів), часом реакції – відклику на команду зміни зображення на екрані (від 1 до 8 мс).



Для опрацювання графічних даних, що виводяться на екран монітора, у комп'ютерах використовується спеціальний пристрій – відеоадаптер. Розрізняють інтегровані відеоадаптери та у вигляді окремої плати. У випадку інтегрованого відеоадаптера він розміщується в мікросхемі процесора або на материнській платі. Інтегровані відеоадаптери використовуються в персональних комп'ютерах, що не пов'язані з якісним опрацюванням великих обсягів графічних даних. Для цих цілей використовують відеоадаптери у вигляді окремої плати (мал. 2.18), що вставляється в один із слотів материнської плати. Відеоадаптер містить спеціальний графічний процесор та додаткову оперативну пам'ять – графічну (відео) пам'ять. Розрізняють графічні адаптери за типом процесора (наприклад, *Radeon R7 370*, *GeForce GTX 950*), обсягом пам'яті (від 1 до 6 Гбайт і більше), системою під'єднання до материнської плати та до монітора тощо.

Для виведення даних на папір, плівку або інші матеріали використовуються принтери і плотери.

Принтери поділяють за принципом дії на *матричні*, *лазерні*, *струменеві*, *термічні*, *3D-принтери* тощо. Основними властивостями принтерів є:

- розмір матеріалу, на який виконується друк;
- кількість кольорів – *монохромні*, *багатокольорові*;
- роздільна здатність – від 600×1200 точок до 5760×1440 точок на дюйм і більше;
- швидкість друку – від 1 до 50 і більше сторінок за хвилину;
- тип роз'ємів для підключення до комп'ютера – *дротове* або *бездротове*.

Останнім часом усе більшої популярності набувають *3D-принтери*, що забезпечують створення виробів складної форми (мал. 2.19).

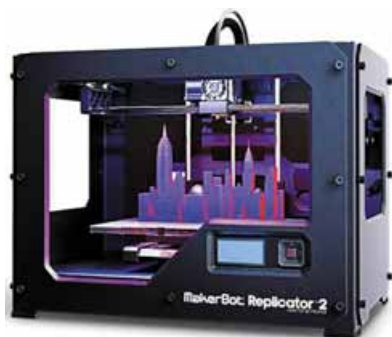
Інтегральний (лат. *integer* – цілий) – нерозривно зв'язаний, цільний.

Інтеграція (лат. *integratio* – поповнення) – об'єднання в ціле окремих частин.

Чіпсет (англ. *chip* – мікросхема, *set* – упорядковувати) – набір мікросхем, що забезпечує обмін даними між пристроями комп'ютера.



Мал. 2.18. Відеоадаптер



Мал. 2.19. 3D-принтер

ПРИСТРОЇ, ЩО ВХОДЯТЬ ДО СКЛАДУ МУЛЬТИМЕДІЙНОГО ОБЛАДНАННЯ

Ви вже знаєте, що серед пристроїв введення і виведення можна виділити пристрої, які працюють з мультимедійними даними. Нагадаємо, що до таких пристроїв належать **фотокамера**, **графічний планшет**, **мікрофон**, **відео-**



камера, веб-камера, мультимедійні (електронні) дошки, навушники, звукові колонки, мультимедійні проектори, екранні панелі тощо. Для пристроїв, що працюють з нерухомими та рухомими зображеннями однією з основних властивостей є роздільна здатність, а для тих, що працюють зі звуком, – смуга частот звукового діапазону, яку може опрацювати пристрій.



Найважливіше в цьому пункті

До пристроїв введення даних належать пристрої введення текстових (клавіатура), графічних (сканер, фотокамера, графічний планшет), звукових (мікрофон), відеоданих (відеокамера, веб-камера, ТВ-тюнер), а також пристрої для забезпечення управління об'єктами в різноманітних програмах – миша, тачпед, мультимедійна (електронна) дошка, сенсорний екран, джойстик, геймпед, кейпед, руль, педалі, танцювальний майданчик тощо.

До пристроїв виведення даних належать пристрої виведення текстових і графічних даних (монітор, принтер, плотер), звукових (навушники, звукові колонки), відеоданих (мультимедійні проектори, екранні панелі) тощо.

Класифікують зазначені пристрої за значенням певних властивостей.



Дайте відповіді на запитання

- 1°. Які пристрої комп'ютера належать до пристроїв введення даних?
- 2°. Які пристрої комп'ютера належать до пристроїв виведення даних?
- 3°. За значенням яких властивостей класифікують клавіатури? Наведіть приклади.
- 4°. Які види маніпуляторів миша використовуються в комп'ютерному класі вашої школи? Назвіть значення їх властивостей.
- 5°. Які значення властивостей можуть мати сканери, що використовуються в комп'ютерних класах школи?
- 6°. Назвіть властивості монітора комп'ютера та наведіть приклади їх значень.
- 7°. Опишіть призначення монітора та відеоадаптера. Які значення властивостей можуть мати сучасні відеоадаптери?
- 8°. Опишіть класифікацію сучасних принтерів. Які види принтерів використовуються у вашій школі?
- 9°. Які значення властивостей повинен, на вашу думку, мати принтер для домашнього користування?
- 10°. Для чого призначені пристрої, що входять до складу мультимедійного обладнання?
- 11°. Порівняйте, що є спільного і відмінного у графічних планшетах і планшетних комп'ютерах. Для яких цілей використовуються перші та другі?



Виконайте завдання

- 1°. На основі відомостей, розміщених у підручнику, підготуйте повідомлення про класифікацію пристроїв введення даних.



- 2*. На основі відомостей, розміщених у підручнику, підготуйте слайд презентації зі схемою класифікації пристроїв уведення даних. Збережіть презентацію у файлі з іменем **завдання 2.2.2** у вашій папці.
- 3*. Використовуючи дані одного із сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад **hotline.ua**, визначте значення властивостей трьох різних маніпуляторів миша, що надійшли останнім часом у продаж у магазини України. Заповніть таблицю:

<i>Властивості</i>	<i>Значення властивостей миші</i>		
Модель			
Тип підключення			
Тип датчиків руху			
Кількість кнопок			
Тип корпусу			



- 4*. Використовуючи дані сайтів з відомостями про сучасні комп'ютерні пристрої, наприклад **hotline.ua**, визначте значення властивостей моніторів трьох різних виробників, що пропонуються магазинами України, заповніть таблицю:

<i>Властивості</i>	<i>Значення властивостей</i>		
Модель			
Виробник			
Довжина діагоналі, <i>дюймів</i>			
Тип матриці			
Максимальна роздільна здатність, <i>пікселів</i>			
Час реакції, <i>мс</i>			

- 5*. На основі відомостей, розміщених у підручнику, підготуйте в одній з прикладних програм схему класифікації сучасних принтерів. Збережіть схему в файлі з іменем **завдання 2.2.5** у вашій папці.



- 6*. Визначте значення властивостей пристроїв вашого домашнього комп'ютера та заповніть таблицю:

<i>Властивості</i>	<i>Значення властивостей</i>		
Модель монітора			
Довжина діагоналі, <i>дюймів</i>			
Тип матриці			
Максимальна роздільна здатність, <i>пікселів</i>			
Час реакції, <i>мс</i>			



- 7*. За відомостями з Інтернету або каталогами комп'ютерних магазинів визначте значення властивостей принтерів трьох різних виробників, що пропонуються магазинами України. Заповніть таблицю:

Властивості	Значення властивостей		
Модель			
Виробник			
Принцип дії			
Роздільна здатність, пікселів			
Швидкість друку, сторінок за хвилину			



- 8*. Підготуйте повідомлення про один з нових видів пристроїв уведення даних – кейпед. Опишіть його будову, призначення та область використання.



- 9*. Підготуйте повідомлення про один з нових видів пристроїв уведення даних – денспед. Опишіть його будову, призначення та область використання.



2.3. ІСТОРІЯ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ. ВИДИ СУЧАСНИХ КОМП'ЮТЕРІВ



- Які види комп'ютерів ви знаєте? Чим відрізняється їх використання?
- Які ви знаєте приклади застосування комп'ютерів у різних галузях людської діяльності?
- Назвіть прізвища українських учених, які зробили значний внесок у розвиток комп'ютерної техніки.

ЕТАПИ РОЗВИТКУ ЗАСОБІВ РЕАЛІЗАЦІЇ ІНФОРМАЦІЙНИХ ПРОЦЕСІВ

Розвиток людства безпосередньо пов'язаний з розвитком засобів передавання, опрацювання та зберігання повідомлень. У цьому розвитку можна виділити кілька етапів (табл. 2.4).

Таблиця 2.4

Етапи розвитку засобів реалізації інформаційних процесів

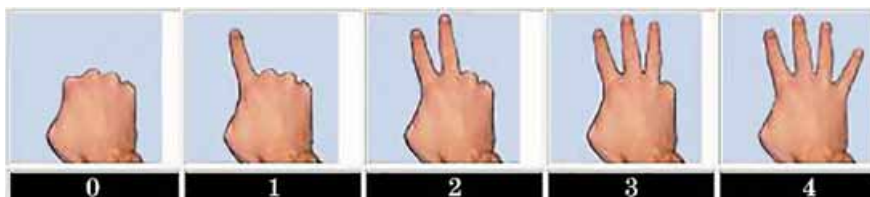
Назва етапу	Період історії людства	Приклади носіїв даних, засобів передавання та опрацювання даних
Етап ручних засобів	Від стародавніх часів до середини XV ст.	Носії даних – глиняні дощечки, папіруси, береста, палиці із зарубками, картини. Засоби передавання – сигнальні вогнища, барабанний бій, персональні посланці, голубина пошта, перші поштові служби для передавання державних документів. Засоби опрацювання – мозок і пальці людини, ручні обчислювальні прилади (абак, рахівниця тощо).



Продовження таблиці 2.4

Назва етапу	Період історії людства	Приклади носіїв даних, засобів передавання та опрацювання даних
Етап механічних засобів	Від середини XV ст. до середини XIX ст.	<i>Носії даних</i> – книги, газети, журнали, фотопластини. <i>Засоби передавання</i> – поштові служби. <i>Засоби опрацювання</i> – друкарські машинки, фотоапарати, арифмометри, машинки для обчислень.
Етап електричних засобів	Від середини XIX ст. до 40-х років XX ст.	<i>Носії даних</i> – платівки, кіноплівки, магнітні плівки. <i>Засоби передавання</i> – телефон, телеграф, радіо. <i>Засоби опрацювання</i> – фонографи, грамофони, електричні друкарські машинки, табулятори, електричні арифмометри, магнітофони, ротапринти.
Етап електронних засобів	Від 40-х років XX ст. до наших днів	<i>Носії даних</i> – магнітні й оптичні диски, електронні мікросхеми. <i>Засоби передавання</i> – телебачення, комп'ютерні мережі, засоби мобільного зв'язку. <i>Засоби опрацювання</i> – електронні калькулятори, комп'ютер.

ІСТОРІЯ ЗАСОБІВ ОПРАЦЮВАННЯ ПОВІДОМЛЕНЬ



Мал. 2.20. Рахунок за допомогою пальців руки

Розглянемо детальніше історію засобів опрацювання повідомлень, призначених для виконання обчислень. Першими засобами опрацювання числових повідомлень були пальці (мал. 2.20). Про це свідчать, наприклад, римські цифри (I, V, X). У стародавній слов'янській нумерації одиниці називали «перстами», тобто пальцями, а назва цифри 5 походить від слова «п'ять» (кисть руки).

Значним кроком уперед у розвитку засобів обчислювання стало створення абака у V ст. до н. е. в Давній Греції. Ідею такого пристрою греки запозичили у стародавніх вавилонян. Більш пізній римський варіант абака зображено на малюнку 2.21. Абак являв собою дошку з паралельними вертикальними заглибленими відрізками, у які клались якісь предмети, найчастіше камінці.



Мал. 2.21. Римський абак

Подібні пристрої у подальшому були створені в різних країнах: Китаї (суаньпань), Японії

(соробан), Росії (рос. *счёты* – **рахівниця**) і використовувалися до кінця ХХ ст.



Мал. 2.22. Рахівниця

Разом з простими і доволі розповсюдженими пристроями типу абака ще в Давній Греції і Римі застосовували різноманітні обчислювальні пристрої з використанням коліщат із зубцями. Про це свідчить знахідка із затонулого античного судна. Її назвали «Антикітерський механізм» за назвою грецького острова, поблизу якого знайшли судно в 1902 р. Але тільки у другій половині ХХ ст. учені змогли за рештками механізму відновити його будову і визначити призначення (мал. 2.22).

Створений за 100–150 р. до н. е. пристрій використовувався як календар, а також для автоматичного визначення положення небесних тіл – Сонця, Місяця та відомих грекам планет – Меркурія, Венери, Марса, Юпітера, Сатурна.



Мал. 2.23. Антикітерський механізм: фрагмент знахідки та комп'ютерна модель

Однак, на жаль, пристрої, подібні до антикітерського механізму, і наукові знання, потрібні для їх виготовлення, були на тривалий час втрачені. Активні роботи щодо вдосконалення обчислювальних пристроїв розпочалися тільки в ХVІІ ст. з розвитком промисловості, торгівлі, будівництва, військової справи.

У 1642 р. французький математик, фізик, інженер і філософ **Блез Паскаль** (1623–1662) подав на розгляд королівської ради свій механічний обчислювальний пристрій. У подальшому він створив кілька десятків таких пристроїв (мал. 2.24). Їх використовували для виконання додавання та віднімання чисел.

Згодом ідеї Паскаля були розвинуті та вдосконалені багатьма математиками та інженерами. Зокрема німецький учений **Готфрід Лейбніц** (1646–1716) створив перший у світі арифмометр – механічний обчислювальний пристрій, що виконував усі чотири арифметичні дії (мал. 2.25).

Удосконалені варіанти арифмометрів різних конструкцій активно використовувалися впродовж кількох століть і були замінені електронними калькуляторами у другій половині ХХ ст.



Мал. 2.24. Обчислювальний пристрій
Блезе Паскаля



Мал. 2.25. Арифмометр
Готфріда Лейбніца

Одночасно з удосконаленням арифмометрів учені намагалися створити обчислювальні пристрої, які могли б автоматично проводити обчислення, виконуючи команди заздалегідь розробленої програми. Одним з перших ідею створення автоматизованої машини для обчислень висловив англійський математик і конструктор **Чарльз Беббідж** (1792–1871) (мал. 2.26). Він назвав свій прилад **аналітична машина**.



Мал. 2.26. Чарльз
Бebbідж

Згідно з проектом Беббіджа аналітична машина мала такі складові частини:

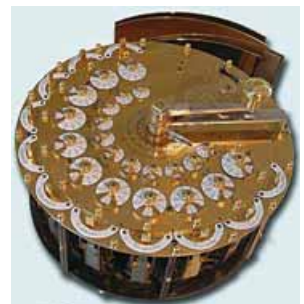
1. «Склад» для зберігання чисел (у сучасній термінології – **пам'ять**).
2. «Млин» для виконання арифметичних операцій над числами (у сучасній термінології – **арифметичний пристрій**).
3. Пристрій, який керує послідовністю виконання арифметичних операцій (у сучасній термінології – **пристрій керування**).
4. Пристрій для введення початкових (вхідних) даних.
5. Пристрій виведення результатів.

Ада Лавлейс (1815–1852) (мал. 2.27), донька відомого англійського поета Джорджа Байрона, працювала разом з Беббіджем над створенням проекту аналітичної машини та його реалізацією. Вона вперше описала основні принципи розробки програм для обчислювальних машин. У зв'язку із цим Ада Лавлейс вважається першим у світі програмістом, і на її честь названо одну із сучасних мов програмування **Ada**.



Мал. 2.27. Ада Лавлейс

До кінця XIX ст. пристрої для обчислень були ручними або механічними (мал. 2.28). І тільки в кінці XIX ст. американський учений **Герман Холлеріт** (1860–1929) запропонував новий пристрій, робота якого базувалася на використанні



Мал. 2.28. Машинка для обчислень І.Г. Мюллера

електричного струму, – **табулятор** (мал. 2.29). Він був призначений для опрацювання даних перепису населення. Дані про конкретну людину записували не на папері, а відмічалися отворами у строго визначених місцях персональної карти – перфокарти.



Мал. 2.29. Табулятор Германа Холлеріта

Перфорація (лат. *perforate* – продірявлювати) – пробивка отворів, сукупність отворів.

З3. У ній використовувалося двійкове кодування. У 1950 р. він створив комп'ютер на електронних лампах – **З4**.



Мал. 2.30. Конрад Цузе

Від інших машин того часу її відрізняло те, що програма, за якою вона працювала, зберігалася в пам'яті самої машини.

У 1943–1946 рр. у США **Джон Моучлі** (1907–1980) і **Преспер Еккерт** (1919–1995) створили **ENIAC** (англ. *Electronic Numerical Integrator and Calculator* – електронний цифровий інтегратор і обчислювач) – **електронну обчислювальну машину** (ЕОМ), що містила 18 000 електронних ламп, важила 30 т і виконувала п'ять тисяч операцій за секунду (мал. 2.31). Після

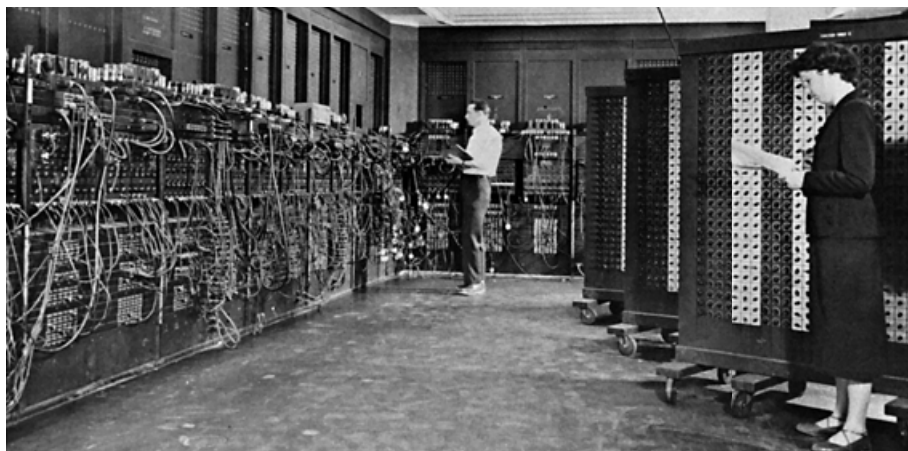
Заснована Холлерітом у 1896 р. фірма з масового випуску табуляторів, зазнавши низку реорганізацій і зміну власників, з 14 лютого 1924 р. стала називатися **IBM Corporation**. Сьогодні – це один з основних виробників потужних комп'ютерів.

До середини ХХ ст. розроблялися різноманітні конструкції механічних і електромеханічних обчислювальних пристроїв. Вони давали змогу значно прискорити процес обчислення.

У кінці 30-х та в 40-х роках ХХ ст. учені та конструктори різних країн створили якісно нові обчислювальні машини, у яких, зазвичай, використовувалися електричні пристрої. Так, у Німеччині в 1941 р. **Конрад Цузе** (1910–1995) створив першу обчислювальну машину на електромеханічних реле, яку назвав

У 1941 році у США **Джон Вінсент Атанасов** (1903–1995) разом зі своїм асистентом **Кліфордом Беррі** (1918–1963) створили перший комп'ютер на електронних лампах з використанням двійкового кодування – **ABC** (англ. *Atanasoff Berry Computer*). Трохи пізніше (1939–1944), також у США, **Говард Ейкен** (1900–1973) у співробітництві з IBM створив обчислювальну машину **Mark-1** на електромагнітних реле з автоматичним керуванням послідовністю операцій.

У 1943–1944 рр. в умовах цілковитої таємності вчені Великобританії створили обчислювальну машину **Colossus**, призначену для дешифрування радіограм фашистської Німеччини.



Мал. 2.31. Обчислювальна машина **ENIAC**

завершення цього проекту вони відразу почали роботу над новим комп'ютером на замовлення військового відомства США – **EDVAC** (англ. *Electronic Discrete Variable Automatic Computer* – електронний дискретно змінний автоматичний обчислювач). До розробки залучили відомого американського математика **Джона фон Неймана** (1903–1957), який у статті «Попередня доповідь про машину EDVAC» сформулював основні принципи побудови універсальної обчислювальної машини. Незважаючи на те, що ці ідеї розробили Еккерт і Моучлі, вони стали відомі всьому світові як «принципи фон Неймана».

Широке розповсюдження комп'ютери отримали після появи великої кількості порівняно дешевих обчислювальних машин – так званих міні-ЕОМ, які в подальшому стали називатися персональними комп'ютерами. Перші з них продавались у вигляді електронних конструкторів, з деталей яких користувач міг самостійно зібрати свій комп'ютер, використавши телевизор як монітор. Першими персональними стали комп'ютери, розроблені фірмами: **MITS** (англ. *Micro Instrumentation and Telemetry Systems* – системи мікроінструментів і телеметрії) – комп'ютер **Altair 8800** (1975 р.); **Apple Computer Company** – комп'ютер **Apple** (1976 р.); **IBM** – комп'ютер **IBM PC** (1981 р.). Широке розповсюдження персональних комп'ютерів значно розширює коло задач, що розв'язуються з використанням ЕОМ.

Детальніше історію розвитку обчислювальної техніки можна розглянути за хронологічною таблицею у додатку 3, який можна переглянути на



Мал. 2.32. **Altair 8800**



Мал. 2.33. **Apple I**



Мал. 2.34. **IBM PC**

сайті **Інформатика для всіх** (*allinf.at.ua*) у розділі **Матеріали до підручників. 8 клас. Підручник.**

ІСТОРІЯ РОЗВИТКУ ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ В УКРАЇНІ

У 1951 р. в Києві під керівництвом Сергія Олексійовича Лебедева (1902–1973) було введено в дію універсальну ЕОМ з програмою, що зберігалася в пам'яті – **МЕОМ (Мала Електронна Обчислювальна Машина)**. Це була перша ЕОМ у Радянському Союзі. Вона мала такі характеристики:

- загальна кількість електронних ламп – близько 6000;
- двійкова система кодування даних;
- кількість розрядів – 16 для кодування числа і один для кодування знака;
- ємність запам'ятовуючого пристрою – 31 число і 63 команди;
- швидкодія – близько 3000 операцій за хвилину.

Перші програми для цієї ЕОМ склала **К.Л. Ющенко (1919–2001)**.

У Києві розробку обчислювальних машин було продовжено в Інституті кібернетики, який очолив видатний учений **В.М. Глушков (1923–1982)**. Під

його керівництвом завершено розробку **ЕОМ Київ (1959 р., розробники Б.В. Гнеденко, Л.М. Дашевський, К.Л. Ющенко)**, створено серію універсальних **ЕОМ Дніпро (1961 р., головний конструктор Б.М. Малиновський)**, **ЕОМ для інженерних розрахунків Промінь (1963 р., серії ЕОМ для інженерних розрахунків МИР (рос. *Машина для инженерных расчётов* – машина для інженерних розрахунків, 1960-ті рр.) та низки ЕОМ для військових цілей. За визначні досягнення в розвитку інформатики в 1996 р. Міжнародна комп'ютерна громада (IEEE Computer Society) нагородила В.М. Глушкова медаллю «Комп'ютерний піонер» (мал. 2.35).**

Піонер (англ. *pioneer* – перший поселенець, дослідник, першовідкривач) – людина, яка першою торує стежку в якій-небудь новій галузі діяльності.

Значний внесок у розвиток комп'ютерної техніки для ракетної галузі зробили колективи науково-виробничих об'єднань, конструкторських бюро з різних міст України. Так у Северодонецькому НВО «Імпульс» вироблялися електронні обчислювальні машини для систем автоматизованого керування виробництвом М6000-М7000, клавішні ЕОМ «Іскра», засоби для автоматичного керування балістичними ракетами. У Київському НВО «Кристал» спроектовано, а в м. Світловодську вироблялися перші в Радянському Союзі і Європі мікрокалькулятори **Електроніка** (мал. 2.36).

У 70–90-ті рр. ХХ ст. Харківське НВО «Хартрон» та Київський радіозавод розробляли і виготовляли ЕОМ для встановлення на ракетно-космічних комплексах.

У 70–90-ті рр. ХХ ст. Харківське НВО «Хартрон» та Київський радіозавод розробляли і виготовляли ЕОМ для встановлення на ракетно-космічних комплексах.



Мал. 2.35. Лицьовий і зворотний бік медалі В.М. Глушкова «Комп'ютерний піонер»

Мал. 2.36. Мікрокалькулятор



Розроблено і налагоджено (Київський науково-дослідний інститут радіоелектроніки та Київський завод «Буревісник», 70–80-ті роки ХХ ст.) серійний випуск ЕОМ «Карат» для керування надводними і підводними кораблями.

У 1975 р. в Інституті кібернетики АН УРСР під керівництвом М.М. Амосова спроектовано перший в СРСР автономний транспортний робот ТАІР (мал. 2.37), який міг рухатися в природному середовищі оминаючи перепони.

Учені та інженери України протягом 60–80-рр. ХХ ст. розробили багато комп'ютерів для різноманітних ракетних комплексів. Так, для найбільшої у світі балістичної ракети Р-36М2 («Сатана») виробництва Дніпропетровського НВО «Південний машинобудівний завод» спеціалістами Харківського НВО «Електроприлад» та ВО «Київський радіозавод» створено бортову ЕОМ, яка забезпечувала керування її запуском і польотом (мал. 2.38).

Нині цю ракету використовують для виведення на орбіту Землі супутників за міжнародною програмою «Дніпро».



Мал. 2.37. Конструктори налагоджують робот ТАІР



Мал. 2.38. Пуск ракети Р-36М2

ВИДИ СУЧАСНИХ КОМП'ЮТЕРІВ ТА ЇХ ЗАСТОСУВАННЯ

Ви знаєте, що найпоширенішими на сьогодні є так звані **персональні комп'ютери (ПК)**, серед яких розрізняють **стаціонарні** та **мобільні (портативні)**. Стаціонарні ПК у свою чергу поділяються на **офісні**, **домашні**, **ігрові** тощо. А мобільні – на **ноутбуки**, **нетбуки**, **планшетні ПК**, **смартфони** тощо.

Для розв'язування задач, що потребують великих обсягів складних обчислень, наприклад для розрахунків прогнозу погоди на кілька днів чи тижнів, використовують потужні комп'ютери. Найбільші з них називають **суперкомп'ютерами**. Наприклад, на середину 2015 року найпотужнішим комп'ютером у світі був комп'ютер **Tianhe-2** (мал. 2.39), створений у Китайському національному університеті оборонних технологій (Гуанчжоу). Він побудований на базі процесорів Intel Xeon із загальною кількістю ядер *3 млн 120 тис.*

У наш час комп'ютери використовують у науці, промисловості, торгівлі, управлінні, банківській системі, освіті, медицині, транспорті, зв'язку, сільсько-



Мал. 2.39. Комп'ютер Tianhe-2



му господарстві, системі соціального забезпечення та інших галузях народного господарства.

Прикладом застосування комп'ютерів у науковій сфері є проведення так званих комп'ютерних експериментів. Адже проведення багатьох наукових досліджень пов'язано зі значними труднощами – матеріальними, технічними, енергетичними тощо. Наприклад, щоб вивчити процеси, що відбуваються під час розщеплення атомних ядер, потрібно будувати атомні реактори, а цей процес тривалий, складний, небезпечний і вимагає великих витрат коштів. У багатьох випадках створити відповідні реальні умови експерименту взагалі неможливо, наприклад, неможливо керувати процесами, які відбуваються в якійсь далекій галактиці. У таких випадках явище вивчають за його **інформаційною моделлю** з використанням комп'ютерної техніки і відповідного програмного забезпечення.

Важливу роль відіграє комп'ютер на виробництві. Моделювання і конструювання різноманітних виробів з використанням комп'ютера значно скорочує термін їх розробки, підвищує їх ефективність і якість, знижує вартість. Наприклад, якщо до застосування комп'ютера в проектуванні нової моделі автомобіля від моменту виникнення ідеї до її виробництва проходило 5–6 років, то тепер – менше одного року.

У сфері обслуговування комп'ютер використовують для зберігання і опрацювання різноманітних даних: текстів, таблиць, баз даних, малюнків і фотографій, мультимедійних даних. Важко сьогодні назвати таку установу, де б не використовували комп'ютери. Картотеки в бібліотеках і лікарнях, виконані на основі комп'ютерної бази даних, у багато разів надійніші й зручніші в роботі, ніж традиційні паперові. Бухгалтер сьогодні використовує комп'ютер і за кілька хвилин отримує результати, на обчислення яких раніше йому були потрібні години або дні. Банкір, не виходячи зі свого кабінету, має змогу стежити за станом справ на біржі, простим натисненням на клавіші перевести на будь-який рахунок певну суму грошей.

Основними напрямками використання комп'ютерної техніки є:

- виконання громіздких обчислень та обчислень з високою точністю;
- створення комп'ютерних моделей об'єктів і проведення комп'ютерних експериментів;
- забезпечення функціонування автоматизованих систем управління;
- забезпечення зберігання та опрацювання великих обсягів даних;
- забезпечення швидкого обміну даними;
- керування промисловою, побутовою та військовою технікою з використанням вбудованих комп'ютерів;
- підтримка вивчення навчальних предметів;
- організація дистанційного навчання учнів, студентів, фахівців, що особливо корисне для віддалених малих населених пунктів, для людей з обмеженими можливостями тощо.

КОНФІГУРАЦІЯ КОМП'ЮТЕРА ПІД ПОТРЕБУ

Залежно від того, для яких цілей буде використовуватися комп'ютер, вибирають його складові (конфігурацію). При цьому особливу увагу звертають на значення таких властивостей:



- **потужність процесора**, що визначається значенням кількох властивостей – тактової частоти, кількістю ядер, обсягом кеш-пам'яті другого та третього рівнів. Що більші значення цих властивостей, тим потужніший процесор;
- **обсяг оперативної пам'яті** – що більше значення, то продуктивність комп'ютера вища;
- **ємність накопичувача на жорстких магнітних дисках** – залежно від обсягів даних, з якими працює користувач, – для відео-, графічних і аудіоданих потрібні НЖМД більшої ємності, ніж для роботи з текстовими даними;
- **наявність окремого відеоадаптера** – позитивно впливає на швидкість опрацювання відеоданих. А швидкість опрацювання даних відеоадаптером залежить від продуктивності процесора відеоадаптера та обсягу відеопам'яті;
- **якість відображення даних монітором**, яка залежить від розмірів монітора, роздільної здатності та швидкості відклику.

Звичайно, що можуть суттєво впливати на вибір комп'ютера і значення інших властивостей, таких як наявність пристрою для роботи з оптичним диском, особливостей материнської плати, тривалість роботи в автономному режимі (для мобільних пристроїв), наявність засобів підключення до різного типу комп'ютерних мереж тощо. На добір комп'ютерів буде суттєво впливати і їх вартість.

Значна частина Інтернет-магазинів пропонує добирати комп'ютери, використовуючи так звані «розумні фільтри». Наприклад, сайт **Hotline** (<http://hotline.ua/computer/>) залежно від призначення пропонує такі групи комп'ютерів: *ПК початкового рівня, Комп'ютер для роботи\навчання, Робоча станція, Базова оптимальна конфігурація, Універсальна оптимальна конфігурація, Прогресивна оптимальна конфігурація, Потужний ігровий ПК*. Сайт **Розетка** (<http://rozetka.com.ua/computers-notebooks>) пропонує лише три групи комп'ютерів: *Початковий рівень, Для роботи та навчання, Ігрові комп'ютери*.

Вартість системних блоків стаціонарних комп'ютерів при цьому змінюється в доволі великому діапазоні – від 2,5 до 120 тис. грн. Наведемо приклади значень основних властивостей настільних персональних комп'ютерів для запропонованої вище класифікації (табл. 2.5).

Таблиця 2.5

Приклади значень властивостей комп'ютерів різного призначення

Назва властивості	Значення властивостей для комп'ютерів різного призначення		
	Офісний (для навчання та роботи)	Домашній	Ігровий
Тип комп'ютера	стаціонарний	стаціонарний	стаціонарний
Модель процесора	Intel Celeron Dual-Core E3400	AMD FX-8320	Intel Core i7-4790K

Назва властивості	Значення властивостей для комп'ютерів різного призначення		
	Офісний (для навчання та роботи)	Домашній	Ігровий
Тактова частота процесора, ГГц	2,6	3,5	4,0
Обсяг оперативної пам'яті, Гбайт	2	8	16
Ємність жорсткого диска, Гбайт	250	1000	1000
Тип відеоадаптера	інтегрований Intel HD Graphics	AMD Radeon R9 270X	NVIDIA GeForce GTX 980 Ti
Обсяг відеопам'яті, Гбайт	додаткова відсутня	2	6
Тип звукового адаптера	інтегрований	Creative X-Fi Xtreme Audio PCI Express	інтегрований HD Audio 7.1
Монітор	LG 19M45A	Philips 233V5LSB	ASUS VX238H
Діагональ монітора, дюймів	18,5	23	23
Роздільна здатність	1366 × 768	1920 × 1080	1920 × 1080
Час відклику, мс	5	4	1

Найважливіше в цьому пункті

Перші пристрої для проведення обчислень були створені людиною кілька тисячоліть тому. Значний внесок у розвиток обчислювальної техніки зробили Б. Паскаль, Г. Лейбніц, Ч. Беббідж, А. Лавлейс, Г. Холлеріт. Перші електронні обчислювальні машини були створені в 40–50-х рр. XX ст. К. Цузе (Z4), Д.В. Атанасовим і К. Беррі (ABC), Г. Ейкеном (Mark-1), групою англійських учених (Colossus), Д. Моучлі та П. Еккертом (ENIAC), колективом радянських учених під керівництвом С. Лебедева (МЕОМ).

В Україні розробили перші в Радянському Союзі комп'ютери. До розробки ЕОМ для різних галузей економіки та для управління військовою технікою залучили десятки підприємств і організацій з різних куточків України. Провідною організацією став Інститут кібернетики Академії наук України під керівництвом В. Глушкова.

Серед комп'ютерів залежно від призначення та швидкості опрацювання виділяють суперкомп'ютери та персональні комп'ютери. Останні поділяються на стаціонарні та мобільні. Стаціонарні комп'ютери поділяють на офісні, домашні, ігрові тощо. А мобільні – на ноутбуки, нетбуки, планшетні ПК, смартфони тощо.



Дайте відповіді на запитання

- 1°. Назвіть етапи розвитку інформаційних технологій.
- 2°. Опишіть, як змінювалися засоби опрацювання повідомлень на різних етапах розвитку інформаційних технологій.
- 3°. Поясніть, у чому заслуга Б. Паскаля в розвитку пристроїв для проведення обчислень, які використовувала людина.
- 4°. Чим відрізнявся арифмометр Лейбніца від пристрою Паскаля?
- 5°. Опишіть складові універсальної машини для обчислень, яку запропонував Ч. Беббідж.
- 6°. Кого вважають першим програмістом у світі? Чому?
- 7°. Опишіть період створення перших електронних обчислювальних машин (40–50-ті рр. ХХ ст.). Які вчені зробили значний внесок у розвиток електронних обчислювальних машин у цей період?
- 8°. Назвіть перші персональні комп'ютери. Коли їх створили?
- 9°. Опишіть розвиток комп'ютерної техніки в Україні.
- 10*. Поясніть, чому існує значна відмінність у значеннях властивостей різних комп'ютерів. Наведіть приклади класифікації персональних комп'ютерів.
- 11°. Що впливає на вибір комп'ютера для певних сфер застосування?



Виконайте завдання

- 1°. Підготуйте схему класифікації комп'ютерів, використавши будь-яку прикладну програму, яку ви знаєте.
- 2°. Виконайте в Інтернеті або друкованих засобах пошук відомостей про перші електронні обчислювальні машини та заповніть порівняльну таблицю:

<i>Властивості</i>	<i>Значення властивостей ЕОМ</i>		
	<i>Z4</i>	<i>Colossus Mark2</i>	<i>ABC</i>
Розробники			
Кількість електронних ламп			
Використання двійкового кодування			
Зберігання програми в пам'яті			



3*. Виконайте порівняння значень основних властивостей перших електронних обчислювальних машин: американської ENIAC та створеної в Україні MEOM. Результати порівняння опишіть у вигляді таблиці в зошиті.



4*. Підготуйте повідомлення про розробку українськими вченими ЕОМ для управління атомними криголамами.

5°. Доберіть комп'ютер для використання як домашній мультимедійний центр. Свій вибір обґрунтуйте.



ПРАКТИЧНА РОБОТА № 2

«Конфігурація комп'ютера під потребу»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Доберіть комп'ютери для людей різних професій і різних сфер застосування:
 - письменник, працюючи у своєму кабінеті, використовує комп'ютер для створення своїх прозових творів, пошуку відомостей, для надсилання й отримання електронних листів тощо;
 - студент використовує комп'ютер для підготовки навчальних завдань, а також для створення власної фонотеки, створення і редагування навчальних відеофільмів;
 - кореспондент Інтернет-видання використовує комп'ютер для написання і своєчасного надсилання до редакції повідомлень з новинами під час відряджень у різні частини світу.
2. За результатами виконання завдання заповніть таблицю значень властивостей.

Назва властивості	Значення властивостей для комп'ютерів		
	письменника	студента	кореспондента
Тип комп'ютера			
Модель процесора			
Тактова частота процесора, ГГц			
Обсяг оперативної пам'яті, Гбайт			
Ємність жорсткого диска, Гбайт			
Тип відеоадаптера			
Обсяг відеопам'яті, Гбайт			
Тип звукового адаптера			
Монітор			
Діагональ монітора, дюймів			
Роздільна здатність			
Час відклику, мс			

3. У текстовому процесорі створіть файл, у якому обґрунтуйте ваш вибір конфігурації для цих комп'ютерів.
4. Файл з обґрунтуванням надішліть на електронну пошту вчителя.



2.4. КЛАСИФІКАЦІЯ ТА ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

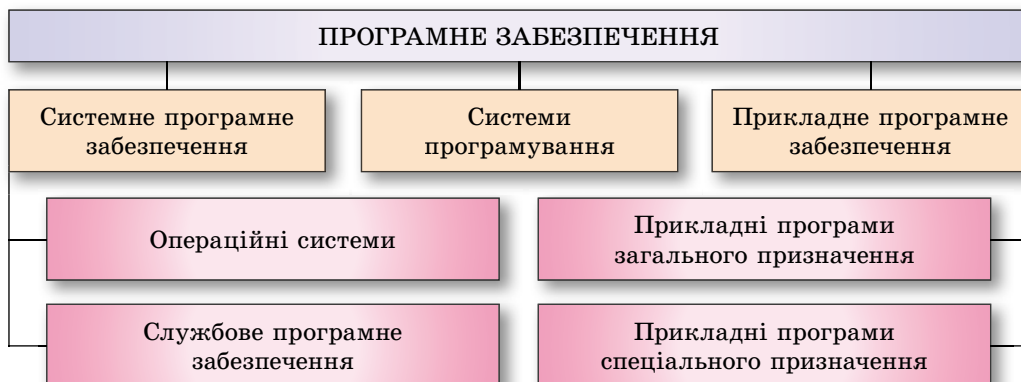


1. Які комп'ютерні програми ви вже знаєте? Для чого вони призначені?
2. Що таке операційна система? Для чого вона призначена?
3. Опишіть способи запуску програм.

КЛАСИФІКАЦІЯ ТА ЗАГАЛЬНА ХАРАКТЕРИСТИКА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Ви вже знаєте, що є різноманітні програми для комп'ютерів. Є програми для опрацювання текстових, графічних, мультимедійних даних, є програми для керування роботою комп'ютера, його окремих пристроїв, таких як принтер, сканер, монітор та ін. Сукупність програм для комп'ютерів ще називають **програмним забезпеченням (ПЗ)**. Тільки під керуванням програмного забезпечення комп'ютер здатний опрацьовувати різноманітні дані.

Сучасне програмне забезпечення поділяється на такі види (мал. 2.40).



Мал. 2.40. Класифікація програмного забезпечення

Системне програмне забезпечення призначене для керування роботою складових комп'ютера та обміном даними між ними, діагностування та усунення недоліків у роботі комп'ютера, автоматизації процесу опрацювання даних, організації обміну даними між користувачем і комп'ютером. До складу системного програмного забезпечення належать операційні системи та різноманітні службові програми.

Системи програмування призначені для створення нових програм з використанням мов програмування, наприклад **Scratch 1.4**, **Free Pascal 2.6**, **DEV-C++ 5.11**, **Microsoft Visual Studio 2013 Professional**, **Android Studio 1.4.0**, **Lazarus 1.4.4**, **Python 2.6.1** та ін.

Прикладне програмне забезпечення призначене для реалізації конкретних задач опрацювання даних, які користувач розв'язує в ході своєї діяльності. Їх поділяють на прикладні програми загального і спеціального призначення.

До прикладних програм **загального призначення** належать програми, які можуть застосовуватися в різних галузях людської діяльності для опрацювання текстів, малюнків, мультимедійних даних, електронних таблиць, створення презентацій тощо.

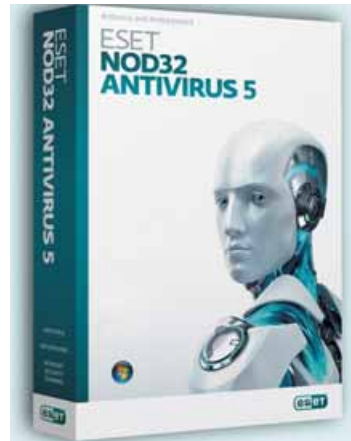
Прикладні програми **спеціального призначення** використовуються для реалізації завдань опрацювання даних у певній галузі діяльності, на конкретному підприємстві, в організації, фірмі або їх підрозділі. До такого типу програм належать програми для створення відеоефектів під час виробництва кінофільмів, креслень машин і механізмів у конструкторських і проектних бюро, діагностування захворювань у медичних закладах, створення шкільного розкладу уроків тощо.

ЛІЦЕНЗІЇ НА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ЇХ ТИПИ

На комп'ютерні програми поширюється дія Закону України «Про авторське право і суміжні права», і їх використання можливе лише за умов дотримання вимог цього закону, а також вимог ліцензії, з якою користувач погоджується, установлюючи програму на свій комп'ютер.

Є кілька видів ліцензій на комп'ютерні програми:

- **комерційна ліцензія** – передбачає, що користувач оплачує вартість використання даної програми на одному чи кількох зазначених в ліцензії комп'ютерах, серед них розрізняють:
 - «коробкові» версії ліцензії продаються, зазвичай, для одного екземпляра програми, містять носій, на якому записана програма, та інструкцію до її використання разом з ключем для встановлення (мал. 2.41);
 - **OEM** (англ. *Original Equipment Manufacturer* – оригінальний виробник обладнання) – ліцензія, що надається на один екземпляр програми разом з певним комп'ютерним обладнанням, наприклад з ноутбуком, підтвердженням ліцензії є спеціальна наклейка (мал. 2.42);
 - **корпоративна ліцензія** – ліцензія на кілька копій програми для використання, наприклад на комп'ютерах навчальних закладів;
- **пробна** або **trial** (англ. *trial* – випробування) – ліцензія на комерційне ПЗ, яка надається для пробного використання програми протягом певного часу (зазвичай близько місяця) або на певну кількість запусків програм. При цьому можуть працювати не всі функції програми;
- **вільного використання** або **freeware** (англ. *freeware* – вільний товар) – ліцензія перед-



Мал. 2.41. «Коробкова» версія антивірусної програми



Мал. 2.42. Наклейка OEM-версії ліцензії операційної системи



бачає вільне використання програм без виплат винагороди автору, але не передбачає можливості внесення змін у програму;

- **з відкритим кодом** або *free* (англ. *free* – вільний) *software* чи *libre* (ісп. *libre* – вільний) *software* – ліцензія на вільне програмне забезпечення, що передбачає не тільки безкоштовне використання програм, але і право на їх модифікацію, внесення змін у програму.

КЛАСИФІКАЦІЯ ТА ОСНОВНІ ФУНКЦІЇ ОПЕРАЦІЙНИХ СИСТЕМ

Як ви вже знаєте з курсу інформатики 6-го класу, існує багато різноманітних операційних систем (ОС). Залежно від призначення їх поділяють на *операційні системи для стаціонарних персональних комп'ютерів* (Windows 10, OS X тощо), *для мобільних комп'ютерів* (Windows Mobile 6.0, iOS 9.0, Android 6.0 тощо), *для серверів і суперкомп'ютерів* (Unix, Linux, Windows Server 2012, Solaris 11 тощо).

Також операційні системи можна поділити за типом ліцензії, що використовуються під час їх розповсюдження. Так, операційні системи **Windows** мають комерційну ліцензію і вільно не розповсюджуються, проте окремі з них можуть використовувати *trial*-ліцензію. До ОС, що розповсюджуються за ліцензією *free software* – вільного програмного забезпечення, у першу чергу належать різні версії операційної системи **Linux**.

Операційні системи призначені для керування роботою пристроїв комп'ютера та забезпечення обміну даними між ними, зберігання даних в оперативній пам'яті та на зовнішніх носіях, забезпечення виконання інших програм та обміну даними між користувачем і комп'ютером.

ПОНЯТТЯ ІНСТАЛЯЦІЇ ТА ДЕЙНСТАЛЯЦІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Для встановлення ОС на комп'ютері потрібно виконати спеціальну операцію, яку називають **інсталяцією операційної системи**.

Для інсталяції ОС необхідно мати оптичний диск або флеш-носій з інсталяційним пакетом – **дистрибутивом** (англ. *distribute* – розподіляти). Під час завантаження з такого диска або флеш-носія програма встановлення ОС запитає в користувача дозвіл на інсталяцію та на наступних кроках запропонує ввести ряд додаткових даних – диск, на який буде встановлено ОС, мову та набір регіональних стандартів тощо. Наприклад, для початку процесу інсталяції ОС **Linux Ubuntu** потрібно вибрати піктограму **Встановити Ubuntu** на **Робочому столі** або в **Панелі запуску** (мал. 2.43). Існують також інші варіанти виконання інсталяції, наприклад з використанням локальної мережі або Інтернету.



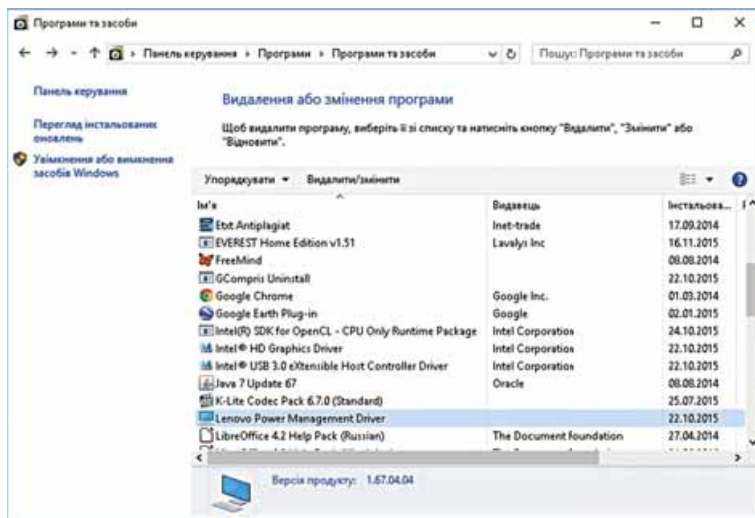
Мал. 2.43. Робочий стіл **Linux Ubuntu** з ярликами програми інсталяції

Під час інсталяції відбувається розміщення складових ОС на вибраному диску, налагодження її взаємодії з апаратною складовою комп'ютера. Диск, на який встановлено операційну систему, називають **системним**.

Будь-яке програмне забезпечення, що використовується в сучасних комп'ютерах, працює під управлінням операційної системи. Кожна з програм створюється для певної операційної системи і не може працювати з іншою, не сумісною. Перед початком використання комп'ютерної програми має відбутися процес її узгодження з операційною системою. Цей процес також називають **інсталяцією** і відбувається відповідно до алгоритму, записаного у спеціальній програмі, наприклад, *setup.exe*, *install.exe* (для **Windows**) або *INSTALL.sh* чи файлів з розширенням *deb* (для **Linux**).

У подальшому потрібно дотримуватися вказівок програми інсталяції. Якщо інсталяція пройшла успішно, то команда для запуску програми буде включена:

- для **Windows** – до меню **Пуск** (як правило, у розділ **Усі програми**) і до списку вже встановлених програм;

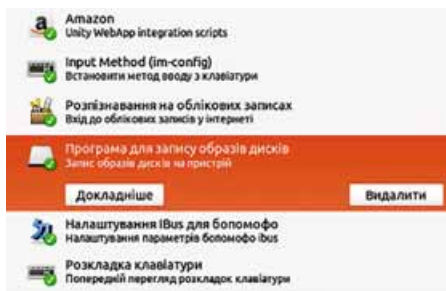


Мал. 2.44. Вікно Програми та засоби

- для **Linux Ubuntu** – до Панелі запуску та до Центру програмного забезпечення.

Видалення (деінсталяція) програм проводиться:


- для **Windows** – у вікні Програми та засоби (**Пуск** ⇒ **Усі програми** ⇒ **Система** ⇒ **Windows** ⇒ **Панель керування** ⇒ **Програми** для **Windows 10** або **Пуск** ⇒ **Панель керування** ⇒ **Програми** для **Windows 7**) слід вибрати програму,



Мал. 2.45. Фрагмент вікна Центру програмного забезпечення



яку потрібно видалити або змінити її налаштування, після чого – кнопку **Змінити/Видалити** (мал. 2.45);

- для **Linux Ubuntu** – у вікні **Центру програмного забезпечення (Панель запуску ⇒ Центр програмного забезпечення **) вибрати програму, яку потрібно видалити, та вибрати кнопку **Видалити** (мал. 2.45).








Вибір операції видалення запускає програму, яка коректно видаляє всі компоненти програми з комп'ютера.

ПРОБЛЕМИ СУМІСНОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Комп'ютерні програми розробляються для певної операційної системи, відповідно до стандартів, які встановлюють розробники ОС. Програми, розроблені для одного типу операційних систем, як правило, є несумісними для інших типів або версій операційних систем і не можуть бути інстальованими. Так, програми, розроблені для ОС **Windows**, не можуть працювати в ОС сімейства **Linux**. Для того щоб певна програма запрацювала в іншій операційній системі, її код потрібно змінити з урахуванням вимог іншої операційної системи. Так, існують, наприклад, браузерери **Chrome**, для **Windows**, **Linux**, **Android** тощо (мал. 2.46).

Увага:

У вас є можливість скачати **Google Chrome** для **linux**, скачати **Google Chrome** для **mac**, скачати **Google Chrome** для **android**, а також скачати **Google Chrome** для **ios** (посилання для завантаження приведено нижче). Скачати **Google Chrome**

Ліцензія:	Freeware
Автор:	Google
Оновлено:	4-02-2015, 18:23
Платформа:	    
Українська мова:	
Російська мова:	

Мал. 2.46. Фрагмент веб-сторінки завантаження браузера **Google Chrome**

Певне програмне забезпечення не завжди коректно працює навіть під управлінням операційних систем одного сімейства, але різних його версій. Так, окремі програми для **Windows 10** не зможуть працювати з **Windows XP**.

СЛУЖБОВЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ

Службове програмне забезпечення – це програми, що призначені для діагностування апаратної і програмної складових комп'ютера, розширення можливостей ОС. За необхідності вони усувають недоліки та оптимізують роботу комп'ютера. Ці програми називають **утилітами** (англ. *utility* – корисність). Частина таких програм включається до складу операційних

систем при інсталяції ОС. Наприклад, до складу операційної системи **Windows 7** входять такі службові програми, як **Відновлення системи**, **Дефрагментація диску**, **Очистка диску**, **Монітор ресурсів**, **Відомості про систему**, **Планувальник завдань**, **Панель керування** та ін.

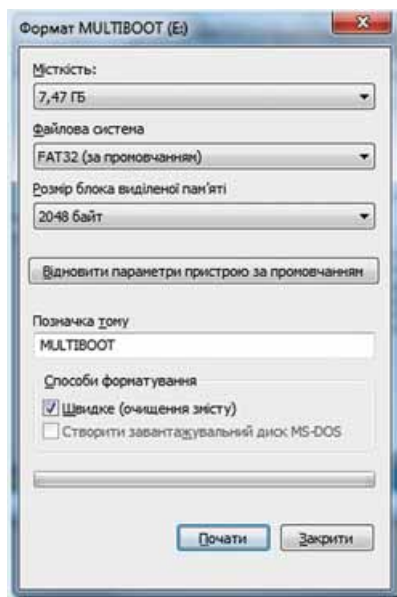
Аналогічні програми є і в інших операційних системах. Так, у **Linux Ubuntu** такими програмами, наприклад, є: **Менеджер архівів**, **Журнал системи**, **Аналізатор використання диску**, **Системний монітор**, **System Testing**, **Диски** та ін.

Додаткові службові програми можна придбати або вільно скопіювати із сайтів розробників та інстальювати для подальшого використання.

ФОРМАТУВАННЯ НОСІЇВ ДАНИХ

Серед службових програм є програми, що виконують обслуговування пристроїв зовнішньої пам'яті. Їх ще називають програмами для роботи з дисками. До таких програм належать програми розмітки, форматування, перевірки, дефрагментації, очищення носіїв даних та ін.

Використання таких програм пов'язане з необхідністю забезпечення тривалого надійного зберігання даних, а також швидкого і точного пошуку потрібних даних.



Мал. 2.47. Вікно **Формат**

Ви знаєте, що для того, щоб швидко щось знайти у сховищі даних, потрібно ці дані упорядковано зберігати. Для забезпечення упорядкованого зберігання даних на жорстких магнітних дисках перед використанням диск розбивають на частини – **логічні диски**. Кількість логічних дисків задає користувач залежно від ємності диска і його призначення. Як правило, це 1–3 логічних диски, які в подальшому будуть розглядатися операційною системою як окремі об'єкти. Перший з них можна відвести для встановлення операційної системи, а інші – для збереження даних користувача.

Після створення логічних дисків потрібно виконати операцію **форматування** кожного з них. У ході форматування виконується розбиття диска на окремі «комірки пам'яті». Такі «комірки» називають «**сектори**». У кожному секторі може зберігатися від 0 до 512 байтів даних. У ході форматування для спрощення доступу до даних на

носіях великої ємності сектори об'єднують у **кластери**. Кластер може складатися з одного або кількох, наприклад чотирьох, секторів. В останньому випадку ємність кластера буде: $512 \text{ байтів} \cdot 4 = 2048 \text{ байтів}$.

Під час форматування також створюється структура таблиці розміщення файлів на носії.



Форматування проводять також і для інших зовнішніх запам'ятовуючих пристроїв. Так, для форматування флеш-носія потрібно виконати таку послідовність дій:

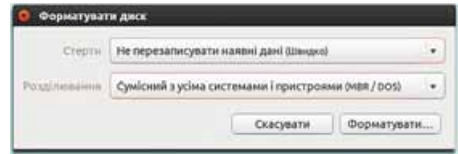
1. Відкрити вікно **Провідника** з переліком пристроїв зовнішньої пам'яті, виконавши – *Пуск* ⇒ *Комп'ютер*.
2. Відкрити контекстне меню пристрою флеш-пам'яті.
3. Вибрати команду **Форматувати**.
4. У вікні, що відкрилося (мал. 2.47), встановити:
 - тип файлової системи – у списку **Файлова система** (для використання на комп'ютерах з різними файловими системами рекомендується залишити за промовчанням файлову систему FAT32);
 - ємність кластера – серед запропонованих значень у списку **Розмір блока виділеної пам'яті**;
 - ім'я пристрою, що буде відображатися під час перегляду зовнішніх запам'ятовуючих пристроїв, – у полі **Позначка тому**;
 - спосіб форматування – встановленням або зняттям позначки прапорця **Швидке (очищення змісту)**;
5. Вибрати кнопку **Почати**.



Для тих, хто працює з Linux

Форматування носіїв даних в **Linux Ubuntu** можна виконати за таким алгоритмом:

1. Відкрити вікно **Диски (Панель запуску ⇒ Програми ⇒ Диски)**.
2. Вибрати потрібний носій у лівій частині вікна.
3. Вибрати кнопку **Додаткові дії** ⚙️.
4. Вибрати команду **Форматувати**.
5. Установити параметри форматування у вікні **Форматувати диск** (див. малюнок):
 - спосіб форматування – у списку **Стерти**;
 - тип файлової системи – у списку **Розділювання**.
6. Вибрати команду **Форматувати**.
7. Підтвердити виконання команди форматування вибором відповідної кнопки в діалоговому вікні.

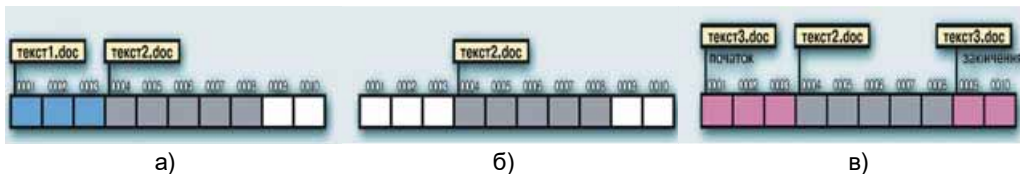


Для тих, хто хоче знати більше

Сучасні ОС автоматично виконують перевірку стану пристроїв зовнішньої пам'яті. Особливо це стосується накопичувача на жорстких магнітних дисках – основного пристрою збереження даних у персональних комп'ютерах. Часто виникає потреба в перевірці логічної структури та фізичного стану носіїв даних. Для цього користувач може скористатися стандартною службовою програмою перевірки носіїв, вибравши в контекстному меню пристрою збереження даних команду **Властивості** і на вкладці **Знаряддя** – кнопку **Виконати перевірку** 🗑️.

Для автоматичного виправлення помилок у файлах і папках, виявлених під час сканування, установіть позначку прапорця **Автоматично виправляти помилки файлової системи**. В іншому разі засіб перевірки диска лише повідомлятиме про помилки, але не виправлятиме їх.

Щоб виконати ретельну перевірку диска, установіть позначку прапорця **Перевіряти й намагатися відновити пошкоджені сектори**. Під час сканування виконується пошук і виправлення фізичних помилок безпосередньо на диску, тому цей процес може тривати набагато довше.



Мал. 2.48. Послідовність операцій запису і видалення файлів

Під час запису даних на носій ОС розбиває файли на частини, що відповідають ємності кластерів, і записує в перші вільні кластери на носіїві. Якщо виконуються тільки операції запису, то файли будуть записуватись у сусідні кластери підряд. Однак, якщо, крім запису даних, виконується і видалення файлів, то, зазвичай, виникає явище **фрагментації файлів**. Наприклад, спочатку було записано на носій файл **текст1.doc**, який займав три кластери, потім було записано ще один файл **текст2.doc**, який зайняв п'ять наступних кластерів (мал. 2.48, а). Згодом перший файл було видалено і звільнено три перші кластери (мал. 2.48, б). Далі було записано файл **текст3.doc**, що мав розмір п'яти кластерів. Для запису цей файл було розбито на дві частини (фрагменти). Перша частина розміром у три кластери була записана в ті, що звільнилися після видалення першого файлу, а друга (два кластери) – записана у вільні кластери після файлу **текст2.doc** (мал. 2.48, в). Що частіше виконуються операції запису та видалення файлів, то більшою стає кількість фрагментів файлів, які знаходяться в несусідніх кластерах.

Для усунення такої ситуації та прискорення обміну даними з носіями даних потрібно періодично виконувати **дефрагментацію носіїв**. У ході дефрагментації спеціальна програма намагається розмістити кожний з файлів у сусідніх кластерах диска. Для запуску цієї програми можна вибрати кнопку **Оптимізувати** на вкладці **Знаряддя** вікна властивостей пристрою збереження даних. У вікні, що відкрилося, слід вибрати потрібний носій даних і вибрати кнопку **Дефрагментувати диск**.

Найважливіше в цьому пункті

Сукупність програм для комп'ютерів називають **програмним забезпеченням (ПЗ)**. Його можна поділити на такі види: *системне програмне забезпечення, системи програмування, прикладне програмне забезпечення*.

Є кілька видів ліцензій на комп'ютерні програми: *комерційна ліцензія («коробкова», OEM, корпоративна), пробна (trial), вільного використання (freeware), з відкритим кодом (free software)*.

Залежно від призначення операційні системи поділяють на *операційні системи для стаціонарних персональних комп'ютерів, для мобільних комп'ютерів, для серверів і суперкомп'ютерів*. Також ОС поділяють залежно від типу ліцензії, що використовуються під час їх розповсюдження.

Для встановлення програм (включаючи операційні системи) потрібно виконати спеціальну операцію, яку називають **інсталяцією**. Програми, розроблені для одного типу операційних систем, як правило, є несумісними для інших типів операційних систем і не можуть бути інсталюваними.



Службове програмне забезпечення – це програми, які призначені для діагностування апаратної і програмної складових комп'ютера, розширення можливостей ОС.

Форматування носія даних – це процес його підготовки до запису даних, що полягає у створенні місць для збереження даних – секторів ємністю 512 байт, які об'єднуються у кластери. У ході форматування також створюється структура таблиці файлів, у яку будуть у подальшому заноситися дані про файли та кластери, у яких вони зберігаються.



Дайте відповіді на запитання

- 1°. Що таке програмне забезпечення?
- 2°. Опишіть класифікацію програмного забезпечення.
- 3°. Що таке системне програмне забезпечення? Наведіть приклади програм, що входять до його складу.
- 4°. Які програми належать до систем програмування? Наведіть приклади.
- 5°. Які з вивчених раніше програм належать до прикладних програм? Які з них є прикладними програмами загального призначення?
- 6°. Для яких цілей використовуються прикладні програми спеціального призначення? Наведіть приклади таких програм.
- 7°. Назвіть види ліцензій для комп'ютерних програм. Чим вони відрізняються?
- 8°. Опишіть послідовність підготовки носіїв даних до використання на прикладі жорстких магнітних дисків.
- 9°. Що таке кластер носія даних? Які значення може набувати ємність кластера?
- 10°. Опишіть послідовність операцій форматування носія даних.



Виконайте завдання



1°. Підготуйте схему повідомлення про класифікацію видів ліцензій програмного забезпечення.

2°. Ємність одного кластера диска 1024 байти. На диск записали файли розміром 2750 байтів і 324 Кбайт. Скільки кластерів займуть ці файли?



3°. Відкрийте вікно програми **Очищення диска** (*Пуск* ⇒ *Усі програми* ⇒ *Стандартні* ⇒ *Службові інструменти* ⇒ *Очищення диска*). Визначте, у чому полягає операція очищення дисків, у яких групах об'єктів відбувається пошук файлів, які можуть бути зайвими. Який загальний розмір файлів, від яких програма пропонує очистити диски?

4°. Проведіть форматування одного зі змінних носіїв (за вказівкою вчителя). Під час форматування дотримуйтеся наведеного в підручнику алгоритму та встановіть такі значення властивостей форматування:

- тип файлової системи – залишити запропоновану операційною системою;
- ємність кластера – 512 байтів;
- позначка тому – *флешка_Прізвище*;



- спосіб форматування – *швидке*.

5*. Дослідіть відмінності в реальних розмірах файлів та розмірах, які вони займають на носіїві даних для трьох файлів, указаних учителем, наприклад, що розміщені в папці **Розділ 2\Вправа 2.5**. Заповніть таблицю та визначте ємність кластера диска.

Ім'я файлу	Розмір файлу	Розмір файлу на диску	Ємність кластера



6*. Складіть схему класифікації операційних систем за наведеним у підручнику описом. Використовуючи матеріали Інтернету, доповніть цю схему класифікацією за значенням ще однієї властивості операційних систем.



2.5. СТИСНЕННЯ І АРХІВУВАННЯ ДАНИХ



1. З якими комп'ютерними програмами ви вже вмієте працювати? Для чого вони призначені?
2. Що таке службове програмне забезпечення? Для чого воно призначене?
3. Що таке архів? Для чого він використовується?

СТИСНЕННЯ ДАНИХ, ВИДИ СТИСНЕННЯ ДАНИХ

Часто виникає потреба у зменшенні розмірів даних, що зберігаються в пам'яті комп'ютера. Для цього використовують спеціальні способи стиснення даних, які називають **алгоритмами (методами) стиснення даних**. Стиснення даних використовується під час створення файлів певних типів, наприклад, графічних типу **TIFF, JPEG, PNG** або звукових типу **MPEG3, WMA**, для передавання файлів мережею тощо.

Стиснення даних – це процес **перекодування даних з метою зменшення розмірів файлів**.

Розрізняють алгоритми стиснення, що забезпечують стиснення **без втрати даних**, і алгоритми, що передбачають **часткову втрату даних**. Алгоритми із частковою втратою даних (інша назва «**з регульованою втратою даних**») використовують, коли цілісність даних не є дуже суттєвою. Наприклад, при стисненні графічних, відео, звукових файлів, оскільки органи чуття людини не завжди здатні помітити незначну відмінність у відтінках кольорів на фотографії, у відтворенні звукових або відеоданих тощо. Або втрата в якості не така важлива, як необхідність зменшити розміри файлів для запису на оптичний диск або для використання в Інтернеті. Однак ці алгоритми не можна застосовувати під час стиснення текстових та числових даних.



Для тих, хто хоче знати більше

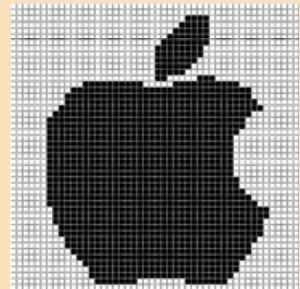
Методи стиснення даних

Алгоритм Д. Хаффмана та **алгоритм Шеннона–Фано** передбачають перекодування даних. Символи, що частіше зустрічаються в повідомленні, перекодовуються меншою кількістю символів, а які рідше – більшою. Так, наприклад, для кодування у **Windows-1251** попереднього речення, яке містить 114 символів, включаючи пробіли, необхідно використати 1 байт даних на кожен символ, або 114 байтів. Якщо ж порахувати, скільки разів той чи інший символ зустрічається в реченні, то виявиться, що всього символів, які є в реченні, – 28, частіше зустрічається символ «пробіл» – 14 разів, символ «і» – 10, «о» – 9, «с» – 7, «в» – 6 і т. д. Найменше, один раз, зустрічаються символи «щ», «б», «з», «–», «.». Якщо застосувати алгоритм стиснення даних для двійкового коду (0 та 1) і позначити символи, що частіше зустрічаються одним символом (1 біт): символ «пробіл» – 0, а символ «і» – 1, наступні за частотою повторюваності – двома символами (2 біти): символ «о» – 00, символ «с» – 01 і т. д. Для символів, що зустрічаються рідше, буде використано чотири двійкових цифри або 4 біти. У результаті загальна довжина коду закодованого таким способом речення буде складати 90 бітів або 11 байтів і 2 біти. Як бачимо загальний обсяг коду речення зменшиться в 10 разів.

Звичайно, що це досить спрощена схема кодування, не слід забувати, що у стиснутому таким способом файлі повинна міститися таблиця кодування і додаткові службові відомості. Тому при кодуванні дуже малих файлів може відбутися не зменшення, а, навпаки, збільшення розмірів файлу.

Для файлів з великим обсягом текстових чи цифрових даних цей алгоритм дає високий ступінь стиснення.

Алгоритм RLE (англ. *Run-Length encoding* – кодування повторів або довжин серій) використовується частіше для кодування графічних даних. Цей метод передбачає заміну послідовності однотипних даних її скороченим описом. Наприклад, у малюнку логотипу корпорації *Apple* (мал. 2.49) використано тільки два кольори – чорний і білий. Якщо позначити чорний колір цифрою «1», а білий – «0», тоді код першого рядка зображення повинен складатися із 48 цифр «0» (48 бітів). Але їх можна замінити описом, указавши код символа і кількість повторів – 0 48. На код опису буде відведено 9 бітів: 1 біт на цифру «0» і по чотири біти на дві цифри кількості повторів. Другий ряд пікселів буде закодовано так само, а третій: 0 29 1 4 0 15–23 біти. Для кольорових зображень з великою кількістю однокольорових фрагментів даний алгоритм забезпечить високу ефективність стиснення.



Мал. 2.49. Логотип Apple

АРХІВУВАННЯ ДАНИХ

Незважаючи на підвищення надійності комп'ютерів і комп'ютерних носіїв даних, повної гарантії збереження даних вони все ж не дають. Втрата даних може призвести до дуже серйозних наслідків. Так, видалення або пошкодження даних про вклади та перерахування коштів клієнтів може призвести до краху банку, втрата даних про продаж квитків – ускладнить перевезення пасажирів, знищення результатів дослідів може звести нанівець багаторічні наукові дослідження. Навіть втрата записника з номерами телефонів друзів завдасть вам клопоту. Тому виникає потреба у створенні копій

даних. Найважливіші дані дублюють, записуючи на інші жорсткі диски, на оптичні диски тощо. Зазвичай, для зручності використання та зменшення обсягів даних, файли і папки під час створення резервних копій упаковують в один файл. Такі копії даних називають *архівами*, а файли, у які вони упаковуються, – *архівними файлами*, або для спрощення – *архівами*.

Створення копій даних з використанням спеціальних програм, що можуть використовувати стиснення даних, називають *архівуванням*.

Більшість операційних систем мають у своєму складі засоби створення резервних копій даних. Ці програми, зазвичай, здійснюють копіювання даних для відновлення операційної системи на тому самому або на іншому диску, а також створюють архіви файлів користувача.

Для створення резервних копій файлів потрібно:

1. Відкрити вікно налаштувань архівування і відновлення файлів (**Пуск** ⇒ **Панель керування** ⇒ **Система й безпека** ⇒ **Резервне копіювання і відновлення**).

Для яких елементів створювати резервні копії?

Установіть прапорці поруч із елементами, які потрібно включати до резервних копій.



Мал. 2.50. Фрагмент вікна **Налаштування резервного копіювання**

2. Вибрати посилання **Налаштування резервного копіювання** (для першого запуску резервного копіювання або посилання **Змінити параметри** для наступних).
3. Указати пристрій, на який буде записано архівний файл.
4. Указати перелік папок з файлами, що будуть включені до резервної копії (мал. 2.50).
5. Змінити, за потреби, розклад здійснення автоматичного резервного копіювання.
6. Розпочати процес створення архіву даних користувача вибором кнопки **Зберегти настройки та запустити резервне копіювання**.

В останніх версіях операційних систем є можливість створювати **образи системи**, у які включаються як копії файлів користувача, так і всі налаштування операційної системи з інсталюваними раніше програмами. Відновлення даних з такого образу забезпечує швидке повернення до попереднього стану всієї системи без необхідності повторної інсталяції програм.



Для створення образу системи у **Windows 7** потрібно:

1. Виконати **Пуск** ⇒ **Панель керування** ⇒ **Система й безпека** ⇒ **Резервне копіювання і відновлення**.
2. Вибрати посилання **Створити образ системи** в лівій частині вікна.
3. Вибрати місце збереження образу та носії (крім системного, що обирається обов'язково), дані з яких будуть включені до образу системи.
4. Вибрати кнопку **Розпочати резервне копіювання**.

У результаті на обраному носії буде створено файл образу системи вашого комп'ютера, що має розширення **iso**. Під час налаштування автоматичного резервного копіювання буде відбуватися **оновлення змісту архівних файлів**. До архівів будуть додаватися нові файли та папки з установлених користувачем місць, а також оновлені версії раніше включених до архівів файлів.

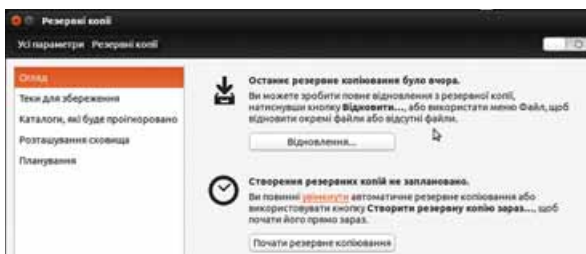
Для відновлення даних з резервної копії або з образу диску потрібно у **вікні резервного копіювання і відновлення** вибрати посилання, що розпочинають відновлення даних **Відновити мої дані**, **Відновити всі файли користувачів** або **Вибрати іншу резервну копію...** У першому випадку будуть відновлені дані тільки користувача, сеанс роботи якого зараз іде, у другому – дані всіх користувачів комп'ютера, що були збережені, і у третьому потрібно вказати місце розміщення збереженої резервної копії даних.



Для тих, хто працює з Linux

Для створення резервних копій слід відкрити вікно системних параметрів (**Панель запуску** ⇒ **Системні параметри**) та у групі **Системна** запустити програму **Резервні копії** (мал. 2.51). У вікні програми можна встановити папки (теки), що будуть включені до резервних копій, розміщення сховища для резервних копій, налаштувати параметри автоматичного резервування тощо. Вибравши кнопку **Відновлення**, можна розпочати відновлення даних із збереженої раніше резервної копії даних.

Для створення образів системи в **Linux Ubuntu** можна використати програми роботи з оптичними дисками – **Brasero** або **КЗб**. Це програми, що, зазвичай, доступні після інсталяції операційної системи, або їх можна завантажити із сайту програмного забезпечення **Linux**, використовуючи **Центр програмного забезпечення Ubuntu**.



Мал. 2.51. Вікно програми **Резервні копії**

АРХІВАТОРИ І ТИПИ АРХІВІВ

Крім засобів операційної системи, існують інші службові програми, що забезпечують архівування даних.

Програми, що використовуються для виконання операцій над архівами, називають *архіваторами*.

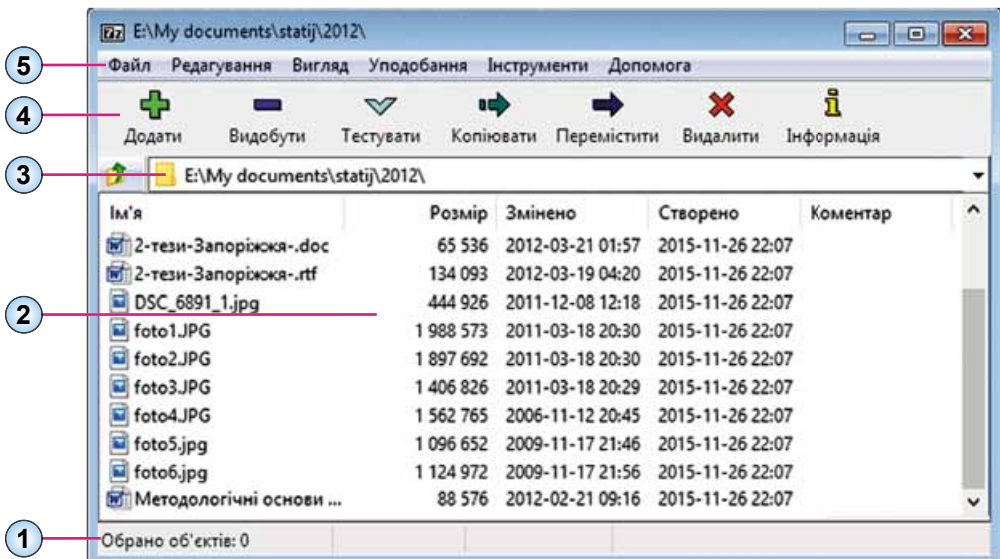
Прикладами архіваторів є програми **WinZIP, WinRAR, 7-zip, PowerArchiver, WinAce, Ark, Vtar, AndroZip, FreeArc** та ін. Результатом роботи цих програм є *архівний файл*, який містить у стисненому або не стисненому стані файли і папки. У процесі архівування можуть бути використані додаткові заходи стосовно захисту даних від несанкціонованого доступу, наприклад встановлення пароля на доступ до даних в архіві.

Залежно від алгоритмів, за якими виконується стиснення і архівування даних, розрізняють такі формати архівних файлів: **ZIP, RAR, TER, ARJ, CAB, LZH, ACE, 7z** та ін. Найчастіше, особливо в мережі Інтернет, використовують архівні файли формату **ZIP**.

ОПЕРАЦІЇ НАД АРХІВАМИ

До основних операцій над архівами належать:

- створення архівів файлів і папок з можливим стисненням даних;
- додавання файлів і папок до вже існуючих архівів та заміна в них уже включених об'єктів;
- перегляд вмісту архівів;
- заміщення й оновлення файлів і папок в архівах;
- видобування з архіву всіх або тільки обраних файлів і папок;
- створення багатотомних архівів (архів розбивається на кілька окремих файлів – томів); розмір томів встановлює користувач;



- | | |
|-----------------|------------------------|
| 1. Рядок стану | 4. Панель інструментів |
| 2. Робоче поле | 5. Рядок меню |
| 3. Рядок адреси | |

Мал. 2.52. Вікно програми 7-zip





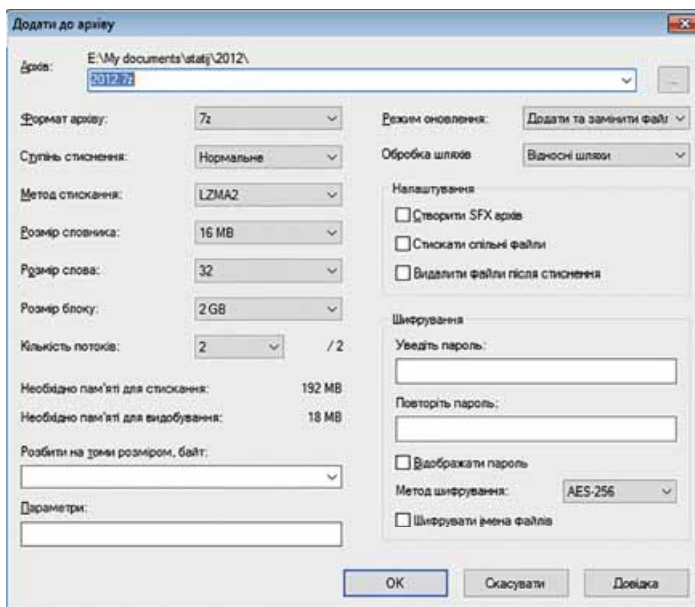
- створення звичайних і багатотомних архівів, які містять програму самостійного видобування файлів і папок, без участі програми-архіватора – так званих **SFX**-архівів (англ. *Self eXtracting* – самовидобування);
- перевірка цілості даних в архівах;
- шифрування даних та імен файлів в архівах та ін.

Розглянемо послідовність виконання операцій над архівами з використанням програми **7-zip**. Ця програма забезпечує роботу з архівними файлами типу *7z*, *ZIP*, *CAB*, *Z*, *RAR*, *ARJ*, *BZIP2*, *TAR*, *LZH* та ін. Особливістю є те, що програма належить до програм з ліцензією **freeware** і може бути вільно завантажена із сайту авторів програми. Існують версії програми як для ОС сімейства **Windows**, так і для **Linux**.

Вікно програми має вигляд, поданий на малюнку 2.52.

Для **створення** архівного файлу потрібно виконати такий алгоритм:

1. Запустити програму **7-zip** (*Пуск* ⇒ *Усі програми* ⇒ *7-zip*).
2. Зробити поточною папку, в якій розміщено файли, що планується включити до архіву.
3. Виділити файли, які планується включити до архіву, та вибрати кнопку  *Додати*.
4. Установити у вікні **Додати до архіву** значення властивостей архіву (мал. 2.53):
 - *Ім'я* – у полі імені файлу;
 - *Місце збереження* – використовуючи кнопку **Переглянути** ;
 - *Формат архіву* – *7z*, *tar*, *wim* або *zip*;
 - *Ступінь стиснення* – *без стиснення*, *найшвидше*, *швидке*, *нормальне*, *максимальне*, *ультра*;
 - *Метод стиснення*;



Мал. 2.53. Вікно **Додати до архіву**


- Розбити на томи розміром, байт – за потреби записати архівний файл на кілька носіїв, що мають порівняно невеликий розмір, наприклад файл з відео на компакт-диск;
- Створити SFX-архів – для випадків, коли потрібно буде виконати видобування файлів з архіву без програми-архіватора;
- Уведіть пароль та Повторіть пароль – для захисту вмісту архіву від доступу сторонніх осіб та інші.

5. Вибрати кнопку **ОК**.


Для **додавання файлів до вже існуючого архіву** потрібно на четвертому кроці виконання алгоритму по створенню архіву вказати ім'я і шлях до вже існуючого файлу-архіву.


Для **перегляду вмісту** архівного файлу потрібно відкрити його у вікні провідника або програми-архіватора.

Для **видобування всіх файлів** з архіву потрібно:

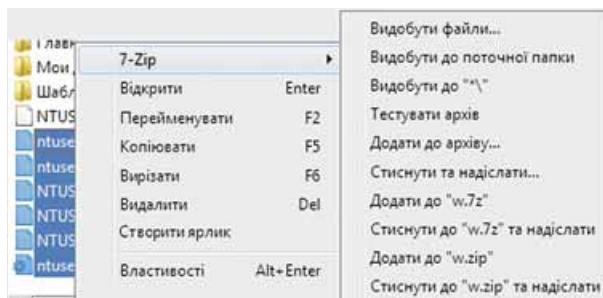
1. Запустити програму **7-zip**.
2. Зробити поточною папку, в якій розміщено файл архіву, та вибрати його.
3. Вибрати кнопку .
4. Указати шлях до папки, у яку буде виконано видобування, та інші значення властивостей у вікні **Видобути**.
5. Вибрати кнопку **ОК**.

Для **видобування окремих файлів і папок** з архіву потрібно:

1. Запустити програму **7-zip**.
2. Виконати **Файл** ⇒ **Відкрити**.
3. Виділити файли і папки, які планується видобувати.
4. Вибрати кнопку .
5. Указати шлях до папки, у яку буде виконано видобування, у вікні **Копіювати**.
6. Вибрати кнопку **ОК**.

Для **видалення файлів з архіву** потрібно відкрити файл архіву в програмі **7-zip**, виділити потрібні файли і вибрати кнопку .

Після правильної інсталяції програми-архіватора до контекстного меню файлів і папок додаються команди роботи з архівами (мал. 2.54), які можна використовувати без запуску програми-архіватора.



Мал. 2.54. Контекстне меню групи файлів



Для тих, хто працює з Linux

Виконати операції з архівами у **Linux Ubuntu** можна з використанням, крім уже описаної програми **7-zip**, програму **Менеджер архівів**. У ході створення архівів може бути обраний один з більш ніж 10 форматів архівних файлів, можна встановити пароль доступу до архіву та розділити архів на кілька томів.



Мал. 2.55. Вікно програми Менеджер архівів

Найважливіше в цьому пункті

Для зменшення розмірів даних використовують **стиснення даних** за спеціальними алгоритмами (методами) стиснення. Стиснення даних – це процес перекодування даних, з метою зменшення розмірів файлів.

Розрізняють алгоритми стиснення, що забезпечують стиснення *без втрати даних*, і алгоритми, що передбачають *часткову втрату даних*.

Створення копій даних з використанням спеціальних програм, що можуть використовувати стиснення даних, називають **архівуванням**. Для створення **резервних копій** та **образів системи** використовуються відповідні службові програми, які, як правило, включені до складу операційних систем.

Програми, що використовуються для виконання операцій над архівами, називають **архіваторами**. Над архівами виконуються такі операції: *створення* (звичайних, багатотомних та SFX-архівів), *перегляд вмісту*, *додавання* нових та *заміна* раніше включених об'єктів, *видалення* об'єктів, *видобування* всіх або окремих об'єктів тощо.

Дайте відповіді на запитання

- 1°. Для чого використовується стиснення даних?
- 2°. У яких випадках можливе використання стиснення із частковою втратою даних?
- 3°. Для чого виконується архівування даних?
- 4*. Що таке архівування і що таке стиснення файлів? Який між ними зв'язок і яка відмінність?
- 5°. Як називають програми, що виконують архівування даних? Які операції над архівами вони можуть виконувати?
- 6°. Назвіть формати архівних файлів, які ви знаєте.
- 7°. Опишіть послідовність дій під час створення архіву кількох файлів.
- 8°. У чому відмінність у діях під час видобування всіх файлів з архіву і кількох файлів з архіву?
- 9°. Як видалити групу файлів з архівів?
- 10°. Опишіть послідовність дій під час створення архіву кількох файлів.
- 11°. Опишіть послідовність дій під час створення архіву кількох файлів з використанням контекстного меню.



Виконайте завдання

- 1°. Запустіть на виконання програму-архіватор. Створіть архів з перших п'яти файлів з розширенням імені **doc**, що зареєстровані в папці **Архів** папки **Розділ 2**. Помістіть цей файл у вашу папку.
- 2°. Видобудьте всі файли з файлу **Архів1.7z** з папки **Розділ 2\пункт 2.5\Архів** у папку **Мій архів**, яку потрібно створити у вашій папці.
- 3°. Додайте до архівного файлу **Архів2.zip**, який знаходиться в папці **Розділ 2\Архів**, два останні файли цієї самої папки. Збережіть змінений архів з тим самим іменем у вашій папці.
- 4°. Запустіть архіватор та:
 - а) створіть архів з перших п'ятнадцяти файлів з розширенням імені **doc**, що зареєстровані в папці **Мої документи** (або з п'ятнадцяти останніх файлів з розширенням імені **doc**, що зареєстровані в папці **Архів** папки **Розділ 2**);
 - б) установіть під час архівування такі значення параметрів: ім'я архіву – *документи*; папка – *Робочий стіл*; формат архіву – *ZIP*; метод стиснення – *без стиснення*;
 - в) використовуючи контекстне меню, видобудьте всі файли зі створеного архіву в папку **DOC**, яку створіть у папці **Мої документи**.
- 5*. Проведіть дослідження щодо ступеня стиснення одних і тих самих файлів при різних режимах стиснення у форматі **7z**. Для дослідження архівування використайте всі файли з розширенням імені **bmp**, що зареєстровані в папці **Розділ 2\пункт 2.5\Малюнки**. Заповніть таблицю. *Примітка: коефіцієнт стиснення – це відношення розміру файлів стиснення до розміру файлів після стиснення.*

№ п\п	Ім'я файлу архіву	Ступінь стиснення	Розмір файлів		Коефіцієнт стиснення
			до стис- нення	після стиснення	
1	Малюнки1.7z				
2	Малюнки2.7z				
3	Малюнки3.7z				
4	Малюнки4.7z				
5	Малюнки5.7z				
6	Малюнки6.7z				



- 6*. Запустіть архіватор та:
 - а) створіть архів з перших двох файлів, що зареєстровані в папці **Зразки музики** (**Пуск** ⇒ **Музика** ⇒ **Зразки музики**), установивши при цьому такі значення параметрів архівації: ім'я архіву – *Archiv02*; формат архіву – *7z*; *SFX-архів*;
 - б) видаліть зі створеного архівного файлу **Archiv02.exe** останній файл;
 - в) додайте до цього архіву третій файл з папки **Зразки зображень** (**Пуск** ⇒ **Музика** ⇒ **Зразки музики**);



г) видобудьте всі файли зі створеного архівного файлу в папку **Музика**, яку створить у папці **Мої документи**.

ПРАКТИЧНА РОБОТА № 3 «Архівування та розархівування даних»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Запустіть на виконання архіватор **7-zip**.
2. Створіть у вашій папці архів з перших шістнадцяти файлів з розширенням імені **doc**, що містяться в папці **Розділ 2\пункт 2.5\Архів**, установивши при цьому такі значення параметрів архівації:
 - формат архіву – *7z*;
 - ступінь стиснення – *швидкий*;
 - створити *SFX-архів*.
3. Виконайте такі дії:
 - протестуйте архів після створення;
 - створіть у вашій папці папку **Копія архіву**, скопіюйте в неї архів і видаліть з нього останні п'ять файлів.
4. Визначте і запишіть у зошит:
 - загальний розмір файлів до архівації;
 - загальний розмір файлів в архіві після архівації;
 - коефіцієнт стиснення.
5. Використовуючи контекстне меню архівного файлу у вікні **Провідника**, видобудьте всі файли зі створеного вами архіву в папку **DOC**, яку створить у вашій папці.
6. Використовуючи архіватор, видобудьте перші десять файлів з архіву, який ви створили, у папку **1_10**, яку створить у вашій папці.
7. Закрийте всі відкриті вікна.

Розділ 3. **Опрацювання текстових даних**

У цьому розділі ви дізнаєтеся про:

вставлення у текстовий документ символів, які відсутні на клавіатурі

стилі та їх використання

створення та опрацювання у текстовому документі колонок, нумерованих і маркованих списків

створення посилань і покажчиків у текстовому документі

перегляд структури документу та автоматичне створення його змісту

вставлення у текстовий документ таблиць, формул, графічних зображень та їх опрацювання

шаблони документів та їх використання

настроювання параметрів сторінок текстового документу та створення колонтитулів

роботу з кількома документами одночасно



3.1. СТВОРЕННЯ ТА ОПРАЦЮВАННЯ В ТЕКСТОВОМУ ДОКУМЕНТІ СИМВОЛІВ, КОЛОНОК, СПИСКІВ




1. Які способи виділення тексту ви знаєте?
2. Які операції редагування символів та абзаців тексту ви знаєте? Як вони здійснюються?
3. Які операції форматування символів і абзаців тексту ви знаєте? Як вони виконуються?
4. Яке призначення маркерів на горизонтальній лінійці?

ВСТАВЛЕННЯ В ТЕКСТОВИЙ ДОКУМЕНТ СПЕЦІАЛЬНИХ СИМВОЛІВ

Ви вже вмiєте створювати та опрацьовувати в текстовому процесорi **Word 2007** нескладнi текстовi документи: вводити текст, редагувати i формувати символи, абзаци та сторiнки, друкувати документ. Однак можливостi текстового процесора значно ширшi. Розглянемо деякi з них.

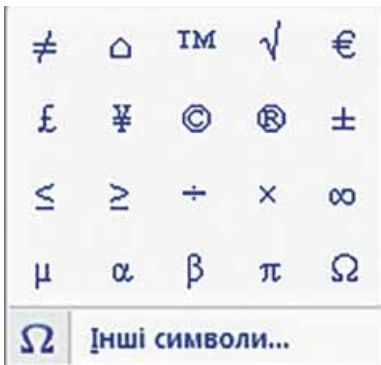
Так, у текстовому процесорi **Word 2007** можна в документ вставляти символи, якi вiдсутнi на клавіатурi, – лiтери iнших алфавiтiв, математичнi символи, умовнi позначки та iн. Наприклад: © ≤ β ∑ € ∞ ≈ √ ∫ ∘ ∫ § ¶ ∫ ∫ ∫.

Для вставлення у текст таких символiв потрiбно виконати алгоритм:

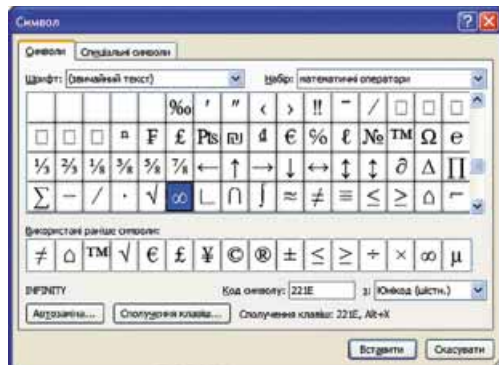
1. Встановити курсор у потрiбне мiсце тексту.
2. Вибрати на вкладцi **Вставлення** в групi **Символи** кнопку **Символ** .
3. Вибрати потрiбний символ у списку, що вiдкриється (мал. 3.1).

Якщо потрiбний символ вiдсутнiй у наведеному списку, то слiд:

1. Вибрати команду **iншi символи**.
2. Переглянути перелiк символiв у дiалоговому вiкнi **Символ** (мал. 3.2) на вкладках **Символи** та **Спецiальнi символи**. За потреби можна у списку поля **Шрифт** або списку **Набiр** вибрати iнше значення, щоб вивести у вiкнi iншiй набiр символiв.
3. Вибрати потрiбний символ.
4. Вибрати кнопку **Вставити**.



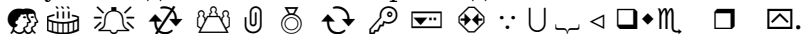
Мал. 3.1. Список символів




Мал. 3.2. Вікно Символ

Тим символам, які зустрічатимуться в тексті часто, можна у вікні **Символ** призначити певні сполучення клавіш (у вікні **Символ** вибрати кнопку **Сполучення клавіш** і заповнити поля вікна **Налаштування клавіатури**). Під час їх натискання потрібний символ буде вставлятися в поточне місце тексту.

У текстовому процесорі **Word 2007** у документ можна вставляти і символи у вигляді малюнків. Наприклад:



Для цього використовують графічні шрифти MS Outlook, Webdings, Wingdings та ін. Під час натискання на клавіші клавіатури в текст вводяться не літери, а різноманітні маленькі малюнки. Ці символи також можна вставити у вікні **Символ**.

Під час редагування тексту, особливо збереженого з Інтернету, бажано включати **режим відображення прихованих символів**, вибравши на вкладці стрічки **Основне** в групі **Абзац** кнопку **Відобразити всі символи** .

До **прихованих** (інша назва **недрукованих**) належать символи, які вводяться користувачем з клавіатури під час набору тексту, але зазвичай не відображаються на екрані й не виводяться на папері під час друкування. Але якщо включити режим відображення цих символів, то у відповідних місцях тексту з'являться спеціальні позначення цих символів (табл. 3.1).

Таблиця 3.1

Приклади прихованих (недрукованих) символів

Символ	Клавіші введення	Позначення	Символ	Клавіші введення	Позначення
Пропуск	Пропуск	.	Нерозривний пропуск	Ctrl + Shift + пропуск	°
Кінець абзацу	Enter	¶	Розрив рядка	Shift + Enter	↵
Табуляція	Tab	→	Розрив сторінки	Ctrl + Enter	— Розрив сторінки —

Включення **режиму відображення прихованих символів** дає змогу бачити на екрані всі символи тексту і краще розуміти структуру документа. Так, наприклад, якщо видалити символ **Кінець абзацу** ¶, то цей абзац об'єднається з наступним. А якщо в будь-якому місці абзацу встановити курсор і натиснути клавішу **Enter**, то він у цій позиції буде розділений на два і в місці розриву з'явиться символ **Кінець абзацу** ¶. А в тих місцях тексту, де натискали клавішу **Пропуск**, буде відображатися прихований символ •, якщо його видалити, то слова в цьому місці «злипнуться».

Редагування тексту і форматування всіх символів, уставлених у текст будь-яким способом, здійснюються за правилами, які ви вже знаєте.



Для тих, хто хоче знати більше

Деякі послідовності символів під час введення **Word 2007** автоматично замінює на спеціальні символи. Приклади таких послідовностей наведено в таблиці 3.2.




Таблиця 3.2


Введення спеціальних символів

Послідовність введених символів	Спеціальний символ	Послідовність введених символів	Спеціальний символ
слово, пропуск, дефіс, пропуск, слово	– тире	--> мінус, мінус, більше	→
:)	☺	:	☹

Режим автоматичної заміни вказаних послідовностей символів на спеціальні в програмі **Word 2007** встановлено за замовчуванням. Користувач може за власним бажанням додати до списку автозамін свої елементи, вибравши кнопку **Автозаміна** в діалоговому вікні **Символ**.


Для тих, хто працює з LibreOffice

Для вставлення символів, яких немає на клавіатурі, в **LibreOffice Writer** потрібно вибрати кнопку **Спеціальні символи**  на панелі інструментів **Стандартна**.

Для відображення прихованих символів потрібно вибрати кнопку **Недруковані символи**  на панелі інструментів **Стандартна**.

СПИСКИ ТА ЇХ СТВОРЕННЯ В ТЕКСТОВОМУ ДОКУМЕНТІ

Особливим видом форматування абзаців текстового документа є оформлення їх у вигляді **списків**. Списками можуть подаватися переліки об'єктів, описи порядку дій тощо. Наприклад, список прізвищ учнів класу, інструкція з користування приладом, перелік правил оформлення документа, список ліків у аптечці, послідовність дій під час приготування якої-небудь страви тощо.

У текстовому процесорі **Word 2007** можна створювати списки трьох типів:

- **Маркований**, у якому кожний елемент списку на початку першого рядка абзацу позначається деяким символом (**маркером**). Наприклад:

<i>Пори року:</i> ▪ Зима ▪ Весна ▪ Літо ▪ Осінь	<i>Пори року:</i> ➤ Зима ➤ Весна ➤ Літо ➤ Осінь	<i>Пори року:</i> ❖ Зима ❖ Весна ❖ Літо ❖ Осінь
---	---	---

- **Нумерований**, у якому на початку першого рядка абзацу вказується його номер. Порядковий номер у списку може задаватися числом, літерою або числівником тощо. Наприклад:

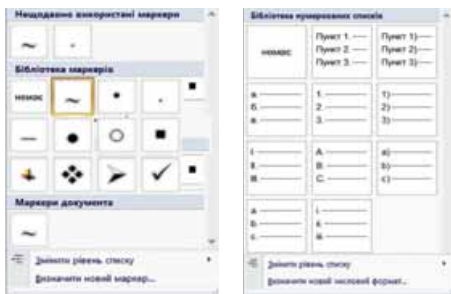
<i>Міста:</i> 1. Запоріжжя 2. Київ 3. Кременчук 4. Чернівці	<i>Об'єкти документа:</i> а) символ б) абзац в) сторінка г) документ	<i>Зміст підручника:</i> Пункт 1. Текстовий процесор Пункт 2. Табличний процесор Пункт 3. Графічний редактор Пункт 4. Мультимедійні презентації
---	--	---

- **Багаторівневий**, у якому в елементи списку можуть бути вкладені інші списки. Максимальна кількість вкладень елементів багаторівневого списку – 9 рівнів. Наприклад:

<p><i>Пори року:</i></p> <ol style="list-style-type: none"> 1. Зима <ul style="list-style-type: none"> • Грудень • Січень • Лютий 2. Весна <ul style="list-style-type: none"> • Березень • Квітень • Травень 3. Літо 4. Осінь 	<p><i>Пори року:</i></p> <ol style="list-style-type: none"> 1. Зима <ol style="list-style-type: none"> а) Грудень б) Січень в) Лютий 2. Весна <ol style="list-style-type: none"> а) Березень б) Квітень в) Травень 3. Літо 4. Осінь 	<p><i>Пори року:</i></p> <ol style="list-style-type: none"> 1. Зима <ol style="list-style-type: none"> 1.1. Грудень 1.2. Січень 1.3. Лютий 2. Весна <ol style="list-style-type: none"> 2.1. Березень 2.2. Квітень 2.3. Травень 3. Літо 4. Осінь
---	---	---

Існує кілька способів оформлення фрагмента тексту у вигляді списку.

I с п о с і б. Основний. До початку введення списку слід розмістити курсор у потрібному місці документа і виконати **Основне** ⇒ **Абзац** ⇒ **Маркери** або **Нумерація**. У поточному місці документа з'явиться маркер або номер того виду списку, який використовувався останнім. Для вибору іншого виду маркера чи номера слід вибрати кнопку біля потрібного типу списку та у відкритому переліку **Бібліотека маркерів** або **Бібліотека нумерованих списків** (мал. 3.3) вибрати потрібний варіант оформлення. Після цього можна вводити перший елемент списку.



Мал. 3.3. Види маркованих і нумерованих списків

Після введення першого елемента списку потрібно натиснути клавішу **Enter** – у наступному рядку автоматично з'являться номер або маркер. Коли останній елемент списку буде введено, потрібно повторно вибрати кнопку відповідного списку на **Стрічці**, або двічі натиснути **Enter**, або видалити номер чи маркер клавішею **Backspace**.

II с п о с і б. Перетворення абзаців на список. Якщо деякі абзаци введеного раніше тексту потрібно оформити як список, то слід:

1. Виділити потрібні абзаци.
2. Вибрати на **Стрічці** кнопку меню відповідного типу списку: **Маркери** або **Нумерація**.
3. Вибрати у списку потрібний вид маркера чи номера.

Після цього біля першого рядка кожного з виділених абзаців з'явиться вибраний маркер або номер.

Для тих, хто хоче знати більше

III с п о с і б. Автоматичне створення списку. Розмістивши курсор у потрібному місці документа, слід увести деякі символи, які визначають вид бажаного списку, і натиснути клавішу **Пропуск** (табл. 3.3). Одразу ж умовні символи пере-



творюються на відповідний маркер або номер. Далі можна розпочинати введення першого елемента списку.



Таблиця 3.3

Символи для створення списку

Символи	Оформлення списку	Символи	Оформлення списку
Для маркованих списків		Для маркованих списків	
зірочка *	•	літера o	◦
мінус -	–	більше >	➤
Для нумерованих списків		Для нумерованих списків	
символ 1 і крапка.	1.	символ 1 і дужка	1)
символ 1 і символ більше >	1>	літера a і дужка	a)



Для тих, хто працює з LibreOffice

У LibreOffice Writer для оформлення деякого фрагмента тексту у вигляді списку потрібно використати кнопки **Маркований список**  або **Нумерація**  на панелі інструментів **Форматування**. Ширші можливості зі встановлення властивостей списків надають користувачу елементи вікна **Нумерація та маркування**, яке можна відкрити, виконавши **Формат** ⇒ **Маркери та нумерація** або вибравши команду **Ще маркери** у списку однієї із зазначених кнопок.

РЕДАГУВАННЯ СПИСКІВ

У текстовому процесорі Word 2007 створені списки зручно редагувати.

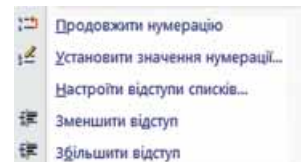
Якщо в будь-якому місці списку потрібно додати ще один рядок, то після попереднього елемента списку слід натиснути клавішу **Enter** – буде вставлено додатковий абзац з номером чи маркером, а нумерація всіх наступних елементів списку автоматично зміниться.

Для того щоб видалити елемент списку, потрібно виділити абзац і натиснути клавішу **Delete** або **Backspace** – елемент списку буде видалено, нумерація автоматично зміниться.

Іноколи потрібно в деякому місці тексту відмінити нумерацію, наприклад, для введення тексту, який не є елементом списку. У такому випадку нумерацію слід видалити повторним вибором кнопки списку на **Стрічці** або натисненням **BackSpace**.

Під час копіювання та переміщення фрагментів нумерованих списків автоматична нумерація може не відповідати потребі користувача. Для змінення нумерації елементів списку потрібно:

1. Вибрати вказівником миші місце на початку рядка перед номером, який потрібно змінити. При цьому нумерація списку виділиться.
2. Відкрити контекстне меню виділеного фрагмента (мал. 3.4).




Мал. 3.4. Контекстне меню номера списку

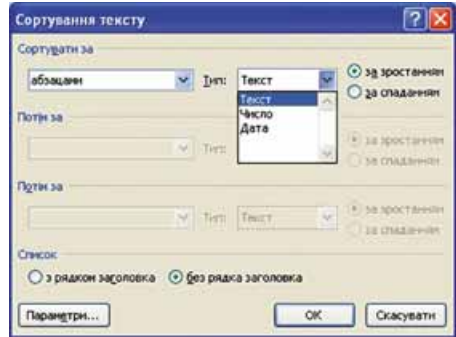
3. Вибрати потрібний варіант змінення номера:

- **Перезапустити з 1** – нумерація елементів списку розпочнеться з першого номера.
- **Продовжити нумерацію** – нумерація наступних елементів списку буде продовжена далі від попереднього номера.
- **Установити значення нумерації** – нумерація елементів списку розпочнеться з номера, який укаже користувач.

Елементи списку можна відсортувати в алфавітному, числовому або хронологічному порядку (за спаданням або за зростанням). При цьому нумерація елементів залишиться послідовною, а абзаци тексту будуть переставлені згідно з вибраним порядком.

Для цього потрібно виконати такий алгоритм:

1. Виділити абзаци, які упорядковуються.
2. Виконати *Основне* ⇒ *Абзаци* ⇒ *Сортування* .



Мал. 3.5. Вікно Сортування тексту

3. Установити в діалоговому вікні *Сортування тексту* (мал. 3.5) такі значення:


- **Сортувати за** – *абзацами*.
- **Тип даних** – *текст, число* або *дата*.
- **Порядок сортування** – *за зростанням* чи *за спаданням*.

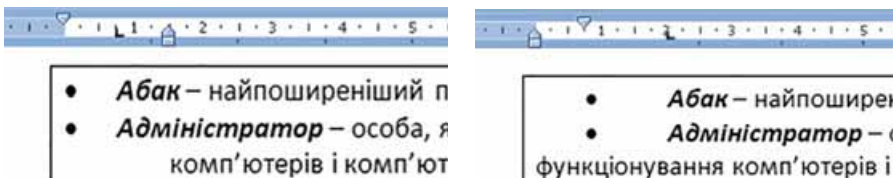
4. Вибрати кнопку **ОК**.

ФОРМАТУВАННЯ СПИСКІВ

За потреби користувач може відформатувати створений список: змінити вид маркера, його формат, спосіб нумерації, спосіб вирівнювання списку, відступи тексту від маркерів і номерів тощо.

Для цього слід виділити потрібні елементи списку і виконати форматування відомими засобами – використати елементи керування *міні-панелі* вкладки *Основне*, діалогових вікон тощо. Так, як і для форматування абзацив, можна використовувати маркери на горизонтальній лінійці для змінення відступів тексту списку від полів (мал. 3.6).

Для встановлення відступу тексту списку від номера чи маркера використовується **табуляція** – засіб, який дає змогу розміщувати об'єкти в рядку в строго визначених місцях (**позиціях табуляції**). На лінійці позиція табуляції, за замовчуванням, відмічається позначкою . Як бачимо на



Мал. 3.6. Маркери і позначка табуляції на лінійці



малюнку 3,6, на першій картинці позначка табуляції стоїть на відмітці 0,75 см, а на другій – на відмітці 2 см. Тому текст елементів списку починається саме з таким відступом.

Для встановлення позначки табуляції слід на лінійці вибрати потрібне місце. Щоб змінити позицію табуляції, потрібно перетягнути відповідну позначку вздовж лінійки у нове місце. Для видалення позиції табуляції достатньо перемістити позначку табуляції за межі лінійки.

Також установити та змінити позиції табуляції можна у вікні **Табуляція**, для чого слід виконати **Основне** ⇒ **Абзац** ⇒ **Табуляція**.

Однією з особливостей форматування списків є те, що формат маркерів (номерів) і формат символів тексту списку може бути різним. Якщо вибрати один з маркерів чи номерів списку, то будуть виділені всі аналогічні об'єкти в усьому списку. Далі можна змінювати їх формат незалежно від формату символів тексту списку (мал. 3.7).



За потреби користувач може створити власний формат оформлення списку, вибравши команду **Визначити новий маркер** або **Визначити новий числовий формат** унизу діалогових вікон відповідних списків (див. мал. 3.3). У вікні, що відкриється, потрібно встановити значення властивостей нового виду списку.

Мал. 3.7. Приклад форматування списку



Для тих, хто хоче знати більше

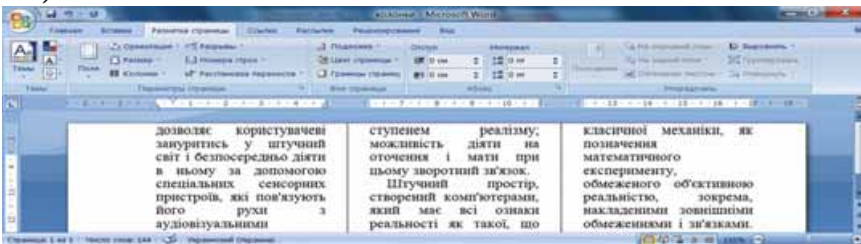
Багаторівневі списки

Для створення багаторівневих списків слід вибрати на **Стрічці** кнопку **Багаторівневий список**. Щоб вибрати вид багаторівневого списку, потрібно відкрити список команд цієї кнопки та вибрати потрібний вигляд списку в розділі **Бібліотека списків**.

Створення багаторівневого списку може починатися з укладення однорівневого списку (маркованого чи нумерованого). Перетворення його на багаторівневий здійснюється зміною рівня вкладеності деяких елементів списку за допомогою кнопок **Зменшити відступ** (перехід на рівень вище) або **Збільшити відступ** (перехід на рівень нижче), які розташовані на **Стрічці** в групі **Абзац**.

СТВОРЕННЯ КОЛОНОК У ТЕКСТОВОМУ ДОКУМЕНТІ

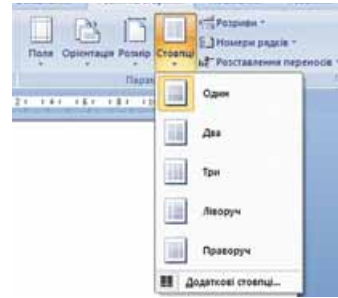
Текстовий процесор **Word** надає можливість розташовувати текст документа в кілька колонок. Як це роблять, наприклад, у газетах та журналах (мал. 3.8).



Мал. 3.8. Приклад тексту, який розміщено у три колонки

Щоб деякий фрагмент тексту розмістити в колонках, його потрібно виділити та виконати послідовність дій **Розмітка сторінок** ⇒ **Параметри сторінки** ⇒ **Колонки**. Потім у списку треба вибрати потрібну кількість колонок (мал. 3.9). Текст автоматично буде розміщений у вибраний спосіб.

Щоб скасувати розташування фрагмента тексту в колонках, цей фрагмент потрібно виділити і встановити для нього режим відображення в одну колонку. При цьому перед першим рядком і після останнього рядка фрагмента тексту залишаються приховані символи **Розрив розділу**, які бажано видалити.



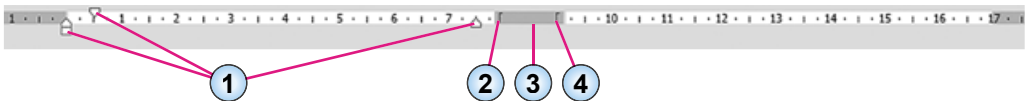
Мал. 3.9. Відкритий список кнопки **Колонки**

Для тих, хто працює з LibreOffice

У LibreOffice Writer для оформлення деякого фрагмента тексту в кілька колонок потрібно виділити цей фрагмент, виконати **Формат** ⇒ **Колонки** та у вікні **Стовпчики** встановити значення властивостей колонок із текстом.

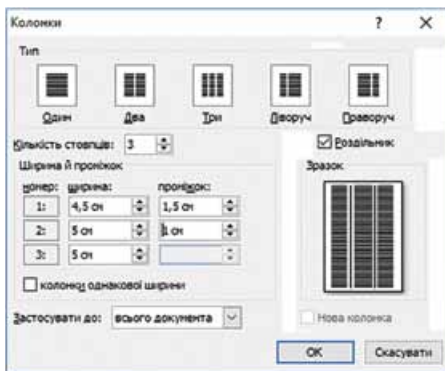
РЕДАГУВАННЯ ТА ФОРМАТУВАННЯ КОЛОНОК

Коли курсор знаходиться в тексті, розміщеному в колонках, на горизонтальній лінійці, крім маркерів абзацу, з'являються *маркери меж колонок* (мал. 3.10), перетягуючи які можна змінювати ширину колонок і відстань між ними.



- 1. Маркери абзаців
- 2. Маркер правої межі першої колонки
- 3. Проміжок між колонками
- 4. Маркер лівої межі другої колонки

Мал. 3.10. Маркери меж колонок на горизонтальній лінійці



Мал. 3.11. Вікно **Колонки**

Форматування колонок можна виконати і в діалоговому вікні **Колонки** (мал. 3.11), яке відкривається **Розмітка сторінок** ⇒ **Параметри сторінки** ⇒ **Колонки** ⇒ **Інші колонки**. Тут можна встановити кількість колонок, ширину кожної з них, відстань між окремими колонками, наявність розділювача (вертикальна лінія між колонками). Після встановлення значень властивостей колонок потрібно вибрати кнопку **ОК**.

**Найважливіше в цьому пункті**

Текстовий процесор **Word 2007** дає змогу вводити в текст символи, які відсутні на клавіатурі. Для цього використовують список таких символів, який відкривається вибором однойменної кнопки на вкладці **Вставлення** в групі **Символи**. Також у текст можна ввести різноманітні маленькі малюночки, вибравши для цього спеціальні шрифти **MS Outlook**, **Webdings**, **Wingdings** тощо.

До **прихованих (недрукованих)** символів належать символи, які вводяться користувачем з клавіатури, але не виводяться на папір під час друкування, і, зазвичай, не відображаються на екрані – пропуск, кінець абзацу, табуляція тощо. Для відображення цих символів у тексті потрібно виконати **Основне** ⇒ **Абзац** ⇒ **Відобразити всі символи** .

Абзаци тексту можна оформити у вигляді маркованих, нумерованих або багаторівневих списків, які використовуються для наочного подання переліку об'єктів. Для цього слід виділити фрагмент тексту з кількох абзаців і в меню кнопок **Маркери** чи **Нумерація** вибрати потрібний вид списку.

Під час редагування списків можна додати нові елементи списку, видалити зайві, впорядкувати за зростанням чи спаданням. Під форматуванням списку розуміють встановлення відступів маркерів від поля, відступів тексту від маркерів, зміну формату номерів та виглядів маркерів, початкового номера списку тощо.

Також можна розмістити фрагмент тексту в кілька колонок. Для цього слід виділити потрібний текст, виконати **Розмітка сторінок** ⇒ **Параметри сторінки** ⇒ **Колонки** ⇒ **Інші колонки**, встановити у вікні **Колонки** кількість колонок, ширину кожної з них, відстань між окремими колонками, наявність розділювача.

**Дайте відповіді на запитання**

- 1°. Як вставити у текст символи, які відсутні на клавіатурі (математичні, латинські, графічні)? Як здійснюється їх редагування і форматування?
- 2°. Які символи називають прихованими (недрукованими)? Наведіть приклади таких символів. Для чого їх використовують?
- 3°. Списки яких типів можна створити у текстовому документі **Word 2007**?
- 4°. Яка назва і призначення кнопок , , групи **Абзац**?
- 5°. У чому полягає автоматична нумерація елементів списку? Як можна змінити номер елемента списку?
- 6°. Як відмінити нумерацію (маркування) після закінчення введення елементів списку?
- 7°. У списку учнів вашого класу було пропущено кілька прізвищ. Як вставити їх у список?
- 8°. Список учнів вашого класу було введено в довільному порядку. Як розмістити прізвища у списку в алфавітному порядку?

- 9°. Як розмістити текст у кілька колонок? Які властивості колонок можна змінити?
- 10°. Як можна встановити ширину колонок, відступи між колонками, кількість колонок?



Виконайте завдання



1°. Відкрийте програму **Word 2007** і введіть по 5 різних графічних символів, використавши графічні шрифти MS Outlook, Webdings, Wingdings. Збережіть документ у файлі з іменем завдання **3.1.1.docx** у вашій папці.



2°. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.1\зразок 3.1.2.docx**). Перший–третій абзаци документа оформіть як маркований список (маркер оберіть самостійно), абзаци четвертий–шостий – як нумерований список (вид нумерації оберіть самостійно). Відмініть нумерацію для заголовку тексту. Збережіть документ у файлі з іменем завдання **3.1.2.docx** у вашій папці.

3°. Створіть новий документ і введіть список прізвищ ваших однокласників у вигляді нумерованого списку. Впорядкуйте їх за алфавітом. Збережіть документ у вашій папці у файлі з іменем завдання **3.1.3-1.docx**. Перетворіть список на маркований. Збережіть документ у вашій папці з іменем завдання **3.1.3-2.docx**.



4°. Створіть новий документ і введіть перелік семи чудес України (відомості знайдіть в Інтернеті) у вигляді маркованого списку. Встановіть такі значення властивостей: відступ маркера від лівого поля – *2 см*, відступ тексту елементів списку від маркера – *1 см*. Збережіть документ у вашій папці у файлі з іменем завдання **3.1.4.docx**.

5°. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.1\зразок 3.1.5.docx**). Оформіть документ за зразком, наведеним у файлі. Збережіть документ у вашій папці у файлі з іменем завдання **3.1.5.docx**.

6*. З'ясуйте, використовуючи **Довідку**, в якому випадку маркери та номери можуть не з'являтися під час створення списку. Як це виправити?



7*. Знайдіть на сайті **Microsoft Office Online** рисункові маркери, за допомогою **Довідки** з'ясуйте, як їх установити на своєму комп'ютері, та виконайте це. Оформіть з їх використанням список областей України. Збережіть документ у вашій папці у файлі з іменем завдання **3.1.7.docx**.

8°. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.1\зразок 3.1.8.docx**). У відкритому документі розмістіть текст другого абзацу у дві колонки однакової ширини без розділювача. Текст четвертого абзацу – в три колонки різної ширини з розділювачем. Збережіть документ у вашій папці у файлі з іменем завдання **3.1.8.docx**.



3.2. ТАБЛИЦІ В ТЕКСТОВИХ ДОКУМЕНТАХ











1. Наведіть приклади використання таблиць у вивченні різних шкільних предметів та в повсякденному житті.
2. Як перемістити курсор у табличному процесорі?
3. Які властивості має графічне зображення в текстовому документі? Які значення вони можуть приймати?

ТАБЛИЦІ ТА ЇХ ВЛАСТИВОСТІ

Для впорядкування і наочного подання в документах різноманітних даних використовують таблиці. Дані, представлені у таблиці, компактні і зручні для сприймання. Наприклад (табл. 3.4):

Таблиця 3.4

Розклад уроків у 8-А класі

№	Понеділок	Вівторок	Середа	Четвер	П'ятниця
1	Інформатика I гр.	Фізична культура		Геометрія	Іноземна мова (англ.)
	Інформатика II гр.				Історія України
2	Фізика	Українська мова	Медіаосвіта	Всесвітня історія	Біологія
3	Фізична культура	Українська література	Алгебра		Історія
10:45-11:15	 Динамічна перерва 				
4	Хімія	Біологія	Хімія	Географія	Українська література
				Основи здоров'я	
5	Алгебра	Іноземна мова (англ.)	Фізика	Зарубіжна література	Українська мова
6	Історія України	Географія	Інформатика I гр.	Трудове на- вчання	Фізична культура
			Інформатика II гр.		
14:00 гуртки					

Таблиця складається зі *стовпців* і *рядків*, на перетині яких знаходяться *клітинки*. Стовпці, рядки, клітинки є об'єктами таблиці. Таблиця у **Word 2007** може містити до 63 стовпців і довільну кількість рядків.

Як видно з наведеного прикладу (табл. 3.4), висота рядків і ширина стовпців таблиці може бути різною. Кілька клітинок можуть бути об'єднані в одну, а деякі з клітинок можуть бути розділені на кілька. Орієнтація тексту в клітинці може бути горизонтальною або вертикальною. Для різних об'єктів таблиці можна встановити межі та заливку різного типу.

Таблиця, як об'єкт текстового документа, має такі властивості:

- **розмір таблиці** – визначає ширину таблиці в *сантиметрах* або у *відсотках* від ширини аркуша;
- **ширина стовпців, висота рядків, розміри клітинки** – визначається в сантиметрах або відсотках від ширини аркуша;
- **спосіб обтікання** таблиці текстом – може набувати таких значень: *без обтікання, з обтіканням навколо таблиці*;
- **межі таблиці** – визначаються *кольором, типом* та *шириною* меж усієї таблиці або окремих її об'єктів;
- **заливка** об'єктів таблиці – визначається *кольором, візерунком* тощо.


СТВОРЕННЯ ТАБЛИЦІ

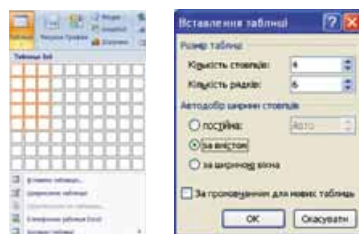
У **Word 2007** існує кілька способів створення таблиці в текстовому документі:

1. Вставити таблицю простої структури.
2. Накреслити таблицю довільної структури.
3. Перетворити фрагмент тексту на таблицю.
4. Вставити таблицю з колекції шаблонів.

Усі команди створення таблиць знаходяться у списку кнопки **Таблиця** групи **Таблиці** вкладки **Вставлення**. Розглянемо два способи, які використовують найчастіше.

І спосіб. Вставити в документ таблицю простої структури можна у такий спосіб:

1. Вибрати в документі місце, де потрібно вставити таблицю.
2. Вибрати на вкладці **Вставлення** у групі **Таблиці** кнопку **Таблиця** .
3. Виділити на схемі потрібну кількість рядків та стовпців і клацнути лівою кнопкою миші. Наприклад, ліворуч на малюнку 3.12 виділено таблицю, що складається із 3 стовпців і 4 рядків.




Мал. 3.12. Вставлення таблиці

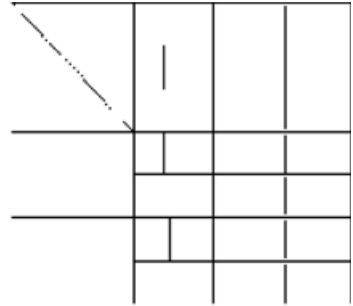
У цей спосіб можна вставити в документ таблицю, у якій не більше ніж 10 стовпців і 8 рядків. Якщо ж потрібно створити більшу таблицю, то її можна вставити в текстовий документ, виконавши **Вставлення** ⇒ **Таблиці** ⇒ **Таблиця** ⇒ **Вставити таблицю**. Потім у відповідних полях діалогового вікна **Вставлення таблиці** (мал. 3.12, праворуч) потрібно вказати кількість стовпців і рядків, спосіб визначення ширини стовпців нової таблиці і вибрати кнопку **ОК**.



П с п о с і б. Якщо таблиця має складну структуру, наприклад, як на малюнку 3.13, то її можна *накреслити* в документі. Для цього потрібно:

1. Виконати **Вставлення** ⇒ **Таблиці** ⇒ **Таблиця** ⇒ **Накреслити таблицю**.
2. Намалювати вказівником, який матиме вигляд олівця, контур усієї таблиці.
3. Намалювати лінії, що розділяють рядки та стовпці.

Якщо під час креслення таблиці були створені зайві або помилкові лінії, їх можна «стерти», використавши інструмент **Гумка**. Для цього слід вибрати на вкладці **Конструктор** у групі **Накреслити межі** кнопку **Гумка** , навести вказівник (його вигляд у цей час буде змінений) на зайву лінію і вибрати її. Після видалення зайвих ліній кнопку **Гумка** слід відключити, вибравши її повторно.



Мал. 3.13. Таблиця складної структури




Для тих, хто хоче знати більше

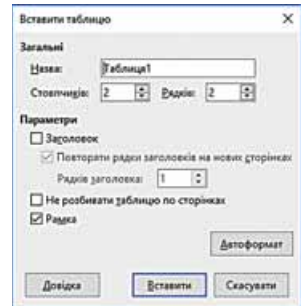
ІІІ с п о с і б. Виділений фрагмент тексту можна перетворити на таблицю, використавши послідовність дій **Вставлення** ⇒ **Таблиці** ⇒ **Таблиця** ⇒ **Перетворити на таблицю**. У діалоговому вікні вказати потрібну кількість стовпців і символи, які у виділеному фрагменті слід вважати роздільниками клітинок (пропуск, табуляція, розрив абзацу, крапки з комою).



Для тих, хто працює з LibreOffice

Для вставлення простої таблиці в **LibreOffice Writer** потрібно вибрати кнопку **Вставити таблицю**  на панелі інструментів **Стандартна** та виділити на схемі потрібну кількість рядків та стовпців і клацнути ліву кнопку миші. У цей спосіб можна вставити в документ таблицю, в якій не більше ніж 10 стовпців і 15 рядків.

Якщо ж потрібно створити більшу таблицю, то її можна вставити в текстовий документ, виконавши **Таблиця** ⇒ **Вставити таблицю**. Потім у відповідних полях діалогового вікна **Вставити таблицю** (мал. 3.14) потрібно вказати кількість стовпців і рядків, інші значення властивостей та вибрати кнопку **Вставити**.



Мал. 3.14. Вікно **Вставити таблицю**

ВВЕДЕННЯ ДАНИХ У ТАБЛИЦЮ І ПЕРЕМІЩЕННЯ ПО ТАБЛИЦІ

Після того як таблицю створено, її потрібно заповнити даними. Текст уводиться в поточну клітинку таблиці за відомими вам правилами введення тексту.

Під час введення даних у клітинки ширина стовпця і висота рядка автоматично змінюються, бо такий режим встановлено в програмі за замовчуванням.

Щоб перемістити курсор у певну клітинку, її потрібно вибрати вказівником або скористатися клавішами керування курсором (табл. 3.5):

Таблиця 3.5






Переміщення курсора в таблиці

Клавіша	Призначення	Клавіша	Призначення
Tab	Наступна клітинка таблиці	Shift + Tab	Попередня клітинка таблиці
→	Наступний символ тексту в клітинці або клітинка	←	Попередній символ тексту в клітинці або клітинка
↓	Наступний рядок тексту в клітинці або таблиці	↑	Попередній рядок тексту в клітинці або таблиці

ВИДІЛЕННЯ ОБ'ЄКТІВ ТАБЛИЦІ

Текст у клітинках таблиці, саму таблицю та її об'єкти можна редагувати і форматувати. Для виконання цих операцій об'єкти таблиці, над якими виконуються дії, потрібно зробити поточними або виділити. Виділення об'єктів можна виконати одним з **двох** способів:

1. Використовуючи мишу:

- Для виділення однієї клітинки таблиці – вибрати внутрішню область клітинки біля її лівої межі, коли вказівник набуває вигляду .
- Для виділення одного рядка – вибрати зовнішню область рядка таблиці біля його лівої межі, коли вказівник набуває вигляду .
- Для виділення одного стовпця – вибрати зовнішню область стовпця таблиці біля його верхньої межі, коли вказівник набуває вигляду .
- Для виділення всієї таблиці – вибрати маркер  над лівим верхнім кутом таблиці, коли вказівник набуває вигляду .
- Для виділення кількох суміжних об'єктів таблиці – виділити область, у яку потрапляють потрібні об'єкти таблиці.
- Для виділення кількох не суміжних об'єктів таблиці – виділити один об'єкт, потім, утримуючи натиснутою клавішу **Ctrl**, виділити решту потрібних об'єктів.

2. Використовуючи елементи керування Стрічки: зробити потрібний об'єкт таблиці поточним, виконати **Табличні знаряддя** ⇒ **Макет** ⇒ **Таблиця** ⇒ **Виділити** та вибрати у списку потрібну команду: **Вибрати клітинку**, **Виділити рядок**, **Виділити стовець** чи **Виділити таблицю**.

Для зняття виділення об'єкта потрібно вибрати місце поза межами виділення.

За вибору будь-якого об'єкта таблиці на **Стрічці** з'являється дві тимчасові вкладки **Конструктор** і **Макет** у тимчасовому розділі **Табличні знаряддя**.



РЕДАГУВАННЯ ТАБЛИЦІ

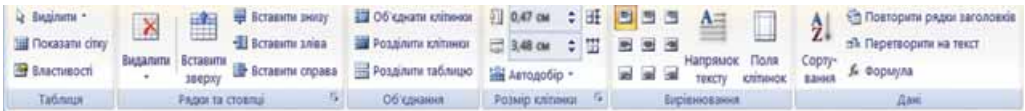
Редагування тексту в таблиці здійснюється звичайними для **Word** способами. Для швидкого очищення вмісту всієї таблиці або її об'єктів достатньо їх виділити і натиснути клавішу **Delete**: дані будуть видалені, а сама таблиця залишиться.

Редагування самої таблиці передбачає додавання або видалення окремих її об'єктів, об'єднання або розділення клітинок таблиці тощо.

Додавати порожні рядки у кінці таблиці можна, використовуючи клавішу **Tab**. Для цього потрібно поставити курсор в останню клітинку таблиці й натиснути цю клавішу. Якщо додатковий рядок потрібен усередині таблиці, то курсор слід поставити в кінці рядка, за межами таблиці й натиснути клавішу **Enter**.

Видалити будь-який виділений рядок чи стовпець таблиці можна, натиснувши клавішу **Backspace**.

Усі елементи керування для виконання операцій редагування таблиці знаходяться на вкладці **Макет** (мал. 3.15). У групі **Рядки та стовпці** роз-



Мал. 3.15. Тимчасова вкладка **Макет**


міщені кнопки для видалення та вставлення відповідних об'єктів. Ці дії застосовуються до виділених або поточних об'єктів таблиці.

Також для виконання операцій редагування можна скористатися командами контекстного меню виділених об'єктів таблиці.


Усю таблицю, як єдиний об'єкт текстового документа, можна переміщувати і копіювати. Для цього таблицю потрібно виділити і застосувати відомі вам способи: перетягуванням, сполученнями клавіш або елементами керування на **Стрічці**.



Для тих, хто хоче знати більше


Якщо кілька клітинок таблиці, які розташовані поруч, потрібно об'єднати в одну, то їх слід виділити та вибрати кнопку **Об'єднати клітинки**  в групі **Об'єднання**. Під час об'єднання вміст клітинок не зникає, а «склеюється» і розміщується в об'єднаній клітинці.


Для розділення однієї чи кількох суміжних клітинок на кілька клітинок потрібно:


1. Виділити потрібні клітинки таблиці.
2. Вибрати кнопку **Розділення клітинки** .
3. Указати в діалоговому вікні, на скільки рядків і стовпців слід розділити клітинки.
4. Указати спосіб розміщення тексту після розділення, використавши позначку відповідного прапорця.

Розділити клітинки чи об'єднати клітинки можна також, використавши інструменти **Накреслити таблицю** та **Гумка** групи **Накреслити межі** вкладки **Конструктор**.

Коли таблицю в документі потрібно розділити на дві для введення між ними тексту або перенесення частини таблиці на іншу сторінку, використовують кнопку

Розділити таблицю  групи **Об'єднання**. Для цього курсор повинен знаходитися в тому рядку таблиці, де планується розрив.

Якщо таблиця велика і займає більше ніж одну сторінку текстового документа, то заголовки стовпців на інших сторінках можна вставити автоматично. Для цього слід виділити рядки заголовка на початку таблиці та виконати **Макет** ⇒ **Дані** ⇒ **Повторити рядки заголовків** .

Для видалення таблиці з документа і представлення її вмісту у вигляді звичайного текста можна виконати автоматичне перетворення таблиці на текст. Для цього потрібно виділити таблицю, виконати **Макет** ⇒ **Дані** ⇒ **Перетворити на текст**  і в діалоговому вікні **Перетворення на текст** указати символ, який слід вставляти в місцях попереднього поділу тексту на окремі клітинки.






ФОРМАТУВАННЯ ТАБЛИЦІ

Під форматуванням таблиці розуміють такі операції: встановлення ширини стовпців та висоти рядків, способу вирівнювання таблиці на сторінці документа і тексту в клітинках, напрямку тексту в клітинці, кольору і товщини меж клітинок, заливки у клітинках тощо.



Для форматування таблиці призначені елементи керування тимчасових вкладок **Макет** (табл. 3.6) та **Конструктор** розділу **Табличні знаряддя** на **Стрічці**. Для форматування тексту в клітинках також можна застосовувати стандартні засоби **Word**.

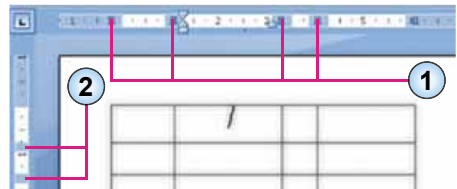
Таблиця 3.6

Призначення деяких елементів керування вкладки **Макет**

Елемент керування	Призначення
Список Вирівнювання 	Для встановлення способу вирівнювання тексту в клітинці – <i>знизу зліва, по центру, по центру справа</i> тощо
Кнопка Напрямок тексту 	Для встановлення способу розташування тексту в клітинці – <i>горизонтальне або вертикальне</i>
Кнопка Поля клітинок 	Для відкриття діалогового вікна, у якому можна встановити <i>розмір полів</i> у клітинках та <i>інтервал між клітинками</i> в таблиці
Лічильники Ширина стовпців  Висота рядків 	Для встановлення точних розмірів окремих об'єктів таблиці

Розміри окремих об'єктів таблиці можна змінити й іншими способами:

- **Перетягування межі об'єкта.** Наведення вказівника на межу рядка чи стовпця змінює його вигляд: на межі рядка вказівник матиме вигляд , на межі стовпця – . Під час перетягування штрихова лінія буде демонструвати нове положення межі.



1. Маркери меж стовпців
2. Маркери меж рядків

Мал. 3.16. Межі рядків і стовпців таблиці на лінійках






- **Перетягування маркера межі.** Коли курсор знаходиться в області таблиці, на горизонтальній лінійці з'являються *маркери меж стовпців*, а на вертикальній – *маркери меж рядків* (мал. 3.16). Коли їх перетягувати, змінюються розміри відповідних стовпців і рядків.

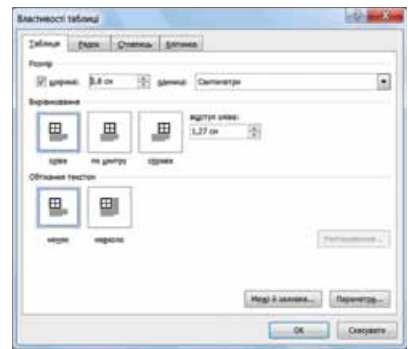
Для меж таблиці або її окремих об'єктів можна встановити колір, товщину, стиль ліній. Внутрішні частини об'єктів таблиці можна залити різними кольорами, використавши різні способи заливки тощо. За замовчуванням у **Word 2007** таблиці призначається обрамлення типу «сітка» – усі клітинки мають межі у вигляді тонких суцільних ліній шириною 0,5 пт без заливки. Змінити значення цих властивостей можна, використовуючи елементи керування тимчасової вкладки **Конструктор** (мал. 3.17).



Мал. 3.17. Тимчасова вкладка **Конструктор**

Для вибору ширини, типу, кольору ліній меж призначені відповідні елементи керування групи **Накреслити межі**. У групі **Стилі таблиць** можна встановити колір заливки об'єктів таблиці (кнопка **Затінення** ) , відобразити або зняти відображення тих чи інших меж (кнопка **Межі** ). Налаштування значень цих властивостей можна виконати і в діалоговому вікні групи **Накреслити межі**.

Встановлення значень, описаних вище, та інших властивостей таблиць, таких як спосіб розташування таблиці на аркуші, спосіб обтікання таблиці текстом тощо, можна виконати на вкладках діалогового вікна **Властивості таблиці** (мал. 3.18). Відкрити це вікно можна відповідною командою контекстного меню таблиці або кнопкою **Властивості**  групи **Таблиця** вкладки **Макет**.



Мал. 3.18. Діалогове вікно **Властивості таблиці**



Для тих, хто хоче знати більше

Також для форматування таблиці можна застосувати один із вбудованих стилів, які встановлюють певні значення параметрів форматування об'єктів таблиці. Стиль можна обрати у списку групи **Стилі таблиць** на вкладці **Конструктор**. Вибір стилю супроводжується динамічним попереднім переглядом майбутнього результату форматування.

Використовуючи прапорці групи **Параметри стилів таблиць** вкладки **Конструктор**, у застосованому стилі можна змінити формати окремих частин таблиці (рядка заголовка та підсумку, першого та останнього стовпців, смуг рядків та стовпців).

Для тих, хто працює з LibreOffice

Редагування і форматування об'єктів таблиці здійснюється з використанням меню **Таблиця** (мал. 3.19) або елементів керування панелі інструментів **Таблиця**.



Мал. 3.19. Панель інструментів **Таблиця**

Найважливіше в цьому пункті

Для впорядкування і наочного подання в документах різноманітних даних використовують таблиці. Таблиця складається зі стовпців і рядків, на перетині яких знаходяться клітинки. Стовпці, рядки, клітинки є об'єктами таблиці. У клітинках таблиці можуть розміщуватися текст, числа, малюнки, формули і навіть інші таблиці.

Таблиця як об'єкт текстового документа має такі властивості: розмір таблиці, ширина стовпців, висота рядків, спосіб вирівнювання таблиці, спосіб обтікання таблиці текстом, межі таблиці, заливка об'єктів таблиці тощо.

Є кілька способів уставити в текстовий документ таблицю. Найбільш загальний з них такий: виконати **Вставлення** ⇒ **Таблиці** ⇒ **Таблиця** ⇒ **Вставити таблицю**, потім у відповідних полях діалогового вікна **Вставлення таблиці** вказати кількість стовпців і рядків, спосіб визначення ширини стовпців нової таблиці і вибрати кнопку **ОК**.

Редагування самої таблиці передбачає додавання або видалення окремих її об'єктів, об'єднання або розділення клітинок таблиці тощо. Під форматуванням таблиці розуміють такі операції: встановлення ширини стовпців та висоти рядків, способу вирівнювання таблиці на аркуші та тексту в клітинках, напрямку тексту в клітинці, кольору і товщини меж клітинок, заливки у клітинках тощо.


Усі елементи керування для виконання операцій редагування та форматування таблиці знаходяться на вкладці **Макет** та **Конструктор** розділу **Табличні знаряддя** на **Стрічці**.



Дайте відповіді на запитання

- 1°. Для чого в документі використовують таблиці? Із чого вони складаються? Які об'єкти містять?
- 2°. Які властивості таблиці та їх можливі значення?
- 3°. Якими способами можна вставити таблицю в текстовий документ?
- 4°. Що розуміють під терміном «таблиця складної структури»? Яким способом її зручніше створити?
- 5°. Як виділити різні об'єкти таблиці?
- 6°. Як можна видалити або вставити рядки чи стовпці в таблиці?
- 7*. У чому полягатиме відмінність натискання клавіш **Delete** та **Backspace** при виділеному фрагменті таблиці?
- 8°. У кінці таблиці потрібно додати ще один рядок. Як це зробити швидко? Як можна діяти, щоб вставити рядок у середині таблиці?



- 9°. Як змінити розміри стовпців чи рядків?
- 10*. Які напрямки тексту можна встановити у клітинці? Як це зробити?
- 11°. Як буде розміщено текст у клітинці, якщо вибрати кнопку ?
- 12°. У яких випадках потрібно об'єднувати клітинки? Коли користувачу потрібно розділити клітинки, як це можна зробити?
- 13°. Де знаходяться елементи керування для форматування таблиць?

**Виконайте завдання**

- 1°. Створіть таблицю за зразком. Збережіть документ у файлі з іменем завдання 3.2.1.docx у вашій папці.

НАРАХУВАННЯ ЗАРОБІТНОЇ ПЛАТИ

Прізвище	1 квартал	2 квартал	3 квартал	4 квартал
Іванов	2200.35	2200.35	2200.35	2200.35
Петров	3378.89	3378.89	3378.89	3378.89
Сидоров	5456.90	5456.90	5456.90	5456.90
Козлов	1456.67	1456.67	1456.67	1456.67
Андрєєв	4589.00	4589.00	4589.00	4589.00

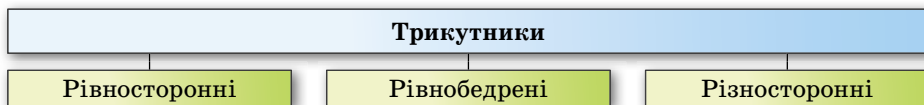


- 2°. Створіть у текстовому документі розклад уроків у вашому класі за зразком таблиці 3.4. Збережіть документ у вашій папці у файлі з іменем завдання 3.2.2.docx та відправте файл на електронну скриньку вчителя.
- 3°. Відкрийте вказаний вчителем файл (наприклад, Розділ 3\Пункт 3.2\зразок 3.2.3.docx). Видаліть останній стовпець і останній рядок таблиці. Вставте перед першим стовпцем додатковий стовпець і введіть у його клітинки номери рядків. Уставте перед першим рядком ще один рядок. Об'єднайте всі клітинки нового рядка і введіть у клітинку назву таблиці. Збережіть документ у вашій папці у файлі з іменем завдання 3.2.3.docx.
- 4°. Відкрийте вказаний учителем файл (наприклад, Розділ 3\Пункт 3.2\зразок 3.2.4.docx). Відформатуйте таблицю за вказаними значеннями властивостей. Збережіть документ у вашій папці у файлі з іменем завдання 3.2.4.docx.

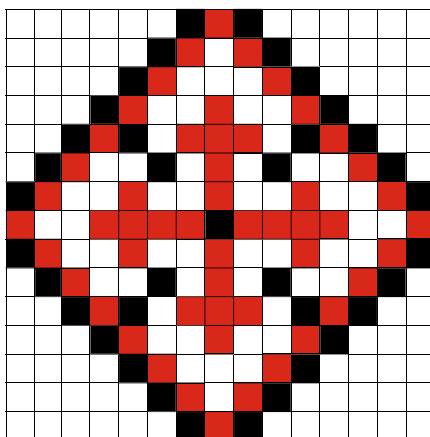
Об'єкт таблиці	Шрифт	Розмір шрифту	Колір символів	Вид накреслення	Вирівнювання
Перший рядок	Comic Sans MS	14	Темно-зелений	Напів-жирний	По центру
Четвертий рядок	Arial	12	Темно-синій	Курсив	За лівим краєм
Перший стовпець	Times New Roman	14	Чорний	Звичайний	По центру



- 5*. Створіть у текстовому документі, використовуючи таблицю, схему класифікації трикутників за рівністю сторін. Збережіть документ у вашій папці у файлі з іменем **завдання 3.2.5.docx**.



- 6*. Створіть у текстовому документі таблицю, зафарбувавши її клітинки за наведеним зразком. Збережіть документ у вашій папці у файлі з іменем **завдання 3.2.6.docx**.



- 7*. Створіть таблицю-календар на поточний місяць. Збережіть документ у вашій папці у файлі з іменем **завдання 3.2.7.docx**.
- 8*. Ви створили новий документ, у якому із самого початку аркуша вставили таблицю і забули написати назву таблиці. З'ясуйте за допомогою **Довідки**, як вставити текст перед таблицею.



3.3. СТВОРЕННЯ, РЕДАГУВАННЯ ТА ФОРМАТУВАННЯ ГРАФІЧНИХ ОБ'ЄКТІВ У ТЕКСТОВОМУ ДОКУМЕНТІ



1. Назвіть властивості графічного зображення в текстовому документі **Word 2007**. Які значення вони можуть мати?
2. Назвіть, які ви знаєте операції редагування графічних зображень у **Word 2007**. Як вони виконуються?
3. Назвіть, які ви знаєте операції форматування графічних зображень у **Word 2007**. Як вони виконуються?

ГРАФІЧНІ ОБ'ЄКТИ В ТЕКСТОВОМУ ДОКУМЕНТІ ТА ЇХ ВЛАСТИВОСТІ

У 6 класі ви навчилися вставляти в текстовий документ графічні зображення, а також розглянули деякі їх властивості. Це були рисунки, схеми, діаграми тощо. Їх використовують для ілюстрації змісту текстового документа. Ви вже знаєте, що графічні об'єкти, вставлені в текстовий



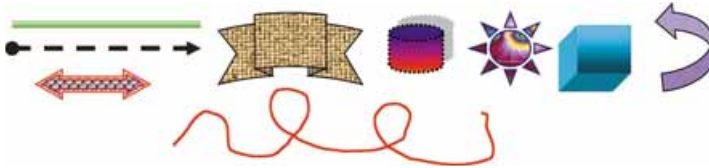
документ, мають певні властивості: *розмір зображення, розташування на сторінці, обтікання текстом, контрастність та яскравість, межі рисунка, заливка* тощо.

У текстовий документ можна вставити різні графічні зображення. Для цього використовують елементи керування групи **Зображення** вкладки **Вставлення**:

- **Рисунок** – для вставляння графічних зображень, які збережені на зовнішніх носіях.
- **Графіка** – для вставляння графічних зображень з готових колекцій картинок, установлених на комп'ютері, чи із сайту **Office Online** корпорації **Microsoft**.
- **Фігури** – для вставляння різноманітних ліній та фігур.
- **SmartArt** – для вставляння різноманітних схем (організаційних, ієрархічних, циклічних тощо).
- **Діаграма** – для вставляння числових діаграм (гістограм, секторних діаграм, графіків тощо).

З перерахованих видів ви вже вмієте вставляти в текстовий документ графічні зображення з колекції **Microsoft ClipArt**, з колекції **SmartArt**, з інших колекцій графічних зображень на зовнішніх носіях, з Інтернету, з власних цифрових фотоальбомів та ін.

У цьому параграфі розглянемо, як створити в текстовому документі графічне зображення, використовуючи різноманітні графічні примітиви (мал. 3.20).



Мал. 3.20. Приклади графічних примітивів з різними значеннями їх властивостей

Графічні примітиви є двох видів – **лінія** та **фігура**. Об'єкт **лінія** характеризується такими властивостями: *форма, товщина, тип, шаблон, колір*. Можливі значення цих властивостей наведено в таблиці 3.6.

Таблиця 3.6

Приклади значень властивостей лінії

Форма лінії	Товщина лінії	Тип лінії	Шаблон лінії	Кольори лінії
Пряма	0,25 пт 0,5 пт 0,75 пт 1 пт 1,5 пт 2,25 пт 3 пт 4,5 пт 6 пт	— — — — — — — — —	— — — — — — — — —	
Крива				
Ламана				

Фігури утворюються замкненими лініями. Ці замкнені лінії – контур фігури, а частина площини, яку вони обмежують, – **внутрішня область фігури** (мал. 3.21). Внутрішня область фігури може бути зафарбована. При цьому можна використати один з 5 способів (ефектів) заливки: *однорідну, градієнтну, заливку візерунком, текстурою, рисунком* (табл. 3.7).



Мал. 3.21. Контур і внутрішня область фігури

Таблиця 3.7

Приклади зафарбування внутрішньої області фігури

Однорідна заливка	Градієнтна заливка	Заливка візерунком	Заливка текстурою	Заливка рисунком
Одним кольором з палітри кольорів	Відтінками кількох кольорів при плавних переходах між ними	Мозаїчним візерунком, який обирається з наданих шаблонів	Деяким зображенням з наданих зразків	Будь-яким зображенням, яке обирається з наявних графічних файлів

СТВОРЕННЯ ГРАФІЧНИХ ПРИМІТИВІВ І ЗОБРАЖЕНЬ У WORD 2007



Мал. 3.22. Інструменти для створення графічних об'єктів

Графічні примітиви, які можна використувати у **Word 2007**, поділяють на кілька груп: лінії, основні фігури, фігурні стрілки, елементи блок-схем, виноски, зірки та стрічки (мал. 3.22).

У **Word 2007** використовується кілька видів ліній – пряма лінія, сполучна лінія, крива, полілінія і мальована крива. Для їх створення використовують такі інструменти:

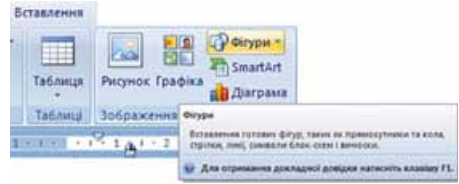
- **Пряма** , **Стрілка** , **Двостороння стрілка** – для малювання прямих ліній. Створення цих об'єктів відбувається аналогічно до малювання прямої лінії у графічному редакторі **Paint**.
- **Мальована крива** – для створення довільної кривої, що імітує малювання ліній олівцем від руки. Створення такого примітиву аналогічно до малювання олівцем у графічному редакторі **Paint**.



- **Крива** \frown – для створення кривої за її місцями згину.
- **Полілінія** --- – для створення ламаних ліній.

Алгоритм створення ламаної (аналогічно створюється і крива):

1. Вибрати на вкладці **Вставлення** у групі **Зображення** кнопку **Фігури** --- (мал. 3.23).



Мал. 3.23. Вставлення фігури

2. Вибрати інструмент **Полілінія**.
3. Вибрати на аркуші початкову точку лінії.
4. Вибрати на аркуші точки, які є вершинами ламаної.
5. Двічі клацнути в кінцевій точці лінії.

Для створення замкненої ламаної або кривої потрібно її кінцеву точку вибрати біля початкової точки цієї лінії. Програма автоматично сполучає ці точки, замкнувши лінію і зафарбувавши внутрішню область.

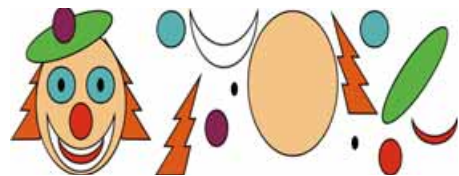
Інші інструменти з груп **Основні фігури**, **Фігурні стрілки**, **Блок-схеми** використовуються для створення примітивів, які мають задану геометричну форму. Наприклад, ромб, трикутник, подвійна стрілка, усміхнене обличчя, дуга, Місяць тощо. Малювання цих фігур виконується за таким алгоритмом:

1. Вибрати на вкладці **Вставлення** в групі **Зображення** кнопку **Фігури**.
2. Вибрати у наведеному переліку потрібну фігуру.
3. Перевести вказівник у робочу область документа.
4. Зафіксувати початкову точку малювання фігури натисканням лівої кнопки миші.
5. Перемістити вказівник у кінцеву точку малювання фігури, утримуючи натиснутою ліву кнопку миші.
6. Відпустити ліву кнопку миші.

Примітки:


- якщо, малюючи овал або прямокутник, утримувати натиснутою клавішу **Shift**, то будуть створюватися фігури правильної форми – круг або квадрат;
- утримання клавіші **Shift** під час побудови ліній та стрілок обмежує кути нахилу величинами, кратними 15° ;
- малювання фігур при натиснутій клавіші **Ctrl** здійснюється від центра фігури, що малюється;
- якщо потрібно послідовно намалювати кілька об'єктів одного виду, то відповідний інструмент слід вибрати правою кнопкою і зафіксувати режим рисування. Додавши всі потрібні об'єкти, натиснути клавішу **Esc**;
- щоб полегшити малювання дрібних деталей зображення, доцільно збільшити масштаб відображення документа.

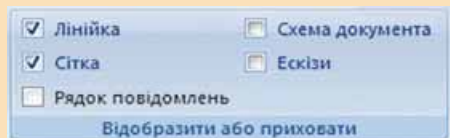
Кожний графічний примітив у текстовому документі створюється окремо, а потім переміщенням, копіюванням, накладанням один на одного створюється графічне зображення (мал. 3.24).



Мал. 3.24. Графічне зображення, створене з графічних примітивів

Для тих, хто хоче знати більше







Під час створення графічного зображення у **Word 2007** зручно користуватися сіткою, яка дає змогу регулювати розміри графічних об'єктів, спрощує малювання прямих ліній та вирівнювання об'єктів. Для відображення сітки потрібно на вкладці **Вигляд** у групі **Відобразити або приховати** встановити позначку прапорця **Сітка** (мал. 3.25). Крок сітки можна змінити у діалоговому вікні **Прив'язування до сітки**, яке відкривається, **Розмітка сторінки** ⇒ **Упорядкування** ⇒ **Вирівнювання**  ⇒ **Настроювання сітки**.



Мал. 3.25. Встановлення відображення сітки

Лінії сітки відображаються на екрані тільки в режимі перегляду **Розмітка сторінки** і не друкуються під час виведення документа на друк. Для того щоб приховати сітку, потрібно зняти позначку відповідного прапорця.


Для тих, хто працює з LibreOffice

Для створення графічних об'єктів у текстовому документі використовують команди меню **Вставка: Зображення, Медіа, Фігури** тощо, або елементи керування панелі інструментів **Графіка** (мал. 3.26). Кнопки зі списками **Основні фігури** , **Фігури-символи** , **Стрілки** , **Зірки** , **Виноски** , **Схеми**  відкривають списки ескізів відповідної групи фігур.

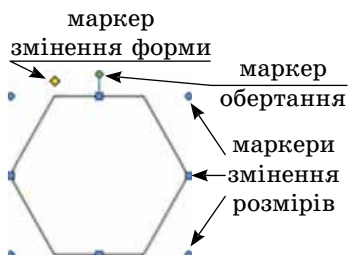


Мал. 3.26. Панель інструментів **Графіка**

РЕДАГУВАННЯ ГРАФІЧНИХ ОБ'ЄКТІВ У WORD 2007

Графічні примітиви є самостійними об'єктами і їх можна редагувати незалежно від інших об'єктів цього зображення. Нагадуємо – усі ці операції здійснюються для виділених об'єктів. Щоб виділити об'єкт, потрібно вибрати його на зображенні. Під час наведення вказівника на об'єкт, він змінює вигляд на такий .

Після вибору об'єкта навколо нього з'являються *маркери змінення розмірів*, а для деяких графічних об'єктів також і *маркер обертання* – це спеціальні позначки у вигляді маленьких кругів і квадратів, які розташовані навколо фігури (мал. 3.27).




Мал. 3.27. Маркери графічного об'єкта

Під час наведення вказівника на маркери змінення розмірів він набуває вигляду двосторонньої стрілки, яка вказує напрямки можливого переміщення. Переміщення маркера в цих напрямках приводить до відповідного змінення розмірів фігури.

Використовуючи маркер обертання, об'єкт можна повернути на довільний кут. Для цього слід навести вказівник на маркер обертання



(вказівник змінить свій вигляд на такий ) , потім переміщувати мишу по колу в потрібний бік, утримуючи натиснутою ліву кнопку.

Для деяких фігур існує ще один маркер – *маркер змінення форми*, який має вигляд маленького жовтого ромба. Використовуючи його, можна змінювати зовнішній вигляд фігур, але їх ширина та висота залишаються незмінними. Наприклад, можна змінити вигляд вістря стрілки, як це показано на малюнку 3.28. Усі ці фігури отримані модифікацією форми першої фігури.




Мал. 3.28. Модифікація фігури маркером змінення форми



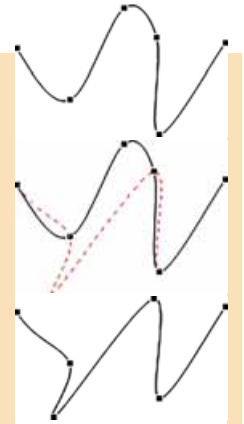
Для тих, хто хоче знати більше

Змінення форми ліній, зокрема кривих, здійснюється редагуванням їх вузлів (мал. 3.29). Для цього слід:

1. Вибрати кнопку **Редагувати фігуру** , яка розміщується на вкладці **Формат** у групі **Вставити фігури**.
2. Вибрати команду **Почати зміну вузлів**.
3. Перетягнути вузли лінії (маленькі чорні квадрати) у потрібне місце.

Операції копіювання, переміщення, видалення графічних об'єктів здійснюються відомими вам способами: з використанням **Буфера обміну**, перетягуванням або клавішами клавіатури.

Якщо графічний об'єкт потрібно перемістити на незначну відстань, то для виділеного об'єкта це можна зробити натисканням клавіш керування курсором у відповідному напрямку. Крок такого переміщення буде дорівнювати встановленому за замовчуванням значенню. Для його зменшення під час переміщення слід додатково утримувати натиснутою клавішу **Ctrl**.



Мал. 3.29. Модифікація кривої з використанням вузлів

ФОРМАТУВАННЯ ГРАФІЧНИХ ОБ'ЄКТІВ У WORD 2007

У створеного графічного об'єкта можна змінювати значення його властивостей, тобто формувати. До операцій форматування графічних примітивів належать: змінення форми і розмірів об'єкта, кольору, типу та шаблону ліній його контуру, кольору й ефектів заливки тощо.

Для форматування графічних примітивів зручно користуватися елементами керування тимчасової вкладки **Формат** (мал. 3.30), яка з'являється на **Стрічці** в розділі **Засоби малювання** під час виділення графічного об'єкта.











Мал. 3.30. Вкладка **Формат**

Призначення цих елементів керування наведено в таблиці 3.8. Під час виконання деяких операцій форматування застосовується динамічний перегляд результатів форматування.

Таблиця 3.8

Призначення елементів керування вкладки **Формат**





Елемент керування	Назва	Призначення
Група Стили фігур		
	Список Експрес-стилі	Для вибору стилю оформлення об'єкта з наданого списку зразків
	Кнопка зі списком Заливка фігури	Для заливки внутрішньої області об'єкта встановленим кольором. Вибір кнопки відкриття списку відкриває панель для вибору кольору та ефекту заливки
	Кнопка зі списком Контур фігури	Для зафарбування контуру фігури встановленим кольором. Вибір кнопки відкриття списку відкриває панель для вибору кольору лінії, її товщини, типу тощо
	Кнопка зі списком Змінити фігуру	Для замінення виділеної фігури на іншу із збереженням форматування поточного об'єкта
Група Ефекти		
	Кнопка Тіньові ефекти	Для встановлення значень властивостей тіні об'єкта
	Кнопка Об'ємні ефекти	Для встановлення значень властивостей об'ємного зображення об'єкта
Група Розмір		
	Лічильник Висота фігури	Для встановлення значення висоти об'єкта
	Лічильник Ширина фігури	Для встановлення значення ширини об'єкта

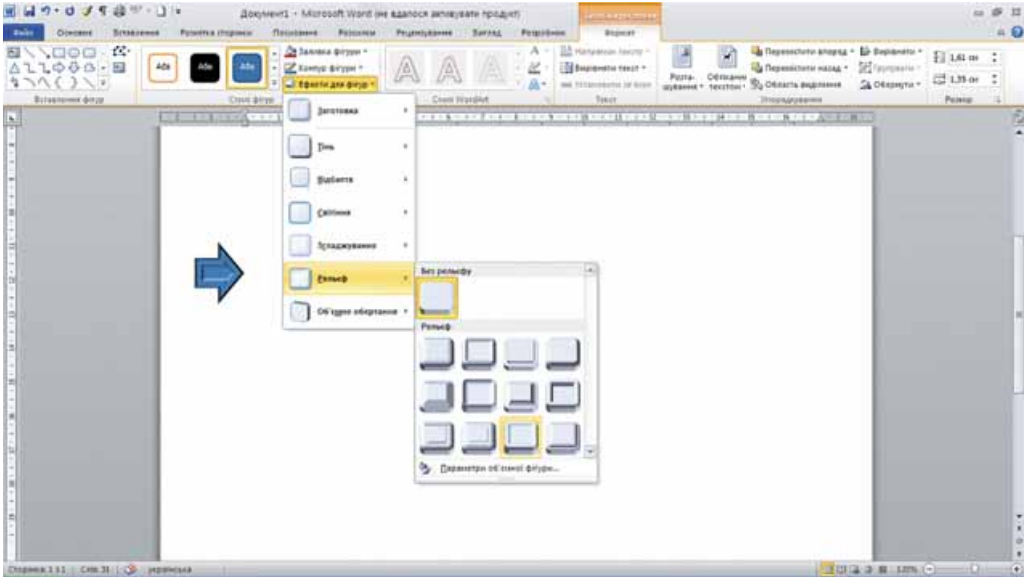
Встановлення значень властивостей графічних об'єктів також можна здійснювати в діалоговому вікні **Формат автофігури**, що відкривається однойменною командою контекстного меню цього об'єкта. Після того потрібно встановити потрібні значення на вкладках **Колір та лінії**, **Розмір** та **Розташування** цього вікна.

До графічних об'єктів, як і для текстових фрагментів, можна застосовувати форматування за зразком, яке дає змогу відформатувати деякий графічний об'єкт так само, як і вже існуючий об'єкт зображення.



Для тих, хто працює з Microsoft Office 2010

Кнопка **Змінити фігуру**  розміщена у списку кнопки **Редагувати фігуру** . Кнопки **Тіньові ефекти**  та **Об'ємні ефекти**  відсутні. Замість них є кнопка зі списком **Ефекти для фігур** (мал. 3.31), яка призначена для вибору готового стилю оформлення або встановлення ефектів тіні, відбиття, світіння, згладжування, рельєфу, об'ємного обертання.



Мал. 3.31. Список кнопки **Ефекти для фігур**

Для тих, хто працює з LibreOffice

Для встановлення значень властивостей графічних об'єктів використовуються елементи керування панелі інструментів **Властивості рисунка** (мал. 3.32).



Мал. 3.32. Панель інструментів **Властивості рисунка**

ОПЕРАЦІЇ НАД ГРУПОЮ ГРАФІЧНИХ ОБ'ЄКТІВ

Якщо деякі операції редагування або форматування потрібно виконати не з одним об'єктом, а з кількома, то для прискорення роботи слід виділити групу цих об'єктів і виконати потрібні операції для всіх об'єктів групи одночасно.

Виділення групи графічних об'єктів у **Word 2007** відбувається при натиснутій клавіші **Ctrl** з використанням миші, аналогічно до виконання подібної операції з об'єктами файлової системи операційної системи **Windows**: папками, файлами, ярликами. Крім того, для виділення групи

графічних об'єктів у деякій прямокутній області малюнка використовуються кнопка **Вибрати об'єкти**, яка розташована на вкладці **Основне** у групі **Редагування** в меню команди **Виділити**. Вибравши цю кнопку, потрібно виділити прямокутну ділянку документа – і всі графічні об'єкти, які повністю потрапили в неї, будуть виділені.

Зауваження: під час вибору цієї кнопки вмикається режим виділення графічних об'єктів, і опрацювання текстових об'єктів документа в цьому режимі неможливе. Відмінити цей режим можна повторним вибором цієї самої кнопки або вибором будь-якого місця в тексті документа.

Для того щоб група виділених об'єктів сприймалась як єдине ціле, ці об'єкти потрібно *згрупувати*. У подальшому всі операції редагування або форматування будуть застосовуватись одразу до всіх об'єктів групи. Таке групування проводиться, коли створення малюнка завершено.

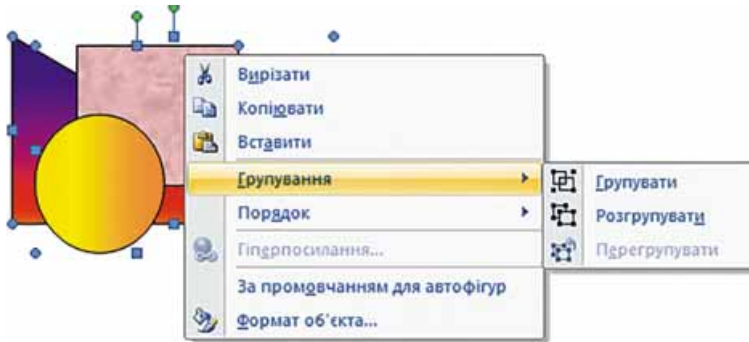
Виконання операції групування здійснюється таким чином:

1. Виділити потрібні графічні об'єкти зображення.
2. Вибрати на вкладці **Формат** у групі **Упорядкування** команду **Згрупувати**.
3. Вибрати команду **Групувати**.

Після групування зберігається можливість виділити будь-який окремий об'єкт і виконати над ним деякі операції.

За потреби об'єкти можна *розгрупувати* або *перегрупувати*, використавши відповідні операції команди **Згрупувати**.

Операції групування і розгрупування об'єктів можна здійснювати і з використанням команди **Групування** контекстного меню об'єктів (мал. 3.33).



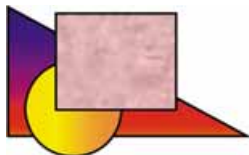
Мал. 3.33. Операції команди **Групування**

Для тих, хто хоче знати більше

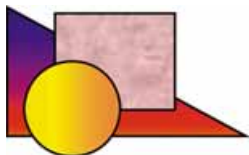
Впорядкування об'єктів графічного зображення

Зображення, створене з окремих графічних об'єктів, має **багатoshарову структуру** – окремі об'єкти можуть бути розташовані поверх інших. Порядок їх накладання один на одного можна змінювати (мал. 3.34), переміщуючи об'єкт відносно інших об'єктів у верхні чи нижні шари.

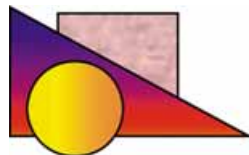
Над об'єктами або групами об'єктів зображення можна виконувати й інші операції – змінювати *порядок їх розташування* на зображенні, *впорядковувати* об'єкти на аркуші, *повертати* та *відображати* тощо.



Початкове зображення групи об'єктів



Переміщення квадрата на шар нижче



Переміщення квадрата на задній план

Мал. 3.34. Приклад переміщення квадрата на нижчі шари зображення

Створені графічні об'єкти або групу об'єктів можна повернути праворуч або ліворуч на 90° та відобразити зверху вниз чи зліва направо (мал. 3.35). Для здійс-



Мал. 3.35. Обертання групи графічних об'єктів

снення цієї операції потрібно скористатися командою **Обернути** у групі **Упорядкувати** вкладки **Формат**, попередньо виділивши потрібні об'єкти.

Виконання всіх цих операцій з групою графічних об'єктів здійснюється з використанням елементів керування групи **Упорядкувати** тимчасової вкладки **Формат** (табл. 3.9) або відповідними командами контекстного меню графічних об'єктів.

Таблиця 3.9

Призначення елементів керування групи Упорядкування вкладки Формат

Елемент керування	Назва	Призначення
	Кнопка зі списком Розташування в тексті	Для вибору розміщення об'єкта на сторінці документа
	Кнопка зі списком На передній план	Для переміщення об'єкта у верхній шар багатшарового зображення
	Кнопка зі списком На задній план	Для переміщення об'єкта у нижній шар багатшарового зображення
	Кнопка зі списком Обтікання текстом	Для вибору способу обтікання об'єкта текстом
	Кнопка зі списком Вирівняти	Для вибору способу взаємного розміщення об'єктів групи на сторінці
	Кнопка зі списком Згрупувати	Для вибору операції над групою об'єктів
	Кнопка зі списком Обернути	Для вибору операцій повертання та відображення об'єкта

Найважливіше в цьому пункті

У текстовий документ можна вставити різні графічні зображення – рисунки, фотографії, схеми тощо. Один з різновидів таких зображень це графічні примітиви – лінії та фігури.

Об'єкт **лінія** характеризується такими властивостями: *форма, товщина, тип, шаблон, колір*. Фігури утворюються замкненими лініями. Ці замкнені лінії – контур фігури, а частина площини, яку вони обмежують, – внутрішня область фігури. Внутрішня область фігури може бути зафарбована.

Графічні примітиви, які можна використовувати у **Word 2007**, поділяються на кілька груп: лінії, основні фігури, фігурні стрілки, елементи блок-схем, виноски, зірки та стрічки. Інструменти для вставлення всіх цих графічних об'єктів знаходяться у списку кнопки **Фігури** на вкладці **Вставлення** в групі **Зображення**.

Кожний графічний примітив у текстовому документі створюється окремо, а потім переміщенням, копіюванням, накладанням один на одного створюється графічне зображення.

Для виконання операцій над об'єктом його потрібно вибрати. Після вибору об'єкта навколо нього з'являються маркери змінення розмірів, маркер обертання, а для деяких фігур – і маркер змінення форми, які дають змогу редагувати створений об'єкт. Операції копіювання, переміщення, видалення графічних об'єктів здійснюються з використанням **Буфера обміну**, перетягуванням або клавішами клавіатури.

Створений графічний примітив можна формувати: змінювати колір, тип та шаблон ліній його контуру, колір та ефекти заливки тощо. Для форматування графічних примітивів використовуються елементи керування тимчасової вкладки **Формат**, яка з'являється на **Стрічці** в розділі **Засоби малювання** під час виділення графічного об'єкта.

Дайте відповіді на запитання

- 1°. З яких графічних примітивів може складатися графічне зображення у **Word 2007**?
- 2°. Які властивості має лінія? Яких значень вони можуть набувати?
- 3°. Які властивості має фігура? Яких значень вони можуть набувати?
- 4°. Лінії яких видів можна побудувати у **Word 2007**? Які інструменти для цього використовують?
- 5°. Для чого використовують клавіші **Ctrl** і **Shift** під час малювання графічних примітивів?
- 6°. Для чого призначені маркери змінення розміру, маркер обертання? Як ними користуватися?
- 7°. Які ефекти заливки можна використовувати в текстовому процесорі **Word 2007**?
- 8°. Як у текстовому документі виділити графічний об'єкт або групу об'єктів?
- 9°. Які операції редагування та форматування можна здійснювати з графічними примітивами в текстовому документі?
- 10°. З якою метою виконують групування об'єктів? Як це зробити?

**Виконайте завдання**

1°. Намалюйте ламану за алгоритмом, наведеним у підручнику. Скопіюйте її кілька разів. Відформатуйте об'єкти по-різному.

2°. Намалюйте зірку. Скопіюйте її 5 разів і розмістіть зірки по колу.

3°. Намалюйте трапецію. Скопіюйте її кілька разів. До кожного об'єкта застосуйте різні види тіней, об'ємних ефектів, ефектів заливки.

4°. Відкрийте вказаний вчителем файл (наприклад, **Розділ 3\Пункт 3.3.\вправа 3.3.4.docx**). Перемістіть об'єкти в зображенні в порядку 3–1–2. Змініть розміри об'єктів зображення: маленькі об'єкти збільшіть, а великі – зменшіть.



5°. Відкрийте вказаний вчителем файл (наприклад, **Розділ 3\Пункт 3.3.\вправа 3.3.5.docx**). Видаліть із зображення всі стрілки. Змініть форму кожного об'єкта за допомогою маркера зміни форми.

6°. Відкрийте вказаний вчителем файл (наприклад, **Розділ 3\Пункт 3.3.\вправа 3.3.6.docx**). Скопіюйте зображений об'єкт тричі. Поверніть чотири зображення в різні боки.

7°. Намалюйте коло і зірку, які розміщені поруч. Згрупуйте їх в єдиний об'єкт.



8*. Знайдіть на вашому комп'ютері графічний файл формату WMF. Уставте його в текстовий документ. Розгрупуйте його елементи і змініть окремі частини малюнка (розміри, кольори, заливки тощо).

**3.4. СТВОРЕННЯ, РЕДАГУВАННЯ ТА ФОРМАТУВАННЯ СПЕЦІАЛЬНИХ ГРАФІЧНИХ ОБ'ЄКТІВ**

1. Як можна ввести в текст математичні символи, наприклад x_3 , x^3 , $\frac{3}{8}$, α ?

2. Які графічні зображення можна вставляти в текстовий документ?

3. Які операції редагування та форматування можна здійснювати над графічними об'єктами в текстовому процесорі **Word 2007**? З використанням яких засобів здійснюються ці операції?

СТВОРЕННЯ ФОРМУЛ У ТЕКСТОВОМУ ДОКУМЕНТІ

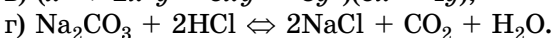
Під час підготовки в текстовому процесорі реферату для уроку чи наукової статті для учнівської конференції в документ інколи потрібно вставляти формули, рівняння, доведення теорем, що використовують особливу символіку: математичну, фізичну, хімічну та ін.

Нескладні формули та вирази мають лінійний вигляд, тобто всі елементи записані в одному рядку, у них немає звичайних дробів, знаків коренів та інших нелінійних структур. Наприклад:

а) $c_2 m_2 (t_2 - \theta) = c_1 m_1 (\theta - t_1)$;

б) $ax^2 + bx + c \geq 0$;

в) $(x^3 + 2x^2y - 5xy^2 - 3y^3)(5x - 4y)$;



Для створення таких виразів використовують різні спеціальні символи, вставлення яких здійснюється у вікні **Символ (Вставка) ⇒ Символи**

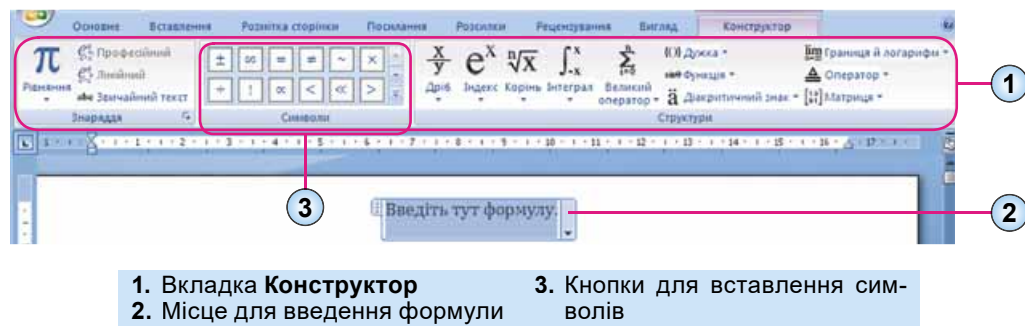
⇒ **Символ** Ω ⇒ **Інші символи**). Уведення символів нижнього і верхнього індексу здійснюється з використанням кнопок **Підрядковий знак** x_2 та **Нарядковий знак** x^2 , які розташовані на вкладці **Основне** в групі **Шрифт**.

Часто трапляються нелінійні, багаторівневі формули та вирази. Наприклад:

$$\begin{aligned}
 & \text{а) } \frac{h_1}{h_2} = \frac{\rho_2}{\rho_1}; \quad \text{б) } E_k = \frac{mv^2}{2}; \quad \text{в) } \eta = \frac{A_1}{A_2} \cdot 100\%; \quad \text{г) } \frac{3x+7}{4} - \frac{x-3}{2} = \frac{5x+2}{8}; \\
 & \text{д) } y = \begin{cases} -3x^3 - 2, & \text{якщо } x < 0, \\ \frac{7x}{4} - 2, & \text{якщо } x \geq 0; \end{cases} \quad \text{е) } \frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a-4b)^2}{9a^4 - b^4}.
 \end{aligned}$$

Вставити в текстовий документ такі формули можна, скориставшись спеціальним засобом **Конструктор формул**. Формули в ньому конструюються з окремих символів та структур, використовуючи відповідні шаблони.

Для відкриття **Конструктора формул** потрібно виконати **Вставлення** ⇒ **Символи** ⇒ **Рівняння** Π . Після цього на **Стрічці** в розділі **Робота з формулами** з'явиться тимчасова вкладка **Конструктор** (мал. 3.36, 1), а в документі – спеціальна область для введення формули (мал. 3.36, 2).



1. Вкладка **Конструктор**
2. Місце для введення формули
3. Кнопки для вставлення символів

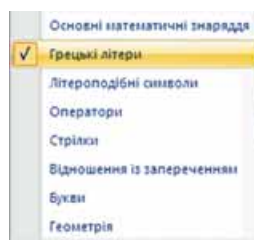
Мал. 3.36. Вкладка **Конструктор** формул

Для запису формули деякі букви, знаки арифметичних операцій та інші символи можна вводити з клавіатури, а спеціальні – вибирати у відповідних списках групи **Символи**.

На **Стрічці** відображаються тільки символи того набору, який використовувався останнім (мал. 3.36, 3). Для перегляду інших частин списку символів поточного набору слід скористатися кнопками \uparrow та \downarrow на смузі прокручування або кнопкою **Додатково** \downarrow для відкриття усього списку символів цього набору. Щоб відкрити перелік усіх наборів символів, потрібно в поточному списку вибрати його заголовок.

Усі символи, які можна вставити у формулу, розділено на 8 основних наборів (мал. 3.37), які у свою чергу можуть бути розділені по окремих блоках (мал. 3.38).

Переглянувши списки та знайшовши потрібний символ, його слід вибрати та вставити у поточне місце виразу.



Мал. 3.37. Список основних наборів символів



Крім окремих символів, у виразі можуть бути різноманітні структури – звичайні дроби, функції, дужки, знаки коренів тощо. Вони вставляються за допомогою елементів керування групи **Структури** вкладки **Конструктор**.

Усі структури розділені за типами, список шаблонів яких відкривається під час вибору відповідних кнопок на вкладці (мал. 3.39). Для того щоб вставити шаблон у формулу, слід відкрити список шаблонів відповідної кнопки і вибрати потрібний.

Далі слід заповнити шаблон даними. Місця для введення чисел та символів у шаблоні позначено спеціальними полями, які обведені пунктирною рамкою (мал. 3.40).

У поля можна вводити не тільки окремі символи, а й вставляти інші шаблони.

Переміщення в межах формули здійснюється клавішами керування курсором або вказівником миші. Для виходу з режиму створення формул потрібно вибрати область за межами формули.

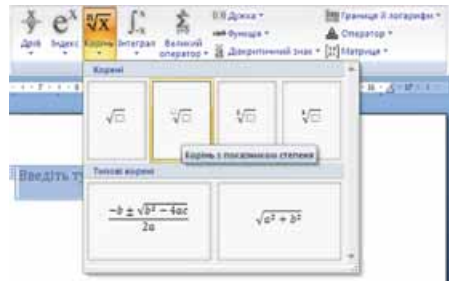
Редагування створеної формули або її фрагментів (видалення, вставлення чи замінування, переміщення та копіювання) здійснюється стандартними засобами **Word 2007**. Під час форматування можна змінити значення властивостей окремих символів формули, встановити міжрядковий інтервал, відступи, вирівнювання, спосіб розташування формули в документі тощо. Для цього використовуються елементи керування **міні-панелі**, вкладки **Основна** та команди контекстного меню (мал. 3.41) формули.

Створену формулу користувач може додати до бібліотеки вбудованих формул. Для цього потрібно:

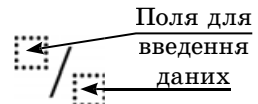
1. Виділити створену формулу.
2. Виконати **Конструктор** ⇒ **Знаряддя** ⇒ **Рівняння**.
3. Вибрати у списку команду **Зберегти виділення в колекції формул**.
4. Увести у діалоговому вікні **Створення стандартного блока ім'я створеної формули**, указати значення інших параметрів.
5. Вибрати кнопку **ОК**.



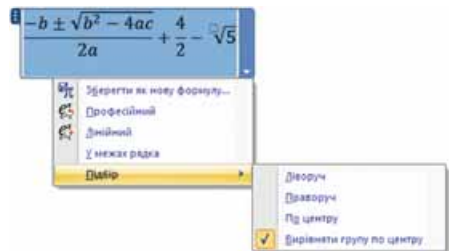
Мал. 3.38. Відкритий список символів



Мал. 3.39. Група **Структури** і список шаблонів кнопки **Корені**



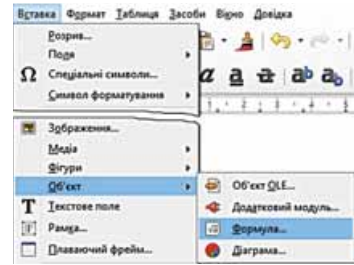
Мал. 3.40. Шаблон кореня з показником степеня



Мал. 3.41. Контекстне меню формули

Для тих, хто працює з LibreOffice

Для вставлення формули до тексту документа потрібно виконати **Вставка** ⇒ **Об'єкт** ⇒ **Формула** (мал. 3.42). У подальшому використовуються шаблони для блоків формул, перелік яких з'являється на панелі в лівій частині вікна програми. Редагування формули в лінійному вигляді здійснюється в нижньому полі вікна програми, а в тексті відображається результат редагування.




Мал. 3.42

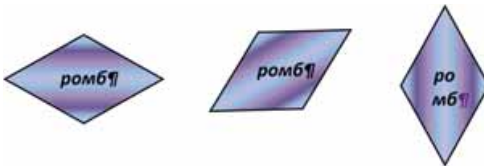
Для тих, хто хоче знати більше

Створення написів у текстовому документі

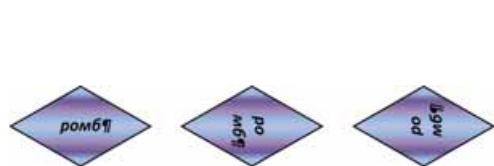
Для розміщення написів у текстовому документі існує кілька способів – текст можна розмістити всередині фігури або створити спеціальні графічні об'єкти **Напис** чи **WordArt**.

У внутрішній області більшості фігур можна розмістити текст. Для додавання тексту до фігури слід у контекстному меню фігури вибрати команду **Додати текст** і ввести його. Якщо повернути або дзеркально відобразити фігуру, розміщення тексту всередині фігури буде відповідно змінюватися (мал. 3.43).


Текст, розміщений у фігурі, можна редагувати та форматовувати стандартними засобами **Word 2007**. Напрямок розташування тексту (горизонтально чи вертикально) можна змінити послідовним вибором кнопки  **Напрямок тексту**, яка розташована на вкладці **Формат** у групі **Текст**, під час виділення фігури. Варіанти розміщення тексту представлено на малюнку 3.44.



Мал. 3.43. Розміщення тексту при змінненні фігури



Мал. 3.44. Напрямки розміщення тексту у фігурі

Якщо потрібно створити текстовий напис як окремий графічний об'єкт у текстовому документі, то для цього слід скористатися спеціальним інструментом **Напис** , що розміщений на вкладці **Вставлення** у групі **Зображення** у списку кнопки **Фігури**.

Редагування і форматування тексту в написі здійснюється відомими способами. У подальшому всі операції з об'єктом **Напис** (зміна розмірів рамки, її розміщення та оформлення) виконуються в такий самий спосіб, що і з іншими графічними об'єктами.

Текстовий процесор **Word** дає змогу вставляти в документи так звані об'єкти **WordArt** (мал. 3.45), які використовуються для художнього оформлення документа.

Об'єкти **WordArt** мають два види властивостей: *текстові* (шрифт, розмір символів, колір, накреслення) і *графічні* (розмір зо-




Мал. 3.45. Приклад об'єкту **WordArt**






браження, розташування, заливка, колір ліній, наявність тіні та об'ємних ефектів тощо).

Створення об'єктів **WordArt** здійснюється за таким алгоритмом:

1. Вибрати на вкладці **Вставлення** у групі **Текст** кнопку **WordArt** .
2. Вибрати в наведеному списку потрібний стиль напису.
3. Увести текст створюваного напису в текстове поле.
4. Вибрати точку за межами поля.

Створений об'єкт **WordArt** буде розміщено в поточному місці документа. Редагування та форматування тексту і графічного об'єкта здійснюється з використанням елементів керування тимчасової вкладки **Формат**.

Найважливіше в цьому пункті

Для створення нескладних формул, які мають лінійний вигляд, використовують спеціальні символи, вставка яких здійснюється у вікні **Символ (Вставлення ⇒ Символи ⇒ Символ  ⇒ Інші символи)**. Уведення символів нижнього і верхнього індексу здійснюється з використанням кнопок **Підрядковий знак ** та **Надрядковий знак **, які розташовані на вкладці **Основне** в групі **Шрифт**.

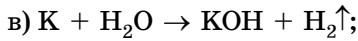
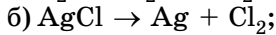
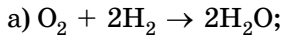
Уведення багаторівневих формул здійснюється спеціальним засобом **Конструктор формул**. Формули в ньому конструюються з окремих символів і структур, використовуючи відповідні шаблони, які потрібно заповнити даними.

Дайте відповіді на запитання

- 1°. Як ввести текст, у якому є хімічні формули?
- 2°. Як вставити в текст такі математичні знаки: \pm \geq \neq \approx ?
- 3°. Використовуючи який інструмент, можна створити у текстовому документі формулу?
- 4°. Які набори математичних символів використовуються у формулах? Як вибрати потрібний символ?
- 5°. Які структури можна вводити у формули? Назвіть кілька з них.
- 6°. Що таке шаблон структури? З яких елементів він складається? Як його використовують?
- 7°. Які операції редагування можна виконувати над формулою? Як вони здійснюються?
- 8°. Які операції форматування можна виконувати над формулою? Як вони здійснюються?

Виконайте завдання

- 1°. Створіть новий текстовий документ. Уведіть такий текст і збережіть документ у вашій папці у файлі з іменем **завдання 3.4.1.docx**. До складу напою «Кока-кола» входять: вода H_2O , вуглекислий газ CO_2 , вугільна H_2CO_3 та фосфорна H_3PO_4 кислоти, сахароза $\text{C}_{12}\text{H}_{22}\text{O}_{11}$, кофеїн $\text{C}_8\text{H}_{10}\text{N}_4\text{O}_2$ тощо.
- 2°. Створіть новий текстовий документ. Уведіть наведені схеми хімічних реакцій. Збережіть документ у вашій папці у файлі з іменем **завдання 3.4.2.docx**.



3°. Створіть новий текстовий документ. Уставте в нього наведені фізичні формули. Збережіть документ у вашій папці у файлі з іменем завдання 3.4.3.docx.

а) період обертання $T = \frac{t}{N}$;

б) співвідношення для сполучних посудин $\frac{h_1}{h_2} = \frac{\rho_2}{\rho_1}$;

в) кінетична енергія $E_k = \frac{mv^2}{2}$;

г) коефіцієнт корисної дії $\eta = \frac{A_1}{A_2} \cdot 100\%$.

4°. Створіть новий текстовий документ. Уставте в нього наведені математичні вирази. Збережіть документ у вашій папці у файлі з іменем завдання 3.4.4.docx.

а) $\frac{3}{4}(12 - x) = \frac{3}{8}$;



в) $\frac{7}{8}(2 - x) = \frac{1}{8}$;

б) $\frac{3}{7}(1 - 2x) + \frac{1}{7} = -2$;



г) $\frac{2}{3}(x - 4) + \frac{1}{3} = 5$.

5°. Створіть новий текстовий документ. Уставте в нього наведені математичні вирази. Збережіть документ у вашій власній папці у файлі з іменем завдання 3.4.5.docx.

а) $y = \begin{cases} x^2 + 3, & \text{якщо } x \leq 0, \\ \frac{x}{2} + 3, & \text{якщо } x > 0; \end{cases}$



в) $y = \begin{cases} 3x - y = 1, \\ x + y = 3, \\ 4y = 8; \end{cases}$

б) $y = \begin{cases} -3x^3 - 2, & \text{якщо } x < 0, \\ \frac{7x}{4} - 2, & \text{якщо } x \geq 0; \end{cases}$



г) $y = \begin{cases} x - y = 1, \\ x + y = 3, \\ 2x = 6. \end{cases}$

6°. Створіть новий текстовий документ. Уставте в нього наведені математичні вирази. Збережіть документ у вашій папці у файлі з іменем завдання 3.4.6.docx.

а) $\frac{5a^4 - 20b^2}{2a^2 + b^2} : \frac{30(a - 4b)^2}{9a^4 - b^4}$;



в) $\frac{a + \frac{25}{a+10}}{\frac{25}{25-a} - a}$;

б) $\frac{x}{7a^2b^3} \cdot \frac{6a^3c^2}{b} = \frac{y}{4c}$;



г) $1 - \frac{1}{1 - \frac{a}{1 - \frac{1}{a+1}}}$.



ПРАКТИЧНА РОБОТА № 4

«Створення текстового документа, що містить об'єкти різних типів»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте текстовий документ з файлу зразок пр4.docx, який знаходиться в папці Розділ 3\практична 4. Оформіть його за наведеним зразком:

Дорогі діти!

Юнацька туристична фірма «Топ-топ» пропонує провести незабутні веселі канікули на легендарному острові Хортиця!

На вас чекають:

- ❖ подорожі,
- ❖ екскурсії,
- ❖ дискотеки,
- ❖ туристичні естафети,
- ❖ спортивні конкурси,
- ❖ козацькі розваги,
- ❖ прогулянки по річці,
- ❖ дивовижні схили,
- ❖ піщані пляжі,
- ❖ історичні місця.

Для проживання можна вибрати:

1. Табори пластунів (7 днів, до 20 дітей)
2. Стаціонарні корпуси (20 днів, до 500 дітей)
3. Дерев'яні будинки (10 днів, до 100 дітей)

Вартість путівок:

Термін	Ціна (за 1 місце в день) у грн.		
			
Березень – травень	20	50	70
Червень – серпень	30	70	100
Вересень – жовтень	20	50	70



Замовити путівки можна:
на сайті www.top-top.zp.ua або
електронною поштою admin@top-top.zp.ua

2. Створіть у документі логотип туристичної фірми «Топ-топ» з графічних примітивів згідно з наведеним зразком:



3. Уведіть у кінці документа фізичні формули відповідно до зразка:

а) густина твердого тіла $\rho = \rho_0 \frac{P}{P - P_1}$;

б) $\text{KMnO}_4 \rightarrow \text{K}_2\text{MnO}_4 + \text{MnO}_2 + \text{O}_2\uparrow$; в) $\frac{3x+7}{4} - \frac{x-3}{2} = \frac{5x+2}{8}$.

4. Збережіть документ у вашій папці у файлі з іменем **практична робота 4.docx**.

3.5. РОЗДІЛИ, КОЛОНТИТУЛИ, СТИЛІ, СТРУКТУРА В ТЕКСТОВИХ ДОКУМЕНТАХ

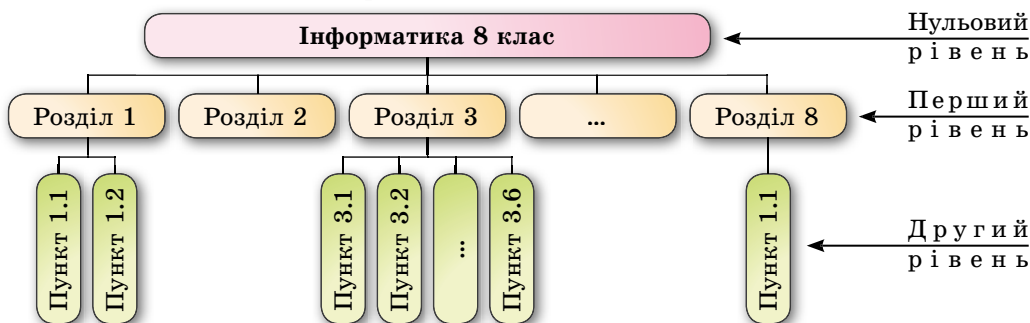


1. Які властивості сторінок ви знаєте? Як встановити значення цих властивостей?
2. Які є режими перегляду текстового документа у **Word 2007**? Як їх встановити?
3. Яким чином структуровано текст вашого підручника з інформатики? Які об'єкти є на його сторінках?

СТРУКТУРА ДОКУМЕНТА

Багатосторінкові документи (реферати, курсові роботи, брошури, книжки тощо) зручно ділити на структурні частини – розділи, глави, параграфи, пункти тощо, створюючи таким чином ієрархічну структуру документа. З такими структурами документів ви стикалися, наприклад, на уроках української літератури, коли складали план написання твору.

Якщо як приклад розглянути структуру цього підручника (мал. 3.46), то на першому рівні ієрархії знаходяться розділи підручника, другий рівень складають пункти, третій – підпункти і т. д.



Мал. 3.46. Приклад структури документа

Структура документа – це ієрархічна схема розміщення складових частин документа.


РОЗДІЛИ В ТЕКСТОВОМУ ДОКУМЕНТІ

Для того щоб поділити текстовий документ на розділи в тому місці, де починатиметься інший розділ, потрібно вставити **Розрив розділу**. Якщо розділів у документі повинно бути кілька, то для кожного з розділів вставляється свій розрив. За відсутності розривів розділів текстовий документ розглядається як один розділ.

Існує кілька видів розривів розділів:

- з наступної сторінки – розділ починається з початку наступної сторінки;
- з поточної сторінки – розділ починається в поточному місці документа;
- з парної сторінки – розділ починається з наступної парної сторінки;
- з непарної сторінки – розділ починається з наступної непарної сторінки.




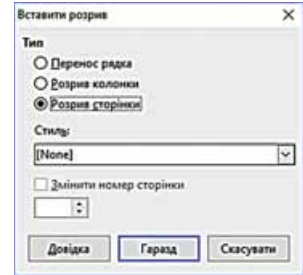
Створення розділів бажано виконувати в режимі перегляду документа **Розмітка сторінки** і при включеному режимі відображення *прихованих* (недрукованих) символів. Далі потрібно виконати **Розмітка сторінки** ⇒ **Параметри сторінки** ⇒ **Розриви**  і вибрати у списку потрібний тип у переліку **Розриви розділів**. У місці розриву розділів з'являється прихований (недрукований) символ **Розрив розділу**.

Зазвичай у текстовому документі є тільки один розділ, і всі сторінки документа відформатовано однаково (розмір аркуша, розміри полів, орієнтація сторінки тощо). Однак під час роботи з багатосторінковими документами може виникнути потреба, щоб деякі сторінки документа мали інші параметри. Наприклад, у тексті з книжковою орієнтацією сторінок потрібно вставити таблицю або схему на сторінці з альбомною орієнтацією або деякий фрагмент документа потрібно розмістити в кілька колонок. Таке форматування документа можливо зробити, розділивши документ на розділи.

Для видалення розриву розділів потрібно видалити в документі вказаний прихований символ. При цьому текст над місцем видаленого розриву стає частиною наступного розділу і його формат автоматично змінюється відповідним чином.

Для тих, хто працює з LibreOffice

Для створення розриву в тексті потрібно виконати **Вставка** ⇒ **Розрив** і вибрати у списку потрібний тип розриву (мал. 3.47). Уставити розрив можна також вибором кнопки **Вставити розрив рядка**  на панелі інструментів **Стандартна**.



Мал. 3.47. Вікно Вставити розрив

ВСТАВЛЕННЯ КОЛОНТИТУЛІВ


У багатосторінкових документах часто використовують *колонтитули* (франц. *colonne* – стовпець, колонка, лат. *titulus* – заголовок) – це службові повідомлення, які розміщують на полях сторінки і повторюються на інших сторінках документа.

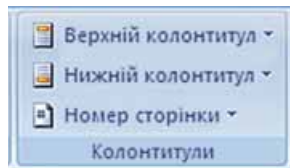
У **Word 2007** розрізняють *верхній*, *нижній* та *бічні* колонтитули. Вони можуть містити номери сторінок, назву документа або поточного розділу, прізвище автора, графічні зображення тощо. Для прикладу – роздивіться сторінку вашого підручника з інформатики.

Колонтитули першої сторінки, парних і непарних сторінок можуть різнитися. Також можуть бути різними колонтитули різних розділів документа, наприклад, як у цьому підручнику – у різних розділів різні колонтитули.

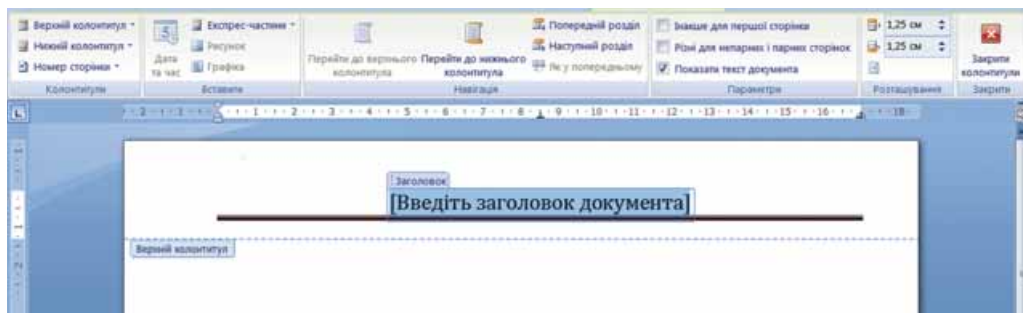
На сторінки текстового документа у **Word 2007** можна вставляти колонтитули, скориставшись готовою колекцією шаблонів колонтитулів, або створити власні колонтитули, які можна зберегти в колекції. Відображаються колонтитули в документі тільки в режимах **Розмітка сторінки** та **Читання**. Опрацювання основного тексту документа під час роботи з колонтитулами неможливе.

Для створення колонтитулів використовуються елементи керування вкладки **Вставлення** групи **Колонтитули** (мал. 3.48).

Після їх створення на **Стрічці** з'являється тимчасова вкладка **Конструктор** (мал. 3.49). Використовуючи відповідні кнопки груп **Колонтитули** та **Вставити**, у колонтитул можна вставити номер сторінки – кнопка **Номер сторінки** , поточну дату та час – кнопка **Дата та час**, рисунок – кнопка **Рисунок** та ін. Розміщення вмісту колонтитула відносно верхнього чи нижнього краю сторінки регулюється відповідними лічильниками у групі **Розташування**.




Мал. 3.48. Група **Колонтитули**




Мал. 3.49. Шаблон верхнього колонтитула і тимчасова вкладка **Конструктор**

Для створення на кожній сторінці документа колонтитула з готової колекції шаблонів потрібно:

1. Вибрати на **Стрічці** вкладку **Вставлення**.
2. Вибрати в групі **Колонтитули** одну з команд **Верхній колонтитул** або **Нижній колонтитул**.
3. Вибрати в запропонованому списку шаблонів колонтитулів один з варіантів.
4. Установити потрібні значення властивостей колонтитула, використавши елементи керування вкладки **Конструктор**.
5. Увести потрібний текст у поля шаблону.
6. Закрити колонтитул, виконавши **Знаряддя для колонтитулів** ⇒ **Конструктор** ⇒ **Закрити** ⇒ **Закрити колонтитул** .

Створений колонтитул можна застосувати до всього документа, до парних чи непарних сторінок або до першої сторінки. Вибрати область застосування створеного колонтитула можна у списку кнопки **Параметри**. Встановлення значень деяких властивостей колонтитулів також можна здійснити і на вкладці **Макет** діалогового вікна **Параметри сторінки**.

Для переходу з поля верхнього колонтитула до поля нижнього колонтитула і назад використовується кнопка **Перейти до Верхнього (Нижнього) колонтитула**  у групі **Навігація** вкладки **Конструктор**.

Якщо колонтитули для різних частин документа повинні бути різними, то перед їх створенням документ потрібно поділити на розділи. Потім колонтитули для різних розділів створюємо окремо.

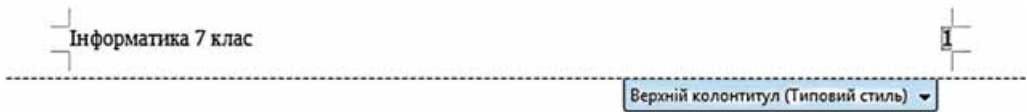


За потреби текст колонтитула можна редагувати і форматувати звичайним способом. Для змінення вставленого колонтитула потрібно виконати **Вставлення** ⇒ **Колонтитули** ⇒ **Верхній (Нижній) колонтитул** ⇒ **Змінити верхній (нижній) колонтитул**.

Для видалення колонтитула слід виконати **Вставлення** ⇒ **Колонтитули** ⇒ **Верхній колонтитул (Нижній колонтитул)** ⇒ **Видалити верхній (нижній) колонтитул**.

Для тих, хто працює з LibreOffice

Для створення колонтитулів використовуються команди **Верхній колонтитул** та **Нижній колонтитул** меню **Вставка**. Після вибору однієї із цих команд, а також стилю колонтитула користувач може відредагувати вміст колонтитулів (мал. 3.50).



Мал. 3.50. Верхній колонтитул

Для вставлення шаблонних значень властивостей документа (номер сторінки, поточна дата, кількість сторінок, дані про автора та ін.) в колонтитул слід використати команди кнопки зі списком **Вставити поле** панелі інструментів **Стандартна**. Для завершення редагування документа слід вибрати місце поза межами колонтитула.

Для використання різних колонтитулів для парних і непарних сторінок та щоб не використовувати колонтитул для першої сторінки встановлення відступів тощо, потрібно виконати **Формат** ⇒ **Сторінка** вибрати команду **Форматувати верхній (нижній) колонтитул** у списку кнопки, що розміщується біля межі колонтитула під час його редагування.

СТИЛІ В ТЕКСТОВОМУ ДОКУМЕНТІ

На попередніх уроках вам доводилося неодноразово створювати нові текстові документи та оформлювати їх належним чином, встановлюючи потрібні значення властивостей для різних типів об'єктів – тексту, таблиць, рисунків, формул, об'єктів **SmartArt** тощо. Якщо документ великий і має значну кількість різноманітних об'єктів, то на форматування кожного окремого об'єкта знадобиться значний час. Тим більше, якщо часто трапляються об'єкти, які повинні бути відформатовані однаково, – стежити за однаковим оформленням однотипних елементів документа досить складно.

Для прискорення оформлення документів та їх об'єктів зручно використовувати стилі. **Стиль** – це набір значень властивостей певного типу об'єктів, який має власне ім'я.

Текстовий процесор **Word 2007** за замовчуванням має власну бібліотеку професійно розроблених стилів, так звані **експрес-стилі**, у яких значення властивостей об'єктів підібрані гармонійно, з урахуванням основних вимог до дизайну. Такі стилі розроблено для різних типів об'єктів текстового документа – тексту, таблиць, графічних об'єктів тощо (табл. 3.10). Кожен

експрес-стиль у бібліотеці має ім'я. Ви вже застосовували їх у форматуванні відповідних об'єктів.

Таблиця 3.10

Приклади стилів різних об'єктів у Word 2007

Тип об'єкта	Ім'я стилю	Зразок оформлення об'єкта	Деякі значення властивостей
Текст	Заголовок 4	Для швидкого оформлення тексту дуже зручно використовувати стилі	Шрифт – <i>Cambria</i> Розмір – 12 Колір – чорний Накреслення – <i>жирний курсив</i> Міжрядковий інтервал – 1,5 рядка Відступ першого рядка – немає Вирівнювання – за шириною Інтервал перед абзацом – 3 пт
Таблиця	Світлий список, акцент 3		Колір ліній – <i>зелений</i> Колір заливки – <i>перший рядок зелений</i> Штрих лінії – <i>суцільна</i> Товщина лінії – 0,5 пт
Фігура	Центральний градієнт – акцент 6		Колір – <i>світло-коричневий</i> Прозорість – 0 % Штрих лінії – <i>суцільний</i> Товщина лінії – 1 пт Колір контуру – <i>темно-коричневий</i> Тіньові ефекти – <i>знизу справа</i>

Використання стилів дає змогу однією дією відразу змінити значення кількох властивостей об'єкта. Наприклад, для того щоб вручну відформатувати текст з такими значеннями властивостей, які має стиль **Заголовок 4** (табл. 3.10), потрібно виконати 8 окремих дій – вибрати відповідні елементи керування на вкладках або міні-панелі для встановлення значень кожної властивості, а під час застосування стилю **Заголовок 4** – тільки одну дію для вибору стилю з бібліотеки.

Особливо зверніть увагу на такі стилі тексту, як **Заголовок 1**, **Заголовок 2**, ..., **Заголовок 9**, які дають змогу автоматизувати створення структури документа.

Колекцію стилів користувач може доповнювати власними стилями, створюючи їх на основі вже існуючих або наново. Під час збереження документа з ним автоматично зберігаються і застосовані стилі, тобто в подальших відкриттях документа зовнішній вигляд документа не змінюватиметься.

ВИКОРИСТАННЯ ЕКСПРЕС-СТИЛІВ

Ескізи експрес-стилів текстових об'єктів документа (символів, абзаців) відображені на вкладці **Основне** в групі **Стилі** (мал. 3.51).



Мал. 3.51. Ескізи експрес-стилів для текстових об'єктів

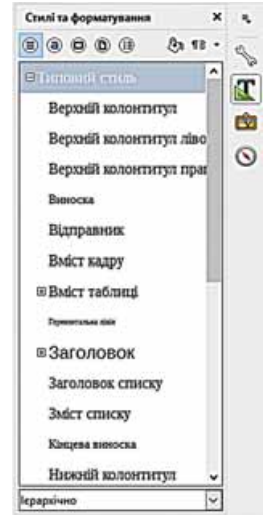


Кожний ескіз із цього списку має оформлення, яке відповідає вибраному стилю, а поруч з іменем стилю міститься значок, який показує тип об'єктів, до яких цей стиль застосовується: до абзаців – ¶; до символів – а, до символів і абзаців – ¶а. Стилі заголовків використовуються для форматування назв структурних частин документа (розділів, пунктів, підпунктів тощо).

У вікні списку експрес-стилів завжди відображається кілька ескізів, які використовувалися останніми. Для перегляду всіх ескізів потрібно використати кнопки ▲ чи ▼ смуги прокручування цього списку. Для відкриття всього списку потрібно вибрати кнопку **Додатково** ▾. Для застосування експрес-стилю потрібно виділити об'єкт і вибрати у списку потрібний ескіз.

Стилі можна перейменувати або видалити, скориставшись командами контекстного меню ескізу стилю. *Слід пам'ятати*, що зазначені операції не можливі над стилями заголовків.

Для відмови від застосування експрес-стилю та встановлення значень властивостей об'єкта за замовчуванням потрібно вибрати команду **Очистити форматування**.



Мал. 3.52. Панель Стилі та форматування

Для тих, хто працює з LibreOffice

Для використання стилів потрібно відкрити панель **Стилі та форматування** (мал. 3.52) у правій частині вікна програми (**Формат** ⇒ **Стилі і форматування** або вибрати кнопку **Стилі і форматування** на правій панелі вікна програми).

Вибір відповідних кнопок відкриває списки: – стилів абзаців, – стилів символів, – стилів кадрів, – стилів сторінки, – стилів списків.

Для роботи з виносками, посиланнями, покажчиками та змістом документа використовуються команди меню **Вставка**.

Для тих, хто хоче знати більше

Теми документа та їх застосування



Мал. 3.53. Бібліотека вбудованих тем документа

Стильове оформлення може бути застосовано не тільки до окремих об'єктів, а й до всього текстового документа як єдиного цілого.

Тема документа – це стиль оформлення текстового документа, який має ім'я і визначає для його об'єктів значення деяких властивостей (колір, розмір та шрифт символів, спосіб вирівнювання абзаців, міжрядковий інтервал, ефекти рисунків тощо). У 5 класі ви застосовували теми для оформлення презентацій.

У **Word 2007** є бібліотека вбудованих тем, список яких відкривається вибором на вкладці **Розмітка сторінки** у групі **Теми** однойменної кнопки **Теми** (мал. 3.53).


Тему можна вибрати на початку створення документа або в процесі його підготовки. Під час перегляду теми відбувається динамічний перегляд вибраного стилю. Вибір теми зумовить зміну стилів оформлення таких об'єктів документа, як текст, таблиці, графічні зображення тощо. Усі об'єкти будуть оформлені в єдиному професійно-розробленому стилі. *Слід пам'ятати*, що змінення теми документа впливає тільки на ті об'єкти, які перед цим були відформатовані з використанням стилів.

СТВОРЕННЯ СТРУКТУРИ ДОКУМЕНТА ТА ЇЇ ВИКОРИСТАННЯ




Створення структури документа відбувається із застосуванням стилів **Заголовок 1, Заголовок 2, ..., Заголовок 9** – назви всіх структурних частин документа слід відформатувати з використанням відповідного стилю. А саме до назв структурних частин першого рівня (розділів) застосувати стиль **Заголовок 1**, назви частин другого рівня (пунктів) відформатувати із застосуванням стилю **Заголовок 2** і т. д.

Для цього потрібно:

1. Виділити назву певної структурної частини документа (розділу, пункту, підпункту).
2. Вибрати на вкладці **Основне** в групі **Стилі** такий стиль заголовка, який відповідає рівню структурної частини в ієрархічній схемі документа (I-й рівень – **Заголовок 1**, II-й рівень – **Заголовок 2** ...).
3. Повторити п. 1–2 цього алгоритму для всіх структурних частин документа.

Для перегляду структури документа, який відформатовано таким чином, використовується режим перегляду **Структура**, який встановлюється вибором кнопки **Структура**  у **Рядку стану** або на вкладці **Вигляд** у групі **Режими перегляду документа**.

У цьому режимі зовнішній вигляд документа відрізняється від його вигляду в режимі **Розмітка сторінки**, у якому ви звикли працювати (мал. 3.54). У режимі **Структура** текст структурних частин документа відображається у так званому **згорнутому вигляді** і на екран виводяться тільки заголовки структурних частин. Для кращої візуалізації ієрархічної структури рівні документа мають різний відступ від лівого поля. Біля кожного рівня документа є відповідна позначка:

-  , якщо в рівня є підрівні або підпорядкований текст;
-  , якщо в рівня відсутній підпорядкований текст;
-  , якщо цей текст має найнижчий рівень (так званий основний текст, до якого не застосовувалися стилі заголовків).

Програма ЮНЕСКО "Інформація для всіх"

1. Преамбула.

1. Інформація і знання є величезним загальнолюдським завоюванням. Вони необхідні для просування освіти, науки, культури і комунікації, для збагачення культурної різноманітності і відкритості управління. Мандат ЮНЕСКО "спирити вільному руху ідей словом і справою" являє собою важливу роль, яку організація повинна грати в тому, щоб зробити інформацію і знання доступними для всіх, з кімшевою метою подолання розриву між інформаційно-багатими й інформаційно-бідними.
2. Для досягнення цієї мети, відповідно до свого конституційного мандата ЮНЕСКО повинна "підтримувати, зберігати, збільшувати, і поширювати знання", гарантуючи і забезпечуючи "збереження і захист" зафіксованих знаць людства. Більш того, Організація повинна сприяти "співпраці між народами... у обміні публікаціями" і іншими інформаційними матеріалами і ініціювати "способи міжнародної співпраці, розраховані на

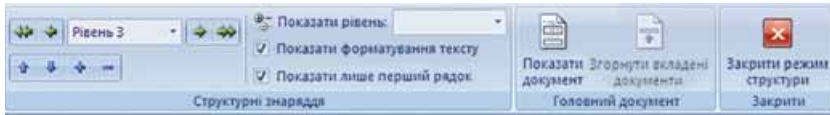
Програма ЮНЕСКО "Інформація для всіх"

1. Преамбула.
 - 1. Інформація і знання є величезним загальнолюдським ...
 - 2. Для досягнення цієї мети, відповідно до свого ...
 - 3. Міжнародна інтелектуальна співпраця є важливою ...
 - 4. Технологічна конвергенція сама, що посилюється, по ...
 - 5. Нові способи доступу, обробки і захисту інформації ...
 - 6. У цьому контексті ЮНЕСКО з своїм мандатом ...
3. Цілі програми.
 - 1. У дусі Міжнародної Декларації Прав Людини, яка формує ...
 - 2. Як одна з основних програм ЮНЕСКО програма ...
 - 3. Зокрема програма "Інформація для всіх" націлена на:
 - о Заохочення і розширення доступу до інформації за ...
 - о Розвиток міжнародної рефлексії і дискусій по етичних ...
 - о Сприяння тренінгу, безперервній освіті і навчанню у сфері ...
 - о Просування використання міжнародних стандартів і ...
 - о Просування мережевої взаємодії у сфері інформації і знань ...

Мал. 3.54. Вигляд тексту документа в режимі **Розмітка сторінки** та **Структура**



Увімкнення цього режиму зумовлює появу на **Стрічці** додаткової вкладки **Структура** (мал. 3.55), елементи керування якої використовуються для відображення тексту тільки потрібних рівнів документа, переміщення структурних частин у документі, змінення їх рівнів, видалення розділів тексту або додавання нових тощо.



Мал. 3.55. Вкладка **Структура**

Основна зручність перегляду документа в режимі **Структура** полягає в тому, що є можливість налаштувати відображення будь-якого рівня документа, приховуючи при цьому ті рівні, які займають нижчу ієрархію. Для цього потрібно на вкладці **Структура** у групі **Структурні знаряддя** вибрати потрібний рівень зі списку **Показати рівень**. Якщо потрібно розгорнути зміст деякої окремої структурної частини документа, слід двічі клацнути на значку **+** біля відповідного заголовка. Повторення цих дій приховає вкладений текст.

Якщо вкладений текст має великий розмір шрифту чи занадто довгий текст, що заважає роботі, можна відключити відображення форматування (зняти позначку прапорця **Показати форматування тексту**), а також відображення всіх рядків тексту, крім першого (встановити позначку прапорця **Показати лише перший рядок**).

Найважливіше в цьому пункті

Структура документа – це ієрархічна схема розміщення складових частин документа. Найчастіше структуру створюють для багатосторінкових документів, розділяючи їх на розділи, глави, параграфи, пункти.

Для того щоб поділити текстовий документ на розділи в тому місці, де починатиметься інший розділ, потрібно вставити **Розрив розділу**. Створення розділів бажано виконувати в режимі перегляду документа **Розмітка сторінки** і при включеному режимі *відображення прихованих (недрукованих) символів*. Далі потрібно виконати **Розмітка сторінки** ⇒ **Параметри сторінки** ⇒ **Розриви** і вибрати у списку потрібний тип в переліку **Розриви розділів**.


У місці розриву розділів з'являється прихований (недрукований) символ **Розрив розділу**. У документі, поділеному на розділи, можна виконати різне форматування окремих частин документа.

У багатосторінкових документах часто використовують **колонтитули** – це службові повідомлення, які розміщують на полях сторінки і повторюються на інших сторінках документа. У **Word 2007** розрізняють *верхній*, *нижній* та *бічні* колонтитули. Вони можуть містити номери сторінок, назву документа або поточного розділу, прізвище автора, графічні зображення тощо. Колонтитули першої сторінки, парних і непарних сторінок можуть різнитися. Також можуть бути різними колонтитули різних розділів документа.

Відображаються колонтитули в документі тільки в **режимах Розмітка сторінки** та **Читання**. Опрацювання основного тексту документа під час

роботи з колонтитулами неможливе. Для створення колонтитулів використовуються елементи керування вкладки **Вставлення** групи **Колонтитули**.

Для прискорення оформлення документів та їх об'єктів зручно використовувати стилі. **Стиль** – це набір значень властивостей певного типу об'єктів, який має власне ім'я. Текстовий процесор **Word 2007** має власну бібліотеку професійно розроблених стилів, так звані **експрес-стилі**, у яких значення властивостей об'єктів підібрані гармонійно, з урахуванням основних вимог до дизайну. Такі стилі розроблено для різних типів об'єктів текстового документа – тексту, таблиць, графічних об'єктів тощо. Кожен експрес-стиль у бібліотеці має ім'я. Ескізи експрес-стилів текстових об'єктів документа відображені на вкладці **Основне** в групі **Стилі**.

Створення структури документа відбувається із застосуванням стилів **Заголовок 1**, **Заголовок 2**, ..., **Заголовок 9** – назви всіх структурних частин документа слід відформатувати з використанням відповідного стилю. Для перегляду структури документа, який відформатовано таким чином, використовується режим перегляду **Структура**, який встановлюється вибором кнопки **Структура**  у **Рядку стану** або на вкладці **Вигляд** у групі **Режими перегляду документа**.

Дайте відповіді на запитання

- 1°. Що таке структура документа? Наведіть приклади структурованих документів.
- 2°. Що таке розділи? Для чого вони застосовуються?
- 3°. Які види розривів розділів можна використовувати в текстовому документі?
- 4°. Що таке колонтитули? Які види колонтитулів можуть бути на сторінці?
- 5°. Яке призначення колонтитулів? Які об'єкти можна вставити в колонтитули?
- 6*. Як встановити різні колонтитули для різних сторінок документа?
- 7°. Що таке стилі? Для чого їх використовують?
- 8°. До яких об'єктів можна застосовувати стильове оформлення у **Word 2007**? Які властивості об'єктів документа вони визначають?
- 9°. Яке призначення стилів заголовків? Скільки таких стилів є у **Word 2007**?
- 10°. Для чого призначений режим відображення документа у вигляді **Структури**?

Виконайте завдання

- 1°. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.5\зразок.docx**). Кожен пункт текстового документа оформіть як окремий розділ, який починається з нової сторінки. Установіть такий формат для розділів документа:
 - а) непарні розділи: орієнтація сторінки – *книжкова*, поля – *по 2 см*, нумерація сторінок – *зверху по центру*;
 - б) парні розділи: орієнтація сторінки – *альбомна*, поля – *по 1 см*, нумерація сторінок – *знизу зліва*.
 Збережіть документ у вашій папці у файлі **завдання 3.5.1.docx**.



2*. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.5\зразок.docx**). Створіть колонтитули: у верхньому запишіть поточну дату з вирівнюванням *по центру*, у нижньому – *ваше прізвище* з вирівнюванням *по лівому краю*. Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.2.docx**.



3*. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.5\зразок.docx**). Відформатуйте документ: розмір сторінки – *A4*; орієнтація – *книжкова*; поля: верхнє – *1,5 см*, нижнє – *1 см*, зовнішнє – *2 см*, внутрішнє – *3 см*; нумерація сторінок – *внизу сторінки*, на *зовнішній стороні*, починаючи з номера *1*; колонтитули: верхній на непарних сторінках – *назва текстового документа*, нижній на парних сторінках – *поточна дата і час*. Збережіть документ у папці **Мої документи** у файлі з іменем **завдання 3.5.3.docx**.

4*. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.5\зразок.docx**). Відформатуйте об'єкти текстового документа з використанням таких стилів:

- Заголовки – стиль *Назва*.
- Текстові фрагменти – стиль *Слабке виокремлення*.
- Списки – *маркований* з маркером ♦.

Збережіть документ у вашій папці у файлі з іменем **завдання 3.5.4.docx**.

5*. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.5\зразок1.docx**), у якому до об'єктів текстового документа було застосовано стильове оформлення. З'ясуйте значення властивостей об'єктів, які встановлюють використані стилі. Запишіть результати в таблицю:

№ об'єкта	Тип об'єкта	Стиль об'єкта	Значення властивостей об'єкта



6*. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.5\зразок1.docx**), у якому до об'єктів текстового документа було застосовано стильове оформлення. Очистіть формати об'єктів документа. Запишіть у таблицю значення властивостей об'єктів за замовчуванням:

№ об'єкта	Тип об'єкта	Стиль об'єкта	Значення властивостей об'єкта



7*. З'ясуйте за допомогою довідки, як створити власний стиль на основі відформатованого фрагмента документа. Складіть відповідний алгоритм. Виконайте це на практиці.

8*. Відкрийте вказаний учителем документ (наприклад, **Розділ 3\Пункт 3.5\зразок.docx**). Відформатуйте документ, використовуючи такі стилі:

- Заголовки 1 рівня – стиль *Заголовок 1*;
- Заголовки 2 рівня – стиль *Заголовок 2*;
- Заголовки 3 рівня – стиль *Заголовок 3*;



- Інші фрагменти тексту – стиль *Звичайний*.
Перегляньте структуру відформатованого документа. Збережіть документ у вашій папці у файлі з іменем завдання **3.5.8.docx**.



3.6. ЗАСОБИ АВТОМАТИЗАЦІЇ СТВОРЕННЯ ТЕКСТОВОГО ДОКУМЕНТА



1. Що таке стилі? Для чого їх використовують?
2. Що таке структура документа? Як її переглянути?
3. Що таке гіперпосилання? Для чого і де воно використовується?

ПОСИЛАННЯ В ТЕКСТОВОМУ ДОКУМЕНТІ

У попередньому пункті ви ознайомилися з деякими особливостями опрацювання багатосторінкових документів – перегляд структури документа, створення розділів, використання стильового оформлення. Крім цього, у текстовому процесорі **Word 2007** існують й інші засоби для спрощення процесу створення великих і складних документів та переміщення ними.

Зазвичай у багатосторінкових документах зустрічаються різні види *посилань* – спеціальний запис у документі, який вказує на якийсь об'єкт цього або іншого документа. Посилання призначені для логічного зв'язування одних об'єктів документа з іншими. Наприклад, у тексті може знаходитися посилання на певний рисунок чи таблицю, посилання на бібліографічні джерела, сайти або інші документи. При клацанні на ці посилання відбувається перехід на вказаний об'єкт.

Текстовий процесор **Word 2007** дає змогу автоматизувати процес створення посилань у документі. Інструменти, які використовуються для цього, знаходяться на вкладці **Посилання**, наприклад:

- група **Зміст** призначена для створення *змісту документа* – переліку назв структурних частин документа із зазначенням номерів відповідних сторінок;
- група **Покажчик** призначена для створення *предметного (алфавітного) покажчика* – алфавітного списку термінів, які використані в документі із зазначенням номерів сторінок, де вони трапляються.

АВТОМАТИЗОВАНЕ СТВОРЕННЯ ЗМІСТУ ДОКУМЕНТА


Якщо для кожного зі структурних частин багатосторінкового документа використане стильове форматування із застосуванням стилів заголовків, то зміст документа можна створити автоматично. Виконується ця операція в режимі перегляду документа **Розмітка сторінки**


Для цього потрібно виконати такий алгоритм:

1. Встановити курсор у тому місці документа, де потрібно розмістити зміст.
2. Виконати **Посилання** ⇒ **Зміст** ⇒ **Зміст** .
3. Вибрати у списку вбудованих зразків потрібний варіант оформлення змісту (мал. 3.56).



Зміст буде представлено у вигляді таблиці, у яку внесено назви структурних частин текстового документа та відповідні номери сторінок, де цей фрагмент тексту починається (мал. 3.57). Зазвичай зміст документа створюється на його початку або в кінці.

Створений зміст можна використовувати для швидкого переміщення по документу, пошуку потрібних структурних частин – розділів, пунктів тощо. Для цього слід у змісті документа вибрати потрібний заголовок та натиснути клавішу **Ctrl**. Зверніть увагу, що вигляд текстового курсору зміниться на такий – .

Якщо в ході роботи над документом його текст та структура змінювалися, то зміст документа потрібно оновити. Для цього потрібно виконати **Посилання** ⇒ **Зміст** ⇒ **Оновити таблицю** .



Мал. 3.56. Список вбудованих зразків оформлення змісту

Зміст	
Програма ЮНЕСКО "Інформація для всіх".....	1
1. Преамбула.....	1
2. Мандат.....	1
3. Цілі програми.....	2
4. Основні розділи програми.....	2
Розділ 1. Розвиток інформаційної політики на міжнародному, регіональному і національному рівнях.....	2
Розділ 2. Розвиток людських ресурсів, навиків і вмінь у століття інформації.....	3
Розділ 3. Посилення ролі інституцій в забезпеченні доступу до інформації.....	3
Розділ 4. Розвиток інструментів, способів і систем інформаційного менеджменту.....	2

Мал. 3.57. Приклад змісту документа

АВТОМАТИЗОВАНЕ СТВОРЕННЯ ПРЕДМЕТНОГО ПОКАЖЧИКА

Предметні покажчики найчастіше використовуються в підручниках, наукових виданнях, словниках. Розміщуються вони зазвичай в кінці документа і виглядають приблизно так (мал. 3.58):

СЛОВНИЧОК

А

Абзац – об'єкт текстового документа, який має такі властивості: вирівнювання, відступи, міжрядковий інтервал та інші, с. 134.

Авторське право – право авторів розповсюджувати матеріали, які вони створили, підписувати їх своїм іменем, отримувати за них винагороду тощо, с. 182.

Алгоритм – скінченна послідовність команд, виконання яких приводить до розв'язання поставленої задачі, с. 19.

Б


Байт – одиниця вимірювання розмірів файлів і ємності носіїв даних, с. 54.

Блок-схема алгоритму – форма подання алгоритму у вигляді блоків. У блок-схемі алгоритму кожна команда записується в геометричній фігурі (блоці) певного вигляду. Блоки з'єднуються між собою стрілками, які вказують, яку команду алгоритму потрібно виконати наступною, с. 24.

Мал. 3.58. Приклад предметного покажчика в підручнику

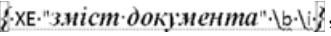
Створення предметного покажчика виконується у два етапи: спочатку потрібно виділити всі терміни, які слід помістити в покажчик, потім вставити покажчик у документ. Розглянемо етапи детальніше.

І е т а п. Виділення елементів покажчика:


1. Виділити термін чи фразу, які потрібно помістити в покажчик, на сторінці, де вони трапляються вперше.
2. Виконати **Посилання** ⇒ **Покажчик** ⇒ **Позначити елемент** .
3. Встановити у вікні **Визначення елемента покажчика** (мал. 3.59), що відкрилось, потрібні значення:

- *основне* – відредагувати, за потреби, основну назву елемента покажчика;
- *додатковий* – ввести, за потреби, додаткові назви елемента покажчика;
- *перехресне посилання* – вибрати назву іншого елемента покажчика, з яким пов'язаний даний елемент;
- *поточна сторінка* – вказувати в покажчику для даного терміна тільки поточну сторінку;
- *діапазон сторінок* – вказувати в покажчику для даного терміна діапазон сторінок, де він трапляється (додатково потрібно встановити закладки для початкової та кінцевої сторінки діапазону);
- *формат номера сторінки* – встановити напівжирного та курсивного накреслення номера сторінки в покажчику.

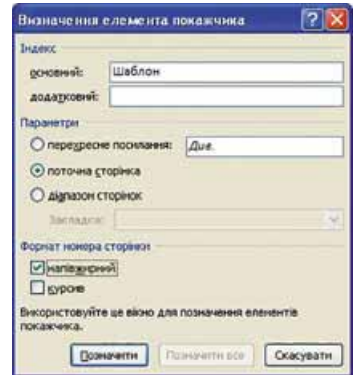
4. Вибрати кнопку **Позначити** (або **Позначити все**, якщо потрібно щоб у покажчику вказувалися всі номери сторінок, де цей термін трапляється).
5. Повторити дії 1–4 для всіх елементів покажчика.

У результаті таких дій всі терміни для предметного покажчика будуть визначені і в тексті документа біля цих слів буде вставлено поле такого вигляду: , однак відобразатися воно буде тільки в режимі відображення прихованих (недрукованих) символів.

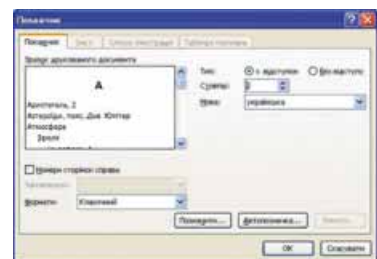
І е т а п. Вставлення предметного покажчика в текстовий документ:

1. Встановити курсор у те місце документа, де потрібно розмістити покажчик (зазвичай у кінці документа).
2. Виконати **Посилання** ⇒ **Покажчик** ⇒ **Покажчик** .
3. Встановити у вікні **Покажчик** (мал. 3.60), що відкрилось, параметри відображення покажчика:

- *Номер сторінки по правому краю* – установити за бажання такий спосіб вирівнювання номерів сторінок у покажчику;
- *Формат покажчика* – вибрати із запропонованого списку бажаний вид;



Мал. 3.59. Вікно **Відмітити елемент покажчика**




Мал. 3.60. Вікно **Покажчик**



- *Тип* – вибрати за бажання один з двох способів розміщення номерів сторінок біля елемента покажчика;
- *Колонки* – указати кількість колонок, у яких буде розміщено покажчик;
- *Мова* – алфавіт, за яким буде впорядковано елементи покажчика.

4. Вибрати кнопку **ОК**.

Таким чином предметний покажчик буде вставлено у вибране місце в документі. Перед і після покажчика в документі буде вставлено приховані символи **Розрив розділу**.

Якщо деякі дані, що входять до покажчика, змінилися (змінився текст або розміщення елемента в документі), то зміст покажчика потрібно оновити. Для цього слід виділити покажчик та виконати **Посилання** ⇒ **Покажчик** ⇒ **Покажчик** .

Для видалення покажчика потрібно виділити його текст і два рядки – перед і після нього з прихованими символами розривів розділів, потім натиснути клавішу **Delete**.



Для тих, хто працює з LibreOffice

Для роботи з покажчиками та змістом документа використовуються команди **Покажчики та зміст** меню **Вставка** (мал. 3.61).

Мал. 3.61. Список команд **Покажчики та зміст** меню **Вставка**

ШАБЛони ТЕКСТОВИХ ДОКУМЕНТІВ

Ви вже навчилися створювати текстові документи та формувати їх з використанням різних засобів текстового процесора. Разом з тим текстовий процесор **Word** для створення документів певних типів надає ще одну можливість – створення документа на основі шаблону. Ви вже створювали за допомогою шаблонів у 5 класі презентації.

Шаблон – це відформатований певним чином документ-заготовка, який зберігається в окремому файлі і використовується як зразок для створення нових документів певного типу.

Наприклад, шаблон для створення календаря, подяки, листа, звіту, вітальної листівки, візитки тощо.

Відкривши шаблон, користувач побачить на екрані текстовий документ, у якому будуть міститися порожні поля для введення тексту, написів, малюнків, колонтитулів, логотипи тощо. До цього документа вже можуть бути застосовані певні теми та стилі оформлення різних об'єктів (мал. 3.62).

Користувачу залишається тільки заповнити відповідні місця документа потрібним текстом і зберегти документ у текстовому файлі – форматування та структурування документа вже визначено самим шаблоном.

Використання шаблонів значно полегшує роботу користувача, оскільки вони є основою для швидкого створення аналогічних за формою, але різних за змістом документів без витрат часу на їх форматування.

Основна відмінність між документами і шаблонами полягає в їх використанні: шаблон – це заготовка документа з готовими елементами тексту та оформлення, яка призначена для подальшого заповнення даними, а документ – це вже підготовлений текст, можливо на основі деякого шаблону.

СВЯТКУВАННЯ
Нового року
МЕНЮ

[ПУНКТ МЕНЮ]
[Щоб замінити текст змійченим (наприклад, цій) словом, виділіть його та введіть новий текст. Не додавайте пробілів праворуч і ліворуч символів виділеного фрагмента.]

[ПУНКТ МЕНЮ]
[Щоб зменшити або збільшити уяву гостей, описи страв мають бути короткими та привабливими.]

[ПУНКТ МЕНЮ]
[Опис пункту меню]

[ПУНКТ МЕНЮ]
[Опис пункту меню]

[ПУНКТ МЕНЮ]



СВЯТКУВАННЯ
Нового року
МЕНЮ

Салат «Зимовий»
курча печіна, корейська морква,
мариновані огірки, гірчична заправка

Салат «Маскарад»
шинка, крабові палички, картопля, зелений
горошок, листя салату, майонез

Десерт «Серпантин»
яблука в шоколаді

Коктейль «Феєрверк»
банан, полуниця, ананас з молоком



Мал. 3.62. Шаблон меню вечірки та створений на його основі документ

Шаблони зберігаються у файлах з розширенням **dotx** та **dotm**. Пам'ятайте, у шаблонах можуть зберігатися макровіруси, тому використовуйте шаблони, які ви отримали з надійного джерела.

У текстовому процесорі **Word 2007** усі шаблони діляться на три групи:

- *інстальовані* – шаблони певних типів документів (листів, факсів, звітів тощо), які інстальовані на комп'ютері у складі пакета **Microsoft Office 2007**;
- *Microsoft Office Online* – шаблони різноманітних типів документів, які розташовані на веб-сайті **Microsoft Office Online**;
- *шаблони користувача* – шаблони, які створені користувачем.

Основним зі стандартних шаблонів **Word 2007** є шаблон документів **Звичайний** (зберігається у файлі **Normal.dotm**), який автоматично відкривається під час запуску програми **Word 2007** і встановлює за замовчуванням такий формат документа (табл. 3.11):

Таблиця 3.11

Значення властивостей об'єктів документа в шаблоні **Normal.dotm**

Сторінка	Абзаци	Символи
<ul style="list-style-type: none"> • Орієнтація аркуша – книжкова • Розмір аркуша – A4 • Верхнє поле – 1,5 см • Нижнє поле – 1 см • Ліве поле – 2 см • Праве поле – 1,5 см 	<ul style="list-style-type: none"> • Міжрядковий інтервал – <i>одинарний</i> • Інтервал після абзацу – 10 пт • Вирівнювання – <i>зліва</i> • Відступи – <i>відсутні</i> 	<ul style="list-style-type: none"> • Шрифт основного тексту – <i>Calibri</i> • Розмір – 11 пт • Колір – <i>чорний</i> • Інтервал – <i>звичайний</i>

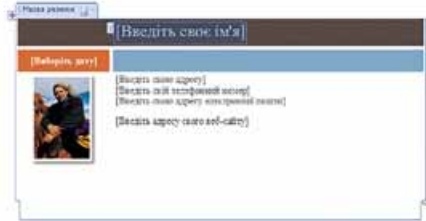
Також цей шаблон визначає стильове оформлення заголовків, списків, таблиць тощо. Введення тексту, форматування окремих об'єктів, структу-



рування тексту користувач здійснює самостійно. На основі цього шаблону **Normal.dotm** можна створювати документи різних типів.

Ще одним зі стандартних шаблонів **Word 2007** є, наприклад, шаблон **Звичайне резюме** (файл **MedianResume.Dotx**). Цей шаблон містить у документі-заготовці кілька текстових полів, у які користувач вносить дані про себе: ім'я та прізвище, контактну інформацію, відомості про освіту, місце роботи тощо (фрагмент шаблону наведено на мал. 3.63). Вставлену фотографію замінює на власну. Дату можна вибрати в календарі, який відкривається за вибору зазначеного поля. Кожна структурна частина документа вже відформатована і розміщена в тексті відповідним чином. Використовуючи цей шаблон, можна швидко підготувати власне резюме.


Резюме (франц. *Résumé*) – вид документа, в якому подаються короткі відомості про навчання, трудову діяльність та професійні успіхи й досягнення особи, яка його складає.

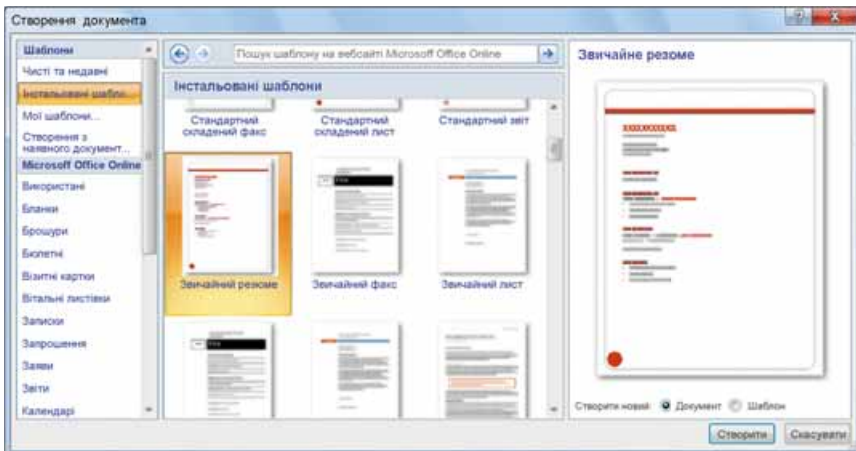


Мал. 3.63. Шаблон **Звичайне резюме**

СТВОРЕННЯ ТЕКСТОВИХ ДОКУМЕНТІВ НА ОСНОВІ ШАБЛОНІВ

Для того щоб у текстовому процесорі **Word 2007** створити документ на основі інстальованих шаблонів, потрібно:


1. Відкрити **Головне меню** програми вибором кнопки **Office**.
2. Вибрати команду **Створити** , що відкриває діалогове вікно **Створення документа**.
3. Вибрати у списку ліворуч у розділі **Шаблони** потрібну групу шаблонів – **Інстальовані шаблони** (мал. 3.64).
4. Вибрати у списку шаблонів потрібний (наприклад, **Звичайне резюме**).
5. Переглянути структуру та зовнішній вигляд шаблону в полі зразків (праворуч у вікні).

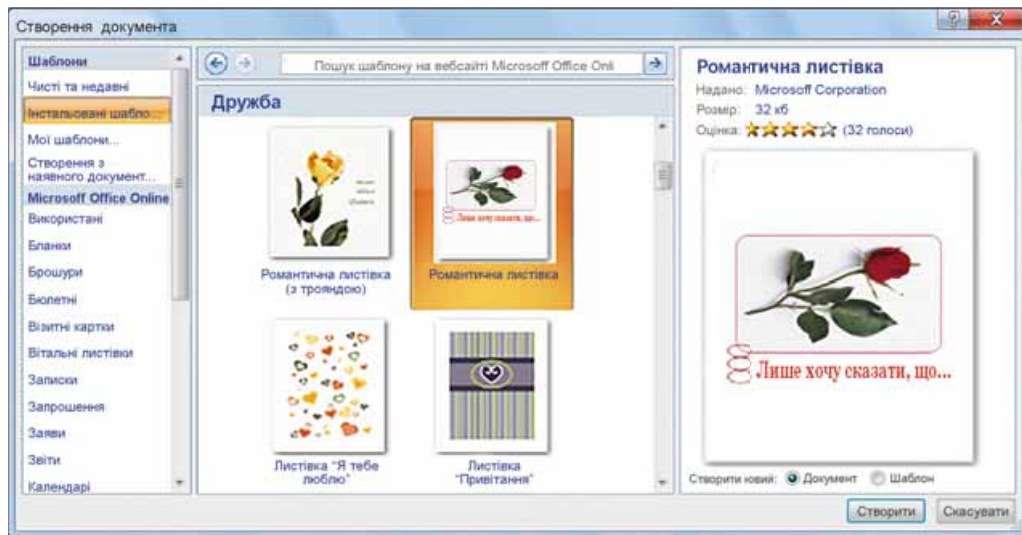


Мал. 3.64. Створення документа на основі інстальованого шаблону

6. Установити в нижній частині поля зразків перемикач **Документ**.
7. Вибрати кнопку **Створити**.
8. Заповнити запропоновані поля потрібними даними.
9. Зберегти документ.

Якщо користувач бажає застосувати один із шаблонів, які розміщені на сайті **Microsoft Office Online**, то слід виконати таку послідовність дій:

1. Відкрити **Головне меню** програми вибором кнопки **Office**.
2. Вибрати команду **Створити** , що відкриває діалогове вікно **Створення документа**.
3. Вибрати у списку ліворуч у розділі **Microsoft Office Online** потрібний тип шаблону (мал. 3.65). Наприклад **Вітальні листівки**.



Мал. 3.65. Створення документа на основі шаблону з Інтернету

4. Дочекатися з'єднання із сервером сайту і відображення списку шаблонів.
5. Вибрати потрібний шаблон.
6. Переглянути структуру та зовнішній вигляд шаблону в полі зразків.
7. Вибрати кнопку **Завантажити**.
8. Дочекатися завантаження шаблону.
9. Заповнити, за потреби, запропоновані поля потрібними даними.
10. Зберегти документ.

Для тих, хто працює з Microsoft Word 2010

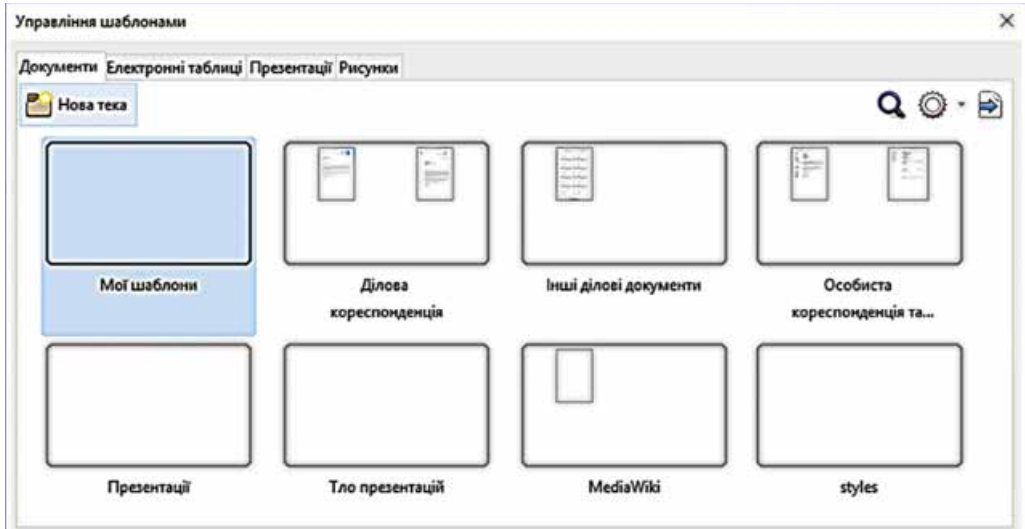
У **Word 2010** для створення документа на основі шаблону потрібно виконати **Файл** ⇒ **Створити** ⇒ **Зразки шаблонів**, вибрати потрібний шаблон, установити перемикач **Документ** та вибрати кнопку **Створити**.

Для вибору шаблону із сайту **Office.com** замість вибору категорії **Зразки шаблонів** потрібно вибрати категорію з розділу **Шаблони Office.com** і завантажити вибраний шаблон.



Для тих, хто працює з LibreOffice

Створення документів на основі шаблонів здійснюється з використанням елементів керування вікна **Управління шаблонами** (*Файл* ⇒ *Шаблони* ⇒ *Управління*) (мал. 3.66).





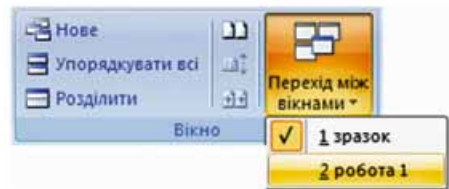
Мал. 3.66. Вікно **Управління шаблонами**

РОБОТА З КІЛЬКОМА ДОКУМЕНТАМИ


Текстовий процесор **Word 2007** надає користувачу можливість відкрити для роботи одночасно кілька документів. Це дає змогу користувачу переглядати документи, порівнювати їх зміст, вставляти фрагменти одного документа в інший, об'єднувати їх зміст тощо.

Для керування відкритими вікнами в програмі існують спеціальні засоби, які розташовані на вкладці **Вигляд** в групі **Вікно** (мал. 3.67). Так, переключення між вікнами можна здійснити, виконавши такий алгоритм:

1. Вибрати кнопку **Перехід між вікнами**  у групі **Вікно** на вкладці **Вигляд**.
2. У списку відкритих вікон вибрати потрібне (поточний документ позначається ).



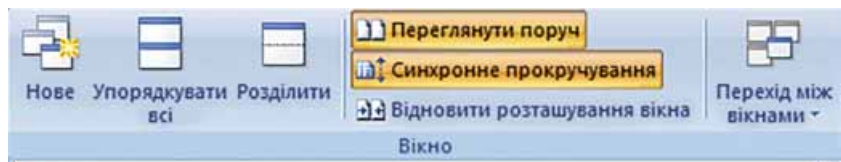
Мал. 3.67. Згорнута група **Вікно**

Коли робота відбувається з двома різними версіями одного і того самого документа, зручно відобразити їх на екрані поруч. Для виконання цієї операції потрібно виконати послідовність дій **Вигляд** ⇒ **Вікно** ⇒ **Переглянути поруч**  і вибрати у списку відкритих документів потрібний. Після цього екран буде розділено вертикально навпіл і в кожній із цих частин




буде відображено вікно одного з указаних документів. Прокручування документів у двох вікнах буде відбуватися синхронно.

Щоб відмінити синхронну прокрутку, потрібно вибрати відповідну кнопку у групі **Вікно** (мал. 3.68).



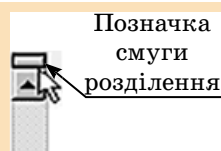
Мал. 3.68. Розгорнута група **Вікно**

Для скасування режиму відображення документів поруч слід повторно вибрати кнопку **Переглянути поруч**  у групі **Вікно**.



Для тих, хто хоче знати більше

Під час опрацювання документа інколи виникає потреба бачити одночасно два різні його фрагменти та переглядати їх незалежно. Для цього зручно розділити вікно документа на дві частини, виконавши послідовність дій **Вигляд** ⇒ **Вікно** ⇒ **Розділити**. На екрані з'явиться смуга розділення вікна, яку слід перетягнути в потрібне місце. Цю ж операцію можна виконати і перетягуванням смуги розділення за допомогою відповідної позначки, яка знаходиться над вертикальною смугою прокрутки (мал. 3.69).



Мал. 3.69.
Позначка смуги розділення вікна



Для повернення до роботи в одному вікні потрібно виконати команду **Зняти поділ** у групі **Вікно** вкладки **Вигляд** або двічі клацнути на смузі розділення вікна.

ФОРМАТИ ФАЙЛІВ ТЕКСТОВИХ ДОКУМЕНТІВ

Стандартним форматом файлів текстових документів, підготовлених у текстовому процесорі **Word 2007** та наступних версій, є формат **DOCX**. Крім того, **Word 2007** забезпечує роботу з документами, збереженими в інших форматах, наприклад **RTF**, **TXT**, **DOC**, **PDF** тощо. Характеристику цих текстових форматів наведено в таблиці 3.12.




Таблиця 3.12

Характеристика найбільш поширених форматів файлів текстових документів у Word 2007

Формат	Стандартний значок	Характеристика
TXT		У файлі зберігається тільки текст з розбиттям на абзаци, без форматування. Може опрацьовуватися різними системами опрацювання текстів.
DOC		У файлі зберігається текст, рисунки, вставлені об'єкти, значення їх властивостей. Є основним форматом для текстових документів, підготовлених у версіях Word до 2003 включно.



Продовження таблиці 3.12

Формат	Стандартний значок	Характеристика
DOCX		У файлі зберігається текст, рисунки, вставлені об'єкти, значення їх властивостей. Забезпечує значне стиснення даних порівняно з іншими форматами (до 70 %). Започаткований у Word 2007 .
RTF		У файлі зберігається текст, рисунки, вставлені об'єкти, значення їх властивостей. Використовується різними системами опрацювання текстів для різних ОС.
PDF		У файлі повністю зберігається форматування документа, редагувати файл неможливо. Для перегляду файлу використовується Adobe Reader .

Для того щоб зберегти текстовий документ не у стандартному форматі, слід у вікні **Збереження документа** відкрити список **Тип файлу** і вибрати потрібний формат.

Для тих, хто працює з LibreOffice

Стандартним форматом файлів текстових документів, підготовлених у текстовому процесорі **LibreOffice Writer**, є формат **ODF** (англ. *Open Document Format for Office Application* – відкритий формат документів для офісних програм) з розширенням імені файлу **ODT**. Крім того, **LibreOffice Writer** забезпечує роботу з документами, збереженими в інших форматах, наприклад **RTF, TXT, DOC, XML** тощо.

Найважливіше в цьому пункті

У великих багатосторінкових документах у **Word 2007** можна створити різні види **посилань** – спеціальні записи, які вказують на деякі об'єкти того або іншого документа. Посилання призначені для логічного зв'язування одних об'єктів документа з іншими. Під час вибору посилання в документі відбувається перехід на вказаний об'єкт.

Інструменти, які призначені для створення посилань, знаходяться на вкладці **Посилання**. З їх використанням можна створити, наприклад, зміст документа (група **Зміст**) та предметний покажчик термінів (група **Покажчик**).

Для створення документів певних типів у текстовому процесорі **Word** можна використовувати **шаблони** – відформатовані певним чином документи-заготовки, які зберігаються в окремих файлах і використовуються як зразок для створення нових документів. Шаблони зберігаються у файлах з розширенням **dotx** та **dotm**.

Для того щоб у текстовому процесорі **Word 2007** створити документ на основі шаблону, потрібно виконати **Office** ⇒ **Створити** ⇒ **Шаблони** ⇒ вибрати потрібний шаблон ⇒ **Створити**.

Текстовий процесор **Word 2007** надає користувачу можливість одночасно працювати з кількома відкритими документами. Це дає змогу користу-

вачу переглядати документи, порівнювати їх зміст, вставляти фрагменти одного документа в інший, об'єднувати їх зміст тощо. Для керування відкритими вікнами у програмі існують спеціальні засоби, які розташовані на вкладці **Вигляд** у групі **Вікно**.

Стандартним форматом файлів текстових документів, підготовлених у текстовому процесорі **Word 2007**, є формат **DOCX**. Крім того, **Word 2007** забезпечує роботу з документами, збереженими в інших форматах, наприклад **RTF**, **TXT**, **DOC**, **PDF** тощо.



Дайте відповіді на запитання

- 1°. Що таке посилання в документі? Для чого вони використовуються?
- 2°. Що таке зміст документа? Для чого він призначений?
- 3°. Як засобами **Word 2007** створити зміст документа?
- 4°. Що таке предметний покажчик? Для чого він використовується?
- 5°. Як створити предметний покажчик у текстовому документі?
- 6°. Що таке шаблон? Для чого його використовують?
- 7°. Які властивості документа може встановлювати шаблон? Які типи шаблонів існують у програмі **Word 2007**?
- 8°. Чим шаблон відрізняється від документа?
- 9°. Яке розширення має файл, у якому зберігається шаблон? Наведіть приклади шаблонів.
- 10°. Які є способи одночасного відображення вікон відкритих документів на екрані? Як їх встановити?
- 11°. У яких форматах може бути збережено файл текстового документа в **Word 2007**? Який формат є стандартним?
- 12°. Як вибрати формат для збереження файлу текстового документа?






Виконайте завдання

- 1°. Відкрийте вказаний учителем документ (наприклад, **Розділ 3\Пункт 3.6\зразок.docx**). Відформатуйте назви структурних елементів документа з використанням стилів заголовків відповідних рівнів. Створіть на початку документа його зміст. Збережіть документ у вашій папці у файлі з іменем **завдання 3.6.1.docx**.
- 2°. Відкрийте вказаний учителем документ (наприклад, **Розділ 3\Пункт 3.6\зразок.docx**). Створіть у кінці документа предметний покажчик для термінів, які виділені в тексті зеленим кольором, та їх означень. Формат покажчика встановіть за власним бажанням. Збережіть документ у вашій папці у файлі з іменем **завдання 3.6.2.docx**.
- 3°. Відкрийте текстовий процесор **Word 2007**. Перегляньте встановлені шаблони документів. З'ясуйте їх кількість, тематику документів, які можна створити на їх основі. Результати запишіть у зошит.
- 4°. Знайдіть на вашому комп'ютері файли шаблонів документів. Запишіть у зошит: у якій папці вони збережені? Скільки файлів-шаблонів? Які їх імена?
- 5°. Створіть лист до оргкомітету конкурсу «Україна має талант» про бажання взяти участь у ньому на основі шаблону *Вишуканий лист*.





Збережіть документ у вашій папці у файлі з іменем завдання **3.6.5.docx**.

-  **6***. Створіть власну візитку, скориставшись відповідним шаблоном із сайту **Microsoft Office Online**. Збережіть документ у вашій папці у файлі з іменем завдання **3.6.6.docx**.
-  **7***. За допомогою **Довідки** з'ясуйте, як створити шаблон на основі існуючого документа. Створіть таким чином власний шаблон оформлення звіту про проходження літньої практики. Збережіть шаблон у колекції.
- 8***. Відкрийте в текстовому процесорі **Word 2007** три текстових файли (наприклад, з папки **Розділ 3\Пункт 3.6\зразок1.docx, зразок2.docx, зразок3.docx**). Опрацюйте переключення між вікнами за допомогою кнопок цих вікон на **Панелі задач** та кнопкою **Перехід між вікнами**. Вставте на початку першого документа перший абзац другого документа, а в кінці першого документа – останній абзац третього документа. Збережіть перший документ у файлі з іменем завдання **3.6.8.docx** у вашій папці.
-  **9***. Відкрийте в текстовому процесорі **Word 2007** два вказаних учителем файли (наприклад, **Розділ 3\Пункт 3.6\зразок1.docx, зразок2.docx**). Відобразіть вікна цих документів поруч. Опрацюйте переміщення по документах при включеному і виключеному режимі **Синхронне прокручування**. Знайдіть відмінності в першому документі в порівнянні з другим і виділіть їх кольоровим фоном. Зніміть режим перегляду документів поруч. Збережіть перший документ у файлі з іменем завдання **3.6.9.docx** у вашій папці.
- 10***. Відкрийте будь-який файл формату **DOCX** (наприклад, **Розділ 3\Пункт 3.6\зразок.docx**). Збережіть його у вашій папці у форматі **DOC** та **TXT**. Порівняйте розміри трьох файлів. Поясніть отриманий результат.

ПРАКТИЧНА РОБОТА № 5

«Структура документа. Автоматизоване створення змісту та покажчиків»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте вказаний учителем файл (наприклад, **Розділ 3\Пункт 3.6\Практична5\зразок.docx**).
2. У режимі перегляду **Структура** впорядкуйте структурні елементи документа у правильному порядку. Змініть за потреби рівень структурних елементів.
3. Відформатуйте назви структурних частин текстового документа з використанням стилів заголовків відповідного рівня.
4. Створіть на початку документа його зміст.
5. Створіть у кінці документа предметний покажчик для всіх слів, які позначено червоним кольором.
6. Збережіть документ у вашій папці у файлі з іменем **практична робота 5.docx**.



Розділ 4. **Опрацювання об'єктів мультимедіа**

У цьому розділі ви дізнаєтеся про:

формати аудіо-
та відеофайлів

сервіси розміщення
аудіо- та відеофайлів
в Інтернеті

програмне забезпечення
для опрацювання
об'єктів мультимедіа

засоби перетворення
аудіо- та відеоформатів

засоби
запису/захоплення
аудіо та відео

створення аудіо-
та відеофрагментів
із застосуванням:

відеоефектів

аудіо- і відеорядів
з узгодженням у часі

переходів між окремими
частинами
відеофрагментів



4.1. ОПРАЦЮВАННЯ ОБ'ЄКТІВ МУЛЬТИМЕДІА



1. Які повідомлення називають мультимедійними?
2. Які програми для опрацювання мультимедійних об'єктів ви знаєте? Для чого призначені програвачі мультимедійних даних?
3. Які пристрої використовуються для створення і опрацювання мультимедійних повідомлень?

ФОРМАТИ АУДІО- ТА ВІДЕОФАЙЛІВ

Раніше ви вже ознайомилися з поняттям типу (формату) файлу. Тип (формат) файлу визначає структуру даних у файлі. Вам відомі різні формати текстових і графічних файлів, файлів-презентацій і електронних таблиць. Аудіо- та відеодані також зберігаються у файлах певних форматів, ці файли, як вам відомо, ще називають мультимедійними. Приклади форматів аудіо- та відеофайлів з описом їх особливостей подано в таблиці 4.1.

Таблиця 4.1

Типи (формати) мультимедійних файлів

Тип (формат)	Опис	Розширення імені
<i>Файли, у яких не використовується стиснення або використовується стиснення без втрати даних</i>		
WAV або WAVE (англ. <i>waveform audio format</i> – аудіоформат типу <i>хвиля</i>)	Формат звукових файлів, що використовується як основний для нестиснутих аудіоданих в операційних системах сімейства Windows	wav
MIDI (англ. <i>Musical Instrument Digital Interface</i> – цифровий інтерфейс музичних інструментів)	Формат файлів, що містять команди для відтворення звуку спеціальним пристроєм або програмою – синтезатором	mid, midi
FLAC (англ. <i>Free Lossless Audio Codec</i> – вільний аудіокодек без втрат)	Формат для стиснення аудіоданих без втрат, що розповсюджується за ліцензією вільного програмного забезпечення	flac
<i>Файли, у яких використовується стиснення із частковою втратою даних</i>		
MP3 , точніше MPEG-1/2/2.5 Layer 3 (англ. <i>Motion Picture Experts Group</i> – експертна група з питань рухомих зображень, <i>Layer 3</i> – третій рівень)	Один з найрозповсюдженіших форматів аудіофайлів. Використовується для передавання звукових даних у глобальних мережах, у файлообмінних системах	mp3
MP4 або MPEG-4 Part 14	Файли цього формату зазвичай використовуються для цифрового телебачення, зберігання відеофільмів тощо	mp4

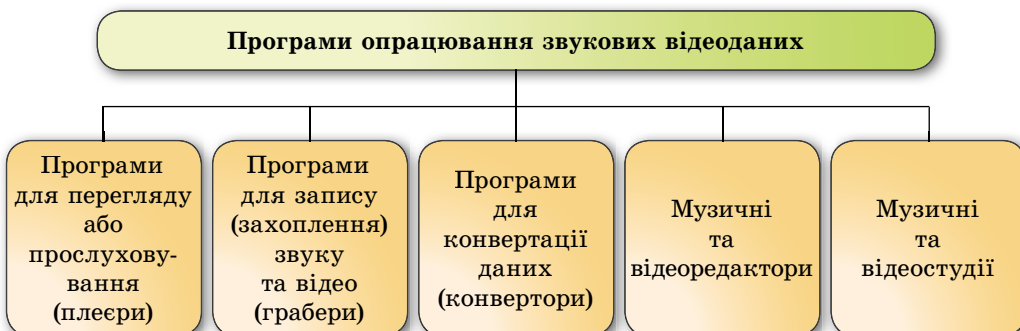
Тип (формат)	Опис	Розширення імені
WMA/WMV (англ. <i>Windows Media Audio/Video</i>)	Формат файлів для зберігання і відтворення аудіо/відеоданих, розроблений корпорацією Microsoft і стандартно використовується в ОС Windows	wma, wmv
QuickTime (англ. <i>Quick Time</i> – швидкий час)	Формат файлів, що базується на технології збереження і відтворення відео- та звукових даних, яку було розроблено корпорацією Apple	mov
Flash відео	Формат файлів, що використовуються зазвичай для розповсюдження відео в Інтернеті. Характеризується високим ступенем стиснення даних	flv

У зв'язку з тим що більшість аудіо- та відеоданих стискаються з використанням різноманітних алгоритмів (виконується кодування даних), для їхнього відтворення потрібно виконати декодування. Програми, що містять алгоритми кодування і декодування мультимедійних даних, називають **кодеками**. Кодеки входять до складу програм, що опрацюовують аудіо- чи відеофайли, а також можуть випускатися як додатковий набір програм, наприклад **K-Lite Codec**.

ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ОПРАЦЮВАННЯ ОБ'ЄКТІВ МУЛЬТИМЕДІА

Ви вже ознайомилися з певними програмами для опрацювання мультимедійних даних – аудіо- та відеопрогравачами. Більш загальну класифікацію таких програм за призначенням подано на малюнку 4.1.

Розглянемо детальніше призначення цих програм та алгоритми роботи з ними.



Мал. 4.1. Схема класифікації програм для опрацювання аудіо- і відеоданих



ЗАПИС (ЗАХОПЛЕННЯ) АУДІО ТА ВІДЕО, СТВОРЕННЯ АУДІО-, ВІДЕОФРАГМЕНТІВ

Для збереження аудіо- і відеоданих, що вводяться з використанням пристроїв введення/виведення мультимедійних даних, призначені спеціальні *програми запису (захоплення) звуку та відео*, які ще називають **граберами** (англ. *grabber* – той, хто захоплює, хапуга, грабіжник), або **рекордерами** (англ. *record* – запис). Дані будуть збережені у відповідних файлах.

Використовуючи ці програми, можна записати звуковий коментар до презентації, відеоконференцію в **Skype**, веб-трансляцію спортивного змагання тощо.

Прикладами програм для захоплення звуку є стандартна програма ОС **Windows Звукозаписувач** (мал. 4.2), **Audacity**, **Audiograbber**, **Streamripper**, **AML Easy Audio Recorder**, **Free Sound Recorder** та ін. Можна також використати Інтернет-сервіси для запису та збереження звуку, наприклад,

<http://vocaroo.com/>;

<http://online-voice-recorder.com>.

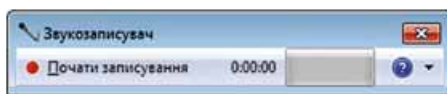
Для того щоб записати звукове повідомлення з використанням мікрофона, потрібно:

1. Підключити мікрофон або скористатися вбудованим (у мобільних комп'ютерах).
2. Запустити програму запису звуку (наприклад, **Пуск** ⇒ **Усі програми Стандартні** ⇒ **Звукозаписувач**).
3. Розпочати запис звуку вибором відповідної кнопки (наприклад, **Почати записування** (мал. 4.3)).
4. Проговорити в мікрофон потрібний текст.
5. Зупинити запис вибором відповідної кнопки (наприклад, **Зупинити записування**).
6. Зберегти звукозапис в аудіофайлі в обраній вами папці.

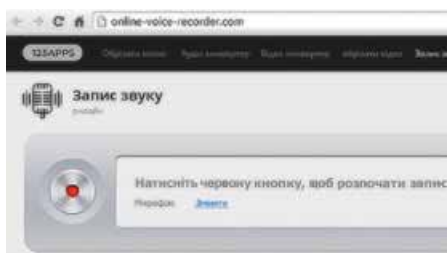
У подальшому файл, який ви створили, можна відкрити і прослухати в середовищі програвача або редактора аудіофайлів.

Під час записування відео може виконуватися запис відео й звуку або тільки відео. Відповідно, до комп'ютера слід під'єднати пристрої введення відео- і звукових даних або тільки відеоданих.

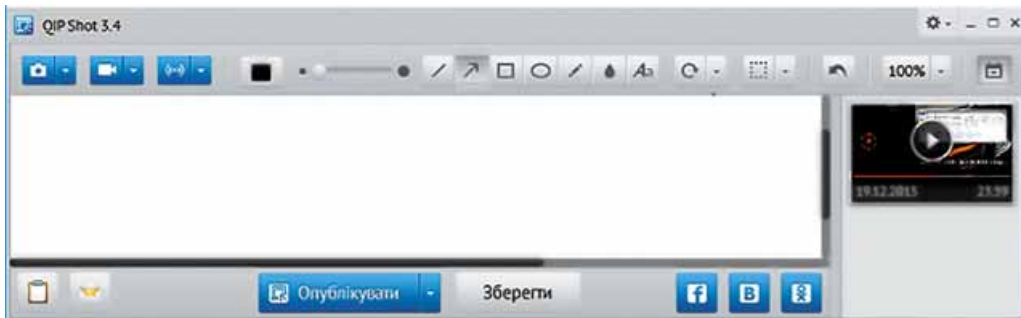
Запис (захоплення) відео можливо виконувати з веб-камери, з вікна програвача відеофайлів чи іншої програми, з веб-сторінки. В усіх випадках потрібно використати відповідну програму. Програмами для запису відео є **CamStudio**, **QIP Shot**, **Free Screen Video Recorder**, **Ezvid** та ін. Наприклад, для запису відео з використанням програми **QIP Shot** (мал. 4.4) потрібно:




Мал. 4.2. Вікно **Звукозаписувача**

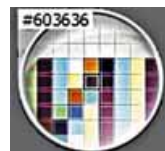


Мал. 4.3. Сервіс Інтернету для запису звуку і відео



Мал. 4.4. Вікно програми QIP Shot

1. Відкрити вікно програми, відео з якої потрібно записати, або відповідний сайт.
2. Запустити програму **QIP Shot**.
3. Вибрати режим запису у списку кнопки переключення режимів зйомки , наприклад режим запису прямокутної області екрана.
4. Указати, використовуючи спеціальний курсор (мал. 4.5) програми, прямокутну область, у якій буде виконано захоплення відео.
5. Розпочати програвання відео у програмі або на сайті-джерелі.
6. Увімкнути запис відео вибором кнопки **Початок** (мал. 4.6).
7. Для завершення запису відео вибрати кнопку **Стоп**.
8. Для збереження створеного відеофрагменту у файлі вибрати кнопку **Зберегти**.
9. Указати папку, в яку буде збережено файл, та ім'я файлу.



Мал. 4.5

Чемпіон України проведе перший спаринг на зборі в Іспанії



дебютний спаринг на першому тренувальному зборі в Іспанії. Суперником чемпіона України виступить представник третьої німецької Бундеслиги дрезденське "Динамо".

Мал. 4.6. Вікно програми QIP Shot у режимі запису прямокутної області




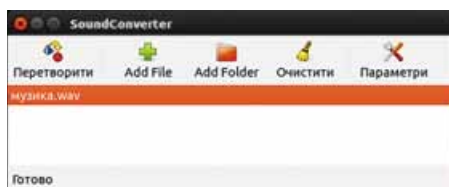
Зазначений алгоритм з невеликими змінами може бути використаний під час запису відео з використанням й інших програм.

ЗАСОБИ ПЕРЕТВОРЕННЯ АУДІО- ТА ВІДЕОФОРМАТІВ

Залежно від потреб користувача щодо використання аудіо- та відеоданих часто виникає потреба змінити формат файлу, у який записано ці дані. Наприклад, якщо ваш мультимедійний програвач не може відтворити потрібний файл. Для таких випадків використовують спеціальні програми – конвертори. Для конвертації аудіофайлів використовують такі програми, як **HAMSTER Free Audio Converter**, **SoundConverter XRECODE II**, **TAudioConverter**, **AudioConverter Studio** та ін.

Для конвертації аудіофайлів, наприклад у програмі **SoundConverter** (мал. 4.7), після запуску програми потрібно:

1. Вибрати кнопку  (англ. *Add File* – додати файл) і у вікні, що відкрилося, вибрати файл, який потрібно конвертувати.
2. Вибрати кнопку **Параметри** та вказати у списку **Format** формат файлу, у який потрібно конвертувати файл.
3. Почати процес конвертації вибором кнопки **Перетворити**.



Мал. 4.7. Вікно програми **SoundConverter** з підготовленим до конвертації файлом

Сконвертований файл за замовчуванням буде записано в ту саму папку, де знаходився початковий файл. Установити іншу папку для запису можна у вікні, що відкривається після вибору кнопки **Параметри**.

Для конвертації відео існує багато різноманітних програм, що розповсюджуються за різними видами ліцензій. Наприклад, **HAMSTER Free Video Converter**, **AVS Video Converter**, **Any Video Converter Free**, **SUPER** та ін.

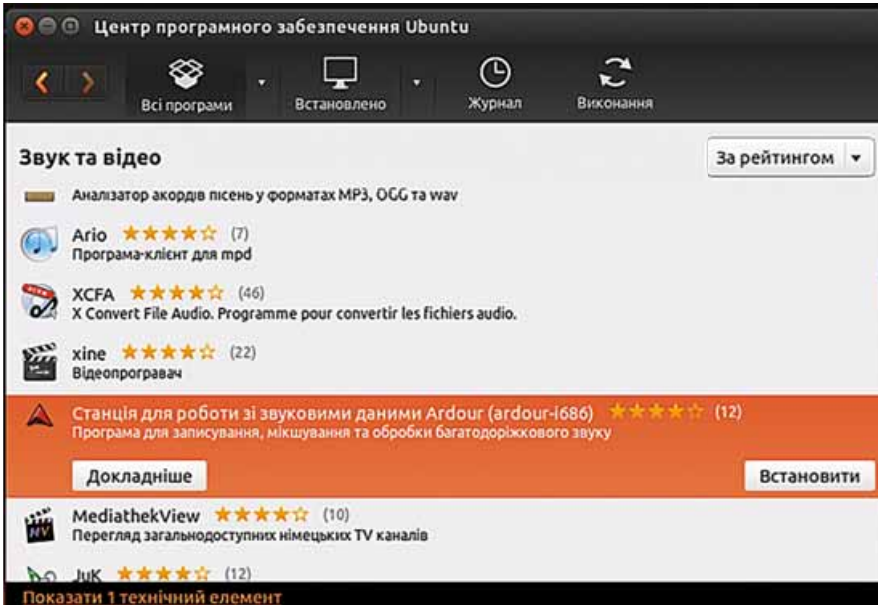
Розглянемо послідовність конвертації відеофайлів у програмі **HAMSTER Free Video Converter**. Після запуску програми потрібно:

1. Додати файли для конвертації, вибравши кнопку **Додати файли** та вибравши у вікні, що відкрилося, потрібні файли.
2. Вибрати кнопку **Далі**.
3. Указати формат файлу, в який будете конвертувати, та за потреби – значення властивостей цього формату (мал. 4.8).
4. Вибрати кнопку **Конвертувати**.
5. Указати папку для запису конвертованого файлу.



Мал. 4.8. Вікно програми **HAMSTER Free Video Converter**

Також існують веб-версії конверторів як аудіо, так і відео, наприклад, на сайтах <http://audio.online-convert.com>, <http://benderconverter.com>, <https://online-audio-converter.com>, <http://www.onlinevideoconverter.com>, <http://www.online-convert.com> та ін.



Мал. 4.9. Вікно Центру програмного забезпечення Ubuntu



Для тих, хто працює з Linux

Для встановлення програм опрацювання аудіо- та відеоданих, якщо вони не були встановлені під час інсталяції ОС, можна використати **Центр програмного забезпечення Ubuntu**. Для цього у списку **Всі програми** слід установити відображення програм роботи зі звуком і відео, вибрати потрібну програму та кнопку **Встановити** (мал. 4.9).

Найважливіше в цьому пункті

Аудіо- та відеодані зберігаються у файлах різних форматів. Файли аудіо мають розширення імені – **wav**, **mid**, **flac**, **mp3** та ін. Файли відео мають розширення імені – **mp4**, **wmv**, **mov**, **flv** та ін.

Програми, що містять алгоритми кодування і декодування мультимедійних даних, називають **кодеками**.

Програми, що опрацюють відео та аудіо, залежно від призначення поділяють на програми для перегляду або прослуховування (**плеєри**), програми для запису (захоплення) звуку і відео (**грабери**, **рекордери**), програми для конвертації даних (**конвертори**), музичні та відеоредактори, музичні та відеостудії.

**Дайте відповіді на запитання**

- 1°. Які типи (формати) аудіофайлів ви знаєте? Які з них не передбачають часткової втрати даних?
- 2°. Які типи (формати) відеофайлів ви знаєте?
- 3°. Що таке кодеки? Для чого вони призначені?
- 4°. Які види програм для опрацювання мультимедійних даних ви знаєте?
- 5°. Для чого призначені програми запису аудіо та відео? Наведіть приклади таких програм.
- 6°. Яка послідовність дій під час запису аудіоданих?
- 7*. Чим відрізняються дії під час запису аудіоданих від дій під час захоплення відео?
- 8°. З яких джерел можна виконати захоплення відео?
- 9°. Яка послідовність дій під час конвертації аудіо- та відеофайлів?

**Виконайте завдання**

- 1°. Запишіть у власному виконанні декламацію вірша Т. Шевченка «Думи мої, думи мої...» (Розділ 4\Завдання 4.1\Думи мої.doc) з використанням програми **Звукозапис**. Збережіть запис у вашій папці у файлі з іменем завдання 4.1.1.wav.
- 2°. Використовуючи Інтернет-ресурс для запису звуку <http://online-voice-recorder.com>, запишіть у власному виконанні декламацію вірша Володимира Сосюри «Осінь» («Облітають квіти...») (Розділ 4\Завдання 4.1\Облітають квіти.doc). Збережіть запис у папці **Документи** у файлі з іменем завдання 4.1.2.mp3.
- 3°. Виконайте запис відео з веб-камери вашого комп'ютера тривалістю 30 с з використанням програми захоплення відео **QIP Shot**. Збережіть запис у папці **Документи** у файлі з іменем завдання 4.1.3.mp4.
- 4°. Виконайте запис відео з вікна відеопрогравача під час відтворення ним файлу **відео 2.avi** (Розділ 4\Завдання 4.1\відео 2.avi) з використанням програми захоплення відео **QIP Shot**. Збережіть запис у папці **Документи** у файлі з іменем завдання 4.1.4.wmv.
- 5*. Виконайте конвертування двох перших файлів формату **MP3** з папки, указані вчителем, наприклад **Розділ 4\Завдання 4.1\Аудіо**, у формат **WMA**, використовуючи один з конверторів звукових файлів. Збережіть ці файли в папці **Завдання 4.1.5**, яку створіть у вашій папці. Порівняйте розміри початкових та отриманих файлів. Чим ви поясните різницю у їх розмірах?
- 6*. Виконайте конвертування двох перших файлів формату **MOV** з папки **Розділ 4\Завдання 4.1\Відео** у формат **WMV**, використовуючи один з конверторів відеофайлів. Збережіть ці файли в папці **Завдання 4.1.6**, яку створіть у папці **Документи**. Порівняйте розміри початкових та отриманих файлів. Чим ви поясните різницю у їх розмірах?



4.2. РОЗРОБКА І СТВОРЕННЯ ВІДЕОФІЛЬМУ



1. Які формати аудіо- та відеофайлів ви знаєте?
2. Які види програм для роботи з мультимедійними даними ви знаєте?
3. Які дії над мультимедійними даними можна виконати з використанням плеєрів, граберів, конверторів?

ПРОГРАМИ ДЛЯ РЕДАГУВАННЯ АУДІО- ТА ВІДЕОДАНИХ

У попередньому пункті ми не розглянули детально два види програм для роботи з мультимедійними даними – музичні та відеоредактори, а також музичні та відеостудії. Ці програми призначені для редагування звукових та відеоданих. Використовуючи їх, можна вставляти, видаляти, копіювати, змінювати тривалість аудіо- чи відеофрагментів, об'єднувати фрагменти, вставляти у відео текстові та графічні об'єкти тощо.

Відмінність між названими програмами в тому, що мультимедійні студії – це багатофункціональні програми. Вони мають розширений набір функцій і використовуються у професійній діяльності на студіях звукозапису, кіно- й анімаційних студіях для створення звукозаписів, аудіо- і відеоконпозицій, анімації, кіно- і відеофільмів, рекламних роликів, відеокліпів тощо. Прикладами програм-музичних студій є **Linux MultiMedia Studio**, **PsycleDelics Psycle**, **Steinberg Cubase**, **Cakewalk Sonar**, **Adobe Audition** та ін. Програмами відеостудій є **Pinnacle STUDIO**, **Kino**, **Adobe Premiere Pro**, **Avid Media Composer** та ін. Зазвичай програми-студії є комерційними програмами.

Для створення і редагування простих звукових та відеофайлів використовують різноманітні мультимедійні редактори. Наприклад, для редагування аудіоданих можна використати **Audacity**, **Free Audio Editor**, **Wave Editor**, а для відеоданих – **Кіностудію Windows**, **Open Shot**, **Virtual Dub**, **ZS4 Video Editor** тощо.

КІНОСТУДІЯ WINDOWS

Кіностудія Windows є новою версією редактора відеоданих **Windows Movie Maker**. Безкоштовну версію програми користувачі **Windows 7** та **Windows 10** можуть завантажити із сайту **Microsoft**. Для попередніх версій ОС **Windows** можна використати **Windows Movie Maker**, що за замовчуванням включається до складу стандартних програм під час інсталяції.

Кіностудія Windows призначена для створення відеофільмів, що можуть містити відеофрагменти, фотографії, звуковий супровід (дикторський текст), різноманітні написи (титри). Створене відео можна опублікувати в Інтернеті або зберегти у файлі одного з можливих форматів.

Загальний вигляд вікна програми **Кіностудія Windows** подано на малюнку 4.10.

ЗАГАЛЬНІ ПОНЯТТЯ ПРО ВІДЕОФІЛЬМ

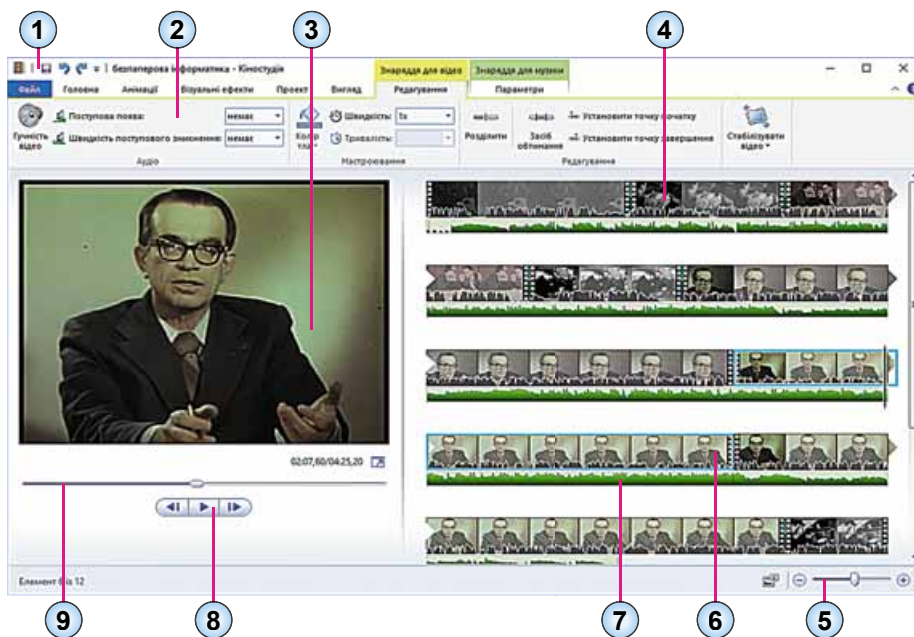
Під **відеофільмом**, як і **кінофільмом** (англ. *cine-film* – кіноплівка, кінофільм), ми здебільшого розуміємо твори кіномистецтва. З розповсюджен-



ням різноманітних пристроїв для запису відеоданих створити відеофільм може будь-який користувач. Такі відеофільми, зазвичай, призначаються для перегляду вдома або в невеликих групах, на відміну від перегляду кінофільмів у кінотеатрах. Для невеликих за розміром кіно- та відеофільмів використовують термін **кліп** (англ. *clip* – стискувати, обрізати), або **відеокліп**.

Відеофільм складається з **кадрів** – окремих фрагментів рухомого (відео) або нерухомого (фото) зображення. Кадри утворюють відеоряд і розміщуються у відеоредакторі на спеціальній відеодоріжці. Відеофільм, зазвичай, містить також такі об'єкти:

- **заголовок** – один або кілька текстових об'єктів, що з'являються на початку відеофільму;
- **титри** – текстові об'єкти, що з'являються або поверх відео та використовуються для пояснення зображення, або в кінці відео та містять відомості про авторів відеофільму;
- **звуківий супровід** може бути кількох видів:
 - записаний разом з відео;
 - дикторський – коментар або пояснення до відео;
 - музичний – музичний твір, що використовується як фоновий під час перегляду відео;



- | | |
|---|-----------------------------------|
| 1. Панель швидкого запуску | 6. Відеоряд (відеодоріжка) |
| 2. Стрічка | 7. Звуківий ряд (звукова доріжка) |
| 3. Вікно перегляду | 8. Кнопки керування переглядом |
| 4. Область монтажу | 9. Повзунок перегляду |
| 5. Повзунок змінення масштабу області монтажу | |

Мал. 4.10. Вікно програми **Кіностудія Windows**



- **візуальні ефекти** – використовуються для зміни зовнішнього викладу зображення, наприклад, переходу до чорно-білого зображення або досягнення ефекту старого кіно;
- **ефекти переходу** – ефекти, що використовуються для переходу від одного кадру до іншого.

Візуальні ефекти та ефекти переходу відображаються, зазвичай, на відеодоріжці. Звуковий супровід може розміщуватися на кількох окремих звукових доріжках. У **Кіностудії Windows** звук, що записано разом з відео, відображається на відеодоріжці.

Під час створення відеофільмів варто дотримуватися певної послідовності:

- **на першому етапі** розробляється сценарій – детальний план послідовності окремих кадрів фільму і розміщення окремих об'єктів. Що детальніший буде сценарій, то простіше буде створювати відеофільм;
- **на другому етапі** виконується підготовка потрібних матеріалів – відбувається зйомка всіх епізодів відповідно до сценарію, добирається музика, фото, малюнки, записується звуковий супровід;
- **на третьому етапі** виконується монтаж – опрацювання та структурування окремих кадрів для отримання цілісного твору – відеофільму. У ході монтажу відзнятий матеріал редагується, уточнюється сценарій, накладається звук і титри – текстові коментарі до кадрів тощо.

РОЗРОБКА СЦЕНАРІЮ ВІДЕОФІЛЬМУ

Розглянемо процес розробки сценарію відеофільму на прикладі створення відеофільму «Безпаперова інформатика» про реалізацію ідей В.М. Глушкова відносно використання комп'ютерних засобів у наші дні. Він говорив, що на початок двадцять першого століття основна маса інформації буде зберігатися в безпаперовому вигляді – у пам'яті ЕОМ.

Для реалізації цих ідей під керівництвом В.М. Глушкова було розроблено серію ЕОМ **МИР** (рос. *Машина для Инженерных Расчётов* – машина для інженерних розрахунків) – прообрази персональних комп'ютерів. Основна ідея таких комп'ютерів – їх широке використання всіма верствами населення, не залежно від рівня освіти і віку.

Враховуючи мету нашого відеофрагменту – демонстрація на уроці інформатики в ході розповіді про академіка В.М. Глушкова, складемо сценарій. Сценарій складається зі сценарних кадрів – опису змістових фрагментів відеофільму та подій, що в них відбуваються.

1. Кадр 1. Титульна сторінка:

- заголовок – *Безпаперова інформатика*;
- колір тла – *світло-зелений*, колір літер – *темно-зелений*;
- тривалість показу – *7 с*;
- звук – *загальний фоновий звук з файлу*;
- візуальний ефект – *Кінематографічні згасання, зліва*;
- ефект переходу на наступний кадр – *розчинення, мозаїка*.

2. Кадр 2. Фотографія В.М. Глушкова:

- фото з файлу;
- тривалість показу – *6 с*;



- звук – дикторський текст, що починається на цьому кадрі і закінчується на наступному; тривалість звучання – визначається швидкістю відтворення тексту;
 - візуальний ефект – відсутній;
 - ефект переходу на наступний кадр – розчинення, мозаїка;
 - дикторський коментар – Директор Інституту кібернетики Академії наук України академік В.М. Глушков одним із завдань використання комп'ютерної техніки вважав перехід до безпаперової інформатики. Він говорив, що на початок двадцять першого століття основні обсяги інформації будуть зберігатися в безпаперовому вигляді – у пам'яті ЕОМ. І людина, яка не буде вмінати працювати з комп'ютером і використовувати цю інформацію, буде подібна до того, хто на початку ХХ століття не вмів ані писати, ані читати.
- 3. Кадр 3. Відео за участі В.М. Глушкова:**
- відео з файлу;
 - тривалість показу – завершується одночасно з дикторським текстом;
 - звук – загальний фоновий звук з файлу;
 - візуальний ефект – відсутній.
- 4. Кадр 4. Заголовок:**
- текст – *Послухаємо самого Віктора Михайловича Глушкова;*
 - колір тла – світло-зелений, колір літер – темно-зелений;
 - тривалість показу – 7 с;
 - звук – загальний фоновий звук з файлу;
 - візуальний ефект – Кінематографічні згасання, зліва;
 - ефект переходу на наступний кадр – розчинення, мозаїка.
- 5. Кадр 5. Відеорозповіді В.М. Глушкова про безпаперову інформатику:**
- відео з файлу;
 - тривалість показу – 48 с;
 - звук – звук з файлу відео;
 - візуальний ефект – відсутній;
 - ефект переходу на наступний кадр – розчинення, мозаїка.
- 6. Кадр 6. Фотографія В.М. Глушкова разом з розробниками комп'ютера МИР-1:**
- фото з файлу;
 - тривалість показу – 6 с;
 - звук – загальний фоновий звук з файлу;
 - підпис – *Для того щоб кожна людина могла працювати з ЕОМ, потрібно перейти до використання малих комп'ютерів – В.М. Глушков висуває ідею створення комп'ютерів МИР;*
 - колір літер – помаранчевий, колір фону – темно-зелений;
 - візуальний ефект – відсутній;
 - ефект переходу на наступний кадр – розчинення, мозаїка.
- 7. Кадр 7. Фотографія комп'ютера МИР-1:**
- фото з файлу;
 - підпис – *Комп'ютер МИР мав малі розміри і зручний механізм введення-виведення даних – електричну друкарську машинку;*



- усі інші значення властивостей аналогічні до кадра 6.
8. Кадр 8. **Фотографія комп'ютера МИР-2:**
- фото з файлу;
 - підпис – *Комп'ютер МИР-2 мав додаткову систему введення даних – світлове перо;*
 - усі інші значення властивостей аналогічні до кадра 6.
9. Кадр 9. **Фотографія комп'ютера МИР-3:**
- фото з файлу;
 - підпис – *Комп'ютер МИР-3;*
 - усі інші значення властивостей аналогічні до кадра 6.
10. Кадр 10. **Заключні титри:**
- титри – *Шкільна відеостудія «Майбутнє»;*
 - оператор – *Михайленко Степан;*
 - автор тексту – *Коваленко Оксана;*
 - монтаж – *Іваненко Микола;*
 - колір тла – *жовто-зелений*, колір літер – *білий;*
 - тривалість показу – *10 с.*

ДОБІРКА МАТЕРІАЛІВ ДО ФІЛЬМУ

Використовуючи матеріали Інтернету, друковані матеріали, доберемо файли для використання у відеофільмі. Їх перелік подано в таблиці 4.2.

Таблиця 4.2

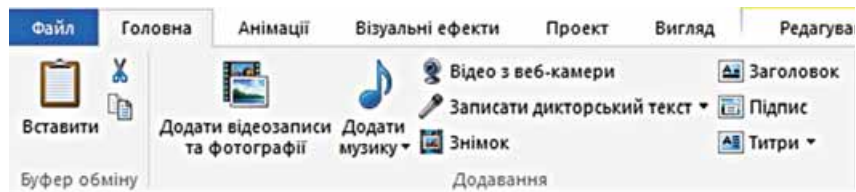
Файли для включення до відеофільму

Ім'я файлу	Папка розміщення	Тривалість, с	Номер кадру
Глушков відео 0.wmv	Розділ 4\Пункт 4.2\Відео	53	3
Глушков відео 1.wmv	Розділ 4\Пункт 4.2\Відео	48	5
Glushkov.jpg	Розділ 4\Пункт 4.2\Фото		2
Glushkov_MIR-1.jpg	Розділ 4\Пункт 4.2\Фото		6
MIR-1.jpg	Розділ 4\Пункт 4.2\Фото		7
MIR-2.jpg	Розділ 4\Пункт 4.2\Фото		8
MIR-3.jpg	Розділ 4\Пункт 4.2\Фото		9
Physical.wma	Розділ 4\Пункт 4.2\Аудіо	220	1–9, крім 5

СТВОРЕННЯ ВІДЕОФІЛЬМУ

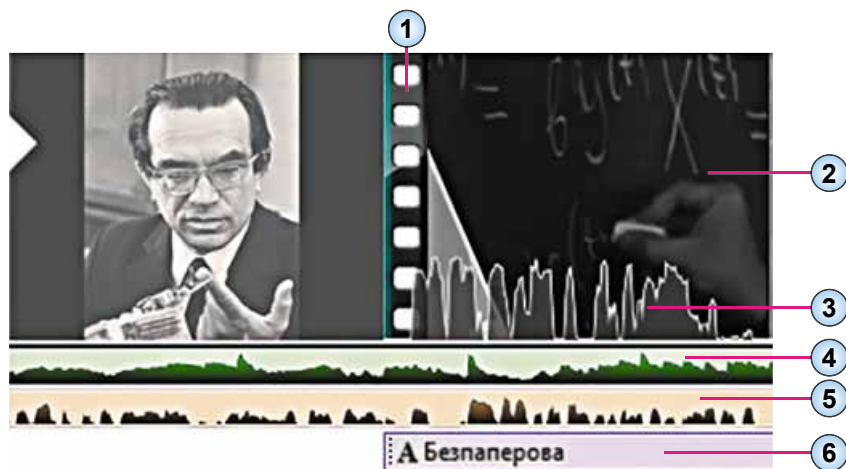
Щоб створити відеофільм з використанням Кіностудії Windows, потрібно:

1. Запустити програму.
2. Зберегти проект відеофільму – **Файл** ⇒ **Зберегти проект**. За замовчуванням програма зберігає файл проекту у файлі з розширенням імені **wlmp**.
3. Уставити зображення, відео- та аудіоматеріали з файлів у послідовності, визначеній за сценарієм, використавши для цього елементи керування групи **Додавання** вкладки **Головна** (мал. 4.11).



Мал. 4.11. Елементи керування групи **Додавання**

4. Додати заголовки фільму, титри, підписи відповідно до сценарію, використавши для цього елементи керування групи **Додавання** вкладки **Головна**.
5. Відредагувати об'єкти фільму, використовуючи елементи керування тимчасових вкладок **Знаряддя для відео**, **Знаряддя для музики**, **Знаряддя для дикторських текстів**, **Знаряддя для текстів**. При цьому слід зважати на те, що в області монтажу відео- і фотоматеріали вставляються у відеоряд (мал. 4.12, 2), звук з файлу – у звуковий ряд (мал. 4.12, 4), звук дикторських коментарів – ще в один звуковий ряд (мал. 4.12, 5), а титри, заголовки і підписи – в окремий ряд із текстом (мал. 4.12, 6). Якщо відео має звук, то звукова доріжка відображається в нижній частині відеоряду (мал. 4.12, 3).
6. Узгодити відеоряд з іншими рядами за часом перегляду, за гучністю звуку тощо, використовуючи переміщення об'єктів по доріжці, перетягуванням місця початку і закінчення об'єкта тощо.
7. За потреби вставити ефекти, що будуть накладатися на окремі кадри (елементи керування вкладки **Візуальні ефекти**).
8. Уставити ефекти переходу між окремими кадрами відеофільму (елементи керування вкладки **Анімація**).
9. Зберегти файл відеофільму – **Файл** ⇒ **Зберегти фільм**.



- | | |
|------------------------------------|--|
| 1. Позначення межі кадрів | 4. Звуковий ряд звукового файлу |
| 2. Відеоряд | 5. Звуковий ряд дикторського коментаря |
| 3. Звукова доріжка відео зі звуком | 6. Ряд тексту |

Мал. 4.12. Об'єкти **Області монтажу**

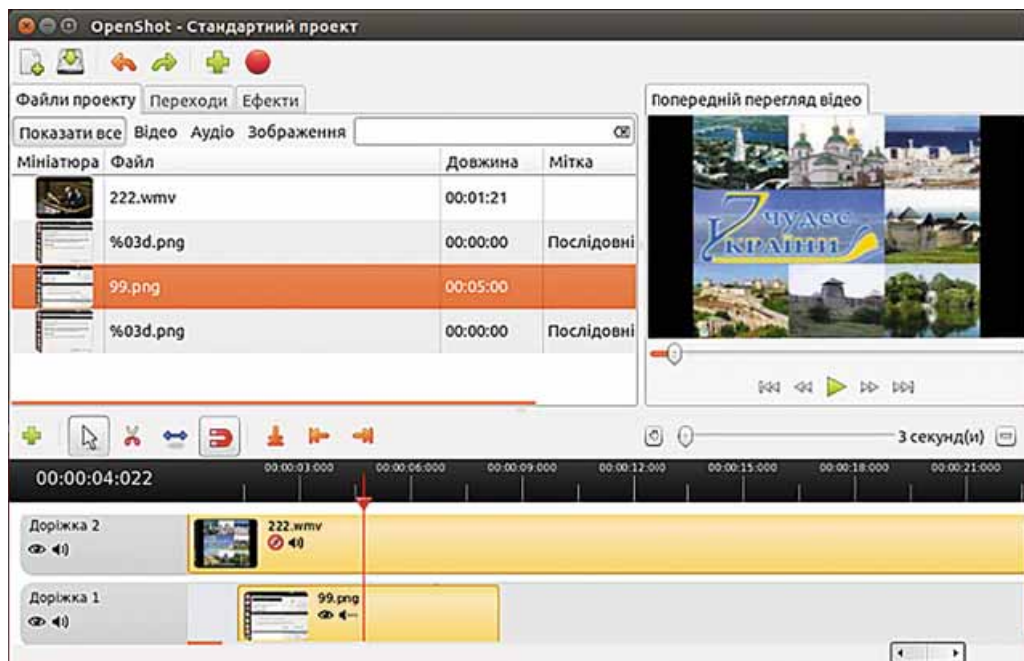


У ході монтажу відеофільму проект слід періодично зберігати. Якщо редагування відеофільму не завершено, то для його продовження слід відкрити збережений файл проекту. Наявність збереженого файлу проекту також дає змогу користувачеві створити кілька варіантів відеофільму з різними ефектами і налаштуваннями.



Для тих, хто працює з Linux

У **Linux** для редагування відео можна використати програму **OpenShot** (див. мал.). За своїм інтерфейсом і основними операціями вона багато в чому подібна до програми **Кіностудія Windows**, описаної в цьому пункті.



Вікно програми **OpenShot**



Найважливіше в цьому пункті

Музичні та відеоредактори, а також музичні та відеостудії – це програми, призначені для редагування звукових та відеоданих. Використовуючи їх, можна вставляти, видаляти, копіювати, змінювати тривалість аудіо- чи відеофрагментів, об'єднувати фрагменти, вставляти у відео текстові та графічні об'єкти тощо.

Відмінність між названими програмами в тому, що мультимедійні студії мають розширений набір функцій і використовуються у професійній діяльності на студіях звукозапису, кіно- й анімаційних студіях.

Кіностудія Windows призначена для створення відеофільмів, що можуть містити відеофрагменти, фотографії, звуковий супровід (дикторський текст), різноманітні написи (титри). Створене відео можна опублікувати в Інтернеті або зберегти у файлі одного з можливих форматів.



Під час створення відеофільмів варто дотримуватися такої послідовності:

- **на першому етапі** розробляється сценарій;
- **на другому етапі** виконується підготовка аудіо-, фото та відеоматеріалів;
- **на третьому етапі** виконується монтаж відеофільму.



Дайте відповіді на запитання

- 1°. Для чого призначені музичні та відеоредактори?
- 2°. У чому відмінність між відеоредакторами і відеостудіями?
- 3°. Які музичні редактори ви знаєте?
- 4°. Які відеоредактори ви знаєте?
- 5°. Для чого призначена **Кіностудія Windows**?
- 6°. Опишіть за малюнком 4.10 основні об'єкти вікна програми **Кіностудія Windows**.
- 7°. Для чого призначена **Область монтажу** в програмі **Кіностудія Windows**?
- 8°. Яка послідовність створення відеофільму?
- 9°. Що таке сценарій відеофільму, для чого він створюється?
- 10°. Як вставити до проекту відеофільму відео або фото з файлу?
- 11°. Як записати дикторський текст?
- 12*. У чому сутність узгодження відео та звуку? Як це виконати в **Кіностудії Windows**?



Виконайте завдання

- 1°. Складіть сценарій і створіть за ним в обраному вами відеоредакторі відеофільм про **Леся Українку** до уроків української літератури. У відеофільмі використайте фрагмент з фільму «Великі українці. Леся Українка» і фотографії поетеси та пам'ятників **Лесі Українці** з папки **Розділ 4\Пункт 4.2\Завдання 4.2.1\Українка**. Підготуйте і включіть до відеофільму титульний слайд та титри супроводу. Самостійно доберіть музику, яка повинна звучати протягом усього відеофільму. Збережіть відеофільм у вашій папці у файлі з іменем **завдання 4.2.1.wmv**.
- 2°. Створіть в обраному вами відеоредакторі відеофільм про визначного мандрівника **Христофора Колумба**, використовуючи матеріал підручника **Всесвітньої історії** для 8-го класу та матеріали, розміщені в папках **Розділ 4\Пункт 4.2\Завдання 4.2.2\Колумб**. Дикторський текст і музичний супровід доберіть самостійно. Збережіть відеофільм у вашій папці у файлі з іменем **завдання 4.2.2.wmv**.
- 3°. Розробіть сценарій та створіть в обраному вами відеоредакторі відеофільм про визначного українського письменника і кінорежисера **Олександра Довженка** для демонстрації на уроці української літератури. Використайте матеріали підручника для 8-го класу з української літератури та матеріали, розміщені в папках **Розділ 4\Пункт 4.2\Завдання 4.2.3\Довженко**. Дикторський текст і музичний супровід доберіть самостійно. Збережіть відеофільм у папці **Документи** у файлі з іменем **завдання 4.2.3.wmv**.





- 4*. Створіть в обраному вами відеоредакторі відеофільм, що буде ілюструвати один з фізичних процесів з курсу фізики 8-го класу. Сценарій і матеріали доберіть самостійно.
- 5*. Створіть в обраному вами відеоредакторі відеофільм про ваш клас або визначну шкільну подію. Сценарій і матеріали доберіть самостійно.

ПРАКТИЧНА РОБОТА № 6

«Створення відеокліпа. Додавання відеоефектів, налаштування часових параметрів аудіо- та відеоряду»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Створіть, використовуючи програму відеоредактора, відеокліп з розповіддю про термометри тривалістю до 2,5 хв.
2. Розробіть сценарій відеокліпа, передбачивши:
 - титульну сторінку з назвою відеокліпа, наприклад **Термометри**;
 - огляд різних видів термометрів, використавши для цього фотографії (4–6 фото) з папки **Розділ 4\Практична 6\Фото**. До кожного фото додайте підписи з назвою типу термометра, що відповідає імені файлу. Тривалість демонстрації кожного з фото – не більше 5 с;
 - розповідь про термометр Галілея, використавши для цього 2–3 фотографії з папки **Розділ 4\Практична 6\Фото** та відео з папки **Розділ 4\Практична 6\Відео**. До фотографій додайте підписи, а до відеофрагмента – дикторський текст. Для підготовки підписів і дикторського тексту використайте матеріал з файлу **Розділ 4\Практична 6\Термометр Галілея.doc**;
 - кінцеві титри з власним прізвищем як сценариста та режисера фільму;
 - музичний супровід усіх частин тексту, за винятком тих, де використовується дикторський текст. Для цього використайте файли музичних творів з папки **Розділ 4\Практична 6\Музика**.
3. Збережіть відеокліп у вашій папці у файл з іменем **практична робота 6**.



4.3. РОЗМІЩЕННЯ МУЛЬТИМЕДІЙНИХ ФАЙЛІВ В ІНТЕРНЕТІ



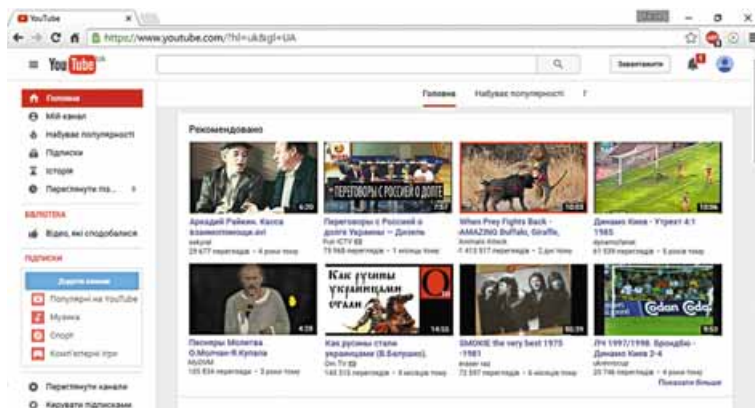
1. Що таке екаунт? Як створити екаунт на поштовому сервері?
2. Як завантажити створений раніше проект у Кіностудії Windows?
3. Як додати музику до фільму в редакторі відео?

РОЗМІЩЕННЯ ВІДЕОФАЙЛІВ У СЕРВІСІ YOUTUBE

Одним з розповсюджених сервісів для розміщення відеофайлів в Інтернеті є сервіс **YouTube** (англ. *You Tube* – твоя труба (слухавка)). Розміщувати




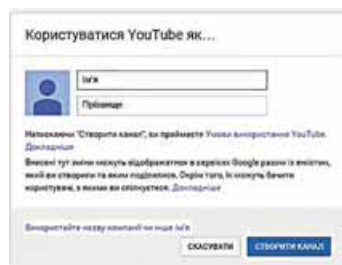
матеріали та надавати доступ до них на цьому сервісі може будь-який користувач, який має екаунт поштової служби **Google** – **Gmail**.



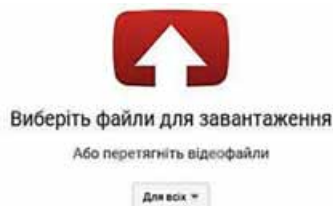
Мал. 4.13. Головна сторінка сайта YouTube

Для розміщення мультимедійних матеріалів в **YouTube** потрібно:

1. Відкрити у браузері головну сторінку сервісу – (мал. 4.13).
2. Вибрати в лівій частині вікна посилання **Мій канал**.
3. Увести у вікні, що відкрилося, логін і пароль пошти **Gmail**.
4. Увести ім'я та прізвище, під яким ваші відеоматеріали будуть розміщуватися в **YouTube** (за замовчуванням будуть використані дані з екаунта **Gmail**) (мал. 4.14).
5. Ознайомитися з умовами використання сервісу за відповідним посиланням.
6. Вибрати кнопку **Створити канал**.
7. Вибрати кнопку **Завантажити** у верхній правій частині вікна.
8. Вибрати у списку кнопки **Конфіденційність** **Для всіх** (мал. 4.15) один з варіантів доступу до перегляду вашого файлу: *Для всіх*, *Не для всіх* або *Приватне*.
9. Вибрати кнопку  та вибрати файл для розміщення.
10. Дочекатися завершення завантаження та ввести значення властивостей вашого відео у вікні налаштувань (мал. 4.16):
 - назву відео (мал. 4.16, 3);
 - опис відео – коротко описати, про що ваше відео (мал. 4.16, 4);
 - ключові слова (теги) для включення вашого відео до швидкого пошуку користувачами за зазначеними тегами (мал. 4.16, 8);



Мал. 4.14. Діалогове вікно створення власного каналу YouTube

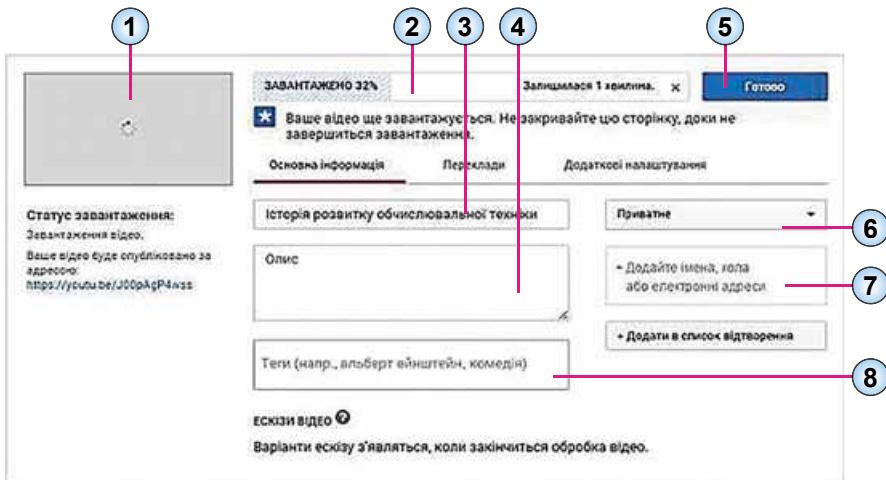


Мал. 4.15. Вікно завантаження файлу



- адреси ваших друзів або кіл спілкування для отримання ними повідомлення про розміщення вами відео з адресою в **YouTube** (мал. 4.16, 7) – у випадку вибору рівня конфіденційності *Приватне*.

11. Вибрати кнопку **Готово**.



- | | |
|---------------------------------|--|
| 1. Вікно ескізу відео | 5. Кнопка Готово |
| 2. Індикатор завантаження файлу | 6. Кнопка зі списком Конфіденційність |
| 3. Поле назви відео | 7. Поле адресантів відео |
| 4. Поле опису відео | 8. Поле ключових слів (тегів) |

Мал. 4.16. Вікно налаштувань параметрів відео

Після завантаження відеофайлу його можна переглянути, а також виконати редагування, використовуючи кнопки, що з'являються в режимі перегляду під екраном (мал. 4.17). Можна змінити:

- назву та опис (кнопка **Інформація та налаштування**);
 - яскравість, контрастність, насиченість кольорів, швидкість перегляду тощо (кнопка **Додаткові можливості**);
 - додати звук (кнопка **Звук**);
 - додати титри (кнопка **Субтитри**)
- тощо.

Конфіденційний (лат. *confidentia* – довіра) – той, що не підлягає розголошенню, обмежений у доступі.



Мал. 4.17. Кнопки засобів редагування відео

Слід зазначити, що в подальшому користувач може завантажити інші файли, використовуючи створений відеоканал на **YouTube**.

Для розміщення матеріалів в Інтернеті можна також використати інші сервіси, наприклад **Vimeo** (<https://vimeo.com>), **dailymotion** (<http://www.dailymotion.com>), **Відео ukrhomenet** (<http://video.ukrhome.net>) або соціальні мережі, наприклад **Facebook**.



ВИКОРИСТАННЯ КІНОСТУДІЇ WINDOWS ДЛЯ РОЗМІЩЕННЯ ВІДЕО В ІНТЕРНЕТІ

Після розробки проекту відеофільму в програмі **Кіностудія Windows** можна не зберігати файл відео в зовнішній пам'яті комп'ютера, а розмістити його в Інтернеті. Для цього потрібно відкрити проект створеного відеофільму та виконати такий алгоритм:

1. Виконати **Файл** ⇒ **Опублікувати фільм**.
2. Вибрати зі списку сервісів (мал. 4.18) той, на сайті якого буде розміщено відеофайл, наприклад **YouTube**.
3. Увести в поля діалогового вікна логін та пароль доступу до вашого відеоканалу на **YouTube** (зазвичай збігається з екаунтом **Gmail**).
4. Увести в поля діалогового вікна значення властивостей відеофільму: назву, опис, теги, вид дозволу на використання, категорію відеофільму.
5. Вибрати кнопку **Опублікувати**.
6. Дочекатися завершення збереження та опублікування фільму.
7. Переглянути відеофільм онлайн, вибравши відповідну кнопку.
8. Закрити вікно браузера.



Мал. 4.18. Список варіантів опублікування фільму

За подібним алгоритмом відбувається і опублікування фільму на інших сервісах. Обов'язковою умовою розміщення мультимедійних даних є створення екаунта на сайті відповідного сервіса.

РОЗМІЩЕННЯ АУДІОФАЙЛІВ В ІНТЕРНЕТІ

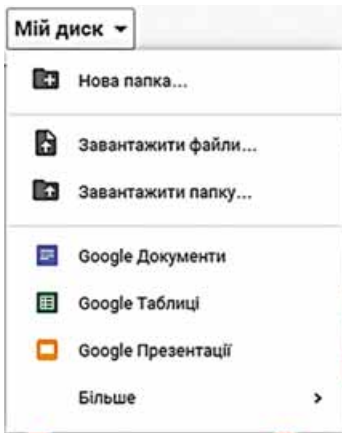
Переважна більшість сервісів для розміщення аудіофайлів призначена для тих, хто створює власні файли з різноманітними музичними творами. Ще одним напрямком розміщення аудіофайлів є створення аудіокнижок і різноманітних оглядів з певних питань: політичних, економічних, технічних тощо.

Зазначені аудіофайли називають **подкастами**. У подальшому термін подкаст поширився і на відеофайли, що розміщують користувачі в Інтернеті.

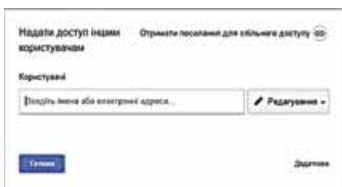
Для розміщення аудіофайлів в Інтернеті існують різні сервіси, такі як: **Канал громадського подкастингу** (<http://cpod.co/>), **Український подкаст-термінал** (<http://podcaster.org.ua>), **Best Podcast Directory** (<http://www.podcast.com>), сторінка медіакомпанії **BBC** для українців, які вивчають англійську (<http://www.bbc.co.uk/ukrainian/learningenglish>) тощо.

Для обміну аудіофайлами з друзями та знайомими можна використати сервіс **Google Диск**. Для розміщення файлів на **Google Диску** потрібно мати екаунт **Google** та виконати такий алгоритм:

Подкаст (англ. *ipod* – медіаплеер корпорації Apple; *broadcasting* – ефірне мовлення) – медіафайл або серія таких файлів, що створюються довільними користувачами (подкастерами) і розповсюджуються з використанням Інтернету.



Мал. 4.19. Список кнопки **Мій диск**



Мал. 4.20. Вікно налаштувань доступу до файлу

на до вашого електронного листа друзям, якщо ви вирішите направити окремі листи, не заповнюючи поле **Користувачі**.

1. Запустити браузер.
2. Відкрити сторінку **Google**, вибрати кнопку **Увійти** та ввести логін і пароль екаунта **Google**.
3. Вибрати кнопку **Додатки Google** та відкрити список сервісів **Google**.
4. Вибрати у списку сервісів **Мій диск**.
5. Відкрити список кнопки **Мій диск** і вибрати команду **Завантажити файли** (мал. 4.19).
6. Вибрати файл, який потрібно завантажити, та натиснути кнопку **Відкрити**.

У подальшому можна налаштувати доступ до ваших аудіофайлів інших користувачів. Наприклад, після вибору кнопки **Надати доступ до файлу** відкриється вікно налаштувань доступу до файлу інших користувачів (мал. 4.20). У поле **Користувачі** можна ввести електронні адреси ваших друзів, яким буде автоматично надіслано електронний лист з адресою аудіофайлу та дозволено його прослуховування.

Вибір посилання **Отримати посилання для спільного доступу** встановлює доступ до файлу тих користувачів, які отримують від вас посилання з адресою розміщення файлу в Інтернеті. При цьому дана адреса автоматично копіюється в **Буфер обміну** і може бути вкледена

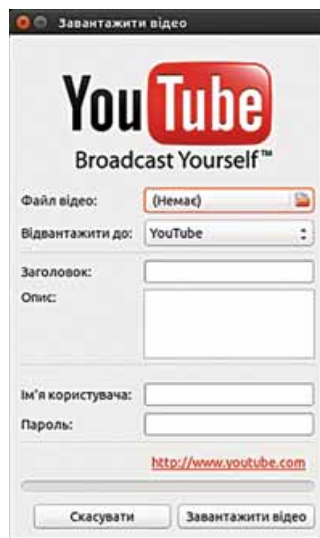


Для тих, хто працює з Linux

У **Linux** для розміщення відео в Інтернеті можна використати редактор відеофільмів **OpenShot**. Після запуску програми потрібно:

1. Виконати **Файл** ⇒ **Завантажити відео**.
2. Указати відеофайл, який буде опубліковано в Інтернеті, для цього скористатися елементом керування **Файл відео** (мал. 4.21).
3. Вибрати сервіс, на якому буде розміщено відеофільм (елемент керування **Відвантажити до**), наприклад **YouTube**.
4. Увести у відповідні поля вікна **Завантажити відео** значення властивостей відеофільму: назву (заголовок) та опис.
5. Увести логін та пароль доступу до сервісу.
6. Вибрати кнопку **Завантажити відео**.

Усі інші дії з розміщеним відеофайлом аналогічні до розглянутих вище.



Мал. 4.21. Вікно **Завантажити відео**

**Найважливіше в цьому пункті**

В Інтернеті є значна кількість сайтів, що надають послуги з розміщення мультимедійних файлів. Зазвичай для розміщення файлів на сайтах потрібно мати відповідний екаунт.

Одним з розповсюджених сервісів для розміщення відеофільмів є **YouTube** (<https://www.youtube.com>). Для завантаження фільмів на сайт **YouTube** потрібно створити власний канал, який у подальшому буде містити добірку відеофайлів користувача. Користувач може керувати доступом до перегляду кожного з відеофайлів, установлюючи певний рівень конфіденційності.

Значна частина редакторів відео мають засоби для розміщення відеофільмів в Інтернеті, наприклад такі програми, як **Кіностудія Windows** та **OpenShot**.

Мультимедійні файли, призначені для розміщення в Інтернеті, називають подкастами, а тих, хто створює такі файли, **подкастерами**.

**Дайте відповіді на запитання**

- 1°. Які ви знаєте сервіси для розміщення мультимедійних файлів в Інтернеті?
- 2°. Як створити канал для розміщення відеофайлів на сайті **YouTube**?
- 3°. Які значення властивостей відео можуть бути встановлені користувачем під час завантаження файлу на сайт **YouTube**?
- 4°. Які рівні конфіденційності можна встановити для перегляду завантаженого на **YouTube** файлу?
- 5°. Які дії потрібно виконати для додавання музичного супроводу до завантаженого на **YouTube** файлу?
- 6°. Які дії потрібно виконати для розміщення відеофайлу на одному із сервісів Інтернету безпосередньо з програми відеоредактора, наприклад з **Кіностудії Windows**?
- 7°. Яка послідовність дій для розміщення аудіофайлів на **Диску Google**?
- 8°. Що таке подкаст? Хто такий подкастер?
- 9°. Як надати доступ до файлу іншим користувачам з використанням **Диску Google**?

**Виконайте завдання**

- 1°. Перегляньте відеофільм за адресою <https://youtu.be/NcrOD9RysUo>. Визначте тривалість демонстрації.
- 2°. Розмістіть на **YouTube** запропонований учителем файл, наприклад **Розділ 4\Пункт 4.3\Відео\Сейм, старе річіще.mov**. Додайте опис, теги та музичний супровід. Надішліть учителю адресу розміщеного файлу.
- 3°. Використовуючи проект відеофільму **Довженко**, створений на минулих заняттях, розмістіть на **YouTube** цей фільм безпосередньо з програми **Кіностудія Windows**.
- 4°. Розмістіть на **YouTube** запропонований учителем файл, наприклад **Розділ 4\Пункт 4.3\Відео\Равлик.mov**. Додайте опис, теги. Знайдіть в Інтернеті відомості про равлика та додайте їх до відеофільму у вигляді титрів.





- 5°. Розмістіть на Диску Google запропонований учителем файл, наприклад **Розділ 4\Пункт 4.3\Аудіо\Щедрівочка.wma**. Надайте дозвіл доступу до цього файлу тим користувачам, хто має посилання на нього. Надішліть електронною поштою адресу файлу в Інтернеті вчителю та одному з ваших друзів.

ПРАКТИЧНА РОБОТА № 7 «Розміщення аудіо- та відеоматеріалів в Інтернеті»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Розмістіть на **YouTube** запропонований учителем файл, наприклад **Розділ 4\Практична 7\Відео\Комахи.wmv**.
2. Установіть такі значення його властивостей:
 - назва – *Комахи*;
 - опис – *Комахи центральної частини України*;
 - теги – *природа, комахи, жук-олень*;
 - рівень конфіденційності – *приватне*;
 - ескіз відео – *жук-олень*;
 - електронні адреси доступу – *електронну адресу вчителя*;
 - титри – з *00.01.0* до *00.09.0* текст *Жук-олень (Lucanus cervus)* – найвідоміший вид з родини рогачів;
 - музика – вид *Кантрі та фольк*, назва *As We Go*.
3. Розмістіть аудіофайл, запропонований учителем, наприклад, **Розділ 4\Практична 7\Аудіо\Музика.mp3**, на Диску Google та надайте доступ до його прослуховування вчителю й одному з учнів вашого класу.

Розділ 5. Основи подійно- та об'єктно-орієнтованого програмування

У цьому розділі ви дізнаєтеся про:

мови програмування та їх складові

середовище розробки проектів **Lazarus**

компоненти:
форма, кнопка, напис; їх властивості та події, що можуть відбуватися з ними

обробників подій

змінення значень властивостей компонентів керування перед виконанням проекту і під час його виконання

вікно повідомлень

створення проектів у середовищі **Lazarus** мовою **Object Pascal**, їх редагування і виконання



5.1. КОМП'ЮТЕРНІ ПРОГРАМИ І МОВИ ПРОГРАМУВАННЯ



1. Що таке алгоритм? Що таке система команд виконавця алгоритму? У чому полягає формальність виконання алгоритму виконавцем?
2. Що таке комп'ютерна програма?
3. Яку мову програмування ви вивчали в попередніх класах? Пригадайте кілька команд цієї мови програмування.

КОМП'ЮТЕРНІ ПРОГРАМИ

Ви вже знаєте, що комп'ютер працює під керуванням програмного забезпечення, яке складається з комп'ютерних програм різноманітного призначення. Працюючи з комп'ютером у школі і вдома, ви використовували текстовий процесор, графічний редактор, програми-архіватори, табличний процесор, редактор комп'ютерних презентацій, навчальні і контролюючі програми, ігрові програми та багато інших.

Ви також знаєте, що **комп'ютерна програма** – це алгоритм опрацювання даних, записаний спеціальною мовою та призначений для виконання комп'ютером.

У процесі своєї роботи програма опрацьовує **дані**.

Дані, які потрапляють до програми від певних пристроїв (наприклад, з клавіатури або від датчика температури) або від іншої програми, або з іншого джерела (наприклад, з текстового файлу), називають **вхідними (початковими) даними**. Деякі програми працюють без вхідних даних.

Під час виконання програми утворюються і опрацьовуються й інші дані, які називають **проміжними даними**.

Якщо метою виконання програми є отримання певних даних, то ці дані називають **вихідними (результуючими) даними**.

Більшість сучасних програм у процесі своєї роботи надають користувачу певний набір засобів для його взаємодії з програмою і пристроями. До цих засобів належать засоби керування (кнопки, меню та ін.), засоби введення даних (поля, лічильники та ін.), засоби виведення даних (написи, поля та ін.) тощо. Сукупність таких засобів, а також методів їх використання утворює **інтерфейс користувача**.

МОВИ ПРОГРАМУВАННЯ

Складаючи алгоритми, призначені для виконання людиною, користуються звичною мовою спілкування людей: українською, російською, англійською, німецькою тощо.

Але для алгоритмів, які повинен виконувати комп'ютер, мова спілкування людей складна, має неоднозначні трактування (наприклад, слова-омоніми). Тому для запису алгоритмів, призначених для виконання комп'ютером, розробляють і використовують спеціальні мови – **мови програмування**.



Мову, яку використовують для запису алгоритмів, призначених для виконання комп'ютером, називають **мовою програмування**.

За останні 70 років створено близько трьох тисяч різних мов програмування. Деякі з них уже не використовують, для деяких постійно з'являються нові версії, що зручніші для складання програм і мають більші можливості. Постійно створюються нові мови програмування.

Деякі мови програмування використовуються для складання програм для розв'язування задач у різних галузях науки, техніки, виробництва, сфери побуту та ін., а деякі створені саме для складання програм для розв'язування спеціального кола задач.

Мови програмування є прикладами так званих **штучних мов**. Іншими прикладами штучних мов є, наприклад, мова запису хімічних реакцій, мова запису розв'язування математичних рівнянь і нерівностей, мова запису поштових індексів на конвертах і листівках, мова запису ходів шахової партії тощо.

Кожна мова програмування має такі складові:

1. **Алфавіт** – множина символів, з яких можна утворювати слова і речення цієї мови.
2. **Словник** – набір слів, які використовуються в цій мові.
3. **Синтаксис** – правила складання і запису мовних конструкцій: не словникових слів і речень.
4. **Семантику** – встановлене однозначне тлумачення мовних конструкцій, правил їх виконання.

У 6–7 класах ви вивчали мову програмування **Scratch**. Пригадаємо, що до алфавіту цієї мови програмування входять цифри, літери української абетки, англійського алфавіту, символи $:$, $=$, $+$, $*$, $<$ та ін. До її словника входять, наприклад, слова **повторювати**, **переміститися**, **якщо**, **відбити**, **коли**. Відповідно до синтаксичних правил цієї мови програмування команда циклу з передумовою має містити слова **повторювати поки** і саме в такому порядку. А відповідно до семантики цієї мови програмування комп'ютер однозначно виконає команду **переміститися на 10 кроків** і перемістить виконавця на 10 кроків у напрямі його руху.

Використання символів, що не входять до алфавіту мови програмування, неправильне написання словникових слів, порушення синтаксичних правил призводять до неможливості виконання комп'ютером відповідної команди. Такі порушення називають **синтаксичними помилками**.

У 8 і 9 класах ви будете вивчати мову програмування **Object Pascal** (англ. *Object Pascal* – об'єктний Паскаль). Цю мову використовують учні і студенти для вивчення основних принципів програмування, а також професійні програмісти для створення різноманітних програм.



Цікаві факти з історії

Мову програмування **Pascal** створила в 1970 р. група співробітників Інституту інформатики Швейцарської вищої технічної школи під керівництвом професора Ніклауса Вірта (нар. у 1934 р., мал. 5.1). Назвали її на честь видатного французького математика, фізика, інженера, літератора, філософа Блезе Паскаля.

Спочатку мова програмування **Pascal** створювалася спеціально для навчання студентів основних принципів і методів процедурного програмування. Але згодом

з'ясувалося, що ця мова програмування досить зручна, і виникла ідея застосувати її не лише для навчання, а й для створення програм для розв'язування практичних завдань з використанням комп'ютера.



Мал. 5.1. Ніклаус Вірт

Так з'явилися перші програми-компілятори для автоматичного перекладання програм, написаних мовою програмування **Pascal**, у програми, які може виконати процесор комп'ютера, а також середовище розробки **Turbo Pascal**, які з часом здобули широку популярність.

З роками мова програмування **Pascal** розвивалася й вдосконалювалася. Нові версії цієї мови, які вже мали елементи **об'єктно-орієнтованої** (орієнтованої на об'єкти та їх властивості) мови програмування, з'явилися у другій половині 80-х років минулого століття. Одна з версій цієї мови одержала назву **Object Pascal**. На початку 2000-х років чергова нова версія **Object Pascal** одержала назву **Delphi**.

СЕРЕДОВИЩЕ РОЗРОБКИ ПРОГРАМ LAZARUS

Для створення, редагування і налагодження програм мовою програмування **Object Pascal** ми будемо використовувати **середовище розробки програм Lazarus**.

Це середовище складається з:


- **текстового редактора** для введення і редагування тексту програм;
- **компілятора** для перекладу програми з мови програмування **Object Pascal** на мову команд, які може виконати процесор комп'ютера;
- **засобів налагодження програми** для пошуку в ній помилок;
- **довідкової системи**

та інших компонентів.

Це середовище розробки надає можливість використовувати велику бібліотеку **візуальних компонентів** (англ. *Lazarus Component Library (LCL)* – бібліотека компонентів **Lazarus**). Це вже відомі вам компоненти: кнопки, поля, написи, прапорці, перемикачі, лічильники, списки, смуги прокручування тощо. Під час розміщення цих компонентів на формі автоматично створюється відповідний фрагмент програми мовою **Object Pascal** і його не потрібно вводити до тексту програми вручну, що значно зменшує, спрощує і пришвидшує роботу зі створення програм. Такий спосіб розробки програм називають **візуальним**.

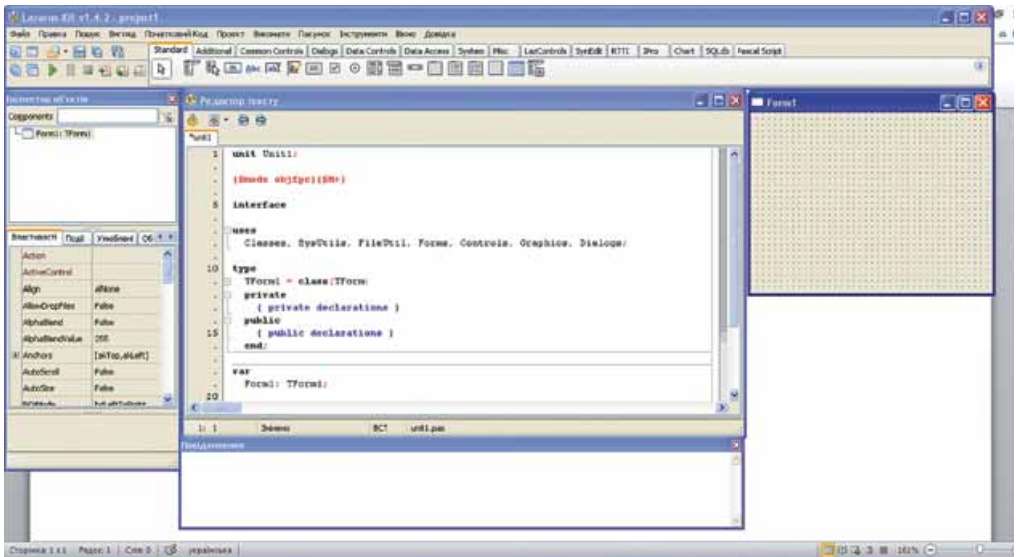
Програми, створені в середовищі **Lazarus**, називають **проектами**. Вони складаються з кількох файлів.

Відкрити вікно середовища **Lazarus** можна:

- використовуючи команду **Lazarus** меню **Пуск** ⇒ **Програми** ⇒ **Lazarus**;
- використовуючи значок  на **Робочому столі**.

У результаті відкривається вікно середовища **Lazarus** (мал. 5.2), на якому розміщено **Головне меню**, панель інструментів, вкладки з компонентами, а також кілька підлеглих вікон, серед яких:

- вікно **Інспектор об'єктів**, у якому можна для кожного компонента переглядати і змінювати значення його властивостей, розпочинати створення обробників подій та ін.;
- вікно **Редактор тексту** з вкладками, на яких розміщуються фрагменти програми мовою **Object Pascal**;
- вікно форми.



Мал. 5.2. Вікно середовища Lazarus і підлеглі вікна

Форма – це об'єкт, на якому можна розмістити різні компоненти, зокрема, кнопки, поля, написи, меню, смуги прокручування та ін. Під час виконання проекту форма відкривається як вікно прикладної програми, на якому відображатимуться всі об'єкти, розміщені на формі.

Форма має **властивості**, які відображаються на вкладці **Властивості** вікна **Інспектор об'єктів**. У лівій частині цієї вкладки знаходиться список властивостей, а в правій – їх значення. Значення деяких властивостей форми потрібно вводити у відповідні поля, а деяких можна як вводити в поле, так і вибирати зі списку.

Наведемо приклади деяких із цих властивостей:

Name (англ. *name* – ім'я) – її значення визначає ім'я форми як об'єкта;

Caption (англ. *caption* – заголовок) – її значення визначає текст у рядку заголовка форми;

Color (амер. *color* – колір) – її значення визначає колір фону форми;

Height (англ. *height* – висота) – її значення визначає висоту вікна в пікселях (**піксель** – це точка графічного зображення на екрані);

Width (англ. *width* – ширина) – її значення визначає ширину вікна в пікселях;

Top (англ. *top* – верх) – її значення визначає відстань (у пікселях) від верхньої межі вікна до верхньої межі екрана;


Left (англ. *left* – лівий) – її значення визначає відстань (у пікселях) від лівої межі вікна до лівої межі екрана.

Зазначимо, що значення властивостей **Top** і **Left** можна змінити перетягуванням форми по екрану, а значення властивостей **Height** і **Width** – зміненням розмірів форми, використовуючи маркери, розташовані на її межах.

Як ми вже зазначали, сучасні середовища візуальної розробки проектів, однією з яких є **Lazarus**, частково автоматизують процес їх створення,



тим самим полегшуючи роботу як професіоналам, так і початківцям. Уже навіть під час відкриття нового проекту, коли ще не було введено жодного рядка тексту програми, **Lazarus** автоматично створює (генерує) певний початковий текст проекту. Цього тексту проекту достатньо, щоб запустити його на виконання.

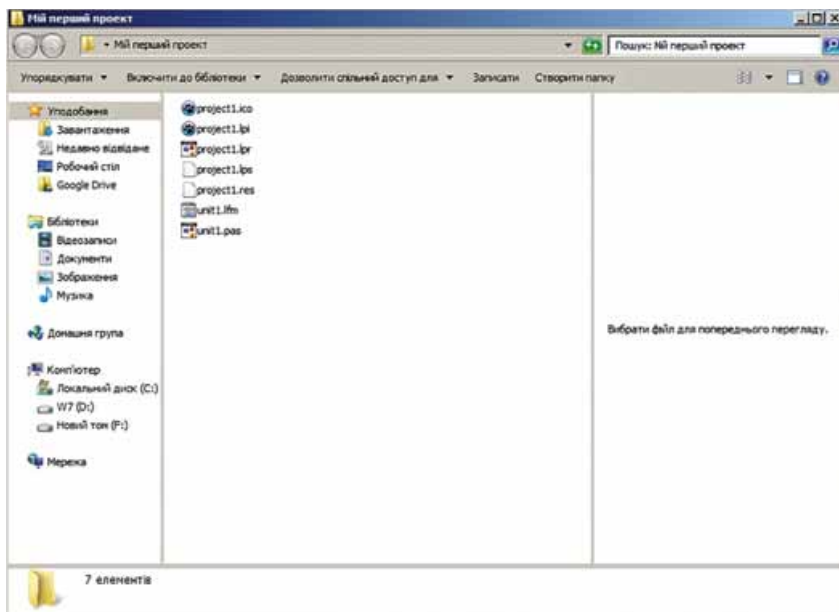
Перед початком виконання проект рекомендується зберегти на носії. Як ми вже зазначали, сучасні проекти складаються не з одного файлу, а з кількох. Тому доцільно для кожного нового проекту створювати окрему папку. Зберегти проект можна, використовуючи команду **Зберегти всі** меню **Файл**, або кнопку **Зберегти всі**  панелі інструментів, або команду **Зберегти проект** меню **Проект**.

Система послідовно пропонує зберегти два файли проекту:

файл з іменем **Project1** (англ. *project* – проект) і з розширенням імені **lpi**;
файл з іменем **Unit1** (англ. *Unit* – блок) і з розширенням імені **pas**.

Розширення імен файлів змінювати не можна, а імена файлів можна змінити.


Усі інші файли проекту автоматично зберігаються у створеній папці, наприклад **Мій перший проект** (мал. 5.3).

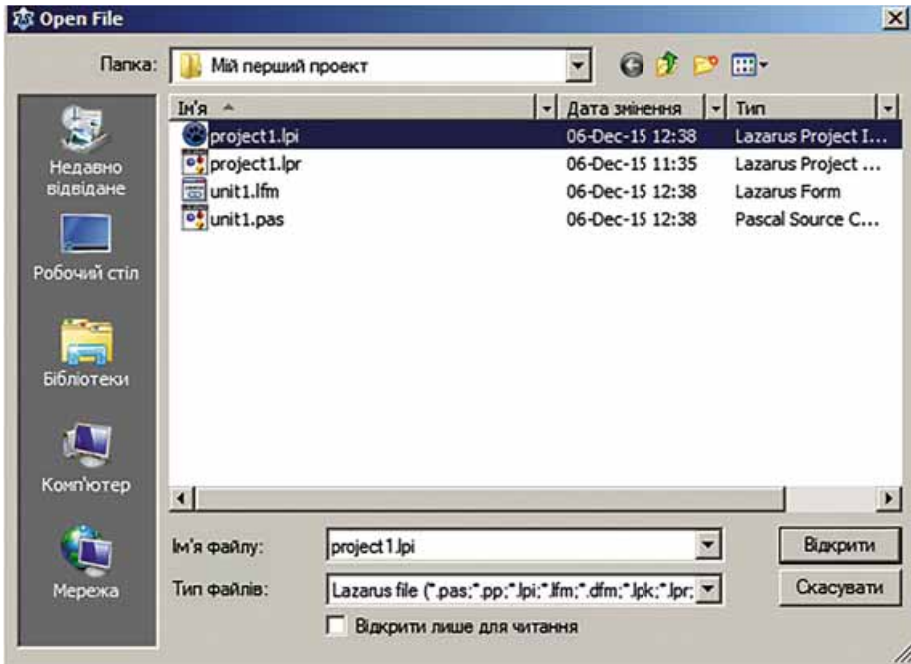


Мал. 5.3. Папка з файлами проекту

Якщо вже збережений проект змінити, то під час збереження його нової версії в ту саму папку без змінення імен його файлів ніякі додаткові вікна відкриватися не будуть.

Уже збережений проект можна зберегти в іншій папці, імена його файлів можна змінити. Для цього доцільно використати команду **Зберегти проект як** меню **Проект**.

Щоб відкрити проект, збережений раніше на носії, потрібно виконати команду **Відкрити** з меню **Файл**, або вибрати кнопку **Відкрити**  на па-

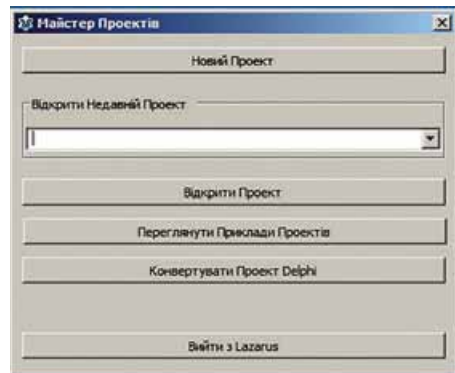


Мал. 5.4. Вікно відкриття проекту

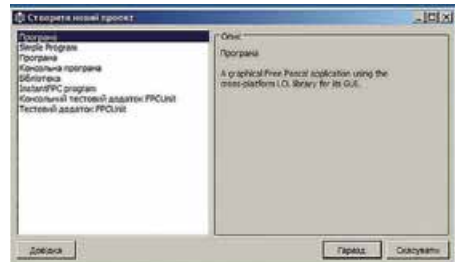
нелі інструментів, або виконати команду **Відкрити проект** з меню **Проект**. Після цього у вікні, що відкриється, вибрати файл з розширенням імені lpi, а далі вибрати кнопку **Відкрити** (мал. 5.4).

Іноколи під час запуску середовища **Lazarus** відкривається останній збережений проект. Закрити його можна командою **Закрити проект** з меню **Проект**. Після виконання цієї команди відкривається вікно **Майстер проектів** (мал. 5.5). У цьому вікні можна вибрати кнопку **Новий проект** для початку роботи над новим проектом, користуючись списком **Відкрити недавній проект**, можна відкрити один з останніх проектів, що відкривалися в середовищі, можна також відкрити будь-який проект, збережений на носії.

Після вибору кнопки **Новий проект** відкривається вікно **Створити новий проект** (мал. 5.6), у якому потрібно у списку вибрати перший рядок **Програма**, після чого вибрати кнопку **Гаразд**.




Мал. 5.5. Вікно Майстер проектів

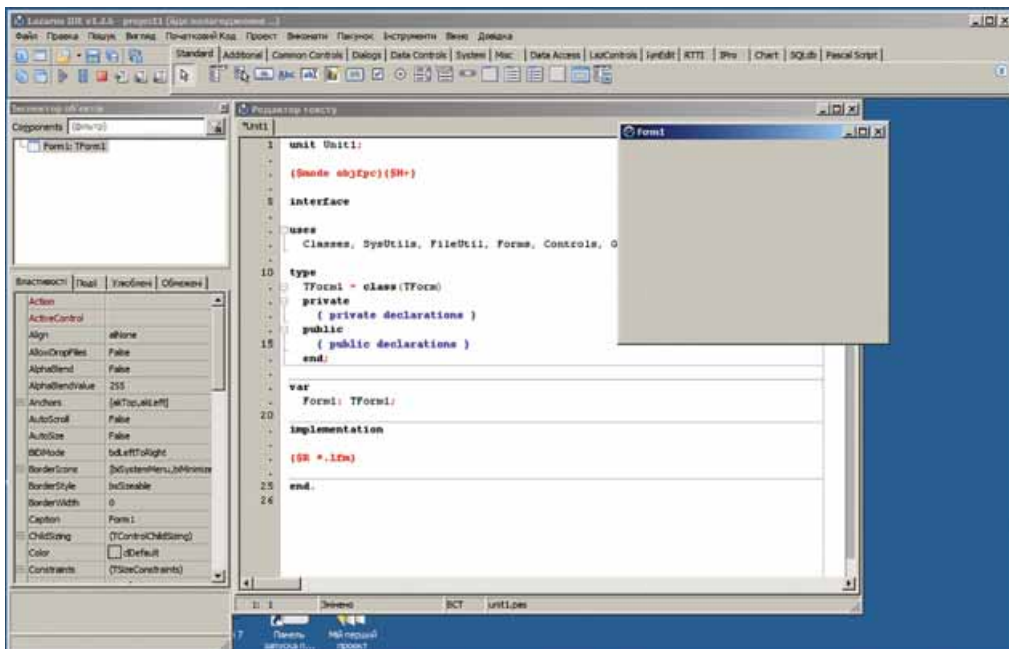


Мал. 5.6. Вікно Створити новий проект

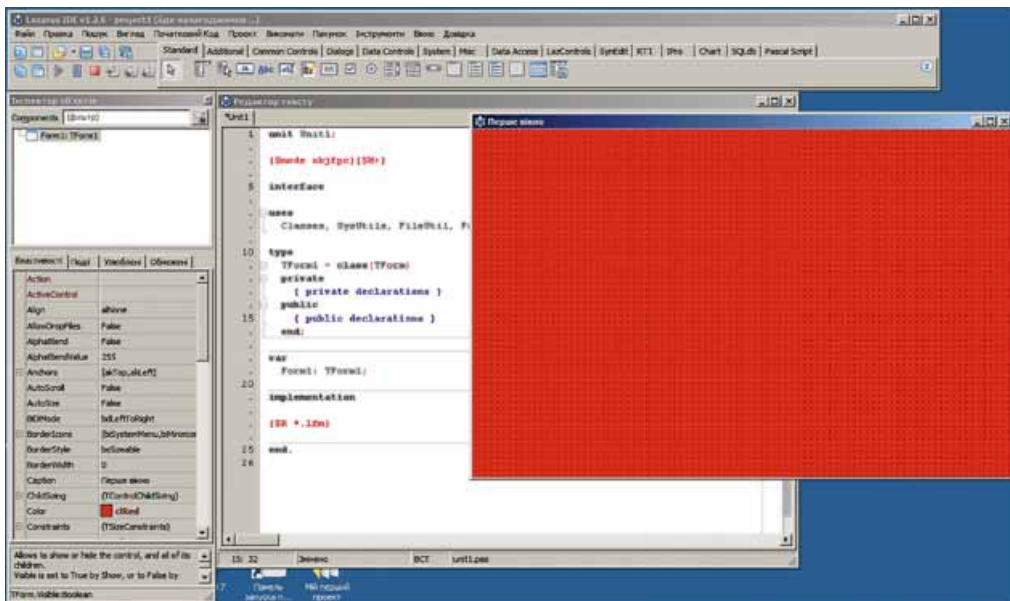


ВИКОНАННЯ ПРОЕКТУ

Розпочати виконання проекту можна, виконавши команду **Виконати** з меню **Виконати**, або вибравши кнопку **Виконати**  на панелі інструментів, або натиснувши клавішу **F9**.



Мал. 5.7. Проект, що створюється за замовчуванням, і результат його виконання



Мал. 5.8. Результат виконання зміненого проекту, створеного за замовчуванням



Під час запуску проекту на виконання розпочинає свою роботу **програма-компілятор**. Насамперед ця програма аналізує текст проекту на наявність синтаксичних помилок. У разі наявності таких помилок ця програма перериває свою роботу і вказує на місце першої з них. Потрібно виправити помилку і знову запустити проект на виконання. Якщо текст проекту не містить синтаксичних помилок, то програма-компілятор перекладає текст проекту з мови **Object Pascal** на мову команд, які може виконати процесор комп'ютера, і передає вже перекладений проект процесору на виконання.

Під час виконання проекту, який автоматично створюється при відкритті нового проекту, на екрані відкривається вікно, значення властивостей якого (колір, розміри, положення на екрані, текст у рядку заголовка та ін.) встановлені за замовчуванням (мал. 5.7).

Для закінчення виконання проекту потрібно закрити вікно його виконання, вибравши кнопку закриття вікна або натиснувши сполучення клавіш **Alt + F4**.

До початкового проекту, як і до будь-якого іншого, можна вносити зміни. Наприклад, можна змінити значення деяких властивостей форми. Для цього на вкладці **Властивості** вікна **Інспектор об'єктів** потрібно встановити, наприклад:

Caption – *Перше вікно*

Color – *clRed* (*color* – колір, *red* – червоний)

Height – *450*

Width – *700*

Left – *600*

Top – *350*

Якщо зберегти нову версію проекту і знову виконати його, теж відкриється вікно, але іншого кольору, інших розмірів, розташоване в іншому місці екрана, з іншим текстом у рядку заголовка (мал. 5.8).

Щоб закрити вікно середовища **Lazarus**, можна виконати команду **Вихід** з меню **Файл** або вибрати кнопку закриття вікна середовища в рядку заголовка вікна **Lazarus**.

Найважливіше в цьому пункті

Комп'ютерна програма – це алгоритм опрацювання даних, записаний спеціальною мовою та призначений для виконання комп'ютером.

У процесі своєї роботи програма опрацьовує **дані**.

Дані, які потрапляють до програми від певних пристроїв (наприклад, з клавіатури або від датчика температури) або від іншої програми, або з іншого джерела (наприклад, з текстового файлу), називають **вхідними (початковими) даними**. Деякі програми працюють без вхідних даних.

Під час виконання програми утворюються і опрацьовуються й інші дані, які називають **проміжними даними**.

Якщо метою виконання програми є отримання певних даних, то ці дані називають **вихідними (результуючими) даними**.

Мову, яку використовують для запису алгоритмів, призначених для виконання комп'ютером, називають **мовою програмування**.

Кожна мова програмування має такі складові:



1. **Алфавіт** – множина символів, з яких можна утворювати слова і речення цієї мови.
2. **Словник** – набір слів, які використовуються в цій мові.
3. **Синтаксис** – правила складання і запису мовних конструкцій: не словникових слів і речень.
4. **Семантику** – встановлене однозначне тлумачення мовних конструкцій, правил їх виконання.

Середовище розробки програм Lazarus складається з:

- **текстового редактора** для введення і редагування тексту програм;
 - **компілятора** для перекладу програми з мови програмування **Object Pascal** на мову команд, які може виконати процесор комп'ютера;
 - **засобів налагодження програми** для пошуку в ній помилок;
 - **довідкової системи**
- та інших компонентів.

Форма – це об'єкт, на якому можна розмістити різні компоненти, зокрема, кнопки, поля, написи, меню, смуги прокручування та ін. Під час виконання проекту форма відкривається як вікно прикладної програми, на якому відображатимуться всі об'єкти, розміщені на формі.

Приклади деяких з властивостей форми:

Name – її значення визначає ім'я форми як об'єкта;

Caption – її значення визначає текст у рядку заголовка форми;

Color – її значення визначає колір фону форми;

Height – її значення визначає висоту вікна в пікселях (**піксель** – це точка графічного зображення на екрані);

Width – її значення визначає ширину вікна в пікселях;

Top – її значення визначає відстань (у пікселях) від верхньої межі вікна до верхньої межі екрана;

Left – її значення визначає відстань (у пікселях) від лівої межі вікна до лівої межі екрана.



Дайте відповіді на запитання

- 1°. Що таке комп'ютерна програма?
- 2°. Які дані, що опрацьовує програма, називають вхідними, проміжними, вихідними?
- 3°. Що таке мова програмування?
- 4°. Які складові частини мови програмування? Опишіть кожен з них.
- 5°. Що таке синтаксична помилка?
- 6°. Для чого призначене середовище **Lazarus**?
- 7°. Які основні складові середовища **Lazarus** і яке їх призначення?
- 8°. Для чого призначена програма-компілятор? Які основні етапи її виконання?
- 9°. Що таке форма? Назвіть кілька її властивостей.
- 10°. Як відкрити проект у середовищі **Lazarus**? Як зберегти проект?

**Виконайте завдання**

1°. Створіть проект, виконання якого відкриє вікно червоного кольору розмірами 400×350 і змініть текст у заголовку вікна на ваше прізвище. Збережіть проект у папці з іменем **Завдання 5.1.1**, створеній у вашій папці.

2°. Створіть проект, виконання якого відкриє вікно синього кольору розмірами 200×450 і змініть текст у заголовку вікна на ваше ім'я. Збережіть проект у папці з іменем **Завдання 5.1.2**, створеній у вашій папці.

3°. Створіть проект, виконання якого відкриє вікно синього кольору розмірами 300×300 , ліва межа якого знаходиться на відстані 300 пікселів від лівої межі екрана і верхня межа якого знаходиться на відстані 400 пікселів від верхньої межі екрана. Збережіть проект у папці з іменем **Завдання 5.1.3**, створеній у вашій папці.



4°. Створіть проект, виконання якого відкриє вікно сірого кольору розмірами 200×100 , ліва межа якого знаходиться на відстані 150 пікселів від лівої межі екрана і верхня межа якого знаходиться на відстані 100 пікселів від верхньої межі екрана. Збережіть проект у папці з іменем **Завдання 5.1.4**, створеній у вашій папці.

**5.2. ПОДІЇ. ОБРОБНИКИ ПОДІЙ**

1. Що таке комп'ютерна програма?
2. Що таке форма? Назвіть кілька її властивостей.
3. Яке призначення програми-компілятора і які етапи її виконання?

ПОДІЇ ТА ЇХ ОБРОБНИКИ

Форма, крім набору властивостей, має ще набір **подій**, які можуть з нею відбуватися під час виконання проекту.

Такими подіями, наприклад, є:

- **Click** (англ. *click* – клацання) – вибір будь-якої точки об'єкта;
- **Close** (англ. *close* – закривати) – закривання форми (вікна) під час завершення виконання проекту;
- **Create** (англ. *create* – створення) – створення (відкриття) форми (вікна) на початку виконання проекту;
- **KeyPress** (англ. *key press* – натиснення клавіші) – натиснення будь-якої клавіші на клавіатурі;
- **MouseMove** (англ. *mouse move* – переміщення миші) – переміщення вказівника по вікні

та інші.

З будь-якою подією, яка може відбуватися з формою, можна пов'язати фрагмент програми, який буде виконуватися одразу після настання цієї події. Такий фрагмент програми називають **обробником події**. Обробник події, пов'язаний з певним об'єктом, називають **методом** цього об'єкта.

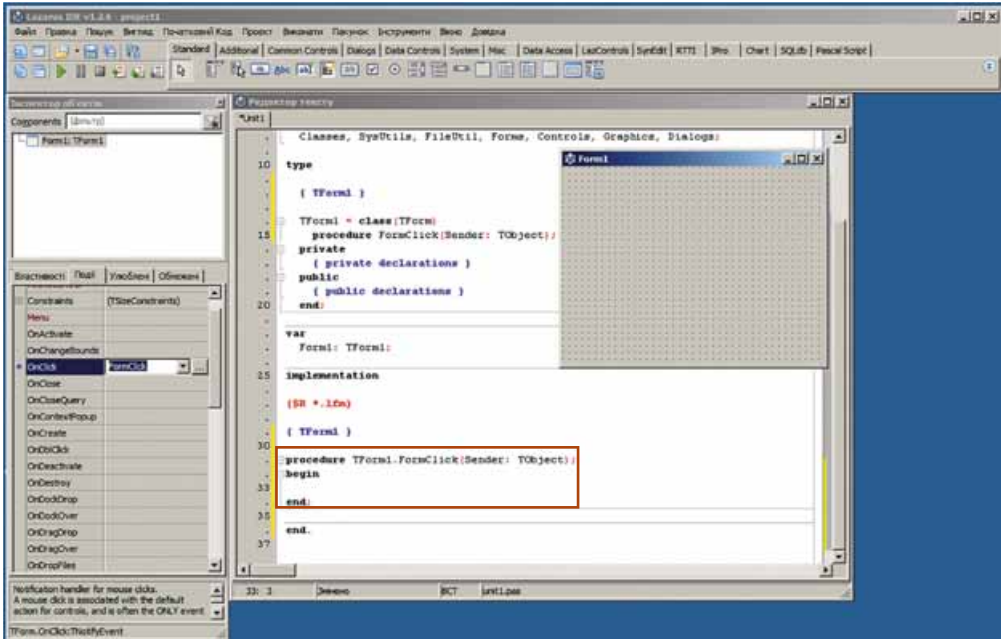
Середовище **Lazarus** за замовчуванням створює обробник події як **процедуру**. **Процедура** – це окремо виділена частина програми, яка має своє ім'я і яку можна за цим іменем запустити на виконання.

Виконання процедури також є подією. Якщо процедура виконується в результаті настання певної події з формою, наприклад події **Click**, то подію самого виконання цієї процедури називають **OnClick** (англ. *on* – на).

Наприклад, якщо потрібно пов'язати деяку процедуру з подією **Click** для форми, то слід:

1. Вибрати форму.
2. Відкрити вкладку **Події** у вікні **Інспектор об'єктів**.
3. Вибрати подію **OnClick**. Поле праворуч від тексту **OnClick** стане поточним і в ньому з'явиться курсор.
4. Клацнути двічі у цьому поточному полі.

У результаті виконання цього алгоритму в полі **OnClick** на вкладці **Події** з'являється текст **FormClick** (англ. *Form Click* – клацання форми), а у вікні редактора тексту проекту на вкладці **Unit1** з'являється заготовка для тексту процедури з іменем **TForm1.FormClick**, команди якої й будуть виконуватися при настанні події **Click** з формою (мал. 5.9).



Мал. 5.9. Заготовка для тексту процедури **TForm1.FormClick**

Команди процедури записуються між двома словами **begin** (англ. *begin* – початок) і **end** (англ. *end* – кінець). *Звертаємо вашу увагу*, що **begin** і **end** – це не команди, ці слова відіграють роль відповідно відкриваючої та закриваючої дужки, що визначають початок і закінчення процедури. Ці слова часто називають **операторними дужками**.

Після слова **end** у кінці процедури повинна стояти крапка з комою (;). Кожна команда процедури також має закінчуватися крапкою з комою (;).



Якщо потрібно, щоб у результаті виконання процедури змінилося значення певної властивості форми, процедура повинна містити команду **присвоювання**. Вона може мати такий вигляд:

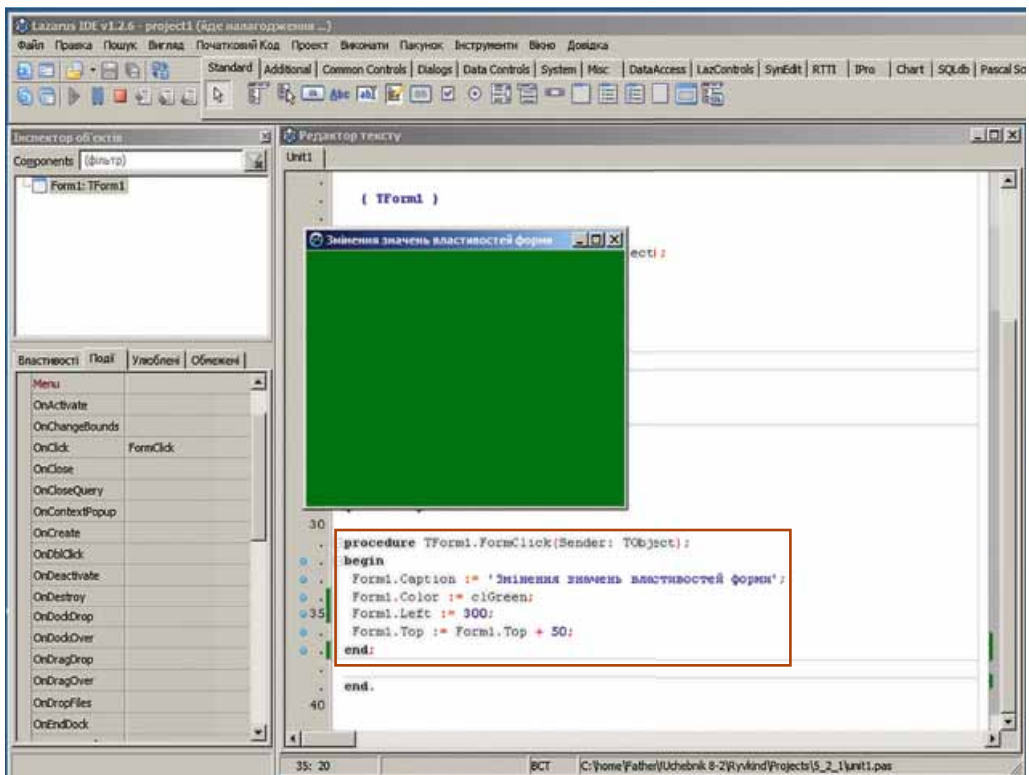
<ім'я об'єкта>. <назва властивості> := <значення або вираз>;
або такий:

<ім'я об'єкта>. <назва властивості> := <ім'я об'єкта>. <назва властивості> + <значення або вираз>;

Наприклад, виконання команди **Form1.Color := clGreen** встановить зеленим колір фону форми. А виконання команди **Form1.Top := Form1.Top + 50** збільшить поточний відступ верхньої межі вікна від верхньої межі екрана на 50 пікселів. Під час виконання останньої команди спочатку до поточного значення властивості **Top** форми буде додано число 50, після чого отримане число стане новим значенням цієї властивості форми.

Командою присвоювання **Form1.Caption := 'Змінення значень властивостей форми'** можна встановити вказаний у лапках текст у рядку заголовка форми, а командою **Form1.Left := 300** – встановити відступ 300 пікселів лівої межі вікна від лівої межі екрана.

Наведемо приклад процедури **TForm1.FormClick**, що складається з команд, наведених вище, у результаті виконання якої зміняться значення властивостей **Caption**, **Color**, **Left** і **Top** форми, а також результат її виконання (мал. 5.10).



Мал. 5.10. Процедура змінення значення властивостей **Caption**, **Color**, **Left** і **Top** форми і результат її виконання

Після запуску проекту на виконання потрібно дочекатися відкриття вікна його виконання і вибрати будь-яку його точку. Таким чином відбудеться подія **Click** для форми і як наслідок відбудеться подія **OnClick**, яка полягатиме у виконанні процедури **TForm1.FormClick** – обробника події **Click** для форми.

ЗАСОБИ СТВОРЕННЯ І РЕДАГУВАННЯ ПРОЕКТУ

Середовище **Lazarus** має зручні засоби допомоги під час написання тексту проекту:

- після введення першої або кількох перших літер імені об'єкта можна натиснути сполучення клавіш **Ctrl + Пропуск** і відкриється список, з якого можна вибрати потрібне ім'я об'єкта (мал. 5.11);
- після введення крапки, що розділяє ім'я об'єкта і назву його властивості, автоматично відкривається вікно зі списком, який містить назви властивостей даного об'єкта (мал. 5.12). Далі назву властивості можна не вводити з клавіатури, а знайти у списку і натиснути клавішу **Enter**.

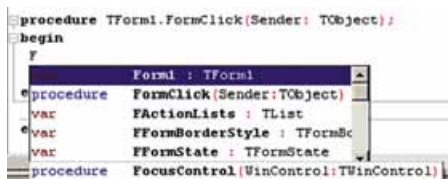
Можна після відкриття списку ввести кілька перших літер з назви властивості. Тоді цей список скорочуватиметься, і у ньому залишатимуться лише ті назви, які починаються з уведених літер. Такий спосіб пришвидшує введення назв властивостей об'єктів, а також запобігає допущенню синтаксичних помилок, особливо з боку тих, хто не вчив англійської мови.

У багатьох випадках система автоматично розставляє в потрібних місцях тексту програми операторні дужки, знак присвоювання, крапку з комою в кінці команди тощо.

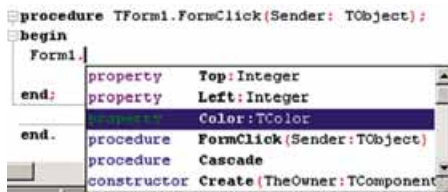
Зазначимо також, що система не розрізняє великі і малі літери. Однак якщо слово утворено з кількох слів, прийнято (і зручно для кращого розуміння) писати першу літеру кожного слова великою.

Ви вже знаєте, що на початку виконання проекту програма-компілятор перевіряє текст проекту на наявність синтаксичних помилок. Якщо такі помилки є, то виконання проекту переривається і у вікні **Messages** (англ. *messages* – повідомлення), яке відкривається під вікном **Редактора тексту**, з'являються повідомлення-підказки про номери рядків і номери позицій у них, де допущені помилки, а також коментарі до кожної із цих помилок. Крім того, кольором виділяється фон першого з тих рядків тексту проекту, що містить помилку, або колір фону рядка, наступного за рядком з помилкою.

Так, на малюнку 5.13 виділено кольором рядок 34, бо саме в ньому містяться перші ситаксичні помилки проекту – неправильно написані слова



Мал. 5.11. Список для вибору імені об'єкта



Мал. 5.12. Список з назвами властивостей об'єкта



Color і **Green**. Крім того, у кінці команди, що знаходиться в рядку 35, пропущено крапку з комою. Для такої помилки у вікні **Messages** з'являється повідомлення про неправильний символ, з якого починається наступний, 36-й рядок. І якщо саме ця помилка була б у проекті першою, то кольором було б виділено саме наступний, 36-й рядок.

Потрібно виправити в тексті проекту всі синтаксичні помилки і знову запустити проект на виконання.

```

- procedure TForm1.FormClick(Sender: TObject);
- begin
  Form1.Caption := 'Зміна значень властивостей форми';
34 Form1.Colour := 'Зміна значень властивостей форми';
35 Form1.Left := 300;
  Form1.Top := Form1.Top + 50;
- end;
-
40
42
end.

```

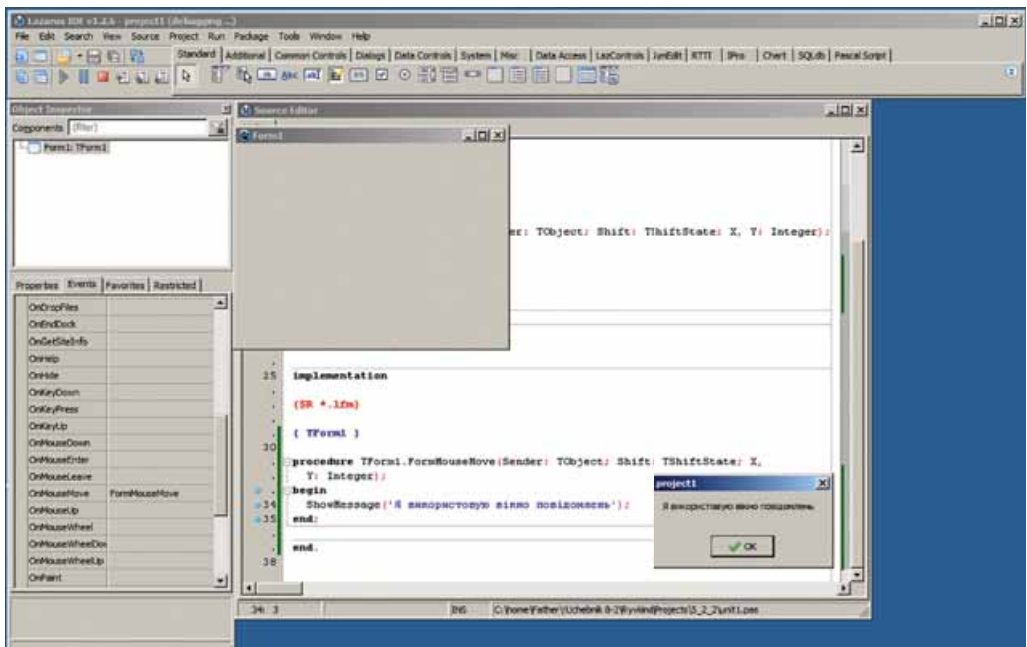
Мал. 5.13. Позначення синтаксичних помилок у середовищі **Lazarus**

ВІКНО ПОВІДОМЛЕНЬ І ЙОГО ВИКОРИСТАННЯ

Для виведення на екран текстових повідомлень можна використати спеціальне вікно – **вікно повідомлень**. Для відкриття такого вікна і виведення на ньому потрібного текстового повідомлення використовується команда **ShowMessage** ('<текст>') (англ. *show message* – показувати повідомлення).

Приклад тексту процедури – обробника події **MouseMove** (англ. *mouse move* – переміщення миші) з такою командою і результат виконання відповідного проекту наведено на малюнку 5.14.

Після ознайомлення з повідомленням потрібно закрити це вікно, вибравши кнопку **OK** або кнопку закриття вікна. Тільки після цього можна закрити вікно виконання проекту.



Мал. 5.14. Використання вікна повідомлень

Звертаємо вашу увагу, що в цьому прикладі використано подію MouseMove. Цікавим є те, що в цьому випадку вікно виконання проекту не вдасться закрити, використовуючи мишу. Адже тільки-но вказівник буде переміщуватися по вікні, настане подія MouseMove і як результат – подія OnMouseMove, тобто запускатиметься на виконання обробник події MouseMove – процедура TForm1.FormMouseMove, яка знову відкриватиме вікно повідомлень. Тому закінчити виконання цього проекту можна, натиснувши сполучення клавіш Alt + F4. Зрозуміло, що це сполучення клавіш можна використовувати для закриття вікна виконання будь-якого проекту.

Найважливіше в цьому пункті

З будь-якою подією, яка може відбутися з формою, можна пов'язати фрагмент програми, який буде виконуватися одразу після настання цієї події. Такий фрагмент програми називають **обробником події**. Обробник події, пов'язаний з певним об'єктом, називають **методом** цього об'єкта.

Середовище Lazarus за замовчуванням оформлює обробник події як процедуру. **Процедура** – це окремо виділена частина програми, яка має своє ім'я і яку можна за цим іменем запустити на виконання.

Команди процедури записуються між двома словами **begin** (англ. *begin* – початок) і **end** (англ. *end* – кінець). *Звертаємо вашу увагу, що ці слова відіграють роль відповідно відкриваючої та закриваючої дужки (операторні дужки), що визначають початок і закінчення процедури.*

Після слова **end** у кінці процедури повинна стояти крапка з комою (;). Кожна команда процедури також повинна закінчуватися крапкою з комою (;).

Якщо потрібно, щоб у результаті виконання процедури змінилося значення певної властивості форми, процедура повинна містити команду **при-своювання**. Вона може мати такий вигляд:

`<ім'я об'єкта>.<назва властивості> := <значення або вираз>;`

або такий:

`<ім'я об'єкта>.<назва властивості> := <ім'я об'єкта>.<назва властивості> + <значення або вираз>;`

Для відкриття вікна повідомлень і виведення на ньому потрібного текстового повідомлення використовується команда **ShowMessage ('<текст>');**

Дайте відповіді на запитання

- 1°. Які події можуть відбуватися з формою?
- 2°. Який зв'язок між подіями Click і OnClick?
- 3°. Що називають обробником події?
- 4°. Як пов'язати з подією її обробник?
- 5°. Що називають методом об'єкта?
- 6°. Що називають процедурою?
- 7°. Що називають операторними дужками? Для чого вони використовуються?
- 8°. Якими командами можна змінити значення властивостей об'єкта? Який їх загальний вигляд? Наведіть приклади таких команд.
- 9°. Якою командою можна відкрити вікно повідомлень?

**Виконайте завдання**

- 1°. Створіть проект, у якому подія **OnClick** для форми встановить її розміри 400 × 300 пікселів, зелений колір фону і змінить текст заголовка форми на назву вашого міста. Збережіть проект у папці з іменем **Завдання 5.2.1**, створеній у вашій папці.
- 2°. Створіть проект, у якому подія **OnKeyPress** для форми встановить її розміри 300 × 200 пікселів, синій колір фону і змінить текст заголовка форми на ваше прізвище. Збережіть проект у папці з іменем **Завдання 5.2.2**, створеній у вашій папці.
- 3°. Створіть проект, у якому подія **OnDbClick** (англ. *double* – подвійний) для форми спричинить змінення кольору її фону, заголовка і збільшить на 200 пікселів відступ її верхньої межі від верхньої межі екрана. Збережіть проект у папці з іменем **Завдання 5.2.3**, створеній у вашій папці.
- 4°. Створіть проект, у якому подія **OnShow** (англ. *show* – показувати) для форми викличе змінення її розмірів, кольору фону і зменшить на 100 пікселів відступ її лівої межі від лівої межі екрана. Збережіть проект у папці з іменем **Завдання 5.2.4**, створеній у вашій папці.
- 5°. Створіть проект, у якому подія **OnClick** для форми встановить її розміри 500 × 500 пікселів, коричневий колір фону і змінить текст заголовка форми на «Проект з двома процедурами», а подія **OnDbClick** відкриє вікно повідомлень з текстом «Я – громадянин України!». Збережіть проект у папці з іменем **Завдання 5.2.5**, створеній у вашій папці.
- 6°. Створіть проект, у якому подія **OnClose** для форми відкриє вікно повідомлень з текстом «Виконання проекту закінчено! До побачення!». Збережіть проект у папці з іменем **Завдання 5.2.6**, створеній у вашій папці.

ПРАКТИЧНА РОБОТА № 8**«Створення об'єктно-орієнтованої програми, що відображає вікно повідомлення»**

Увага! Під час роботи з комп'ютером дотримуйтеся правил поведінки та санітарно-гігієнічних норм.

1. Відкрийте вікно **Lazarus**.
2. Установіть такі значення властивостей форми:
 - колір фону – *сірий*;
 - ширина – *200 пікселів*;
 - висота – *100 пікселів*;
 - відступ лівої межі – *150 пікселів*;
 - відступ верхньої межі – *100 пікселів*;
 - текст у рядку заголовка – *Практична № 8*.
3. Збережіть проект у папці з іменем **Практична 8_1**, створеній у власній папці.
4. Виконайте проект.



5. Закрийте вікно виконання проекту.
6. Створіть обробник події **Click** для форми, виконання якого встановить червоний колір фону вікна, встановить відступ верхньої межі вікна 200 пікселів від верхньої межі екрана, збільшить його ширину на 300 пікселів, зменшить на 50 пікселів відступ лівої межі вікна від лівої межі екрана, відкриє вікно повідомлень з текстом «Ми вивчаємо мову програмування Object Pascal!».
7. Збережіть проект у папці з іменем **Практична 8_2**, створеній у вашій папці.
8. Виконайте проект.
9. Закрийте вікно виконання проекту.
10. Додайте до проекту обробник події **KeyPress** для форми, виконання якого змінить колір фону вікна на синій, збільшить на 300 пікселів відступ верхньої межі вікна від верхньої межі екрана, збільшить його ширину на 200 пікселів.
11. Збережіть проект у папці з іменем **Практична 8_3**, створеній у вашій папці.
12. Виконайте проект.
13. Закрийте вікно виконання проекту.
14. Закрийте вікно середовища **Lazarus**.




5.3. КНОПКА



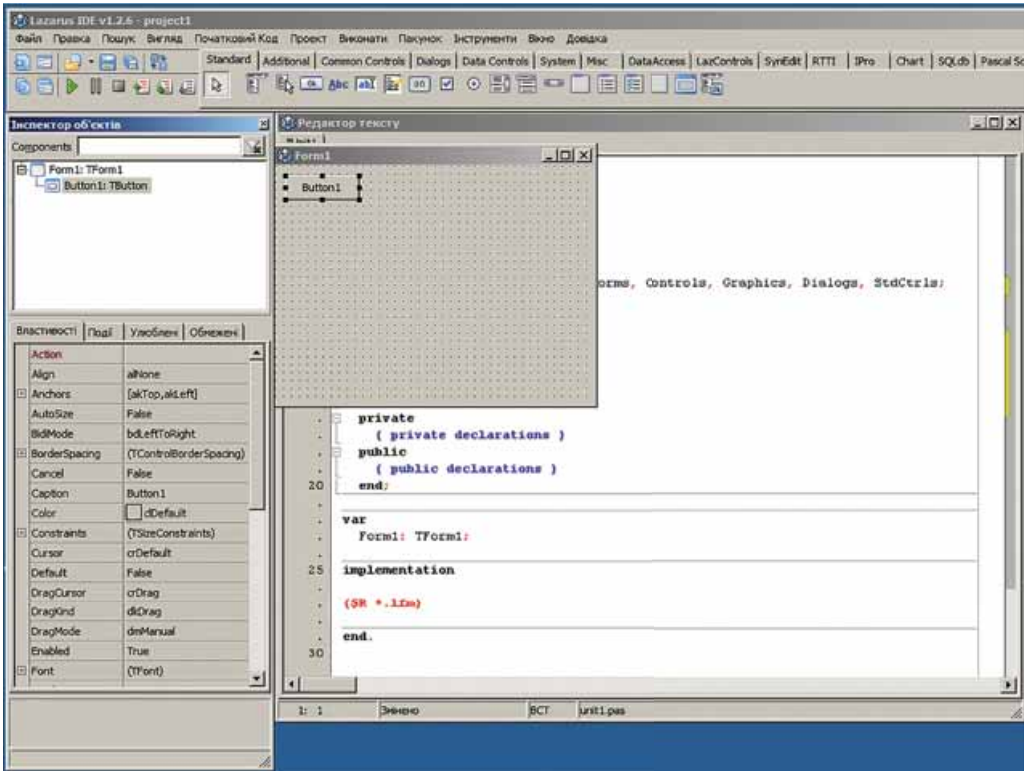
1. Які властивості форми ви знаєте? Як змінити значення її властивостей до початку виконання проекту і під час його виконання?
2. Які події можуть відбуватися з формою (вікном)? У чому полягає кожна з них?
3. Що таке обробник події? Як його створити? Як його виконати?

КНОПКА, ЇЇ ВЛАСТИВОСТІ ТА ОБРОБНИКИ ПОДІЙ ДЛЯ КНОПКИ

Одним з компонентів, який можна розмістити на формі, є **кнопка**.

Щоб розмістити кнопку на формі, потрібно підвести вказівник до її зображення  на вкладці **Standard** (англ. *standart* – стандарт) вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші. Після цього кнопка з'явиться у верхній лівій частині форми (мал. 5.15), а її ім'я додасться до списку компонентів проекту в полі **Components** (англ. *components* – компоненти) вікна **Інспектор об'єктів**. За потреби кнопку можна перетягнути в будь-яке інше місце форми.

Якщо вибрати кнопку на формі або вибрати її ім'я в полі **Components** вікна **Інспектор об'єктів**, то на вкладці **Властивості** цього вікна відобразиться список властивостей кнопки та їхніх значень. Як і для форми, значення деяких властивостей кнопки встановлені за замовчуванням. Значення властивостей кнопки можна змінювати. Для деяких властивостей їх



Мал. 5.15. Кнопка на формі

значення потрібно вводити у відповідні поля, для інших – можна вибирати зі списку; можна змінювати перетягуванням кнопки по формі або зміненням її розмірів, використовуючи маркери на її межах.

Усі властивості форми, які ми розглядали (**Caption, Color, Height, Width, Top, Left, Name**), є і у кнопки. Але змінення значення властивості **Color**, яке встановлено за замовчуванням, не змінює стандартний колір кнопки, значення властивості **Caption** визначає текст на самій кнопці, а значення властивостей **Top** і **Left** визначають відстань верхньої і лівої межі кнопки не від відповідних меж екрана, як для форми, а від відповідних меж форми, на якій розташовано цю кнопку.

Розглянемо ще кілька властивостей, які мають і кнопка, і форма:

- **Enabled** (англ. *enabled* – доступний) – її значення визначає, чи доступний об'єкт для операцій над ним. Ця властивість може набувати лише одного з двох значень: **True** (англ. *true* – істина, правда) – об'єкт доступний або **False** (англ. *false* – хибна, неправда) – об'єкт недоступний;
- **Visible** (англ. *visible* – видимий) – її значення визначає, чи видимий даний об'єкт у вікні. Ця властивість може також набувати лише одного з двох значень: **True** – об'єкт видимий або **False** – об'єкт невидимий.
- **Font** (англ. *font* – шрифт) – її значення визначає значення властивостей шрифту, яким буде виводитися текст на об'єкті. На відміну від усіх попередніх властивостей, ця властивість є **комплексною**. Так її назива-

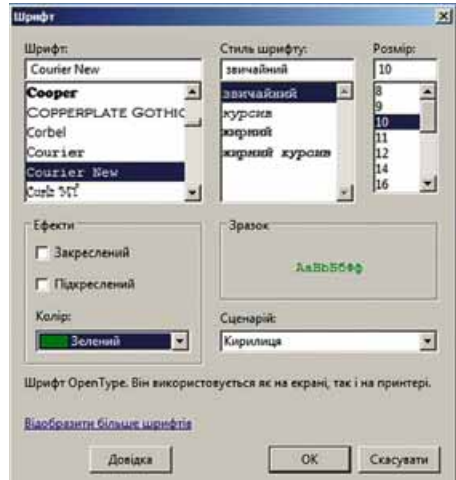
ють тому, що вона складається з кількох властивостей шрифту: **Color**, **Height**, **Name**, **Size** (англ. *size* – розмір) та ін. Значення кожної з них можна змінити, якщо відкрити їх список вибором кнопки ліворуч від назви властивості **Font**, або у вікні **Шрифт** (мал. 5.16), що відкривається вибором кнопки у правій крайній частині рядка **Font**. Ця кнопка стає доступною, якщо зробити поточним поле цього рядка.

Змінити, наприклад, колір шрифту в ході виконання проекту можна командою `Button1.Text.Color := clBlue`.

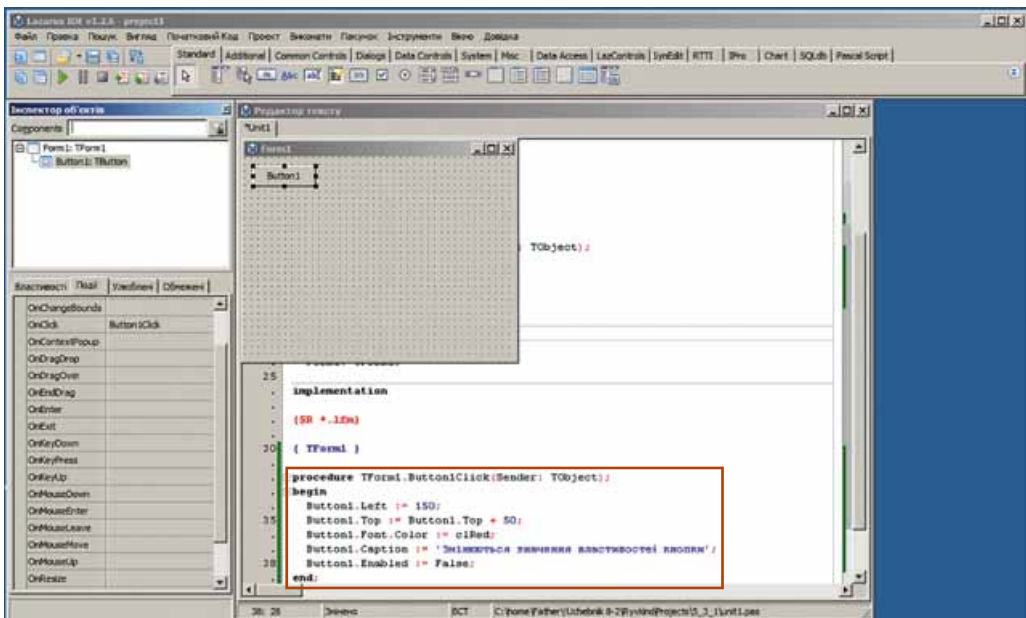
Якщо виділити кнопку, то на вкладці **Події** вікна **Інспектор об'єктів** можна, як і для форми, вибрати подію, для якої буде створено заготовку для тексту відповідної процедури – обробника цієї події. Як і для форми, можна ввести до цієї процедури команди змінення значень властивостей кнопки і форми та виконати створений проект.

На малюнку 5.17 наведено приклад обробника події **Click** для кнопки, який містить команди:

- встановлення відступу 150 пікселів лівої межі кнопки від лівої межі вікна;
- збільшення на 50 пікселів відступу верхньої межі кнопки від верхньої межі вікна;



Мал. 5.16. Вікно **Шрифт** для встановлення значень властивостей шрифту




Мал. 5.17. Обробник події **Click** для кнопки



- встановлення червоного кольору для шрифту, яким буде виводитися текст на кнопці;
- виведення на кнопці вказаного тексту;
- встановлення для кнопки режиму «Недоступна».



Найважливіше в цьому пункті

Щоб розмістити кнопку на формі, потрібно підвести вказівник до її зображення  на вкладці **Standard** вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші.

Кнопка, як і форма, має властивості **Caption, Color, Height, Width, Top, Left, Name**. Але змінення значення властивості **Color**, яке встановлено за замовчуванням, не змінює стандартний колір кнопки, значення властивості **Caption** визначає текст на самій кнопці, а значення властивостей **Top** і **Left** визначають відстань верхньої і лівої межі кнопки не від відповідних меж екрана, як для форми, а від відповідних меж форми, на якій розташовано цю кнопку.

Кнопка, як і форма, має також властивості:

- **Enabled** – її значення визначає, чи доступний об'єкт для операцій над ним. Ця властивість може набувати лише одного з двох значень: **True** (англ. *true* – істина, правда) – об'єкт доступний або **False** (англ. *false* – хиба, неправда) – об'єкт недоступний;
- **Visible** – її значення визначає, чи видимий даний об'єкт у вікні. Ця властивість може також набувати лише одного з двох значень: **True** – об'єкт видимий або **False** – об'єкт невидимий.
- **Font** – її значення визначає значення властивостей шрифту, яким буде виводитися текст на об'єкті. Ця властивість є **комплексною**. Так її називають тому, що вона складається з кількох властивостей шрифту: **Color, Height, Name, Size** та ін.



Дайте відповіді на запитання

- 1°. Як розмістити кнопку на формі?
- 2°. Які ви знаєте властивості кнопки?
- 3°. Що визначають значення відомих вам властивостей кнопки?
- 4°. Чим відрізняється змінення значень властивості **Color** для форми і для кнопки?
- 5°. Чим відрізняється властивість **Caption** для форми і для кнопки?
- 6°. Чим відрізняються властивості **Top** і **Left** для форми і для кнопки?
- 7°. Які значення можуть набувати властивості кнопки **Enabled** і **Visible**? Що визначає кожне із цих значень для кожної із цих властивостей?
- 8°. Чому властивість **Font** називають комплексною?



Виконайте завдання

- 1°. Створіть проект, у якому подія **Click** для кнопки встановить її розміри 40×30 пікселів і змінить текст на кнопці на назву вашого міста. Збережіть проект у папці з іменем **Завдання 5.3.1**, створений у вашій папці.



2°. Створіть проект, у якому подія **KeyPress** для кнопки встановить її розміри 20×40 пікселів і змінить текст на кнопці на ваше прізвище. Збережіть проект у папці з іменем **Завдання 5.3.2**, створеній у вашій папці.

3°. Створіть проект, у якому подія **MouseMove** для кнопки змінить текст на кнопці і збільшить відступ її верхньої межі на 20 пікселів. Збережіть проект у папці з іменем **Завдання 5.3.3**, створеній у вашій папці.



4°. Створіть проект, у якому подія **Click** для кнопки спричинить змінення її розмірів і зменшить відступ її лівої межі на 20 пікселів. Збережіть проект у папці з іменем **Завдання 5.3.4**, створеній у вашій папці.

5°. Створіть проект, у якому подія **Click** для кнопки встановить її розміри 50×50 пікселів, встановить коричневий колір тексту на кнопці і змінить текст на кнопці на «Мій проект», а подія **KeyPress** перемістить кнопку на 30 пікселів управо і на 40 пікселів униз. Збережіть проект у папці з іменем **Завдання 5.3.5**, створеній у вашій папці.

6*. Створіть проект, у якому розташуйте на формі три кнопки. Установіть такі значення їх властивостей, щоб на початку виконання проекту перша кнопка була видима і доступна, друга – видима і недоступна, третя – невидима і недоступна. Для кожної кнопки створіть обробник події **Click**, щоб:

а) вибір першої кнопки робив її невидимою і недоступною, другу кнопку – видимою і доступною, третю кнопку – видимою і недоступною;

б) вибір другої кнопки робив її невидимою і недоступною, третю кнопку – видимою і доступною, першу кнопку – видимою і недоступною;

в) вибір третьої кнопки робив її невидимою і недоступною, першу кнопку – видимою і доступною, другу кнопку – видимою і недоступною.

Збережіть проект у папці з іменем **Завдання 5.3.6**, створеній у вашій папці.



5.4. НАПИС



1. Які властивості кнопки ви знаєте? Як змінити значення її властивостей до початку виконання проекту і під час його виконання?

2. Які події можуть відбуватися з кнопкою? У чому полягає кожна з них?

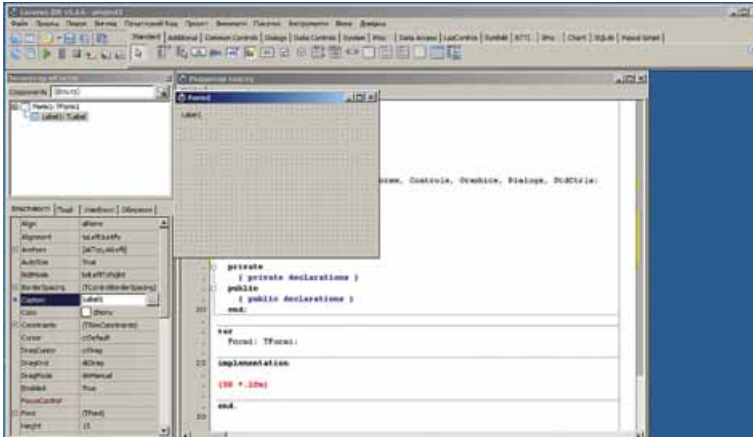
3. Які відмінності властивостей кнопки і форми ви знаєте?

НАПИС, ЙОГО ВЛАСТИВОСТІ ТА ОБРОБНИКИ ПОДІЙ ДЛЯ НАПISУ

Ще одним компонентом, який можна розмістити на формі, є **напис**. Цей компонент використовується для виведення текстових повідомлень.



Щоб розмістити напис на формі, потрібно підвести вказівник до кнопки **Abc TLabel** (англ. *label* – напис) на вкладці **Standard** вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші. Після цього напис з'явиться у верхній лівій частині форми (мал. 5.18), а його ім'я додається до списку компонентів проекту в полі **Components** вікна **Інспектор об'єктів**. За потреби напис можна перетягнути в будь-яке інше місце форми.



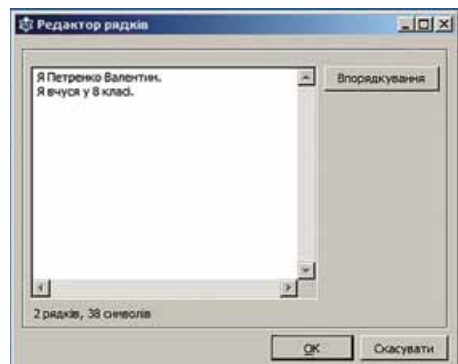
Мал. 5.18. Напис на формі

Виділити напис можна, вибравши його на формі або вибравши його ім'я в полі **Components** вікна **Інспектор об'єктів**. При вибраному написі на вкладці **Властивості** цього вікна відображається список його властивостей та їх значень. Значення властивостей напису можна змінювати. Для деяких властивостей їх значення потрібно вводити у відповідні поля, для інших – можна вибирати зі списку; можна змінювати використовуючи вказівник напису по формі.

Властивості напису **Top**, **Left**, **Name**, **Enabled**, **Visible**, **Font** аналогічні до відповідних властивостей кнопки. Значення властивості **Color** напису визначає колір його фону. Змінення значення цієї властивості приводить до зміни кольору фону напису, як це відбувається для форми.

На відміну від форми і кнопки, значення властивостей **Height** і **Width** для напису не можна змінити на вкладці **Властивості** або перетягуванням. Висота і ширина напису визначаються текстом, розташованим у цьому написі. Якщо змінити текст у полі **Caption**, то відповідно до його довжини зміниться ширина напису.

Текст напису можна вводити і редагувати як у полі **Caption**, так і у вікні **Редактор рядків** (мал. 5.19). У цьому вікні можна задавати також і багаторядкові тексти в написі.



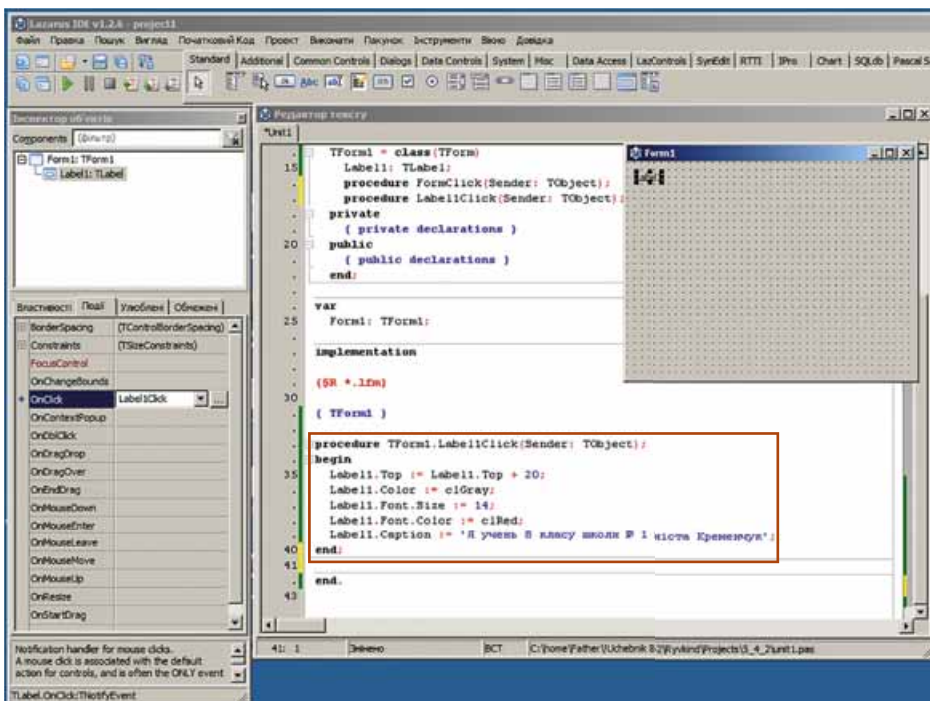
Мал. 5.19. Задання тексту напису у Редакторі рядків



Якщо виділити напис, то на вкладці Події вікна Інспектор об'єктів можна, як і для форми та кнопки, вибрати подію, для якої система створить заготовку процедури – обробника цієї події (мал. 5.20). Як і для форми та кнопки, можна ввести до цієї процедури команди змінення значень властивостей напису, а також виконати створений проект.

Наведена на малюнку 5.20 процедура – обробник події Click на написі містить команди:

- збільшення на 20 пікселів відступу верхньої межі напису від верхньої межі вікна;
- встановлення сірого фону напису;
- встановлення розміру 14 шрифту для виведення тексту напису;
- встановлення червоного кольору шрифту для виведення тексту напису;
- виведення у напис вказаного тексту.



Мал. 5.20. Обробник події Click для напису

Результат виконання цієї процедури наведено на малюнку 5.21.



Мал. 5.21. Результат виконання процедури, наведеної на малюнку 5.20



Для тих, хто хоче знати більше

Щоб у команді виведення тексту в напис текст у потрібному місці переносився на новий рядок, потрібно вставити до команди символ переходу на новий рядок (#13).

Наприклад, якщо в наведеному на малюнку 5.20 прикладі команду виведення тексту в напис змінити на таку:

`Label1.Caption := 'Я учень 8 класу школи № 1' + #13 + 'міста Кременчук';`, то результат виконання цієї процедури виглядатиме, як на малюнку 5.22.



Мал. 5.22. Результат виконання процедури, наведеної на малюнку 5.20, з перенесенням тексту



Найважливіше в цьому пункті

Напис – це компонент, який використовується для виведення текстових повідомлень.

Щоб розмістити напис на формі, потрібно підвести вказівник до кнопки **ABC TLabel** на вкладці **Standard** вікна середовища **Lazarus** і двічі клацнути лівою кнопкою миші.

Властивості напису **Top**, **Left**, **Name**, **Enabled**, **Visible**, **Font** аналогічні відповідним властивостям кнопки. Значення властивості **Color** напису визначає колір його фону. Змінення значення цієї властивості приводить до змінення кольору фону напису, як це відбувається для форми.

На відміну від форми і кнопки, значення властивостей **Height** і **Width** для напису не можна змінити на вкладці **Властивості** або перетягуванням. Висота і ширина напису визначаються текстом, розташованим у цьому написі. Якщо змінити текст у полі **Caption**, то відповідно до його довжини зміниться ширина напису.

Текст напису можна вводити і редагувати як у полі **Caption**, так і у вікні **Редактор рядків**.

Виділити напис можна, вибравши його на формі або вибравши його ім'я в полі **Components** вікна **Інспектор об'єктів**.

Якщо виділити напис, то на вкладці **Події** вікна **Інспектор об'єктів** можна вибрати подію, для якої система створить заготовку процедури – обробник цієї події.






Дайте відповіді на запитання

- 1°. Для чого використовується напис?
- 2°. Як розмістити напис на формі?
- 3°. Що визначають значення властивостей **Top**, **Left**, **Name** для напису?
- 4°. Що визначають значення властивостей **Enabled** і **Visible** для напису?
- 5°. Що визначають значення комплексної властивості **Font** для напису?
- 6°. Чим відрізняється властивість **Color** для напису і для кнопки?
- 7°. Чим відрізняються властивості **Top** і **Left** для напису і для форми?
- 8°. Чим відрізняються властивості **Height** і **Width** для напису і для форми?
- 9°. Як можна змінити текст у написі?



Виконайте завдання

- 1°. Створіть проект, у якому подія **Click** для напису встановить 120 пікселів відступ його верхньої межі від верхньої межі вікна і змінить текст на написі на назву вулиці, на якій знаходиться ваша школа. Збережіть проект у папці з іменем **Завдання 5.4.1**, створеній у вашій папці.
-  2°. Створіть проект, у якому подія **KeyPress** для напису встановить 80 пікселів відступ його лівої межі від лівої межі вікна і змінить текст на написі на ваше ім'я. Збережіть проект у папці з іменем **Завдання 5.4.2**, створеній у вашій папці.
- 3°. Створіть проект, у якому подія **MouseMove** для напису змінить текст на ньому, установить червоний колір тексту і збільшить відступ напису від верхньої межі вікна на 20 пікселів. Збережіть проект у папці з іменем **Завдання 5.4.3**, створеній у вашій папці.
-  4°. Створіть проект, у якому подія **Click** для напису зменшить його відступ від лівої межі вікна на 20 пікселів, установить розмір шрифту 12, колір шрифту синій і виведе в напис назву вашої школи. Збережіть проект у папці з іменем **Завдання 5.4.4**, створеній у вашій папці.
- 5°. Створіть проект, у якому подія **Click** для напису встановить коричневий колір фону і змінить текст на написі на «Мій проект», а подія **KeyPress** для напису перемістить його на 30 пікселів праворуч і на 40 пікселів униз. Збережіть проект у папці з іменем **Завдання 5.4.5**, створеній у вашій папці.
- 6°. Створіть проект, у якому подія **Click** для кнопки перемістить її на 60 пікселів праворуч і на 10 пікселів уверх і виведе в напис ваше прізвище та ім'я зеленим кольором на сірому фоні. Збережіть проект у папці з іменем **Завдання 5.4.6**, створеній у вашій папці.
-  7°. Створіть проект, у якому подія **Click** для кнопки перемістить вікно на 60 пікселів праворуч і на 80 пікселів уверх і виведе в напис ваше прізвище та ім'я сірим кольором на зеленому фоні. Збережіть проект у папці з іменем **Завдання 5.4.7**, створеній у вашій папці.

ПРАКТИЧНА РОБОТА № 9

«Створення програми з кнопками та написами»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте вікно середовища **Lazarus**.
2. Розмістіть на формі дві кнопки і напис.
3. Установіть такі значення властивостей першої кнопки:
 - ширина – 60 пікселів;
 - висота – 20 пікселів;
 - відступ від лівої межі форми – 120 пікселів;
 - відступ від верхньої межі форми – 100 пікселів;
 - текст на кнопці – *Форма*.
4. Установіть такі значення властивостей другої кнопки:



- ширина – 100 пікселів;
 - висота – 30 пікселів;
 - відступ від лівої межі форми – 300 пікселів;
 - відступ від верхньої межі форми – 100 пікселів;
 - текст на кнопці – *Напис*.
5. Установіть такі значення властивостей напису:
 - ширина – 120 пікселів;
 - висота – 40 пікселів;
 - відступ від лівої межі форми – 150 пікселів;
 - відступ від верхньої межі форми – 200 пікселів;
 - текст у написі – назва вашого класу.
 6. Збережіть проект у папці з іменем **Практична 9_1**, створеній у вашій папці.
 7. Виконайте проект.
 8. Закрийте вікно виконання проекту.
 9. Створіть обробник події **Click** для першої кнопки, виконання якого встановить ширину форми 800 пікселів, висоту форми 400 пікселів, колір фону форми зелений, ширину цієї кнопки 200 пікселів, збільшить її висоту на 10 пікселів, перемістить її на 50 пікселів ліворуч і на 30 пікселів уверх, зробить її недоступною.
 10. Створіть обробник події **MouseMove** для другої кнопки, виконання якого встановить червоний колір фону напису, відступ напису від верхньої межі вікна 200 пікселів, зменшить відступ напису від лівої межі вікна на 50 пікселів, установить колір тексту червоний і виведе у напис текст «Ми вивчаємо мову програмування Object Pascal!».
 11. Збережіть нову версію проекту.
 12. Виконайте нову версію проекту.
 13. Закрийте вікно виконання проекту.
 14. Закрийте вікно середовища **Lazarus**.



Розділ 6. Алгоритми роботи з об'єктами та величинами

У цьому розділі ви дізнаєтеся про:

змінні величини та константи, їх властивості

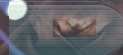
налагодження проектів

компоненти поле, прапорець, поле з розкритим списком, їх властивості й використання

використання у проектах розгалуження, циклів з лічильником і з передумовою

змінні та їх властивості

створення графічних об'єктів у ході виконання проектів





6.1. ПОЛЕ. ВЕЛИЧИНИ, СТАЛІ І ЗМІННІ ВЕЛИЧИНИ. ВЛАСТИВОСТІ ЗМІННИХ



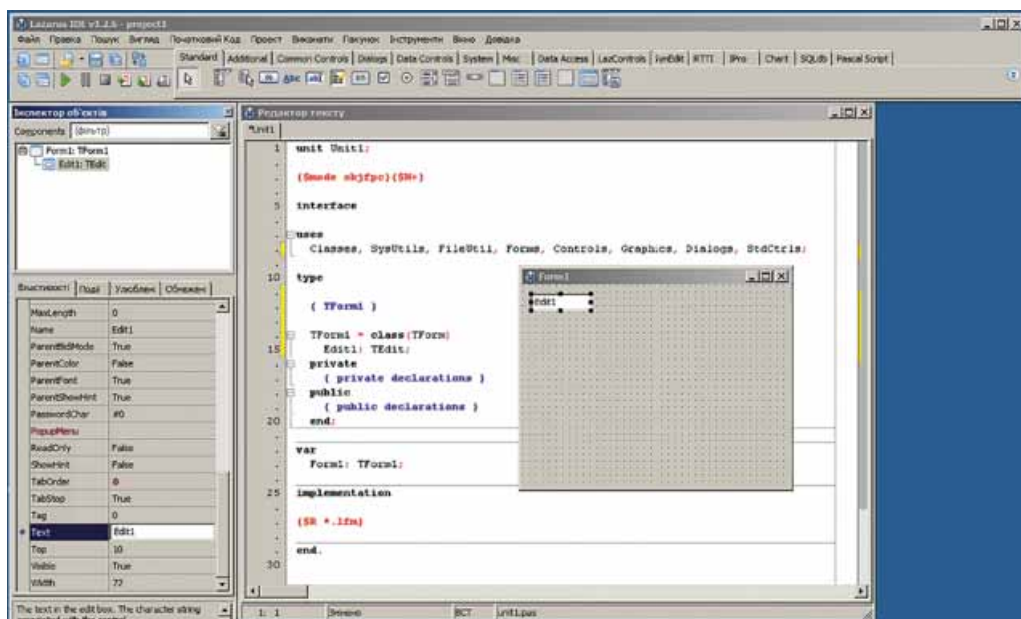
1. Для чого використовується команда присвоювання? Який вигляд може мати ця команда?
2. Які властивості має компонент **напис**? Що визначають значення цих властивостей?
3. Які властивості має компонент **кнопка**? Що визначають значення цих властивостей?

ПОЛЕ, ЙОГО ВЛАСТИВОСТІ ТА ОБРОБНИКИ ПОДІЙ ДЛЯ ПОЛЯ

Ще одним компонентом, який можна розмістити на формі, є **поле**. Поле – це компонент, який використовується для введення даних.

Щоб розмістити поле на формі, потрібно підвести вказівник до кнопки **TEdit** (англ. *edit* – редагувати) на вкладці **Standard** вікна середовища **Lazarus** і двічі клацнути ліву кнопку миші. Після цього поле з'явиться у верхній лівій частині форми (мал. 6.1), а його ім'я додається до списку компонентів проекту в полі **Components** вікна **Інспектор об'єктів**. За потреби поле можна перетягнути в будь-яке інше місце форми.

Виділити поле можна, вибравши його на формі або вибравши його ім'я в полі **Components** вікна **Інспектор об'єктів**. При вибраному полі список його властивостей та їх значень відображається на вкладці **Властивості** цього вікна. Значення властивостей поля можна змінювати. Для деяких властивостей їх значення потрібно вводити у відповідні поля на вкладці



Мал. 6.1. Поле на формі



Властивості, для деяких – можна вибирати зі списку; можна змінювати перетягуванням поля по формі.

Властивості поля **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font** аналогічні до відповідних властивостей напису. Як і для форми, і для кнопки, значення властивостей поля **Height** і **Width** можна змінити або на вкладці **Властивості**, або перетягуванням спеціальних маркерів на межах поля. Текст у полі є значенням його властивості **Text** (англ. *text* – текст). Його можна вводити і редагувати у відповідному полі вкладки **Властивості**. Часто перед початком виконання проекту значення цієї властивості доцільно зробити порожнім.

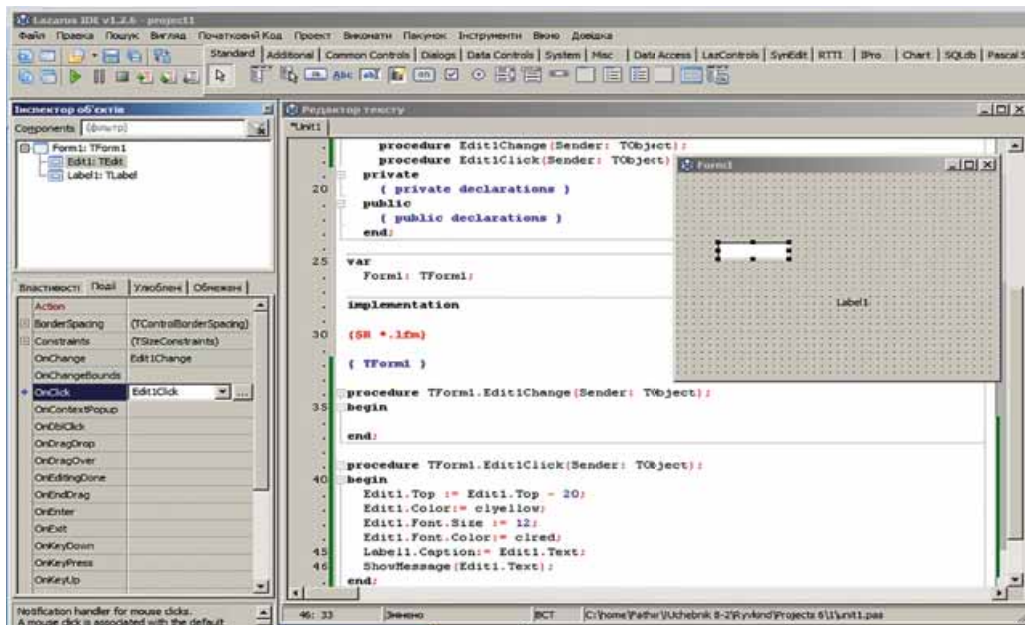
Якщо виділити поле, то на вкладці **Події** вікна **Інспектор об'єктів** можна, як і для розглянутих раніше компонентів, вибрати подію, для якої система створить заготовку процедури – обробника цієї події. Наведена на малюнку 6.2 процедура – обробник події **Click** для поля – містить команди:

- зменшення на 20 пікселів відступу верхньої межі поля від верхньої межі вікна;
- установлення жовтого фону поля;
- установлення розміру 12 шрифта для тексту в полі;
- установлення червоного кольору шрифта для тексту в полі;
- виведення тексту, який введений у поле, у напис і у вікно повідомлень.

Після запуску проекту на виконання можна ввести в поле довільний текст (на мал. 6.3 у поле



Мал. 6.3. Вікно виконання проекту після введення тексту в поле і до настання події **Click** для поля

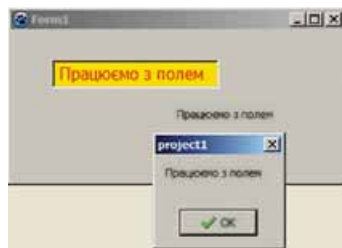


Мал. 6.2. Обробник події **Click** для поля



введено текст «Працюємо з полем»), установити вказівник на полі й клацнути ліву кнопку миші. Результат виконання процедури – обробника події **Click** для поля – наведено на малюнку 6.4.

Незважаючи на те що основним призначенням поля є введення даних, воно може використовуватися і для виведення даних. Для цього в ході виконання проекту значенню його властивості **Text** потрібно присвоїти певний текст. Наприклад, `Edit1.Text := 'Ми учні 8 класу'`. Якщо ширина поля буде меншою від довжини тексту, то, на відміну від напису, ширина поля не збільшиться і частина тексту буде знаходитися за межами поля та буде невидима користувачеві.



Мал. 6.4. Вікно виконання проекту після настання події **Click** для поля

ВЕЛИЧИНИ. СТАЛІ ТА ЗМІННІ ВЕЛИЧИНИ

Інформатика, математика, фізика та інші науки використовують величини.

Величина – це кількісно виражене значення властивості об'єкта.

Прикладами величин в інформатиці є, наприклад, ширина і висота вікна, колір його фону, напис на кнопці, її розташування на формі та ін. Прикладами величин у математиці є довжина і ширина прямокутника, його площа, міра кута та ін. Прикладами величин у фізиці є довжина шляху, час, швидкість, густина речовини та ін.

Величину, значення якої не змінюється, називають **сталою величиною**, або **константою**.

Прикладом математичної константи є, наприклад, число π . Ви знаєте, що наближене значення цього числа дорівнює 3,1416. Прикладами фізичних констант є густини речовин, наприклад густина повітря наближено дорівнює $1,293 \text{ кг/м}^3$.

Величину, значення якої може змінюватися, називають **змінною величиною**, або **змінною**.

ВИКОРИСТАННЯ ПОЛЯ ДЛЯ ВВЕДЕННЯ ЧИСЕЛ

Створимо проект, у якому буде обчислюватися сума двох довільних дійсних чисел.

Розмістимо на формі два поля, у які користувач під час виконання проекту вводитиме доданки, напис для виведення суми і кнопку, після вибору якої відбуватиметься додавання введених у поля чисел і виведення суми в напис.

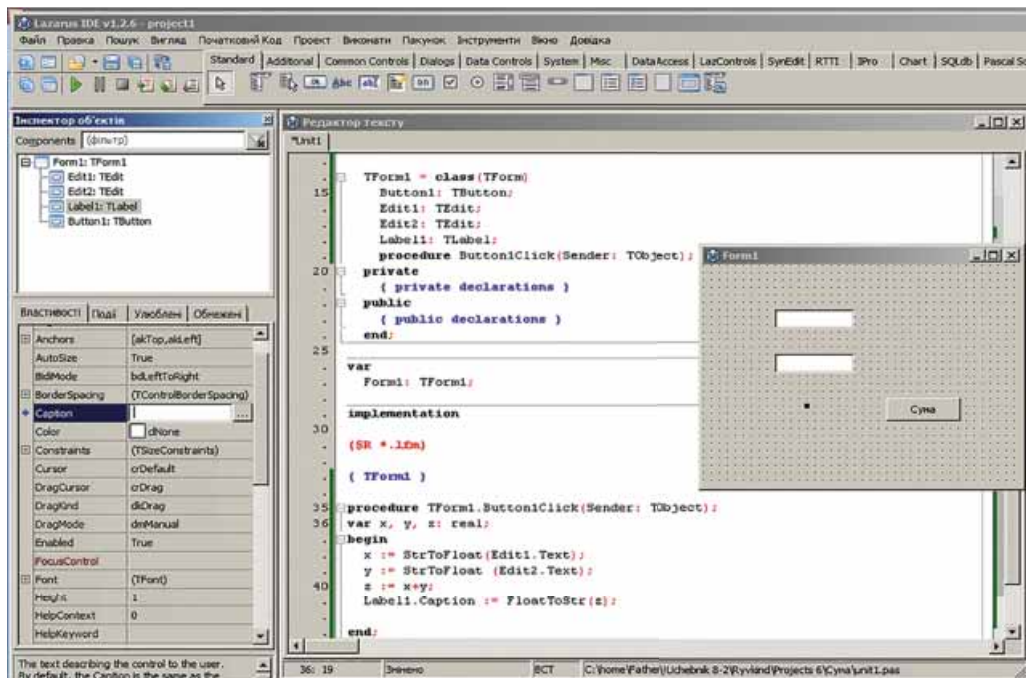
Форму для такого проекту з розміщеними на ній об'єктами, указаними вище, а також процедуру для додавання двох дійсних чисел наведено на малюнку 6.5.

Розглянемо наведену процедуру.

Хоча користувач під час виконання проекту вводить у поля числа, однак система сприймає їх як тексти. Ці тексти є значеннями властивості **Text** полів з іменами **Edit1** і **Edit2**. Тому перші дві команди процедури



$x := \text{StrToFloat}(\text{Edit1.Text})$ і $y := \text{StrToFloat}(\text{Edit2.Text})$ (англ. *string to float* – рядок у число з плаваючою десятковою комою) переводять текстове подання дійсних чисел безпосередньо в дійсні числа і присвоюють їх змінним x і y відповідно.



Мал. 6.5. Форма з двома полями, написом і кнопкою та процедура для додавання двох дійсних чисел

Третя команда додає ці числа і присвоює суму змінній z .

У четвертій команді `Label1.Caption := FloatToStr(z)` (англ. *float to string* – число з плаваючою десятковою комою в рядок) переводить обчислену суму із числа в його текстове подання і присвоює цей текст значенню властивості `Caption` напису з іменем `Label1`.

У мові програмування **Object Pascal** для будь-якої змінної повинен бути вказаний її тип.

Тип змінної визначає:

- яких значень може набувати ця змінна;
- які операції над нею можна виконувати;
- який обсяг оперативної пам'яті буде виділено для зберігання значення цієї змінної.

Імена змінних та їх типи вказуються в проекті після ключового слова `var` (англ. *variable* – змінна).

Значення змінних x , y , z , які використано у процедурі, за умовою задачі можуть бути довільними дійсними числами. Для таких змінних можна використати тип `real` (англ. *real* – дійсний). Змінні цього типу можуть набувати цілих і дробових значень, модуль яких не перевищує $1,7 \cdot 10^{38}$.



Для кожного із цих значень буде виділено 8 байтів пам'яті. Змінні, що використовуються у процедурі, та їх типи описуються в рядку **var**, який розташований після рядка заголовка процедури (мал. 6.5).

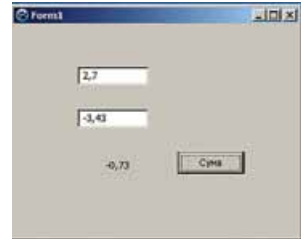
Якщо змінні можуть набувати тільки цілих значень, то для них можна використовувати тип **integer** (англ. *integer* – цілий) або тип **longint** (англ. *long integer* – довге ціле). Змінні першого із цих типів можуть набувати цілих значень від **-32 768** до **32 767**, а другого – цілих значень від **-2 147 483 648** до **2 147 483 647**. Для кожного із цих значень буде виділено 4 байти пам'яті.

У рядку **var**, який розташований поза процедурами (мал. 6.5), описуються ті змінні та їх типи, які можуть використовуватися в усіх процедурах проекту.

Якщо в рядку **var** описуються змінні різних типів, то він матиме вигляд, наприклад такий: **var x, y: real; t: integer;**

Під час виконання створеного проекту відкривається вікно. Користувач вводить у поля два числа (на мал. 6.6 це числа 2,7 і -3,43), вибирає кнопку із заголовком *Сума* і одержує результат (-0,73). Не закриваючи вікна виконання проекту, користувач може ввести в поля інші числа, знову вибрати кнопку із заголовком *Сума* і одержати новий результат. І так можна повторювати довільну кількість разів.

Щоб під час виконання проекту користувачеві було зрозуміліше, що робити, доцільно на формі розмістити ліворуч від кожного поля і від напису, призначеного для виведення результату, написи з пояснювальними текстами, наприклад: *Перший доданок*, *Другий доданок*, *Сума* (мал. 6.7).



Мал. 6.6. Вікно виконання проекту додавання двох дійсних чисел



Мал. 6.7. Вікно виконання проекту додавання двох дійсних чисел з пояснювальними текстами

ТИП РЕЗУЛЬТАТІВ ОПЕРАЦІЙ НАД ЗМІННИМИ ЧИСЛОВИХ ТИПІВ

Ви вже знаєте, що над змінними числових типів можна виконувати операції додавання, віднімання, множення і ділення.

Якщо в командах $x := a + b$, $x := a - b$, $x := a * b$ і $x := a / b$ змінні a і b є змінними типу **real**, то і тип змінної x має бути визначений як **real**.

Якщо в командах $x := a + b$, $x := a - b$, $x := a * b$ змінні a і b є змінними типу **integer (longint)**, то тип змінної x може бути визначений і як **integer (longint)**, і як **real**.

А от якщо в команді $x := a / b$ змінні a і b є змінними типу **integer (longint)**, то тип змінної x має бути визначений як **real**.

ВИКОРИСТАННЯ КОНСТАНТ У ПРОЕКТІ

У проекті, крім змінних, можна використовувати константи.

Створимо проект, у якому потрібно обчислити масу сосни, якщо відомо її об'єм.

З курсу фізики відомо, що густина сосни є константою і дорівнює 520 кг/м^3 , а масу можна обчислити за формулою $m = \rho V$.

Вважатимемо, що значення об'єму вводиться в кубічних метрах у поле з іменем **Edit1**, а значення маси виводиться в кілограмах у напис з іменем **Label1**.

Тоді відповідний фрагмент проекту матиме такий вигляд:

```
const p = 520;
var x, m: real;
begin
  x := StrToFloat(Edit1.Text);
  m := x*p;
  Label1.Caption := FloatToStr(m);
end;
```

Найважливіше в цьому пункті

Поле – це компонент, який використовується для введення даних.

Властивості поля **Top**, **Left**, **Name**, **Color**, **Enabled**, **Visible**, **Font** аналогічні до відповідних властивостей напису. Значення властивостей поля **Height** і **Width** можна змінити або на вкладці **Властивості**, або перетягуванням маркерів на межах поля. Текст у полі є значенням його властивості **Text**. Його можна вводити і редагувати у відповідному полі вкладки **Властивості**.

Поле може використовуватися і для виведення даних. Для цього його властивості **Text** потрібно присвоїти певний текст.

Величина – це кількісно виражене значення властивості об'єкта.

Стала величина (константа) – величина, значення якої не змінюється.

Змінна величина (змінна) – величина, значення якої може змінюватися.

У мові програмування **Object Pascal** для будь-якої змінної повинен бути вказаний її **тип**. **Тип змінної** визначає:

- яких значень може набувати ця змінна;
- які операції над нею можна виконувати;
- який обсяг оперативної пам'яті буде виділено для зберігання значення цієї змінної.

У мові програмування **Object Pascal** можуть бути використані константи.

Дайте відповіді на запитання

- 1°. Для чого може використовуватися поле?
- 2°. Як розмістити поле на формі?
- 3°. Що визначають значення властивостей **Top**, **Left**, **Name**, **Color**, **Text**, **Enabled** і **Visible** для поля?
- 4°. Що визначають значення комплексної властивості **Font** для поля?
- 5°. Чим відрізняються властивості **Top** і **Left** для поля та для форми?
- 6°. Чим відрізняються властивості **Height** і **Width** для поля та для напису?
- 7°. Як можна змінити текст у полі?
- 8°. Якими командами можна перевести текстове подання числа безпосередньо в число і навпаки?
- 9°. Що визначає тип змінної?
- 10°. Де і як вказуються імена змінних та їх типи?

**Виконайте завдання**

1°. Створіть проект, у якому подія **Click** для поля встановить 100 пікселів відступ його верхньої межі від верхньої межі вікна і змінить текст у полі на слово «Інформатика». Збережіть проект у папці з іменем **Завдання 6.1.1**, створеній у вашій папці.



2°. Створіть проект, у якому подія **KeyPress** для поля встановить 120 пікселів відступ його лівої межі від лівої межі вікна і змінить текст у полі на ваше ім'я. Збережіть проект у папці з іменем **Завдання 6.1.2**, створеній у вашій папці.



3°. Створіть проект, у якому подія **MouseMove** для поля змінить текст у ньому, встановить червоний колір тексту, сірий колір фону і перемістить поле на 100 пікселів праворуч і на 80 пікселів униз. Збережіть проект у папці з іменем **Завдання 6.1.3**, створеній у вашій папці.



4°. Створіть проект, у якому подія **Click** для поля перемістить його на 80 пікселів праворуч і на 20 пікселів вверх і виведе в поле ваше прізвище та ім'я зеленим кольором на жовтому фоні. Збережіть проект у папці з іменем **Завдання 6.1.4**, створеній у вашій папці.

5°. Створіть проект для обчислення значення виразу $(2a + b) \cdot (a + c)$. Збережіть проект у папці з іменем **Завдання 6.1.5**, створеній у вашій папці.

6°. Створіть проект для віднімання двох дійсних чисел. Збережіть проект у папці з іменем **Завдання 6.1.6**, створеній у вашій папці.

7°. Створіть проект для обчислення значення виразу $(3a - 2b) \cdot c$. Збережіть проект у папці з іменем **Завдання 6.1.7**, створеній у вашій папці.

8°. Створіть проект для обчислення довжини кола, використовуючи константу. Збережіть проект у папці з іменем **Завдання 6.1.8**, створеній у вашій папці.

ПРАКТИЧНА РОБОТА № 10**«Складання та виконання лінійних алгоритмів опрацювання величин»**

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте вікно середовища **Lazarus**.

2. Розмістіть на формі кнопку і поле.

3. Установіть такі значення властивостей кнопки:

- ширина – 70 пікселів;
- висота – 30 пікселів;
- відступ від лівої межі форми – 150 пікселів;
- відступ від верхньої межі форми – 200 пікселів;
- текст на кнопці – *Властивості поля*.

4. Установіть такі значення властивостей поля:

- ширина – 200 пікселів;
- висота – 50 пікселів;
- відступ від лівої межі форми – 70 пікселів;



- відступ від верхньої межі форми – 100 пікселів;
 - текст у полі – назва вашого класу.
5. Збережіть проект у папці з іменем **Практична 10_1**, створеній у вашій папці.
 6. Виконайте проект.
 7. Закрийте вікно виконання проекту.
 8. Створіть обробник події **Click** для кнопки, виконання якого встановить червоний колір фону поля, відступ поля від верхньої межі вікна 150 пікселів, зменшить відступ поля від лівої межі вікна на 50 пікселів, установить колір тексту червоний і виведе в поле текст «Ми використовуємо властивості поля!».
 9. Збережіть нову версію проекту.
 10. Виконайте нову версію проекту.
 11. Закрийте вікно виконання проекту.
 12. Створіть новий проект для обчислення значення виразу $a \cdot (4b - 3c)$.
 13. Збережіть проект у папці з іменем **Практична 10_2**, створеній у вашій папці.
 14. Виконайте проект.
 15. Закрийте вікно виконання проекту.
 16. Закрийте вікно середовища **Lazarus**.



6.2. НАЛАГОДЖЕННЯ ПРОЕКТУ



1. Яке призначення компілятора? Які особливості його роботи?
2. Які помилки називають синтаксичними? Як вони виявляються на етапі компіляції проекту?
3. Як відкрити раніше створений проект?

ТЕСТУВАННЯ ПРОЕКТУ

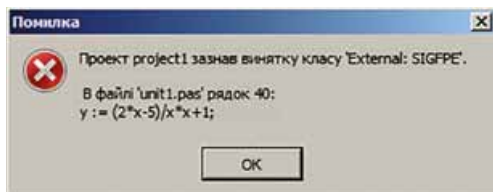
Ви вже знаєте, що під час створення проекту можуть виникати синтаксичні помилки. Синтаксичні помилки, які не помітив автор проекту, знаходить компілятор на етапі компіляції під час запуску проекту на виконання. Потрібно виправити всі такі помилки і знову запустити на виконання нову версію проекту.

Якщо виконання проекту неочікувано переривається або результат його виконання неправильний, то говорять, що проект містить **логічні помилки**.

Одним зі способів виявлення логічної помилки є створення **тестових наборів** вхідних даних. Для кожного із цих наборів даних визначають очікуваний результат виконання проекту і порівнюють його з реальним результатом виконання проекту. У разі їх розбіжностей потрібно знайти команди у проекті, виконання яких є причиною появи цих розбіжностей.

Процес пошуку логічних помилок у проекті з використанням тестових наборів вхідних даних називають **тестуванням проекту**.

Пояснимо цей спосіб на такому прикладі. Нехай потрібно скласти проект для обчислення значень функції $y = \frac{2x - 5}{x^2 + 1}$. Процедура для обчислення



Мал. 6.8. Вікно з повідомленням про помилку

```

procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text);
  y := (2*x-5)/x*x+1;
  Label1.Caption := FloatToStr(y);
end;

```

Мал. 6.9. Текст процедури із зафарбованим рядком, що містить помилку

значень цієї функції, у якій значення змінної x вводиться в поле, матиме такий вигляд:


```

procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text);
  y := (2*x-5)/(x*x+1);
  Label1.Caption := FloatToStr(y);
end;

```

Часто учні, записуючи команду присвоювання для обчислення значення відповідного виразу $y := (2*x - 5)/(x*x + 1)$, забувають узяти знаменник (а то й чисельник) у дужки, тобто вводять, наприклад, команду $y := (2*x - 5)/x*x + 1$ або $y := 2*x - 5/x*x + 1$. Таку логічну помилку нескладно помітити, застосувавши вдалу добірку тестів. Так, під час тестування для $x = 0$ стане очевидним, що проект не видає очікуваного правильного результату -5 , а неочікувано перериває своє виконання, відкриваючи вікно з повідомленням про помилку (мал. 6.8).

У цьому вікні, крім повідомлення про помилку, вказується номер і вміст рядка з помилкою. Після вибору кнопки **ОК** це вікно закривається, а рядок з помилкою в тексті процедури стає зафарбованим (мал. 6.9).

Щоб перервати виконання проекту в такому випадку, потрібно натиснути сполучення клавіш **Ctrl + F2** або вибрати кнопку  **Завершити**. Далі потрібно зрозуміти, у чому полягає помилка, виправити її і знову запустити проект на виконання. У розглянутому випадку це відбувається тому, що в ході виконання проекту із цим значенням x має виконатися операція ділення на 0, яку виконати неможливо.

Якщо як тестовий приклад буде взято інше значення x , наприклад -1 , то виконання проекту не переривається, вікно з повідомленням про помилку не відкривається, але як результат буде виведено число -6 , хоча очікуваним правильним результатом є число $-3,5$. У такому випадку також потрібно уважно переглянути текст процедури, виявити помилку і виправити її.



Для тих, хто хоче знати більше

Досить часто для полегшення пошуку логічних помилок у проект додають **коментарі**. **Коментар** – це текст, який ігнорується компілятором і включається до тексту проекту з метою полегшити його розуміння та пошук логічних помилок.



Щоб компілятор відрізняв команди від коментарів, коментарі потрібно брати у фігурні дужки, наприклад **{Команда обчислення значення у}** (мал. 6.10). Інколи коментарі замість фігурних дужок беруть у круглі дужки із зірочками, наприклад **(*Команда виведення значення у*)**.

Якщо текст коментаря короткий і не займає більше ніж один рядок, то можна замість дужок поставити перед ним дві похилі риски, наприклад **//Команда введення значення х**.

```
procedure TForm1.Button1Click(Sender: TObject);
var x, y: real;
begin
  x := StrToFloat(Edit1.Text); //Команда введення значення х
  (*Команда обчислення значення у)
  y := (2*x-5)/(x*x+1);
  (*Команда виведення значення у*)
  Label1.Caption := FloatToStr(y);
end;
```

Мал. 6.10. Текст процедури з логічною помилкою і коментарями

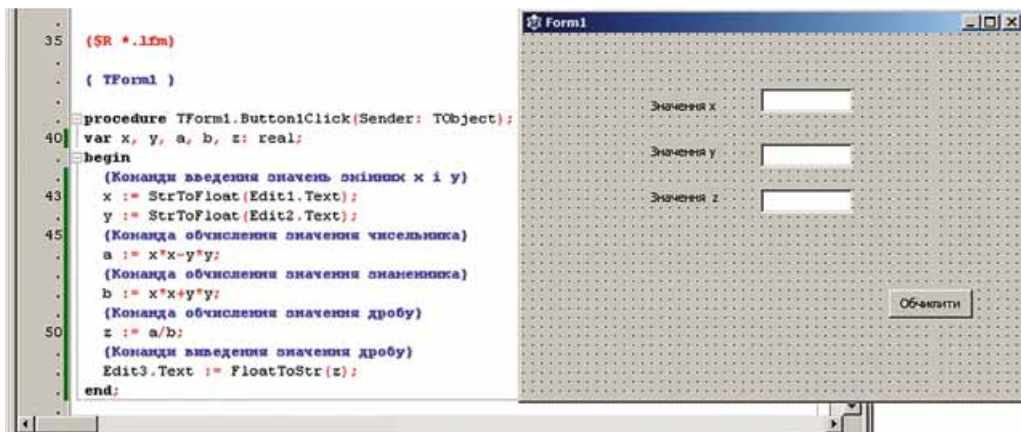
Якщо коментар займає не більше ніж один рядок, то він може знаходитися після команди, а може розпочинатися з початку рядка. Будь-який коментар не може знаходитися всередині команди.

ПОКРОКОВЕ ВИКОНАННЯ ПРОЕКТУ

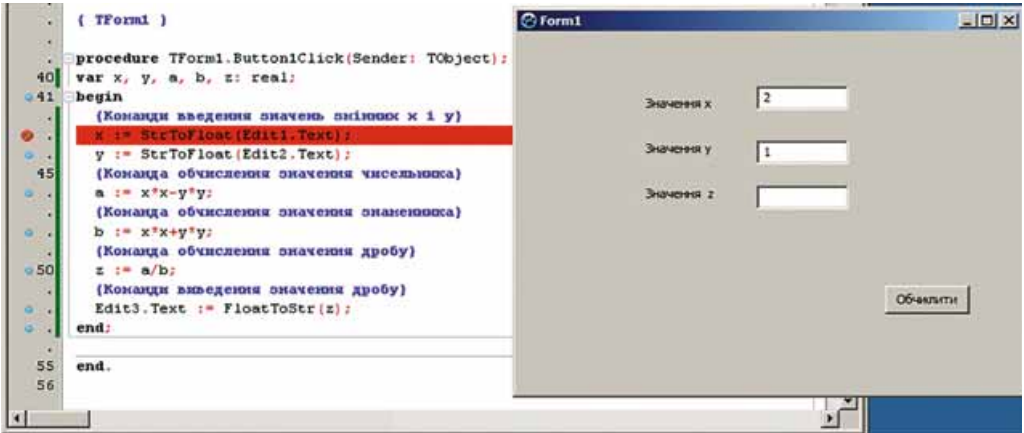
Ще одним способом виявлення в проекті логічних помилок є його **покрокове виконання**, або **трасування**. **Крок виконання проекту** – це виконання команд, записаних в одному рядку проекту. У рядку може бути записано одну команду або кілька.

Розглянемо проект для обчислення значення дробу $z = \frac{x^2 - y^2}{x^2 + y^2}$. Відповідну процедуру для обчислення значення цього дробу і форму подано на малюнку 6.11. Для кращого розуміння процесу покрокового виконання проекту в процедурі значення чисельника і знаменника дробу обчислено окремими командами.

Щоб швидше перейти до покрокового виконання наведеної процедури, доцільно здійснити виконання тієї частини проекту, що передуює виконанню цієї процедури, в автоматичному режимі. Для цього потрібно встановити **точку зупинки** в першому рядку процедури (мал. 6.12) і запустити проект на виконання. Якщо в деякому рядку проекту встановлено точку зупинки, то після запуску проекту на виконання в автоматичному режимі



Мал. 6.11. Проект для обчислення значення дробу $z = \frac{x^2 - y^2}{x^2 + y^2}$



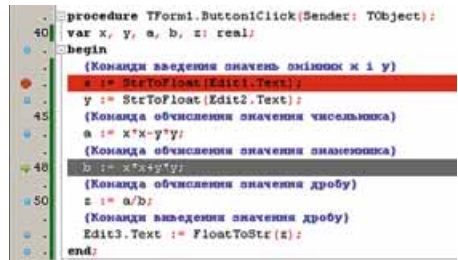
Мал. 6.12. Точка зупинки

виконається частина проекту від самого початку і до цього рядка, не виконуючи команди саме цього рядка.

Для встановлення точки зупинки в потрібному рядку слід вибрати номер цього рядка в тексті проекту. Після цього фон вибраного рядка буде зафарбовано червоним кольором.

Після запуску проекту на виконання потрібно у вікні, що відкрилося, ввести у відповідні поля значення змінних x і y та вибрати кнопку **Обчислити**. Після цього розпочинається виконання наведеної процедури. Але оскільки в першій команді цієї процедури встановлено точку зупинки, то виконання процедури призупиниться і можна розпочинати її покрокове виконання. Для виконання одного кроку проекту потрібно виконати **Виконати** \Rightarrow **Переступити** або натиснути клавішу **F8**. Під час покрокового виконання фон рядка проекту, команди якого будуть виконані під час наступного кроку виконання, фарбується в сірий колір (мал. 6.13).

У будь-який момент покрокове виконання проекту можна перервати, натиснувши сполучення клавіш **Ctrl + F2**, або вибравши кнопку **Завершити**, або вибравши кнопку **Виконати**. У перших двох випадках виконання проекту переривається, а у третьому виконання проекту продовжується в автоматичному режимі.



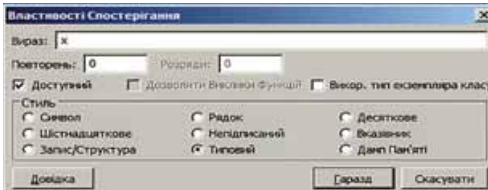
Мал. 6.13. Покрокове виконання

ПЕРЕГЛЯД ЗНАЧЕНЬ ЗМІННИХ ПІД ЧАС ТРАСУВАННЯ

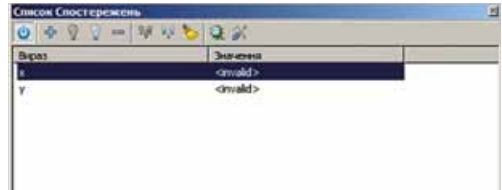
Покрокове виконання проекту не має жодного сенсу, якщо в ході такого виконання не бачити і не спостерігати, як змінюються значення змінних. Саме таке спостереження може допомогти нам зрозуміти, які команди є причиною логічної помилки.

Щоб під час покрокового виконання проекту спостерігати за зміненням значень змінних, потрібно імена цих змінних помістити до вікна **Список Спостережень**. Це можна зробити, використавши вікно **Властивості Спостереження**.

Щоб помістити ім'я змінної до вікна **Властивості Спостереження**, потрібно натиснути сполучення клавіш **Ctrl + F5** або виконати **Виконати** ⇒ **Додати спостереження** (мал. 6.14). Ім'я змінної потрібно ввести в поле **Вираз** цього вікна і вибрати кнопку **Гаразд**.



Мал. 6.14. Вікно **Властивості Спостереження**

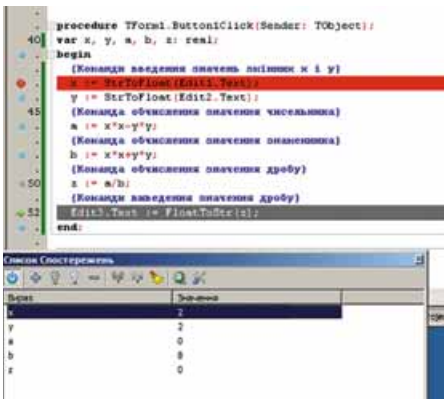


Мал. 6.15. Вікно **Список Спостережень**

Після цього відкривається вікно **Список Спостережень** (мал. 6.15) із внесеним до нього іменем змінної. Таким чином можна до цього вікна внести імена всіх тих змінних, за значеннями яких доцільно спостерігати під час покрокового виконання програми. Вибір кнопки **+** **Додати** цього вікна також відкриває вікно **Властивості Спостереження**, вибір кнопки **-** **Вилучити** вилучає із цього вікна виділений рядок з іменем змінної та її значенням, а вибір кнопки **🗑️** **Видалити все** повністю видаляє список змінних, за якими ведеться спостереження.

До початку виконання проекту значення змінних, які використовуються у проекті, не визначені. Тому у вікні **Список Спостережень** ми бачимо значення цих змінних **invalid** (англ. *invalid* – недійсний, неіснуючий).

Під час покрокового виконання проекту значення цих змінних змінюватимуться, що й можна спостерігати у вікні **Список Спостережень** (мал. 6.16). Уважно стежачи за цими змінами і порівнюючи ці значення з очікуваними, можна помітити, у момент виконання якої команди проекту значення певної змінної стає неправильним. Саме ця команда й може бути причиною логічної помилки.



Мал. 6.16. Значення змінних у вікні **Список Спостережень** під час покрокового виконання проекту

Найважливіше в цьому пункті

Логічна помилка – це помилка у проекті, що призводить до неочікуваного переривання його виконання або до неправильного результату його виконання.



Тестовий набір вхідних даних – це набір значень вхідних даних, для яких можна заздалегідь визначити правильний результат виконання проекту, після чого порівняти його з результатом виконання проекту для цих значень вхідних даних.



Тестування проекту – це процес пошуку логічних помилок у проекті з використанням тестових наборів вхідних даних.

Коментар – це текст, який ігнорується компілятором і включається до тексту проекту з метою полегшення його розуміння і пошуку логічних помилок. Коментар потрібно взяти у фігурні дужки {}, або у круглі дужки із зірочками (* *), або відокремити від команд двома похилими рисками //.

Точка зупинки – рядок у проекті, перед виконанням якого виконання проекту призупиняється.

Крок виконання проекту – це виконання команд, записаних в одному рядку проекту. У рядку може бути записано одну команду або кілька.

Для виконання одного кроку проекту потрібно виконати **Виконати** ⇒ **Переступити** або натиснути клавішу F8.

Покрокове виконання проекту можна перервати, натиснувши сполучення клавіш **Ctrl + F2**, або вибравши кнопку  **Завершити**, або вибравши кнопку  **Виконати**.

Щоб під час покрокового виконання проекту стежити за зміненням значень змінних, потрібно імена цих змінних помістити до вікна **Список Спостережень**. Це можна зробити, використавши вікно **Властивості Спостереження**.

Щоб помістити ім'я змінної до вікна **Властивості Спостереження**, потрібно натиснути сполучення клавіш **Ctrl + F5** або виконати **Виконати** ⇒ **Додати спостереження**. Ім'я змінної потрібно ввести в поле **Вираз** цього вікна і вибрати кнопку **Гаразд**.



Дайте відповіді на запитання

- 1°. У чому полягає процес тестування проекту?
- 2°. Поясніть, для чого виконується тестування проекту.
- 3°. Що таке коментар у проекті? Як можна коментар включити до тексту проекту?
- 4°. Поясніть, для чого в текст проекту включають коментарі.
- 5°. Для чого проект виконують покроково?
- 6°. Поясніть, у чому полягає процес покрокового виконання проекту.
- 7°. Що таке крок виконання проекту? Як здійснити його виконання?
- 8°. Що таке точка зупинки? Для чого вона використовується?
- 9°. Як перервати режим покрокового виконання проекту?
- 10°. Як переглянути значення змінних під час покрокового виконання програми?



Виконайте завдання

- 1°. Відкрийте проект, наприклад з папки **Тема 6\Завдання 6.2\Проект 6.2.1**, призначений для розв'язування такої задачі: «*Два потяги виїхали одночасно назустріч один одному. Визначити, через який час вони зустрінуться, якщо відома відстань між ними в момент по-*



чатку руху та швидкість кожного потяга». Виконання обробника події **Click** для кнопки повинно привести до виведення шуканого часу, якщо значення відстані та швидкостей вводяться в поля. Для заданих початкових даних обчисліть результати та запишіть у зошит:

а) відстань – 1000, швидкості – 120 та 130;

б) відстань – 600, швидкості – 80 та 120;

в) відстань – 1200, швидкості – 115 та 85.

Виконайте проект для наведених наборів тестових даних. Перевірте, чи збігаються результати, отримані під час виконання проекту, з попередньо обчисленими результатами. Поясніть, у чому може бути помилка.



2°. Створіть три тестових набори вхідних даних для тестування проекту, призначеного для розв'язання задачі: «*Два потяги виїхали одночасно з однієї станції у протилежних напрямках. Потрібно знайти відстань, на якій вони будуть один від одного через заданий час, якщо відома швидкість кожного потяга*». Обчисліть очікувані результати за складеними тестами. Відкрийте проект, наприклад з папки **Тема 6\Завдання 6.2\Проект 6.2.2**. Протестуйте його на ваших тестових даних. Зробіть висновок, чи містить проект логічні помилки. Якщо так, то в чому може бути помилка.

3*. Відкрийте проект, наприклад з папки **Тема 6\Завдання 6.2\Проект 6.2.3**. Перегляньте текст проекту та на основі аналізу коментарів зробіть припущення, яку задачу розв'язує цей проект. Спираючись на ваше припущення, складіть три тестових набори вхідних даних для тестування проекту. Обчисліть очікувані результати. Протестуйте проект на ваших тестових даних. Зробіть висновок, чи правильним було ваше припущення.

4°. Відкрийте проект, наприклад з папки **Тема 6\Завдання 6.2\Проект 6.2.4**. Виконайте проект у покроковому режимі. Зверніть увагу на відображення ходу виконання проекту.



5°. Відкрийте проект, наприклад з папки **Тема 6\Завдання 6.2\Проект 6.2.5**. У ході покрокового виконання проекту простежте за зміненням значень змінних s_1 та s_2 .

6°. Учні на свято купили цукерки та печиво. Відомо ціни цукерок і печива та маси куплених солодоців кожного виду. Створіть проект для знаходження повної вартості покупки. Збережіть проект у папці з іменем **Завдання 6.2.6**, створеній у вашій папці. Виконайте проект у покроковому режимі, простежте за значеннями змінних, що містять значення вартості кожного товару.

7°. Відкрийте проект, наприклад з папки **Тема 6\Завдання 6.2\Проект 6.2.7**. У ході покрокового виконання проекту простежте за зміною значень змінних x , a та s . З'ясуйте, як значення, яких послідовно набуває змінна a , пов'язані зі значенням змінної x . Змініть проект так, щоб у результаті виконання змінна s містила суму $x + x^2 + x^4$ для введеного значення x . Додайте до тексту програми коментар, у якому вкажіть, які зміни було внесено у програмний код. Збережіть проект у папці з іменем **Завдання 6.2.7**, створеній у вашій папці.



- 8*. Відкрийте проект, наприклад з папки **Тема 6\Завдання 6.2\Проект 6.2.8**. Цей проект призначено для знаходження, на скільки більше часу витратив один з потягів на шлях між містами, ніж другий, якщо задано відстань між містами і швидкості кожного з потягів. З'ясуйте, чи містить проект логічні помилки. Якщо так, визначте, які команди їх спричинили, і виправте їх. Запишіть тестові набори даних, на яких ви виявили логічні помилки. Збережіть виправлений проект у папці з іменем **Завдання 6.2.8**, створеній у вашій папці.

ПРАКТИЧНА РОБОТА № 11

«Налагодження проекту»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте вікно середовища **Lazarus**.
2. Відкрийте проект для визначення у скільки разів добуток двох заданих дійсних чисел більший за суму цих чисел. Наприклад, проект з папки **Тема 6\Практична 11**.
3. Створіть три тестових набори вхідних даних для тестування проекту. Запишіть їх у зошит. Обчисліть очікувані результати й запишіть їх. Протестуйте проект на ваших тестових даних. Порівняйте результати виконання проекту з очікуваними результатами. Запишіть висновок.
4. Додайте коментарі для команд, у яких відбувається обчислення суми та добутку.
5. Виконайте проект у покроковому режимі. У ході покрокового виконання проекту стежте за значеннями суми та добутку. Запишіть, які повідомлення відображаються у стовпчику **Значення** для вказаних змінних перед початком покрокового виконання проекту, у ході виконання для одного з наборів початкових даних та після завершення виконання.
6. Збережіть проект у папці з іменем **Практична 11**, створеній у вашій папці.



6.3. ЛОГІЧНІ ВИРАЗИ. ЗМІННІ ЛОГІЧНОГО ТИПУ. ЛОГІЧНІ ОПЕРАЦІЇ



1. Що таке висловлювання? Яке висловлювання вважається істинним, а яке – хибним?
2. Які ви знаєте властивості змінної?
3. Що визначає тип змінної? Які ви знаєте типи числових змінних?

ВИСЛОВЛЮВАННЯ ЯК ЛОГІЧНИЙ ВИРАЗ

Ви вже знаєте, що **висловлювання** – це речення, яке містить твердження про певний об'єкт або про зв'язки між об'єктами і про яке можна одно-



значно сказати, **істинне** воно чи **хибне**. Прикладом істинного висловлювання є, наприклад, висловлювання «Київ – столиця України», а прикладом хибного – висловлювання « $3 > 5$ ».

Основною властивістю висловлювання є його **істинність**. Якщо висловлювання істинне, то вважають, що значення його властивості істинність дорівнює **true** (англ. *true* – правда). Якщо висловлювання хибне, то вважають, що значення його властивості істинність дорівнює **false** (англ. *false* – хиба, хибність).

Тобто значення властивості **істинність** висловлювань: «Київ – столиця України», « $2 * 5 - 4 = 6$ », « $4 < 12$ » дорівнює **true**, значення властивості **істинність** висловлювань: «Цей підручник для учнів сьомого класу», « $2 * 7 + 3 = 12$ », « $3 > 5$ » дорівнює **false**.

Логіка (грец. *λόγος* – слово, смисл, думка, мова) – наука про форми, методи і закони правильного мислення, про способи міркування.

Висловлювання можна розглядати як **логічний вираз**.

Логічним виразом називають вираз, який може набувати одне з двох значень: **true** або **false**.

Якщо висловлювання істинне, то вважають, що значення відповідного логічного виразу дорівнює **true**. Тобто, «Київ – столиця України» = **true**, « $2 * 5 - 4 = 6$ » = **true**, « $4 < 12$ » = **true**.

Якщо висловлювання хибне, то вважають, що значення відповідного логічного виразу дорівнює **false**. Тобто «Це підручник для учнів сьомого класу» = **false**, « $2 * 7 + 3 = 12$ » = **false**, « $3 > 5$ » = **false**.

ЗМІННІ ЛОГІЧНОГО ТИПУ

Значення логічного виразу можна присвоювати певній змінній. Тобто можна використовувати такі команди присвоювання: $x := 2 + 12/3 = 6$, $y := 32 < 13$. Тоді змінна x має значення **true**, а змінна y має значення **false**.

Змінні, які можуть набувати одне з двох значень **true** або **false**, називають **змінними логічного типу**.

Логічний тип змінних позначається **boolean**. Наприклад, `var x, y: boolean`.

Часто для змінних логічного типу використовуються команди `x := true` або `x := false`.

Змінні логічного типу також вважаються логічними виразами.

ОПЕРАЦІЇ НАД ЛОГІЧНИМИ ВИРАЗАМИ

Над логічними виразами можна виконувати **логічні операції**: **заперечення**, **кон'юнкція** (лат. *conjunctio* – об'єднання), **диз'юнкція** (лат. *disjunctio* – роз'єднання) та ін.

Запереченням логічного виразу x називають логічний вираз, значення якого дорівнює **true**, якщо значення логічного виразу x дорівнює **false**, і дорівнює **false**, якщо значення логічного виразу x дорівнює **true**.

Заперечення логічного виразу x позначають **not x**.

x	not x
true	false
false	true

Таблиця істинності для заперечення



Тобто якщо $x = \text{true}$, то $\text{not } x = \text{false}$, і якщо $x = \text{false}$, то $\text{not } x = \text{true}$. Можна скласти таку таблицю, яку називають **таблицею істинності** для операції заперечення.

Для утворення заперечення висловлювання часто використовують частку **не**. Наприклад, запереченням висловлювання $A = \text{«Існує найбільше просте число»}$ є висловлювання $\text{not } A = \text{«Не існує найбільшого простого числа»}$.

Інколи заперечення логічного виразу x позначають \bar{x} або $\neg x$.

Кон'юнкцією двох логічних виразів x і y називають логічний вираз, значення якого є **true**, якщо значення кожного з логічних виразів x і y є **true**; і є **false**, якщо значення хоча б одного з логічних виразів x або y є **false**.

Кон'юнкція двох логічних виразів x і y позначається $x \text{ and } y$ (англ. *and* – і, та).

Для утворення кон'юнкції двох висловлювань часто використовують сполучник **і**. Наприклад, висловлювання $\text{«Число 36 кратне 3 і число 36 кратне 2»}$ є кон'юнкцією двох висловлювань: $\text{«Число 36 кратне 3»}$ та $\text{«Число 36 кратне 2»}$.

Інколи кон'юнкція двох логічних виразів x і y позначається $x \wedge y$.

Диз'юнкцією двох логічних виразів x і y називають логічний вираз, значення якого є **true**, якщо значення хоча б одного з логічних виразів x або y є **true**; і є **false**, якщо значення кожного з логічних виразів x і y є **false**.

Диз'юнкція двох логічних виразів x і y позначається $x \text{ or } y$ (англ. *or* – або).

Для утворення диз'юнкції висловлювань часто використовують сполучник **або**. Наприклад, висловлювання $\text{«}11 \leq 21\text{»}$ (11 менше або дорівнює 21) є диз'юнкцією двох висловлювань: $\text{«}11 < 21\text{»}$ та $\text{«}11 = 21\text{»}$.

Інколи диз'юнкція двох логічних виразів x і y позначається $x \vee y$.

Часто використовують **логічні вирази**, що містять кілька логічних операцій. Наприклад: $(x \text{ or } y) \text{ and } (x \text{ or } z)$.

У таких виразах для логічних операцій, розглянутих вище, визначено такий **пріоритет виконання операцій**: спочатку виконуються операції заперечення, потім – операції кон'юнкції, потім – операції диз'юнкції. Для змінення цього порядку виконання логічних операцій використовують дужки.

Для обчислення значень логічного виразу можна використати таблицю істинності. У ній потрібно розглянути всі можливі значення змінних логічного типу, що входять до цього виразу.

x	y	$x \text{ and } y$
true	true	true
true	false	false
false	true	false
false	false	false

Таблиця істинності для кон'юнкції

x	y	$x \text{ or } y$
true	true	true
true	false	true
false	true	true
false	false	false

Таблиця істинності для диз'юнкції

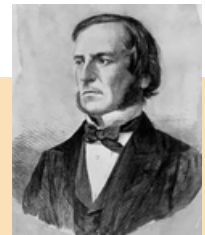


<i>x</i>	<i>y</i>	<i>z</i>	1	2	3
			<i>x or y</i>	<i>x or z</i>	<i>(x or y) and (x or z)</i>
true	true	true	true	true	true
true	true	false	true	true	true
true	false	true	true	true	true
false	true	true	true	true	true
true	false	false	true	true	true
false	true	false	true	false	false
false	false	true	false	true	false
false	false	false	false	false	false



Цікаві факти з історії

Означення операцій над змінними логічного типу і властивості цих операцій сформулював англійський математик та філософ **Джордж Буль** (1815–1864) (мал. 6.17). У 1854 р. вийшла його основна робота «Дослідження законів думки, на яких засновано математичні теорії логіки та ймовірності». У ній досліджується система, яку сьогодні називають «алгеброю висловлень» або «булевою логікою». Булева логіка стала основним математичним інструментом для створення комп'ютерів.



Мал. 6.17.
Джордж Буль



Найважливіше в цьому пункті

Логічним виразом називають вираз, який може набувати одне з двох значень: **true** або **false**.

Висловлювання є прикладом **логічного виразу**. Якщо висловлювання істинне, то вважають, що значення відповідного логічного виразу дорівнює **true**. Якщо висловлювання хибне, то вважають, що значення відповідного логічного виразу дорівнює **false**.

Змінні, які можуть набувати одне з двох значень **true** або **false**, називають **змінними логічного типу**. Логічний тип змінних позначається **boolean**. Змінні логічного типу також вважаються логічними виразами.

<i>x</i>	<i>not x</i>
true	false
false	true

Таблиця істинності для заперечення

<i>x</i>	<i>y</i>	<i>x and y</i>
true	true	true
true	false	false
false	true	false
false	false	false

Таблиця істинності для кон'юнкції

<i>x</i>	<i>y</i>	<i>x or y</i>
true	true	true
true	false	true
false	true	true
false	false	false

Таблиця істинності для диз'юнкції



У логічних виразах для логічних операцій, розглянутих вище, визначено такий **пріоритет виконання операцій**: спочатку виконуються операції заперечення, потім – операції кон'юнкції, потім – операції диз'юнкції. Для змінення цього порядку виконання логічних операцій використовують дужки.

Для обчислення значень логічного виразу можна використати таблицю істинності. У ній потрібно розглянути всі можливі значення змінних логічного типу, що входять до цього виразу.






Дайте відповіді на запитання

- 1°. Який вираз називають логічним? Наведіть приклади.
- 2°. Які змінні називають змінними логічного типу?
- 3°. Як описуються змінні логічного типу?
- 4°. Що таке заперечення логічного виразу?
- 5°. Що таке кон'юнкція двох логічних виразів?
- 6°. Що таке диз'юнкція двох логічних виразів?
- 7°. Яким є пріоритет виконання логічних операцій?
- 8*. Що вивчає наука логіка?



Виконайте завдання

- 1°. Укажіть порядок виконання логічних операцій у логічних виразах:
 - а) $a \text{ or } not \ a \ \text{and} \ b$;
 - б) $not \ a \ \text{or} \ not \ b \ \text{and} \ a$;
 - в) $(a \ \text{or} \ not \ b) \ \text{and} \ (not \ a \ \text{or} \ not \ b)$;
 - г) $not \ a \ \text{and} \ not \ b \ \text{or} \ c \ \text{and} \ d$.
- 2°. Складіть таблиці істинності логічних виразів зі змінними логічного типу a і b :

а) $a \ \text{or} \ a \ \text{and} \ b$;	 г) $not \ a \ \text{or} \ not \ b$;
б) $a \ \text{or} \ not \ a \ \text{and} \ b$;	 д) $(a \ \text{or} \ not \ b) \ \text{and} \ (a \ \text{or} \ not \ b)$;
в) $a \ \text{or} \ b \ \text{or} \ not \ b$;	 е) $a \ \text{and} \ not \ b \ \text{or} \ not \ a \ \text{and} \ b$.
- 3°. Сформулюйте заперечення висловлювань та з'ясуйте їх істинність:
 - а) Число 3 є дільником числа 545.
 - б) Автомобіль не має права їхати на червоне світло.
 - в) Існують паралелограми з прямими кутами.
 - г) Рівняння $2x^2 - 3x + 1 = 0$ має цілий корінь.
 - д) Не існує натурального числа, що ділиться на 2.
 - е) Існує ціле число, яке ділиться на всі цілі числа.
- 4°. Серед зазначених висловлювань знайдіть кон'юнкції і диз'юнкції та визначте їх істинність:
 - а) Число 27 кратне 3 і кратне 9.
 - б) $17 < 42 < 18$.
 - в) Число 2 просте або парне.
 - г) ABC є гострокутним, прямокутним або тупокутним.
 - д) Діагоналі будь-якого паралелограма перпендикулярні і точкою перетину поділяються навпіл.
 - е) $7^2 = 49$ і $(-7)^2 = 49$.
 - є) $21 \leq 21$.
 - ж) $21 \leq 18$.



- 5*. Нехай $A = \text{«Сергієнко вивчає англійську мову»}$, $B = \text{«Сергієнко має оцінку 8 з інформатики»}$. Сформулюйте висловлювання:
- а) A and not B
 - б) not A or B
 - в) A or not A and B
 - г) not A and not B and A
- 6*. Визначте істинність або хибність висловлювань для наведених значень змінних:
1. « $x > 0$ і $y > 0$ або $x < 0$ і $y < 0$ »
 2. « $x > 0$ і не $y < 0$ або $x < 0$ та $y > 0$ »
- Значення змінних:
- а) $x = 5, y = 8$;
 - б) $x = 5, y = -8$;
 - в) $x = -5, y = 8$;
 - г) $x = -5, y = -8$.



6.4. АЛГОРИТМИ З РОЗГАЛУЖЕННЯМИ



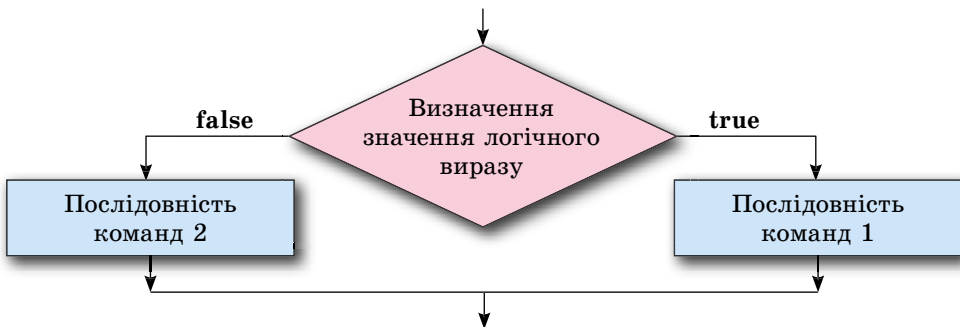
1. Які вирази називають логічними? Яких значень вони можуть набувати?
2. Назвіть логічні операції. Наведіть таблицю істинності для кожної з них.
3. Що таке розгалуження? Які види розгалуження ви знаєте? Чим вони різняться між собою?

АЛГОРИТМИ З РОЗГАЛУЖЕННЯМ

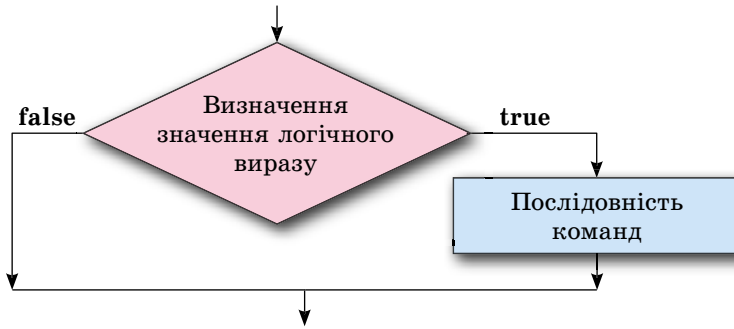
Ви вже знаєте, що алгоритми можуть містити розгалуження. Ви також знаєте, що розгалуження як фрагмент алгоритму починається з команди перевірки умови і результатом виконання цієї команди може бути або **істина** (Так, true), або **хиба** (Ні, false). І залежно від результату виконання цієї команди будуть виконуватися ті чи інші команди.

У загальному випадку в цій команді визначається значення певного логічного виразу, яке може бути або true, або false.

У загальному випадку блок-схеми розгалужень мають такий вигляд (мал. 6.18 і 6.19):



Мал. 6.18. Повне розгалуження



Мал. 6.19. Неповне розгалуження

РОЗГАЛУЖЕННЯ В ОБ'ЄКТ PASCAL

Команда повного розгалуження в мові програмування **Object Pascal** має такий загальний вигляд:

```

If<логічний вираз>
Then begin
    <послідовність команд 1>
end
Else begin
    <послідовність команд 2>
end;

```

(англ. *if* – якщо, *then* – тоді, *else* – інакше).

Виконання цієї команди відбувається так: обчислюється значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням; якщо це значення **false**, то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням.

Є в **Object Pascal** і команда неповного розгалуження:

```

If<логічний вираз>
Then begin
    <послідовність команд>
end;

```

Виконання цієї команди відбувається так: обчислюється значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд** і після цього виконується команда, наступна за розгалуженням; якщо це значення **false**, одразу виконується команда, наступна за розгалуженням.

Звертаємо вашу увагу:

- якщо після ключових слів **Then** і **Else** слідує тільки по одній команді, то операторні дужки **begin** і **end** можна не ставити;
- перед **Else** ставити крапку з комою не можна.

Розглянемо задачу, алгоритм розв'язування якої містить розгалуження.

Задача 1. Обчислити значення виразу $(a + 2b) / (a - 3b)$, де a, b – дійсні числа.

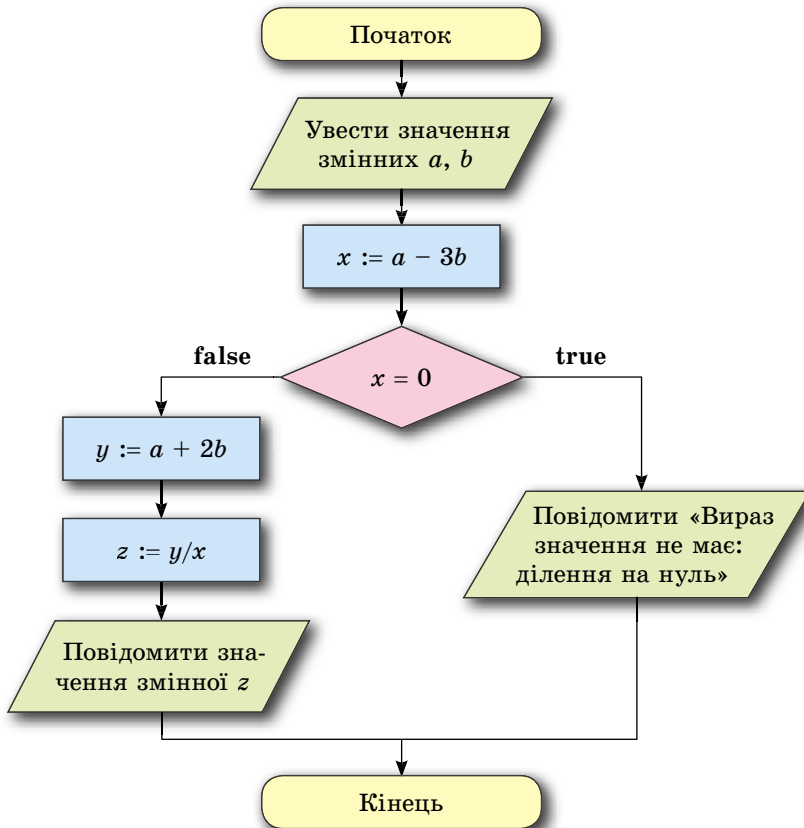
Оскільки вираз містить дію ділення на вираз, який може дорівнювати нулю (наприклад, при $a = 3$ і $b = 1$), алгоритм розв'язування цієї задачі



Розділ 6

повинен містити розгалуження, у якому буде ураховано два можливих випадки: значення виразу $a - 3b$ дорівнює 0 і значення виразу $a - 3b$ не дорівнює 0.

Блок-схема алгоритму розв'язування цієї задачі матиме такий вигляд (мал. 6.20):



Мал. 6.20. Блок-схема алгоритму розв'язування задачі 1

Якщо записати мовою **Object Pascal** алгоритм, наведений на малюнку 6.20, то він матиме такий вигляд:

```
a := StrToFloat(Edit1.Text);  
b := StrToFloat(Edit2.Text);  
x := a-3*b;  
If x = 0  
Then Label1.Caption := 'Вираз значення не має: ділення на нуль'  
Else begin  
    y := a+2*b;  
    z := y/x;  
    Label1.Caption := FloatToStr(z);  
end;
```

Звертаємо вашу увагу, що обчислення значення виразу $a - 3b$ можна було б не здійснювати в окремій команді і не запам'ятовувати як значення



змінної x , а здійснити безпосередньо в команді розгалуження. Також можна обчислення значення виразу $(a + 2b) / (a - 3b)$ записати не двома командами, а однією. Тоді цей фрагмент проекту матиме такий вигляд:

```

a := StrToFloat(Edit1.Text);
b := StrToFloat(Edit2.Text);
If a-3*b = 0
Then Label1.Caption := 'Вираз значення не має: ділення на нуль'
Else begin
    z := (a+2*b) / (a-3*b);
    Label1.Caption := FloatToStr(z);
end;
```

Розглянемо задачу, під час розв'язування якої в команді розгалуження використовується кон'юнкція.

Задача 2. Обчислити значення заданої функції

$$y = \begin{cases} 2x - 12, & \text{якщо } -3 < x < 5, \\ 7 - 8x, & \text{для всіх інших } x. \end{cases}$$

Фрагмент проекту для розв'язування цієї задачі матиме такий вигляд:

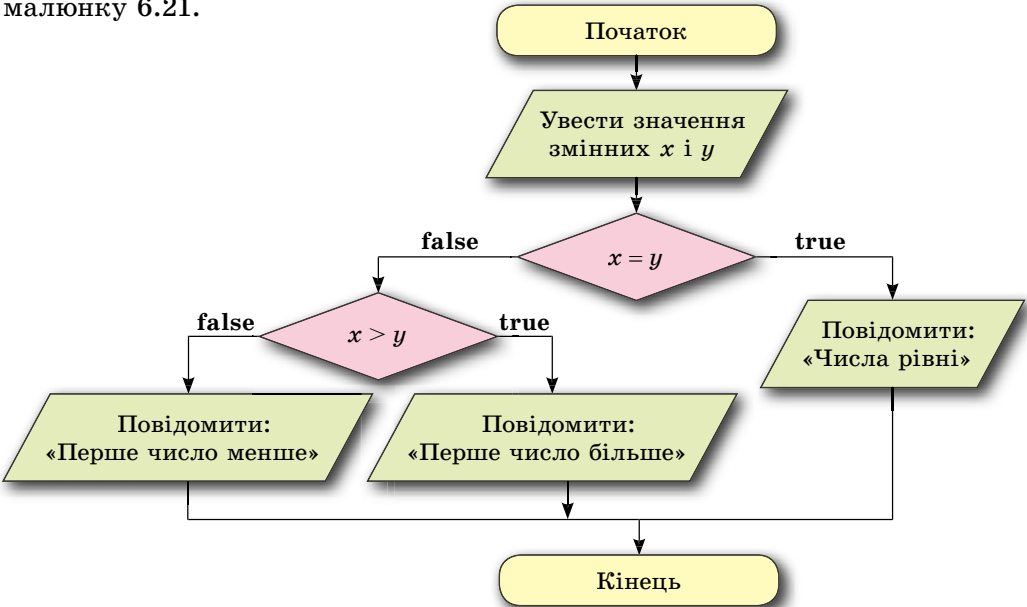
```

x := StrToFloat(Edit1.Text);
If (x > -3) and (x < 5)
Then y := 2*x - 12
Else y := 7-8*x;
Label1.Caption := FloatToStr(y);
```

Розглянемо, нарешті, задачу, під час розв'язування якої використовується дещо складніше розгалуження.

Задача 3. Дано два числа. Визначити, чи рівні вони. Якщо ні, то яке з них більше?

Блок-схема алгоритму розв'язування цієї задачі матиме вигляд як на малюнку 6.21.



Мал. 6.21. Блок-схема алгоритму розв'язування задачі 3



У цьому алгоритмі всередині одного розгалуження використовується інше розгалуження. Ось який вигляд має фрагмент відповідного проекту:

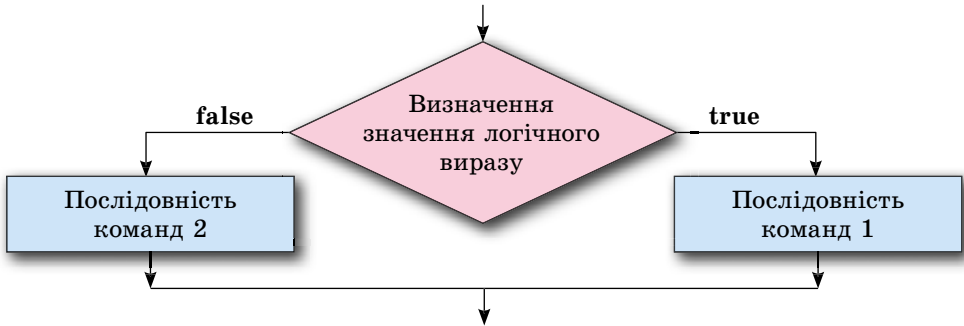
```

If x = y
Then Label1.Caption := 'Числа рівні'
Else If x > y
Then Label1.Caption := 'Перше число більше'
Else Label1.Caption := 'Перше число менше';

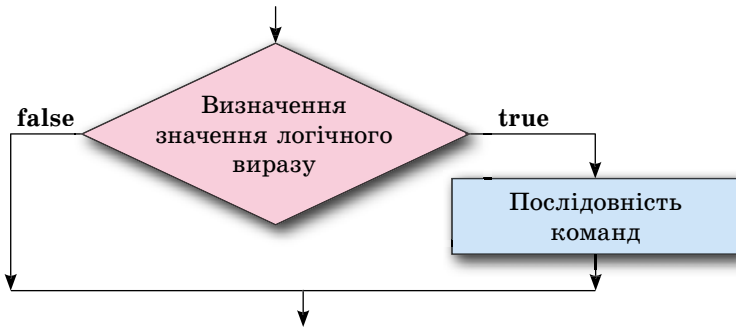
```

Найважливіше в цьому пункті

У загальному випадку блок-схеми розгалужень матимуть такий вигляд (мал. 6.22 і 6.23):



Мал. 6.22. Повне розгалуження



Мал. 6.23. Неповне розгалуження

Команда повного розгалуження в мові програмування **Object Pascal** має такий загальний вигляд:

```

If <логічний вираз>
Then begin
    <послідовність команд 1>
end
Else begin
    <послідовність команд 2>
end;

```



Виконання цієї команди відбувається так: обчислюється значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд 1** і після цього виконується команда, наступна за розгалуженням; якщо це значення **false**, то виконується **послідовність команд 2** і після цього виконується команда, наступна за розгалуженням.

Є в **Object Pascal** і команда неповного розгалуження:

```
If <логічний вираз>
Then begin
    <послідовність команд>
end;
```

Виконання цієї команди відбувається так: обчислюється значення логічного виразу; якщо це значення **true**, то виконується **послідовність команд** і після цього виконується команда, наступна за розгалуженням; якщо це значення **false**, одразу виконується команда, наступна за розгалуженням.



Дайте відповіді на запитання

- 1°. Який вигляд має блок-схема повного розгалуження? Опишіть особливості її виконання.
- 2°. Який вигляд має блок-схема неповного розгалуження? Опишіть особливості її виконання.
- 3°. У чому полягають відмінності у виконанні повного і неповного розгалуження?
- 4°. Який вигляд має команда повного розгалуження в **Object Pascal**?
- 5°. Який вигляд має команда неповного розгалуження в **Object Pascal**?
- 6°. У чому полягають відмінності виконання команд повного і неповного розгалуження в **Object Pascal**?



Виконайте завдання

- 1°. Поясніть хід виконання команди:

```
If<логічний вираз 1>
Then begin
    <команди 1>
end
Else If<логічний вираз 2>
Then begin
    <команди 2>
end
Else begin
    <команди 3>
end;
```

- 2°. Поясніть хід виконання послідовності команд:






```
If<логічний вираз 1>
Then begin
    <команди 1>
end;
If<логічний вираз 2>
```



Then begin

<команди 2>

end;

- 3*. Поясніть, чим виконання команди в завданні № 1 відрізняється від виконання послідовності команд у завданні № 2.
- 4*. Складіть проект для обчислення значення виразу $(a + b) - c : (a - b)$. Збережіть проект у папці з іменем **Завдання 6.4.4**, створеній у вашій папці.
-  5*. Складіть проект для обчислення значення виразу $a + b - c : (a - 2b)$. Збережіть проект у папці з іменем **Завдання 6.4.5**, створеній у вашій папці.
- 6*. Створіть проект, який визначає менше з двох довільних дійсних чисел або виводить повідомлення, що числа рівні. Збережіть проект у папці з іменем **Завдання 6.4.6**, створеній у власній папці.
-  7*. Створіть проект, який визначає більше з двох нерівних дійсних чисел. Збережіть проект у папці з іменем **Завдання 6.4.7**, створеній у вашій папці.
-  8*. Створіть проект, який обчислює модуль довільного дійсного числа. Збережіть проект у папці з іменем **Завдання 6.4.8**, створеній у вашій папці.
- 9*. Створіть проект, який визначає найменше число серед трьох нерівних чисел. Збережіть проект у папці з іменем **Завдання 6.4.9**, створеній у вашій папці.
- 10*. Наведіть приклади правил з української мови, математики, інших предметів, що містять розгалуження.
- 11*. Наведіть приклади життєвих ситуацій, які можна описати алгоритмом з розгалуженням.
- 12*. Створіть проект, у якому кнопка буде переміщуватися на 20 пікселів праворуч під час наведення на неї вказівника. Після того як кнопка зникне за правою межею вікна, вона повинна з'явитися біля лівої її межі. Збережіть проект у папці з іменем **Завдання 6.4.12**, створеній у вашій папці.
- 13*. Створіть проект для знаходження x з рівняння $ax = b$. Збережіть проект у папці з іменем **Завдання 6.4.13**, створеній у вашій папці.
-  14*. Створіть проект для знаходження x з рівняння $ax + b = c$. Виконайте його при: 1) $a = 2; b = -8; c = 18$; 2) $a = 20; b = 5; c = 5$; 3) $a = 0; b = 12; c = 3$; 4) $a = 0; b = 10; c = 10$. Збережіть проект у папці з іменем **Завдання 6.4.14**, створеній у вашій папці.
- 15*. Створіть проект для визначення, чи можна з трьох відрізків із заданими довжинами утворити трикутник. Збережіть проект у папці з іменем **Завдання 6.4.15**, створеній у вашій папці.
-  16*. Створіть проект, у якому можна обчислити значення функції

$$y = \begin{cases} 15 - 3x, & \text{якщо } -1 < x < 4, \\ 6 + 4x, & \text{для всіх інших } x. \end{cases}$$

Збережіть проект у папці з іменем **Завдання 6.4.16**, створеній у вашій папці.



6.5. ПРАПОРЦІ. ПОЛЕ З РОЗКРИВНИМ СПИСКОМ. ПЕРЕМИКАЧІ




1. Який вигляд мають команди повного і неповного розгалуження в **Object Pascal** і як вони виконуються?
2. Які властивості напису і поля ви знаєте, що визначають їх значення, як їх можна змінити?
3. Для чого використовувалися в програмах, з якими ви працювали, прапорці, поля з розкритими списками, перемикачі?

ПРАПОРЦІ ТА ЇХ ВИКОРИСТАННЯ

Розглянемо компонент **прапорець**. Значення деяких його властивостей, зокрема властивості **Checked** (англ. *checked* – перевірено), можуть бути використані в логічному виразі в команді розгалуження.

Для розміщення на формі прапорця потрібно двічі клацнути на кнопці **TCheckBox** (англ. *checkbox* – коробка для перевірки) на вкладці **Standard** з компонентами. Після цього прапорець з'явиться у верхній лівій частині форми (мал. 6.24), а його ім'я додасться до списку компонентів проекту в полі **Components** вікна **Інспектор об'єктів**. За потреби поле можна перетягнути в будь-яке інше місце форми.

Властивості прапорця **Caption, Top, Left, Name, Color, Enabled, Visible, Font, Width, Height** аналогічні до відповідних властивостей напису. Зокрема, значення властивостей **Width** і **Height** прапорця визначаються довжиною тексту праворуч від прапорця і розміром шрифту, яким виведено цей текст. Сам цей текст є значенням властивості **Caption** прапорця. Цей текст, як і для напису, можна вводити безпосередньо на вкладці **Властивості** в полі **Caption**, а можна вводити у вікні **Редактор рядків**, яке відкривається вибором кнопки  праворуч від поля **Caption**.



Мал. 6.24. Прапорець на формі

Властивість **Checked** прапорця може набувати одне з двох значень: **true** (позначку прапорця встановлено, прапорець встановлено) або **false** (позначку прапорця не встановлено, прапорець знято).

Якщо на формі розміщено кілька прапорців, то серед них можна встановити один або кілька, не встановлювати жодного або встановити всі. Якщо прапорець встановлено, то у нього всередині з'являється позначка .

Встановленням позначок прапорців можна встановити значення однієї або кількох властивостей об'єкта з деякого визначеного набору властивостей.

Розглянемо, як можна використати прапорці для реалізації розгалуження.

Нехай, наприклад, нам потрібно в ході виконання проекту змінювати значення або тільки однієї з властивостей вікна (або заголовка, або колір, або розміри), або кількох з них, або не змінювати значення жодної з них.

Для цього розмістимо на формі кнопку і три прапорці (мал. 6.25). Під час розміщення прапорців на формі для кожного з них за замовчуванням значення властивості **Checked** встановлюється **false**. Оскільки під час виконання проекту жодний прапорець може бути не встановлений, то доцільно залишити такі початкові значення властивості **Checked** для всіх прапорців.

Створимо процедуру для опрацювання події **Click** для кнопки і до її тексту помістимо такі три послідовні (не вкладені одна в одну) команди розгалуження:

```
If CheckBox1.Checked
Then Form1.Caption := 'Заголовок змінений';
  If CheckBox2.Checked
  Then Form1.Color := clRed;
  If CheckBox3.Checked
  Then begin
    Form1.Width := 500;
    Form1.Height := 200;
  end;
```



Мал. 6.25. Форма з кнопкою і трьома прапорцями

У кожній із цих команд, незалежно одна від одної, буде визначатися, вибрано відповідний прапорець чи ні. І залежно від цього, у кожній з них будуть чи не будуть змінюватися значення відповідних властивостей вікна.

ПОЛЕ З РОЗКРИВНИМ СПИСКОМ ТА ЙОГО ВИКОРИСТАННЯ



Мал. 6.26. Поле з розкривним списком на формі

Ще одним компонентом, значення властивостей якого можна використати в логічному виразі в команді розгалуження, є поле з розкривним списком.

Для розміщення на формі поля з розкривним списком потрібно двічі клацнути на кнопці **TComboBox** (англ. *combobox* – поле зі списком) на вкладці **Standard** з компонентами. Після цього поле з розкривним списком з'явиться у верхній лівій частині форми (мал. 6.26), а його ім'я додасться до списку компонентів проекту в полі **Components**

вікна **Інспектор об'єктів**. За потреби поле з розкривним списком можна перетягнути в будь-яке інше місце форми.

Властивості **Top**, **Left**, **Height**, **Width**, **Name**, **Color**, **Enabled**, **Visible**, **Font**, **Text** поля з розкривним списком повністю аналогічні до відповідних властивостей поля.

Задати вміст розкривного списку можна у вікні **Редактор рядків** (мал. 6.27), яке відкривається вибором кнопки **...** праворуч від поля **Items** (англ. *items* – предмети) вкладки **Властивості**.

Під час виконання проекту можна відкрити цей список і вибрати будь-який його рядок. При цьому відповідний текст потрапляє в поле і стає значенням властивості **Text** поля з розкривним списком.

Розглянемо, як можна використати поле з розкривним списком у команді розгалуження.



Нехай, наприклад, у ході виконання проекту потрібно змінювати значення тільки однієї з властивостей вікна: або заголовок, або колір, або розміри.

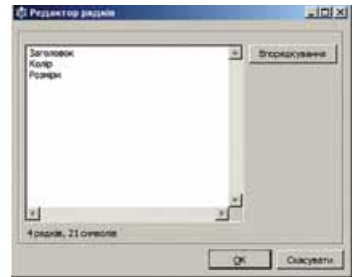
Розмістимо на формі поле з розкритим списком і створимо відповідний список (мал. 6.27). Розмістимо також на формі кнопку і складемо процедуру – обробник події **Click** для цієї кнопки, під час виконання якої вибиратимемо один з рядків розкритого списку (мал. 6.28). Залежно від вибраного рядка змінюватимуться або заголовок форми, або її колір, або її розміри.

Виберемо перший пункт списку *Заголовок* і клацнемо кнопку. Тоді під час виконання процедури значення логічного виразу (**ComboBox1.Text = 'Заголовок'**) = **true**, а значить виконуватиметься команда **Form1.Caption := 'Заголовок змінений'**, яка змінить заголовок вікна, після чого виконання команди розгалуження і всієї процедури закінчиться.

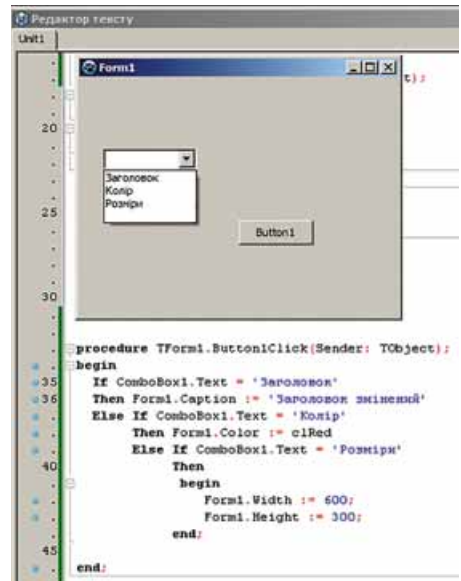
Якщо вибрати другий пункт списку *Колір* і клацнути кнопку, то під час виконання процедури значення логічного виразу (**ComboBox1.Text = 'Заголовок'**) = **false**. Тоді обчислюється значення логічного виразу **ComboBox1.Text = 'Колір'**, і воно дорівнює **true**. А значить виконуватиметься команда **Form1.Color := clRed**, яка змінить колір вікна, після чого виконання команди розгалуження і всієї процедури закінчиться.

Якщо вибрати третій пункт списку *Розміри* і клацнути кнопку, то під час виконання процедури значення логічного виразу (**ComboBox1.Text = 'Заголовок'**) = **false**. Після цього обчислюється значення логічного виразу **ComboBox1.Text = 'Колір'**, і воно теж дорівнює **false**. Після цього обчислюється значення логічного виразу **ComboBox1.Text = 'Розміри'**, і воно дорівнює **true**. А значить будуть виконуватися команди **Form1.Width := 600** і **Form1.Height := 300**, які змінять ширину і висоту вікна, після чого виконання команди розгалуження і всієї процедури закінчиться.

Якщо не вибрати жодного пункту списку і клацнути кнопку, то значення всіх логічних виразів у розгалуженні будуть **false** і в цьому випадку не змінюватимуться значення жодної властивості вікна і виконання процедури закінчиться.



Мал. 6.27. Вікно **Редактор рядків** зі списком




Мал. 6.28. Процедура змінення значення заголовка, або кольору, або розмірів вікна і вікно виконання проекту




Для тих, хто хоче знати більше

Перемикачі та їх використання

Для вибору одного варіанта з кількох можливих, наприклад установити для певної властивості об'єкта одне значення з деякого визначеного набору значень, можна використати **перемикачі**.

Для розміщення на формі перемикача потрібно двічі клацнути на кнопці  **TRadioButton** (англ. *radiobutton* – кругла кнопка) на вкладці **Standard** з компонентами.

Якщо на формі розміщено кілька перемикачів, то серед них можна вибрати тільки один перемикач. Якщо перемикач вибрано, то в нього всередині з'являється позначка .

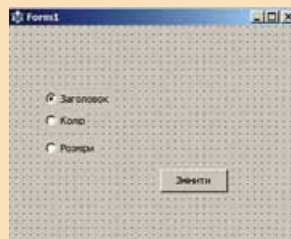
Усі ті властивості, які ми розглядали для прапорця, має і перемикач, у тому числі й властивість **Checked**, значення якої можна використати в логічному виразі в команді розгалуження.

Розглянемо, як можна змінювати значення тільки однієї з властивостей вікна: або заголовок, або колір, або розміри, використавши при цьому не поле з розкритим списком, а перемикачі.

Розмістимо на формі кнопку і три перемикачі (мал. 6.29). Під час розміщення перемикачів на формі для кожного з них за замовчуванням встановлюється значення властивості **Checked** – **false**. До запуску проекту змінимо значення цієї властивості в першого перемикача на **true**. Після цього на формі він матиме вигляд вибраного. Таким він буде і після запуску проекту на виконання.

Створимо процедуру – обробник події **Click** для кнопки і до її тексту помістимо таку команду:

```
If RadioButton1.Checked  
Then Form1.Caption := 'Заголовок змінений'  
Else If RadioButton2.Checked  
Then Form1.Color := clRed  
Else begin  
    Form1.Width := 600;  
    Form1.Height := 300;  
end;
```



Мал. 6.29. Форма з кнопкою і трьома перемикачами

Розпочинатиметься виконання цієї команди з визначення значення властивості **RadioButton1.Checked**.

Якщо при настанні події **Click** для кнопки вибраним буде перший перемикач, то **RadioButton1.Checked = true**. Тому виконається команда змінення заголовка вікна, а інші частини цієї команди розгалуження виконуватися не будуть.

Якщо при настанні події **Click** для кнопки вибраним буде другий перемикач, то **RadioButton1.Checked = false**. Тому будуть виконуватися команди після першого ключового слова **Else**. Цією командою є також команда розгалуження, що є частиною першої команди розгалуження. Під час її виконання визначається значення властивості **RadioButton2.Checked**. Оскільки буде вибраний другий перемикач, то це значення дорівнює **true**. Тому виконається команда змінення кольору вікна, а інші частини команди виконуватися не будуть.

Якщо при настанні події **Click** для кнопки вибраним буде третій перемикач, то **RadioButton1.Checked = false**. Тому будуть виконуватися команди після першого ключового слова **Else**, тобто внутрішня команда розгалуження. Під час її виконання визначається значення властивості **RadioButton2.Checked**. Оскільки другий



перемикач не вибраний, то її значення **false**. Тоді будуть виконуватися команди після другого ключового слова **Else**, тобто дві команди змінення розмірів вікна.

Якщо перед запуском цього проекту на виконання у жодного перемикача не змінювати значення властивості **Checked**, установлені за замовчуванням, то при запуску проекту жоден з них не буде вибраним. Якщо і під час виконання проекту не вибрати жодного з перемикачів і вибрати кнопку, то зміняться розміри вікна (поміркуйте самостійно, чому відбудуться саме такі змінення). Щоб цього не сталося, потрібно або обов'язково вибирати один з перемикачів, або після другого ключового слова **Else** вставити ще одну команду розгалуження:


```

If RadioButton1.Checked
Then Form1.Caption := 'Заголовок змінений'
Else If RadioButton2.Checked
Then Form1.Color := clRed
Else If RadioButton3.Checked
Then
    begin
        Form1.Width := 600;
        Form1.Height := 300;
    end;


```

Найважливіше в цьому пункті

Значення властивості **Checked** прапорців може бути використано в логічному виразі команди розгалуження. Воно дорівнює **True**, якщо прапорець встановлено, або **False**, якщо прапорець не встановлено (знято).

Для розміщення на формі прапорця потрібно двічі клацнути на кнопці  **TCheckBox** на вкладці **Standard** з компонентами. Якщо на формі розміщено кілька прапорців, то серед них можна встановити один або кілька, не встановлювати жодного або встановити всі. Тому прапорці використовують для того, щоб у ході виконання проекту вибрати або один з можливих варіантів, або кілька, або не вибрати жодного.

Для опрацювання стану прапорців (встановлено або не встановлено) у проект включають для кожного з них послідовно записані команди розгалуження, які незалежні одна від одної.

Для того щоб у ході виконання проекту вибрати лише один з можливих варіантів, можна використати поле з **розкритим списком**. Для розміщення на формі поля з розкритим списком потрібно двічі клацнути на кнопці  **TComboBox** на вкладці **Standard** з компонентами.

Дайте відповіді на запитання

- 1°. Для чого використовують прапорці?
- 2°. Які значення може мати властивість **Checked** для прапорця?
- 3°. Для чого використовують поле з розкритим списком?
- 4°. Як потрібно записувати команди розгалуження, у яких використовується значення властивості **Checked** для прапорців?
- 5°. Як доцільно записати команди розгалуження, у яких використовуються значення властивостей поля з розкритим списком?



- 6*. У чому полягають відмінності використання команд розгалуження з використанням значень властивостей прапорців і поля з розкритим списком?
- 7*. Для чого використовують перемикачі?
- 8*. Які значення може мати властивість **Checked** для перемикача?



Виконайте завдання

- 1*. Намалюйте блок-схему і поясніть хід виконання послідовності команд:

```
If CheckBox1.Checked
Then begin
    <команди 1>
end;
If CheckBox2.Checked
Then begin
    <команди 2>
end;
```



- 2*. Намалюйте блок-схему і поясніть хід виконання команд:

```
If ComboBox1.Text = '<Текст 1>'
Then begin
    <команди 1>
end
Else If ComboBox1.Text = '<Текст 2>'
Then begin
    <команди 2>
end
Else If ComboBox1.Text = '<Текст 3>'
Then begin
    <команди 3>
end;
```

- 3*. Намалюйте блок-схему і поясніть хід виконання команди:

```
If RadioButton1.Checked
Then begin
    <команди 1>
end
Else If RadioButton2.Checked
Then begin
    <команди 2>
end
Else
begin
    <команди 3>
end;
```

- 4*. Поясніть відмінності у виконанні команди розгалуження із завдання № 2 і послідовності двох команд розгалуження із завдання № 1.
- 5*. Створіть проект, у якому можна обчислити суму, або різницю, або добуток, або частку від ділення двох довільних дійсних чисел залежно від вибраного пункту розкритого списку. Збережіть проект у папці з іменем **Завдання 6.5.5**, створеній у вашій папці.



6*. Створіть проект, у якому можна обчислити суму, різницю, добуток, частку від ділення двох довільних дійсних чисел залежно від установлених позначок прапорців. Збережіть проект у папці з іменем **Завдання 6.5.6**, створеній у вашій папці.

7*. Робітник виготовляє деталі. Відомо кількість реально виготовлених деталей за місяць і вартість однієї деталі. Якщо робітник виконує план, то він одержує премію в розмірі 10 % від заробленої в результаті виготовлення деталей суми. Якщо робітник перевиковує план, то його премія становить 30 % від заробленої суми. А якщо він недовиконає план, то з нього знімається 20 % від заробленої суми. Створіть проект, який визначає суму грошей, яку заробив робітник за місяць. Збережіть проект у папці з іменем **Завдання 6.5.7**, створеній у вашій папці.



8*. Робітник має фіксовану заробітну плату за роботу протягом місяця. Якщо в робітника є діти, він отримує доплату в розмірі 20 % від його місячної заробітної плати. Якщо в робітника протягом місяця були порушення трудової дисципліни, на нього накладають штраф у розмірі 15 % від його місячної заробітної плати. Створіть проект, який визначає суму грошей, яку заробив робітник за місяць. Збережіть проект у папці з іменем **Завдання 6.5.8**, створеній у вашій папці.



6.6. ЦИКЛ З ЛІЧИЛЬНИКОМ



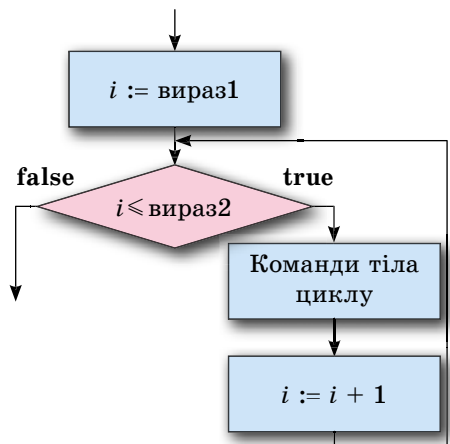
1. Які процеси називають циклічними? Наведіть приклади.
2. Що таке цикл в алгоритмі? Що таке тіло циклу?
3. Який вигляд має команда циклу **Повторити N разів**? Як вона виконується?

КОМАНДА ЦИКЛУ З ЛІЧИЛЬНИКОМ В ОБ'ЄКТ PASCAL

У мові програмування **Object Pascal** є кілька команд, які можуть реалізувати цикл. Одна з них – команда циклу з лічильником. Її доцільно використовувати в тих випадках, коли кількість повторень команд тіла циклу відома ще до початку виконання команди циклу. Загальний вигляд цієї команди такий:

```
for <ім'я змінної> := <вираз1> to <вираз2> do
begin
    <команди тіла циклу>
end;
```

Рядок **for <ім'я змінної> := <вираз1> to <вираз2> do** (англ. *for* – для, *to* – до, *do* – робити, виконувати) називають рядком заголовка команди циклу з лі-



Мал. 6.30. Блок-схема команди циклу з лічильником



чильником. Змінну в рядку заголовка команди циклу з лічильником, що стоїть перед знаком присвоювання, називають **лічильником циклу**.

Блок-схему команди циклу з лічильником показано на малюнку 6.30.

Лічильник циклу, вираз1 і вираз2 мають набувати тільки цілих значень. Якщо тіло циклу складається лише з однієї команди, операторні дужки **begin і end** можна не ставити.

Виконується команда циклу з лічильником так:

1. Надати лічильнику циклу **i** значення **вираз1**.
2. Обчислити значення логічного виразу **i ≤ вираз2**.
3. Якщо значення логічного виразу **true**, то виконати команди тіла циклу і перейти до виконання команди 4, якщо **false** – перейти до виконання команди, наступної за командою циклу.
4. Збільшити значення лічильника циклу на 1.
5. Перейти до виконання команди 2.

Для ілюстрації виконання команди циклу з лічильником розглянемо фрагмент проекту, в якому обчислюється сума $1^2 + 2^2 + 3^2 + 4^2$.

```

var s, i: integer;
begin
  s := 0;
  for i := 1 to 4 do
    begin
      s := s + i*i;
    end;
  Label1.Caption := IntToStr (s);
end;

```

Виконаємо цей фрагмент програми.

Команда	Результат виконання
<code>s := 0</code>	<code>s = 0</code>
<code>i := 1</code>	<code>i = 1</code>
<code>i <= 4</code>	<code>(1 <= 4) = true</code>
<code>s := s + i*i</code>	<code>s = 0 + 1*1 = 1</code>
<code>i := i + 1</code>	<code>i = 2</code>
<code>i <= 4</code>	<code>(2 <= 4) = true</code>
<code>s := s + i*i</code>	<code>s = 1 + 2*2 = 5</code>
<code>i := i + 1</code>	<code>i = 3</code>
<code>i <= 4</code>	<code>(3 <= 4) = true</code>
<code>s := s + i*i</code>	<code>s = 5 + 3*3 = 14</code>
<code>i := i + 1</code>	<code>i = 4</code>
<code>i <= 4</code>	<code>(4 <= 4) = true</code>
<code>s := s + i*i</code>	<code>s = 14 + 4*4 = 30</code>
<code>i := i + 1</code>	<code>i = 5</code>
<code>i <= 4</code>	<code>(5 <= 4) = false</code>
<code>Label1.Caption := IntToStr (s)</code>	<code>Label1.Caption = 30</code>



Звертаємо вашу увагу, що після закінчення виконання команди циклу з лічильником значення лічильника циклу (змінної *i*) на 1 більше, ніж значення **вираз2**, і це значення, за потреби, можна використовувати в наступних командах.



Для тих, хто хоче знати більше

Існує різновид команди циклу з лічильником, у якому лічильник не збільшується на 1, а зменшується на 1:

```
For <ім'я змінної> := <вираз1> downto <вираз2> do
begin
  <команди тіла циклу>
end;
```

(англ. *down to* – униз до).

Цей цикл продовжує виконуватися, якщо значення лічильника циклу більше або дорівнює (не менше) значенню **вираз2**.



Найважливіше в цьому пункті

Загальний вигляд команди циклу з лічильником такий:

```
for <ім'я змінної> := <вираз1> to <вираз2> do
begin
  <команди тіла циклу>
end;
```

Рядок **for <ім'я змінної> := <вираз1> to <вираз2> do** називають **рядком заголовка команди циклу з лічильником**. Змінну в рядку заголовка команди циклу з лічильником, що стоїть перед знаком присвоювання, називають **лічильником циклу**. **Лічильник циклу, вираз1 і вираз2** мають набувати тільки цілих значень.



Дайте відповіді на запитання

- 1°. Що таке цикл в алгоритмі?
- 2°. Який вигляд має команда циклу з лічильником в **Object Pascal**?
- 3°. Як виконується команда циклу з лічильником в **Object Pascal**?
- 4°. Який вигляд має блок-схема циклу з лічильником? Поясніть виконання фрагмента алгоритму.
- 5°. Чи можуть команди тіла циклу з лічильником виконуватися тільки один раз? Поясніть свою відповідь. Наведіть приклад.
- 6°. Чи можуть команди тіла циклу з лічильником не виконуватись жодного разу? Поясніть свою відповідь. Наведіть приклад.
- 7*. Чи може виконання циклу з лічильником ніколи не закінчитися? Поясніть свою відповідь. Наведіть приклад.
- 8*. У чому полягають відмінності у виконанні основних алгоритмічних структур: слідування, розгалуження, цикл.



Виконайте завдання

1*. Складіть блок-схему фрагмента алгоритму і виконайте його.

<p>а) $n := 1;$ $\text{for } i := 1 \text{ to } 4 \text{ do}$ $n := n * i;$</p>	<p>б) $p := 0; a := 2;$ $\text{for } i := 1 \text{ to } 5 \text{ do}$ begin $a := 3 * a + 1;$ $p := p + a;$ $\text{end};$</p>
--	---



2*. Складіть блок-схему фрагмента алгоритму і виконайте його.

<p>а) $n := 0;$ $\text{for } i := 1 \text{ to } 4 \text{ do}$ $n := n + i * i * i;$</p>	<p>б) $p := 0; a := 4;$ $\text{for } i := 1 \text{ to } 5 \text{ do}$ begin $a := 2 * a - 1;$ $p := p + a;$ $\text{end};$</p>
--	---

3*. Складіть блок-схему алгоритму знаходження суми п'яти чисел, перше з яких дорівнює 7, а кожне із чотирьох наступних на 3 більше від попереднього. Виконайте алгоритм.



4*. Складіть блок-схему алгоритму знаходження суми шести чисел, перше з яких дорівнює 100, а кожне з п'яти наступних у 5 разів менше від попереднього. Виконайте алгоритм.

5*. Складіть блок-схему алгоритму розв'язання задачі: Визначити, на скільки квадратів можна розрізати прямокутну смужку паперу із заданими довжиною та шириною, якщо на кожному кроці від смужки відрізати квадрат, сторона якого дорівнює меншій зі сторін смужки, що залишилася після попереднього відрізу. Виконайте алгоритм для двох різних значень початкових даних. Підберіть ці значення так, щоб команди тіла циклу виконались кілька разів; один раз.

6*. Створіть проект для обчислення суми n чисел, перше з яких дорівнює x , а кожне наступне на 5 більше від попереднього. Збережіть проект у папці **Завдання 6.6.6**, створеній у вашій папці.



7*. Створіть проект для обчислення суми n чисел, перше з яких дорівнює x , а кожне наступне дорівнює квадрату попереднього. Збережіть проект у папці **Завдання 6.6.7**, створеній у вашій папці.

8*. Створіть проект для розв'язання задачі: Клієнт поклав у банк певну суму грошей. Щорічно банк додає до суми 20 % від суми, що зберігається на рахунку після попереднього року. Визначити суму, яка буде на рахунку через n років. Збережіть проект у папці **Завдання 6.6.8**, створеній у вашій папці.



9*. Створіть проект для розв'язання задачі: Клієнт поклав у банк певну суму грошей. Щорічно банк додає до суми заздалегідь визначений відсоток від суми, що зберігається на рахунку після попереднього року. Визначити, на скільки збільшиться сума на рахунку через n років. Збережіть проект у папці **Завдання 6.6.9**, створеній у вашій папці.



6.7. ЦИКЛ З ПЕРЕДУМОВОЮ



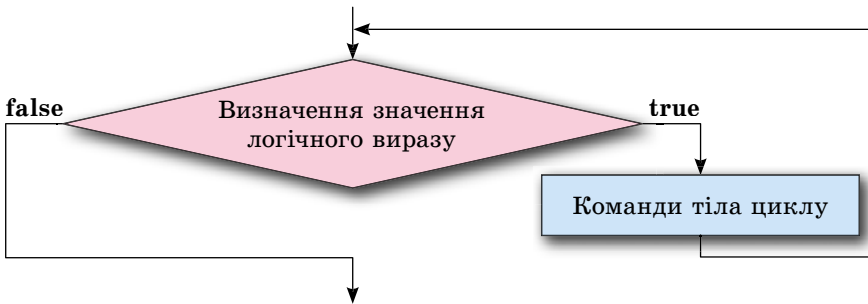
1. Який вигляд має і як виконується команда циклу з лічильником в **Object Pascal**?
2. Який вигляд має і як виконується блок-схема циклу з передумовою?
3. У яких випадках в алгоритмах використовується цикл з лічильником, а в яких – цикл з передумовою?

ЦИКЛ З ПЕРЕДУМОВОЮ

Ви вже знаєте, що цикл з передумовою як фрагмент алгоритму починається з команди перевірки умови й результатом виконання цієї команди може бути або **істина (Так, true)**, або **хиба (Ні, false)**. І залежно від результату виконання цієї команди – виконуватимуться команди тіла циклу або команда алгоритму, наступна за циклом.

У загальному випадку у цій команді визначається значення певного логічного виразу, яке може бути або **true**, або **false**.

Загальний вигляд блок-схеми циклу з передумовою такий (мал. 6.31):



Мал. 6.31. Блок-схема циклу з передумовою

КОМАНДА ЦИКЛУ З ПЕРЕДУМОВОЮ В ОБ'ЄКТ PASCAL

Команда циклу з передумовою в мові програмування **Object Pascal** має такий вигляд:

```

While<логічний вираз>
Then begin
  <команди тіла циклу>
end;
  
```

(англ. *while* – поки).

Виконання цієї команди відбувається так: обчислюється значення логічного виразу; якщо це значення **true**, то виконуються **команди тіла циклу**, після чого знову обчислюється значення логічного виразу, і якщо це значення знову **true**, то знову виконуються **команди тіла циклу**, після чого знову обчислюється значення логічного виразу; якщо значення логічного виразу дорівнює **false**, то **команди тіла циклу** не виконуються, а виконується команда, наступна за циклом.

Звертаємо вашу увагу:

- якщо в тілі циклу лише одна команда, то операторні дужки **begin** і **end** можна не ставити;
- серед команд тіла циклу можуть бути й лінійні фрагменти, і розгалуження, й інші цикли.

Розглянемо задачу, алгоритм розв'язування якої містить цикл з передумовою.

Задача. Скільки потрібно взяти доданків, перший з яких дорівнює заданому дійсному числу, а кожний наступний на 3 більший від попереднього, щоб їхня сума перевищила 100?

Даними цієї задачі є перший доданок. Вводитимемо його в поле.

Перед початком циклу присвоїмо змінній *s*, яку використаємо для збереження суми, значення першого доданка. А змінній *n*, яку використаємо для збереження кількості взятих доданків, присвоїмо значення 1, бо в сумі вже враховано один (перший) доданок.

До суми потрібно додавати наступні доданки, поки ця сума буде меншою, або дорівнюватиме 100. Тому логічний вираз у заголовку циклу матиме вигляд $s \leq 100$. У тілі циклу шукатимемо наступний доданок, додаватимемо його до поточного значення суми і збільшуватимемо кількість доданих доданків на 1.

Відповідний фрагмент проекту матиме такий вигляд:

```
var a, s: real; n: integer;
begin
  a := StrToFloat (Edit1.Text);
  s := a; n := 1;
  while s <=100 do
  begin
    a := a+3;
    s := s+a;
    n := n+1;
  end;
  Label1.Caption := IntToStr(n);
end;
```

Найважливіше в цьому пункті

Загальний вигляд блок-схеми циклу з передумовою показано на малюнку 6.31.

Команда циклу з передумовою в мові програмування **Object Pascal** має такий вигляд:

```
While<логічний вираз>
Then begin
  <команди тіла циклу>
end;
```

Виконання цієї команди відбувається так: обчислюється значення логічного виразу; якщо це значення **true**, то виконуються **команди тіла циклу**, після чого знову обчислюється значення логічного виразу; якщо значення логічного виразу дорівнює **false**, то **команди тіла циклу** не виконуються, а виконується команда, наступна за циклом.



Дайте відповіді на запитання

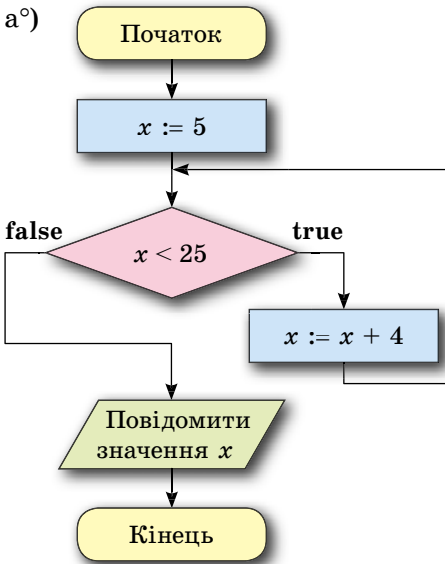
- 1°. Який вигляд має блок-схема циклу з передумовою? Поясніть виконання цього циклу.
- 2°. Чи можуть команди тіла циклу з передумовою не виконуватись жодного разу? Поясніть свою відповідь. Наведіть приклади.
- 3°. Чи може виконання циклу з передумовою ніколи не закінчитися? Поясніть свою відповідь. Наведіть приклади.
- 4°. Чим відрізняються між собою цикл з лічильником і цикл з передумовою?
- 5°. У чому полягають відмінності у виконанні основних алгоритмічних структур: слідування, розгалуження, цикл з лічильником і цикл з передумовою?
- 6°. Який загальний вигляд команди циклу з передумовою в **Object Pascal**?
- 7°. Як виконується команда циклу з передумовою в **Object Pascal**?



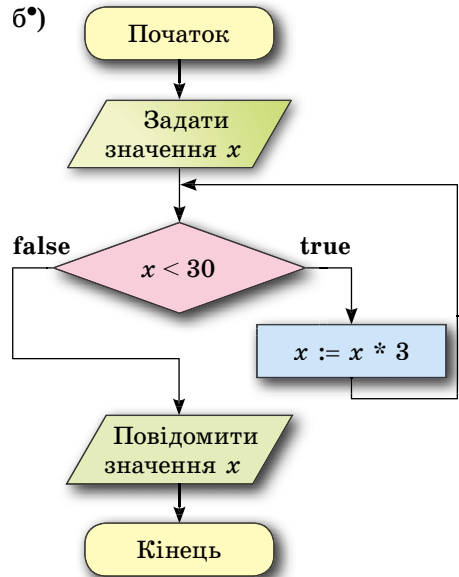
Виконайте завдання

1. Виконайте алгоритми.

а°)



б°)



2°. Виконайте фрагмент програми та з'ясуйте, якими будуть значення змінних після його завершення:

- | | |
|--|--|
| <p>а) <code>k := 1; a := 12;</code>
 <code>while a < 100 do</code>
 <code>begin</code>
 <code> a := 2*a - 4;</code>
 <code> k := k + 1;</code>
 <code>end;</code></p> | <p>б) <code>s := 0; a := 1; k := 1;</code>
 <code>while a < 50 do</code>
 <code>begin</code>
 <code> s := s + a;</code>
 <code> k := k + 1;</code>
 <code> a := k*k;</code>
 <code>end;</code></p> |
|--|--|



3*. Виконайте фрагмент програми та з'ясуйте, якими будуть значення змінних після його завершення:

<pre>а) k := 1; a := 100; while a > 10 do begin a := a/2; k := k + 1; end;</pre>	<pre>б) s := 0; a := 5; while a <= 100 do begin s := s + a; a := a*2; end;</pre>
--	--

4*. Складіть блок-схему алгоритму знаходження суми чисел, перше з яких дорівнює 7, кожне наступне на 5 більше від попереднього і всі вони не перевищують 100. Виконайте алгоритм.



5*. Складіть блок-схему алгоритму знаходження суми додатних чисел, перше з яких дорівнює 50, кожне наступне на 8 менше від попереднього. Виконайте алгоритм.

6*. Створіть проект, у якому можна обчислити кількість двоцифрових чисел серед чисел, перше з яких дорівнює 10, а кожне наступне більше від попереднього на задане додатне число. Збережіть проект у папці з іменем **Завдання 6.7.6**, створеній у вашій папці.



7*. Створіть проект для обчислення середнього арифметичного додатних чисел серед чисел, перше з яких дорівнює 100, а кожне наступне менше від попереднього на задане додатне число. Збережіть проект у папці з іменем **Завдання 6.7.7**, створеній у вашій папці.

8*. Створіть проект для розв'язання задачі: Клієнт поклав у банк певну суму грошей. Щорічно банк додає до суми наперед визначений відсоток від суми, що зберігається на рахунку після попереднього року. Через скільки років сума на рахунку перевищить S грн? Збережіть проект у папці з іменем **Завдання 6.7.8**, створеній у вашій папці.

ПРАКТИЧНА РОБОТА № 12

«Складання та виконання проектів з повтореннями та розгалуженнями для опрацювання величин»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте середовище **Lazarus**.
2. Створіть проект для розв'язування задачі: *Відомі довжини сторін двох прямокутників. Обчислити площі прямокутників та порівняти їх. Визначити, чи є прямокутники квадратами. Визначити, чи вміститься один прямокутник усередину другого.*

1. Розмістіть на формі поля для введення даних написи для виведення результатів і три кнопки.

2. У проекті:

- складіть обробник події **Click** для першої кнопки, виконання якого приведе до обчислення площ прямокутників та їх порівняння;
- складіть обробник події **Click** для другої кнопки, виконання якого, залежно від вибраних прапорців, приведе до визначення того, чи є відповідний прямокутник квадратом;
- складіть обробник події **Click** для третьої кнопки, виконання якого приведе до визначення, чи вміститься перший прямокут-



- ник усередину другого або другий прямокутник усередину першого, залежно від вибраного перемикача.
- Збережіть проект у папці з іменем **Практична 12_1**, створеній у вашій папці.
 - Створіть проект для розв'язування задачі: *Перед початком повені рівень води у річці становив H метрів. Під час повені щогодини рівень води збільшувався на P відсотків від рівня попередньої години. Яким буде рівень води через N годин після початку повені? Через скільки годин після початку повені рівень води буде не менше ніж K метрів?*
 - Розмістіть на формі поля для введення початкових даних написи з текстами, що пояснюватимуть призначення полів, і три кнопки.
 - Установіть на першій кнопці напис *Завдання 1*, на другій – *Завдання 2*, на третій – *Спочатку*, у полів – *порожній текст*.
 - Складіть обробник події **Click** для першої кнопки, виконання якого приведе до виведення у вікно повідомлення відповіді на перше запитання задачі.
 - Виконайте проект і переконайтеся, що результати його роботи правильні.
 - Складіть обробник події **Click** для другої кнопки, виконання якого приведе до знаходження відповіді на друге запитання задачі і виведення його в окремий напис.
 - Виконайте проект і переконайтеся, що результати його роботи правильні.
 - Складіть обробник події **Click** для третьої кнопки, виконання якого приведе до очищення тексту в полях та написі з відповіддю на друге запитання задачі.
 - Збережіть проект у папці з іменем **Практична 12_2**, створеній у вашій папці.
 - Закрийте середовище **Lazarus**.



6.8. ВІДОБРАЖЕННЯ ГРАФІЧНИХ ОБ'ЄКТІВ ЗАСОБАМИ МОВИ ПРОГРАМУВАННЯ



- Які графічні примітиви можна створювати у графічному редакторі та текстовому процесорі?
- Які властивості графічних примітивів ви знаєте?
- Які спільні властивості напису, кнопки і форми ви знаєте?

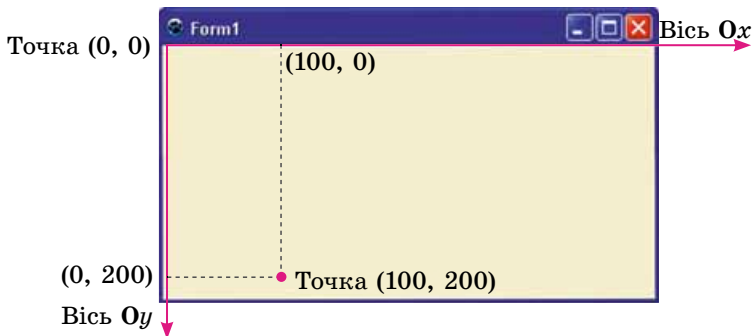
ЗАСОБИ ДЛЯ МАЛЮВАННЯ В ОБ'ЄКТ PASCAL

Графічні об'єкти у проектах, розроблених у **Lazarus**, можна створювати та відображати різними способами. Одним з них є малювання графічних примітивів за їх координатами на координатній площині.

Частина такої координатної площини розміщена, наприклад, на формі. Початок координат збігається з верхнім лівим кутом форми. Вісь Ox спрямована вправо, вісь Oy – униз. Частина координатної площини, що відобра-



жається на формі, має розміри ширини і висоти форми. На малюнку 6.32 червоним кольором позначено точку з координатами (100, 200).



Мал. 6.32. Частина координатної площини на формі

Для малювання на формі використовується її властивість **Canvas** (англ. *canvas* – полотно, канва).

Кожна точка (піксель) на формі має свої координати та колір. Координати можуть бути задані лише цілими числами. Колір можна задати константою типу **TColor**. Приклади констант і відповідних їм кольорів наведено в таблиці 6.1.

Таблиця 6.1

Константи для кольорів у Lazarus

Константа	Колір	Константа	Колір
<i>clBlack</i>	Чорний	<i>clSilve</i>	Сріблястий
<i>clWhite</i>	Білий	<i>clGray</i>	Сірий
<i>clRed</i>	Червоний	<i>clGreen</i>	Зелений
<i>clBlue</i>	Синій	<i>clNavy</i>	Темно-синій
<i>clLime</i>	Салатовий	<i>clOlive</i>	Оливковий

Для того щоб установити для точки з координатами (*x*, *y*) деякий колір, наприклад червоний, потрібно виконати команду:

```
Canvas.Pixels[x, y] := clRed;
```

Для створення графічних примітивів можна використовувати команди, які наведено в таблиці 6.2.

Таблиця 6.2

Команди для малювання графічних примітивів на координатній площині форми

Команда	Пояснення
<code>Canvas.MoveTo (x, y)</code>	Вибір на формі точки (<i>x</i> , <i>y</i>), з якої буде розпочинатися малювання ліній

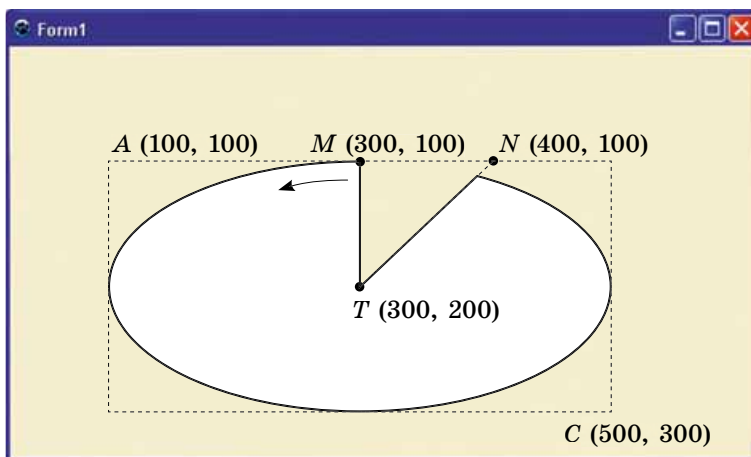


Продовження таблиці 6.2

Команда	Пояснення
Canvas.LineTo (x, y)	Малювання відрізка від останньої точки, у якій відбувалось малювання, до точки (x, y) . Використання послідовно кількох таких команд приведе до малювання відрізка або ламаної
Canvas.Rectangle $(x1, y1, x2, y2)$	Малювання прямокутника, у якого протилежні вершини розміщені в точках $(x1, y1)$ та $(x2, y2)$, а сторони паралельні осям координат
Canvas.Ellipse $(x1, y1, x2, y2)$	Малювання еліпса, вписаного у прямокутник, у якого протилежні вершини розміщені в точках $(x1, y1)$ та $(x2, y2)$, а сторони паралельні осям координат. Прямокутник при цьому не відображається
Canvas.Pie $(x1, y1, x2, y2, x3, y3, x4, y4)$	Малювання замкненої фігури – сектора еліпса. Точки $(x1, y1)$ та $(x2, y2)$ задають прямокутник, у який вписано еліпс. Початкова точка дуги визначається перетином еліпса з прямою, що проходить через його центр і точку $(x3, y3)$. Кінцева точка дуги визначається перетином еліпса з прямою, що проходить через його центр і точку $(x4, y4)$. Дуга малюється проти годинникової стрілки від початкової до кінцевої точки (мал. 6.33).

У наведеному на малюнку 6.33 прикладі еліпс вписаний у прямокутник, що має протилежні вершини в точках $A (100, 100)$ та $C (500, 300)$. Сектор еліпса утворений відрізками, що сполучають центр прямокутника (точку $T (300, 200)$) з точками $M (300, 100)$ та $N (400, 100)$, та дугою еліпса, що починається в точці $M (300, 100)$, а закінчується при перетині еліпса з відрізком TN .

Для кожного примітива можна задати значення властивостей його контуру та внутрішньої області. Значення властивостей контуру та внутріш-



Мал. 6.33. Сектор еліпса, створений командою `Canvas.Pie (100, 100, 500, 300, 300, 100, 400, 100)`

ньої області графічного примітива збігаються зі значеннями властивостей форми, наведеними у таблиці 6.3.

Таблиця 6.3

Властивості форми для встановлення значень властивостей контуру та внутрішньої області графічних примітивів

Властивість	Призначення та приклади значень	Значення за замовчуванням
Canvas.Pen.Color	Колір лінії контуру	Чорний
Canvas.Pen.Width	Товщина ліній контуру	1 піксель
Canvas.Pen.Style	Стиль лінії контуру, наприклад, <i>psSolid</i> – суцільна лінія, <i>psDash</i> – штрихова лінія, <i>psDot</i> – пунктирна лінія та ін.	Суцільна
Canvas.Brush.Color	Колір заливки внутрішньої області	Білий
Canvas.Brush.Style	Стиль заливки внутрішньої області, наприклад, <i>bsSolid</i> – суцільна заливка, <i>bsCross</i> – заливка у вигляді сітки з горизонтальними та вертикальними лініями, <i>bsDiagCross</i> – заливка у вигляді сітки з діагональними лініями та ін.	Суцільна

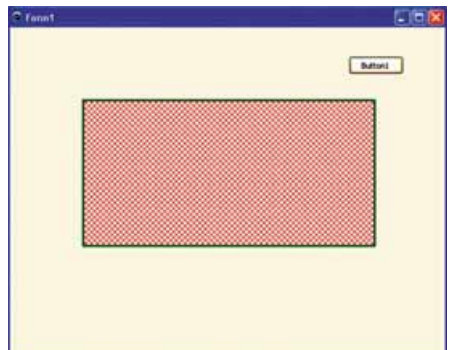
Створюючи зображення, що складається з графічних примітивів, варто спочатку встановити значення властивостей контуру та внутрішньої області фігури, після чого відобразити саму фігуру. На малюнку 6.34 зображено результат виконання наступної процедури створення кольорового прямокутника:

```

procedure TForm1.Button1Click(Sender: TObject);
begin
    Canvas.Pen.Color := clGreen; // колір контуру зелений
    Canvas.Pen.Width := 3; // товщина контуру 3 пікселі
    Canvas.Brush.Color := clRed; // колір заливки червоний
    Canvas.Brush.Style := bsDiagCross; (* стиль заливки – сітка з діагональними лініями *)
    Canvas.Rectangle (100, 100, 500, 300); (* прямокутник з протилежними вершинами в точках (100, 100) та (500, 300) і сторонами, паралельними осям координат *)
end;
    
```

Для того щоб малюнок відображався на формі одразу після запуску проекту на виконання, потрібно команди малювання розміщувати у процедурі – обробнику події **Paint** для форми.

Мал. 6.34. Результат виконання процедури малювання графічного примітива





КОМПОНЕНТ ФІГУРА ДЛЯ СТВОРЕННЯ ГРАФІЧНИХ ПРИМІТИВІВ

Графічні примітиви в проектах **Lazarus** також можна створювати, використавши компонент **Фігура**.

Щоб розмістити **Фігуру** на формі, потрібно двічі клацнути на кнопці **TShape** (англ. *shape* – форма, фігура) на вкладці **Additional** (англ. *additional* – додатковий) вікна середовища **Lazarus**.

Властивості **Фігури** **Name**, **Enabled**, **Visible** аналогічні до відповідних властивостей форми. Як і для форми, значення властивостей **Фігури** **Top**, **Left**, **Height** і **Width** можна змінити або на вкладці **Властивості**, або перетягуванням.

Компонент **Фігура** має властивості, що визначають колір і стиль контуру та внутрішньої області (табл. 6.4).

Таблиця 6.4

Властивості компонента **Фігура**

Властивість	Що визначає її значення
Pen.Color	Колір контуру
Pen.Width	Товщина контуру
Pen.Style	Стиль лінії контуру
Brush.Color	Колір заливки внутрішньої області
Brush.Style	Стиль заливки внутрішньої області

Значення цих властивостей можна встановлювати на вкладці **Властивості** у вікні **Інспектор об'єктів** або з використанням команди присвоєвання. Ці значення можуть бути такими самими, як і для графічних примітивів.

Крім того, **Фігура** має властивість **Shape**, що визначає її форму. Ця властивість може набувати, наприклад, значень **stRectangle** – прямокутник, **stCircle** – коло, **stSquare** – квадрат, **stDiamond** – ромб, **stTriangle** – трикутник та ін. За замовчуванням значення цієї властивості визначає квадрат з контуром чорного кольору товщиною 1 піксель і суцільною заливкою білого кольору.

На малюнку 6.35 наведено приклад зображення, створеного з використанням кількох об'єктів **Фігура** з різними значеннями властивостей.



Мал. 6.35. Створення зображення з використанням об'єктів **Фігура**

ВІДОБРАЖЕННЯ ЗОБРАЖЕНЬ ІЗ ЗОВНІШНІХ ФАЙЛІВ

У проектах можна відображати зображення, що містяться в зовнішніх файлах. Для цього призначений компонент **Область зображення**. Для його розміщення на формі потрібно двічі клацнути на кнопці **TImage** (англ. *image* – зображення) на вкладці **Additional** у вікні середовища **Lazarus**.

Властивості **Області зображення** **Name**, **Enabled**, **Visible** аналогічні до відповідних властивостей форми. Як і для форми, значення властивостей області зображення **Top**, **Left**, **Height** і **Width** можна змінити або на вкладці **Властивості**, або перетягуванням.

Відображення зображення, що міститься в зовнішньому файлі, визначається значеннями таких властивостей (табл. 6.5):

Таблиця 6.5

Властивості компонента **Область зображення**

Властивість	Пояснення
Picture	Вибір файлу з графічним зображенням для відображення в Області зображення
AutoSize	При значенні true компонент Область зображення набуває розмірів зображення, отриманого з файлу. При значенні false розміри компонента не залежать від розміру зображення, а визначаються значеннями відповідних властивостей
Stretch	При значенні true зображення, отримане з файлу, набуває розмірів Області зображення . При значенні false зображення зберігає свої розміри і не залежить від розмірів компонента
Proportional	При значенні true ширина та висота зображення, отриманого з файлу, будуть змінюватися пропорційно при розміщенні в Області зображення . При значенні false ширина та висота зображення можуть змінюватися не пропорційно

Приклад відображення у проєкті зображення, вставленого із зовнішнього файлу, наведено на малюнку 6.36.

Найважливіше в цьому пункті

Графічні примітиви можна малювати за їх координатами на координатній площині, частина якої розміщена на формі. Для цього використовується її властивість **Canvas**. Кожна точка на формі має свої координати та колір. Для кожного графічного примітива можна задати значення властивостей його контуру та внутрішньої області.

Графічні примітиви в проєктах **Lazarus** можна створювати також, використавши компонент **Фігура**. Цей компонент має властивості, значення яких визначають колір і стиль контуру та внутрішньої області.

Можна відображати зображення, що містяться в зовнішніх файлах. Для цього призначений компонент **Область зображення**.



Мал. 6.36. Відображення у проєкті зображення із зовнішнього файлу

Дайте відповіді на запитання

- 1°. Як розміщена на формі координатна площина?
- 2°. Яка властивість форми використовується для малювання графічних примітивів?

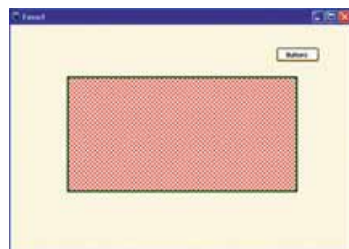


- 3°. Які команди використовуються для малювання графічних примітивів – лінія, ламана, прямокутник, еліпс, сектор?
- 4°. Як змінити контур і внутрішню область графічних примітивів?
- 5°. Який компонент використовується для створення графічних примітивів? Як його встановити на формі?
- 6°. Значення яких властивостей **Фігури** визначають вигляд її контуру та внутрішньої області?
- 7°. Значення якої властивості **Фігури** визначають її форму? Яких значень вона може набувати?
- 8°. Який компонент використовується для відображення на формах графічних зображень, що містяться в зовнішніх файлах? Як його встановити на формі?
- 9°. Як вибрати файл для відображення в **Області зображення**?



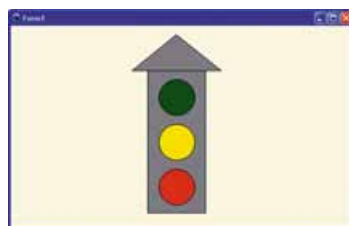
Виконайте завдання

- 1°. Створіть проект, при виконанні якого після вибору кнопки буде намальовано на формі прямокутник за зразком, що наведено на малюнку. Збережіть проект у папці з іменем **Завдання 6.8.1**, створеній у вашій папці.
- 2°. Створіть проект, після запуску якого на формі буде намальовано зображення таких графічних примітивів:
 - а) відрізок від точки (10, 10) до точки (200, 100) червоного кольору, тип лінії – штрих-пунктир, товщина лінії – 5;
 - б) ламана із чотирьох відрізків синього кольору, стиль лінії – крапками, товщина – 3;
 - в) прямокутник з протилежними вершинами в точках (50, 50) та (250, 150), лінії контуру – червоні суцільні товщиною 1, заливка сріблястого кольору, стиль заливки – горизонтальні лінії;
 - г) еліпс, вписаний у прямокутник з вершинами в точках (100, 20) та (200, 300), лінії контуру – жовті пунктирні товщиною 2, заливка салатого кольору, стиль заливки – вертикальні лінії;
 - г) сектор еліпса, вписаного у прямокутник з вершинами (50, 50) та (150, 150), початкова точка дуги (150, 100), кінцева – (100, 150), лінії контуру – білі суцільні товщиною 4, заливка оливкового кольору, стиль заливки – нахилені лінії.
 Збережіть проект у папці з іменем **Завдання 6.8.2**, створеній у вашій папці.



До завдання 1

- 3°. Створіть проект, при виконанні якого буде побудовано зображення з фігур відповідно до зразка на малюнку. Збережіть проект у папці з іменем **Завдання 6.8.3**, створеній у вашій папці.
- 4°. Створіть проект, при виконанні якого з фігур буде побудовано зображення:
 - а) вантажівки;



До завдання 3



- б) робота;
- в) комп'ютера;
- г) ракети.

Стиль і колір фігур виберіть самостійно. Збережіть проект у папці з іменем **Завдання 6.8.4**, створеній у вашій папці.

- 5°. Створіть проект, при виконанні якого на формі відобразатиметься зображення, що міститься у файлі, наприклад **Розділ 6\Пункт 6.8\завдання 6.8.5.jpg**. Збережіть проект у папці з іменем **Завдання 6.8.5**, створеній у вашій папці.
- 6°. Створіть проект, при виконанні якого на формі як фон буде розміщено зображення з файлу, наприклад **Розділ 6\Пункт 6.8\завдання 6.8.6.jpg**. Поверх нього з фігур буде створено зображення автомобіля. Збережіть проект у папці з іменем **Завдання 6.8.6**, створеній у вашій папці.
- 7*. Розмістіть на формі кнопку, фігуру, три прапорці та поле з розкривним списком. Створіть проект, у якому при виборі кнопки залежно від вибраного значення у списку фігура змінюватиме форму, набуваючи значення квадрат, коло або трикутник, а залежно від вибраних прапорців контур матиме товщину 1 або 4, колір заливки буде жовтим або зеленим, стиль заливки – суцільний або діагональні лінії. Збережіть проект у папці з іменем **Завдання 6.8.7**, створеній у вашій папці.

ПРАКТИЧНА РОБОТА № 13

«Складання та виконання алгоритмів з графічним відображенням даних»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте вікно **Lazarus**.
2. Створіть проект, після запуску якого на виконання:
 - а) у верхній частині форми з графічних примітивів буде зображено сонце;
 - б) у нижній частині форми буде розміщено зображення з файлу **Розділ 6\Пункт 6.8\практична 13.jpg**;
 - в) з фігур буде створено зображення будинку.
3. Збережіть проект у папці з іменем **Практична 13**, створеній у вашій папці.
4. Запустіть проект на виконання.
5. Проаналізуйте отриманий результат.
6. Закрийте вікно виконання проекту.
7. Закрийте вікно середовища **Lazarus**.

Розділ 7. **Опрацювання числових даних**

У цьому розділі ви дізнаєтеся про:

абсолютні, відносні
й мішані посилання
на комірки та діапазони
комірок

створення та
налагодження діаграм
різного типу, вибір типу
діаграми

встановлення параметрів
сторінки та друкування
електронної таблиці

умовне форматування

призначення й
використання
математичних,
статистичних, логічних
функцій табличного
процесора

упорядковування даних
у таблицях

автоматичні та
розширені фільтри,
проміжні підсумки



7.1. АБСОЛЮТНІ, ВІДНОСНІ Й МІШАНІ ПОСИЛАННЯ. ВСТАНОВЛЕННЯ ПАРАМЕТРІВ СТОРІНКИ. ДРУКУВАННЯ ЕЛЕКТРОННОЇ ТАБЛИЦІ



1. Як у табличному процесорі записується посилання на клітинку? Для чого воно використовується?
2. Що таке модифікація формул? Коли вона відбувається? За якими правилами здійснюється?
3. Які властивості має сторінка текстового документа? Яких значень можуть набувати ці властивості? Як установити значення властивостей сторінки?

ВІДНОСНІ, АБСОЛЮТНІ ТА МІШАНІ ПОСИЛАННЯ

У 7 класі ви вже вивчали, що в електронних таблицях у формулах можна використовувати посилання на клітинки чи діапазони клітинок. Воно забезпечує автоматичне переобчислення значень формул, де використовуються посилання. Також ви знаєте, що під час копіювання формул посилання в них модифікуються. Але інколи потрібно, щоб під час копіювання формул певні посилання не модифікувалися.

Посилання, яке модифікується під час копіювання формули, називають відносним.

Посилання, яке не модифікується під час копіювання формули, називають абсолютним.

Для того щоб посилання було абсолютним, потрібно перед номером стовпця та перед номером рядка ввести символ \$.

Наприклад, посилання **B10** є відносним, а посилання **\$B\$10** – абсолютним.

Розглянемо приклад, який пояснює доцільність використання абсолютних посилань.

Нехай у клітинках діапазону **C3:C7** (мал. 7.1) міститься кількість відмінників у п'яти 8–10-х класах школи, а в клітинці **C8** обчислена їх загальна кількість. У клітинках діапазону **D3:D7** потрібно обчислити відсоток відмінників у кожному із цих класів по відношенню до загальної кількості відмінників.

	A	B	C	D	E
1					
2			Відмінники	Відсоток	
3		8 А	3	18,75	
4		8 Б	1	6,25	
5		9 А	5	31,25	
6		9 Б	4	25,00	
7		10	3	18,75	
8		Усього	16		

Для обчислення цих відсотків у клітинках діапазону **D3:D7** повинні знаходитися такі формули:

$$D3 = C3/C8*100$$

$$D4 = C4/C8*100$$

$$D5 = C5/C8*100$$

$$D6 = C6/C8*100$$

$$D7 = C7/C8*100$$

Якщо проаналізувати всі ці формули, то можна зробити висновок, що посилання в чисельнику змінюються,

Мал. 7.1. Використання відносних і абсолютних посилань



а посилання у знаменнику – ні. Якщо ми введемо в клітинку D3 формулу $=C3/C8*100$ і скопіюємо її в клітинки діапазону D4:D7, то ці посилання у формулі будуть модифікуватися. Потрібно зробити так, щоб посилання C3 під час копіювання формули модифікувалося, а посилання C8 – ні. Тобто формула у клітинці D3 повинна мати такий вигляд: $=C3/(\$C\$8*100)$.

Якщо в посиланні символ \$ додано тільки перед номером стовпця або тільки перед номером рядка, наприклад \$B10 або B\$10, то під час копіювання такі посилання модифікуються частково: змінюється лише номер стовпця або лише номер рядка, біля якого не стоїть символ \$. Такі посилання називають *мішаними*.

Посилання, у якому під час копіювання формули модифікується або номер стовпця, або номер рядка, називають *мішаним*.

Змінити вид посилання у формулі можна під час його введення або редагування послідовним натисненням клавіші F4. При цьому види посилань змінюються по черзі в порядку: відносне – абсолютне – мішане.

Якщо у формулі для посилання на клітинку використати її ім'я, то під час копіювання цієї формули це посилання модифікуватися не буде. Отже, **ім'я клітинки у формулі є абсолютним посиланням на цю клітинку**. Тому в розглянутому вище прикладі можна було клітинці C8 надати ім'я, наприклад **Відмінники**, і тоді формула у клітинці D3 мала б такий вигляд: $=C3/Відмінники*100$.

Приклади абсолютних, відносних і мішаних посилань показано в таблиці 7.1.

Таблиця 7.1

Типи посилань

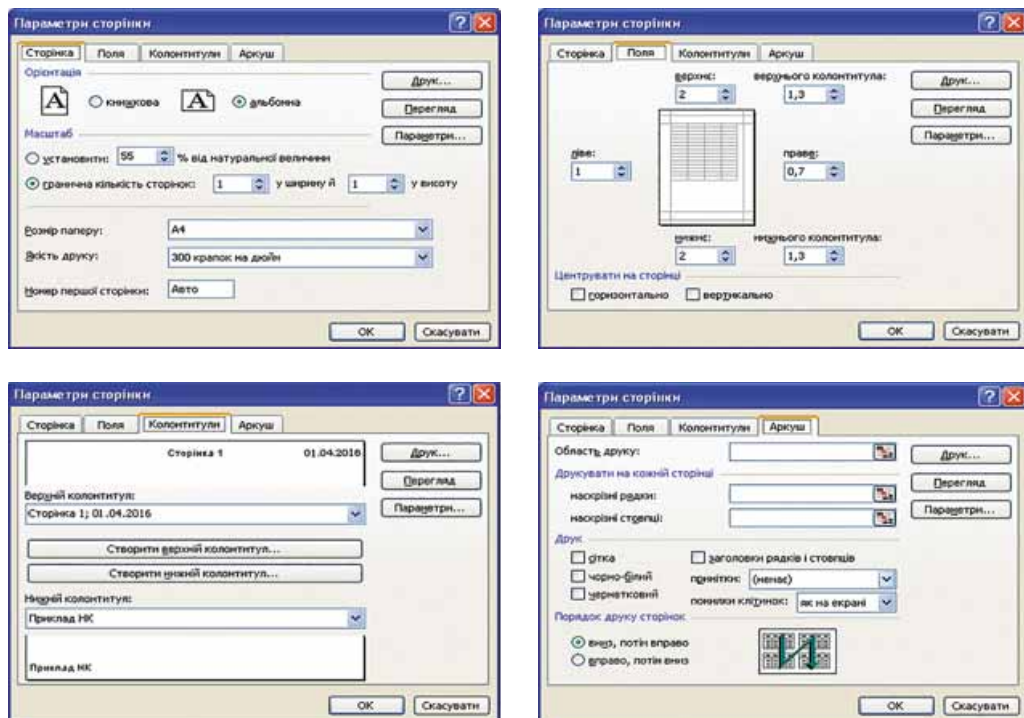
Посилання	Тип	Пояснення
C3	Відносне	Під час копіювання формули може змінюватися і номер стовпця, і номер рядка
\$A\$1	Абсолютне	Під час копіювання формули номер стовпця і номер рядка незмінні
курс	Абсолютне	Під час копіювання формули номер стовпця і номер рядка незмінні
C\$1	Мішане	Під час копіювання формули номер стовпця може змінюватися, а номер рядка змінюватися не може
\$A3	Мішане	Під час копіювання формули номер стовпця незмінний, а номер рядка може змінюватися

ВСТАНОВЛЕННЯ ЗНАЧЕНЬ ВЛАСТИВОСТЕЙ СТОРІНКИ

Ви вже вмієте друкувати текстові документи і знаєте, що перед цим потрібно встановити значення властивостей сторінки документа (розмір аркуша, орієнтацію сторінки, поля тощо), потім у режимі **Попереднього перегляду** переглянути зовнішній вигляд документа і згодом надрукувати документ.

Аналогічним є й алгоритм друкування електронної таблиці. Однак друкування в **Excel 2007** має певні відмінності порівняно з аналогічною операцією у **Word 2007**.

Установлення значень властивостей сторінки здійснюється елементами керування групи **Параметри сторінки** на вкладці **Розмітка сторінки** на **Стрічці** або на вкладках діалогового вікна **Параметри сторінки** (мал. 7.2).



Мал. 7.2. Вкладки діалогового вікна **Параметри сторінки**

На вкладці **Сторінка** можна встановити значення таких властивостей:

- орієнтація аркуша, на якому здійснюватиметься друк;
- масштаб у відсотках для тієї частини аркуша електронної таблиці, яку буде надруковано;
- розмір аркуша паперу для друку;
- якість друку;
- номер першої сторінки

тощо.

На вкладці **Поля** можна встановити значення таких властивостей аркуша, на якому здійснюватиметься друк: розміри полів, розміри колонтитулів, центрування таблиці на аркуші.


На вкладці **Колонтитули** можна створити верхній і нижній колонтитули, обравши відповідну кнопку, або вибрати потрібний колонтитул у полях зі списком **Верхній колонтитул** або **Нижній колонтитул**.

На вкладці **Аркуш** можна встановити значення таких властивостей аркуша електронної таблиці:

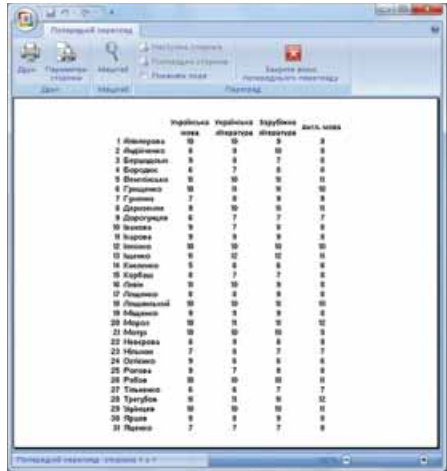


- область друку – діапазон клітинок аркуша електронної таблиці, які виводитимуться на друк;
- друк наскрізних рядків і стовпців на кожній сторінці – діапазон клітинок, які будуть повторюватися на кожній сторінці, коли надрукована частина таблиці на папері займає більше ніж одну сторінку;
- порядок друку сторінок – вниз, потім вправо чи вправо, потім вниз тощо.

ДРУКУВАННЯ ЕЛЕКТРОННОЇ ТАБЛИЦІ

Для того щоб побачити, який вигляд матиме електронна таблиця на папері, потрібно перейти до режиму **Попередній перегляд** (мал. 7.3), який встановлюється *Office* ⇒ *Друк* ⇒ *Попередній перегляд*, або в режимі перегляду електронної таблиці **Розмітка сторінки** (мал. 7.4), який встановлюється вибором кнопки  у Рядку стану.

У режимі **Попередній перегляд** буде відображено таблицю, підготовлену до друку, яка розділена на сторінки. Якщо область друкування не вміщується на одній сторінці аркуша для друкування, *Excel 2007* автоматично розділяє цю область на кілька сторінок (такий випадок показано на мал. 7.4).



Мал. 7.3. Режим Попередній перегляд

С6		fx		Андійченко																
1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q			
2								Оцінки за I семестр												
3																				
4			Українська мова	Українська література	Зарубіжна література	Англ. мова	Історія України	Всесвітня історія	Алгебра	Геометрія	Фізика	Хімія	Інформатика	Біологія	Економіка	Фізична культура				
5	1	Дніперова	10	9	9	9	9	9	7	9	6	7	9	7	7	11				
6	2	Андійченко	9	10	9	10	10	11	11	9	9	9	10	11	7	9				
7	3	Бірюцька	9	9	7	9	11	11	5	6	5	6	6	6	7	9				
8	4	Борозок	6	7	8	8	9	9	8	10	10	11	11	10	10	10				
9	5	Венгінський	11	10	11	11	10	10	9	9	9	9	9	11	11	11				
10	6	Грищенко	10	11	11	10	9	9	8	7	7	7	9	12	12					
11	7	Гришко	7	8	9	9	9	9	7	7	7	6	8	9	10	11				
12	8	Деризеня	9	10	11	11	11	11	9	9	8	8	9	9	11	10				
13	9	Дорожничев	6	7	7	7	8	8	7	7	6	6	7	8	8	9				
14	10	Іванова	9	7	8	8	9	9	7	7	7	7	9	9	8	11				
15	11	Ізарова	9	9	9	8	7	7	7	8	9	7	7	9	9	9				
16	12	Ілленко	10	10	10	10	10	11	9	9	8	9	8	8	9	11				
17	13	Іщенко	12	12	11	10	10	11	11	11	11	11	12	12	12	12				
18	14	Киселев	5	6	6	6	7	7	5	5	5	5	6	6	6	7				
19	15	Корбаш	8	7	7	8	9	8	8	8	7	6	7	8	9	10				
20	16	Леві	11	10	9	9	9	9	9	9	10	10	11	11	11	11				
21	17	Лещенко	9	9	9	9	9	9	9	9	9	9	9	9	10	10				
22	18	Лещенський	10	10	11	10	9	9	9	8	8	7	8	7	9	10				
23	19	Мищенко	9	9	9	9	10	10	7	7	8	8	9	11	11	11				
24	20	Мороз	10	11	11	12	12	12	11	11	11	10	9	9	11	11				
25	21	Мотуза	10	10	10	9	9	9	8	9	9	9	10	10	10	11				
26	22	Немерова	8	9	9	9	8	9	9	9	9	8	10	10	10	11				
27	23	Нільман	7	8	7	7	8	8	9	9	8	8	9	9	9	9				
28	24	Овсієнко	9	6	6	6	7	8	9	9	10	10	10	10	11	11				
29	25	Рогова	9	7	8	8	9	10	10	11	11	10	9	9	9	10				
30	26	Рябов	10	10	10	11	11	11	9	9	9	9	11	10	10	10				
31	27	Тіщенко	6	6	7	7	8	8	8	9	10	10	11	11	11	9				
32	28	Тригубов	11	11	11	12	12	12	10	10	10	11	11	10	10	10				
33	29	Урбанець	10	10	10	11	11	11	9	9	9	8	10	10	9	9				
34	30	Ярцев	8	9	9	9	9	9	9	9	9	7	7	7	9	9				
35	31	Яценко	7	7	7	8	8	7	7	6	6	5	5	8	8	10				

Мал. 7.4. Режим перегляду Розмітка сторінки



Мал. 7.5. Вікно Друк

У режимі **Попередній перегляд** можна переглядати сторінки по одній, а в режимі **Розмітка сторінки** – одразу бачити кілька сторінок, розділених штриховими лініями. Переміщуючи штрихові лінії, можна змінювати розподіл вмісту по сторінках.

Після закриття режиму **Попередній перегляд** на аркуші таблиці з'являються вертикальні й горизонтальні штрихові лінії, які демонструють розбиття таблиці на сторінки.

Якщо на **Панелі швидкого доступу** в Excel 2007 є кнопка **Швидкий друк**, то її вибір або виконання *Office* ⇒ *Друк* ⇒ *Швидкий друк* здійснює друкування однієї копії частини поточного аркуша книги, яка заповнена даними, зі значеннями властивостей друку за замовчуванням.

Якщо потрібно встановити інші параметри друку, то слід виконати *Office* ⇒ *Друк* ⇒ *Друк* і у вікні **Друк**, що відкрилося (мал. 7.5), установити потрібні значення: доступний принтер; кількість копій, які буде надруковано; об'єкт для друку (виділений діапазон, виділені аркуші чи всю книгу). За допомогою перемикачів **Усі** і **Сторінки** можна вказати номери окремих сторінок чи їх діапазонів для друку.

Найважливіше в цьому пункті

У формулах можна використовувати різні види посилань на клітинки: **відносні** (модифікуються під час копіювання формули), **абсолютні** (не модифікуються під час копіювання формули), **мішані** (модифікуються частково).

Для того щоб посилання було абсолютним, потрібно перед номером стовпця та номером рядка ввести символ **\$**. У мішаних посиланнях символ **\$** вводиться або біля номера стовпця, або біля номера рядка.

Якщо у формулі для посилання на клітинку використати її ім'я, то під час копіювання цієї формули це посилання модифікуватися не буде, воно вважається абсолютним.

Встановлення значень параметрів сторінки електронної таблиці для друку здійснюється на **Стрічці** елементами керування групи **Параметри сторінки** на вкладці **Розмітка сторінки** або на вкладках діалогового вікна **Параметри сторінки** цієї групи.

Для того щоб переглянути, який вигляд матиме електронна таблиця на папері, потрібно перейти до режиму **Попередній перегляд**, який встановлюється *Office* ⇒ *Друк* ⇒ *Попередній перегляд*, або в режимі перегляду електронної таблиці **Розмітка сторінки**, який встановлюється вибором кнопки у **Рядку стану**.

Для того щоб надрукувати електронну таблицю, потрібно виконати *Office* ⇒ *Друк* ⇒ *Швидкий друк* (зі значеннями властивостей сторінки за замовчуванням) або *Office* ⇒ *Друк* ⇒ *Друк*.

**Дайте відповіді на запитання**

- 1°. Які види посилань на клітинки можна використовувати у формулах в електронних таблицях?
- 2°. Як записують відносне, абсолютне та мішане посилання на клітинку B2?
- 3°. Чим відрізняється використання різних видів посилань?
- 4°. До якого виду посилань належить ім'я клітинки?
- 5°. Які властивості сторінки електронної таблиці?
- 6°. Як установити значення властивостей друку електронної таблиці?
- 7°. Для чого використовують попередній перегляд електронної таблиці? Які його можливості?
- 8°. Як попередньо переглянути, на які сторінки буде розділено електронну таблицю під час друкування?
- 9°. Як виконати швидкий друк електронної таблиці?
- 10°. Як надрукувати електронну таблицю?

**Виконайте завдання**

- 1°. У клітинці E3 знаходиться формула: $=\$B\$3+C4$. Запишіть, який вигляд матиме ця формула, якщо її скопіювати у клітинку:

а) E7;	в) C3;	д) K15;	є) C3;
б) E7;	г) E11;	е) T34;	ж) A2.
- 2°. У клітинці K8 знаходиться формула: $=E5+\$C3$. Запишіть, який вигляд матиме ця формула, якщо її скопіювати у клітинку:

а) C9;;	в) B4;	д) D8;	є) I2;
б) K7;	г) E9;	е) T23;	ж) F5.
- 3°. Відкрийте вказаний учителем файл (наприклад, Розділ 7\пункт 7.1\зразок 7.1.3.xlsx).
 - а*) На аркуші **Аркуш1** у клітинку F3 запишіть формулу для знаходження прибутку фірми за рік, яку потім скопіюйте з використанням маркера заповнення у клітинки F4:F9. Які посилання повинна містити ця формула: абсолютні, відносні чи мішані? Відповідь поясніть.
 - б*) На аркуші **Аркуш2** у клітинку E4 запишіть формулу для визначення ціни товару у гривнях (курс євро знаходиться у клітинці B3). Скопіюйте формулу з використанням маркера заповнення в клітинки E5:E12. Які посилання повинна містити ця формула: абсолютні, відносні чи мішані? Відповідь поясніть.
 - в*) На аркуші **Аркуш3** у клітинку D10 уведіть формулу для обчислення загальної кількості учнів у 8-х класах. Надайте цій клітинці ім'я **Учні_8**. Використайте це ім'я у формулах для обчислення відсотків кількості учнів 8-А класу по відношенню до загальної кількості учнів 8-х класів. Скопіюйте цю формулу з використанням маркера заповнення у клітинки для знаходження відсотків кількості учнів інших 8-х класів по відношен-



ню до загальної кількості учнів 8-х класів. Чи відбулася модифікація формули? Відповідь поясніть.

- Збережіть книгу у вашій папці у файлі з іменем **завдання 7.1.3.xlsx**.
- 4*. Створіть таблицю для визначення вартості українських банкнот у доларах, євро та фунтах стерлінгів. Передбачте зберігання курсів долара, євро та фунта стерлінгів у окремих клітинках. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.1.4.xlsx**.
- 5*. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\пункт 7.1\зразок 7.1.5.xlsx**). Виконайте двома способами попередній перегляд сторінок друкування електронної таблиці з аркуша **Аркуш1**. У кожному режимі **Попереднього перегляду** перегляньте всі сторінки. Виконайте друк цього аркуша електронної таблиці.

ПРАКТИЧНА РОБОТА № 14

«Розв'язування задач на обчислення»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Практична 14\зразок.xlsx**).
2. На аркуші **Аркуш1** у клітинку **F2** уведіть формулу для знаходження шляху, що пройшов перший турист за три дні. Скопіюйте формулу з використанням маркера заповнення у клітинки **F3:F7**.
3. На аркуші **Аркуш2** у клітинку **E3** уведіть формулу для визначення ціни товару в доларах (курс долара знаходиться у клітинці **B2**), яку потім скопіюйте з використанням маркера заповнення у клітинки **E4:E10**.
4. На аркуші **Аркуш3** у клітинку **B10** уведіть формулу для обчислення загальної кількості учнів у школі. Надайте цій клітинці ім'я **Учні**. Використайте це ім'я у формулах для обчислення відсотків кількості учнів паралелі 1-х класів по відношенню до загальної кількості учнів школи. Скопіюйте цю формулу з використанням маркера заповнення у клітинки для знаходження відсотків кількості учнів інших паралелей класів по відношенню до загальної кількості учнів школи.
5. Збережіть книгу у вашій папці у файлі з іменем **практична 14.xlsx**.

7.2. ПРИЗНАЧЕННЯ Й ВИКОРИСТАННЯ МАТЕМАТИЧНИХ, СТАТИСТИЧНИХ, ЛОГІЧНИХ ФУНКЦІЙ ТАБЛИЧНОГО ПРОЦЕСОРА



1. Які функції в Excel ви знаєте? Як вставити функцію у формулу в Excel?
2. Які параметри має функція в Excel? Що може бути аргументом функції?
3. Як записується команда розгалуження в мові **Object Pascal**? Як вона виконується?

Ви вже знаєте, що табличний процесор **Excel 2007** має бібліотеку вбудованих функцій, до якої входить більше ніж 300 різноманітних функцій. Усі вони для зручності розподілені на категорії: *математичні, статистичні, текстові, логічні, фінансові* та ін.



У 7 класі ви вже вивчали функції в табличному процесорі та розглядали використання функцій **SUM** – для знаходження суми чисел, **AVERAGE** – для знаходження середнього арифметичного, **MAX** та **MIN** – для визначення максимального та мінімального значень серед указаних чисел. Перша з них належить до категорії математичних функцій, інші – до категорії статистичних.

МАТЕМАТИЧНІ ФУНКЦІЇ

Розглянемо ще кілька математичних функцій (табл. 7.2).

Таблиця 7.2

Деякі математичні функції

Функція	Кількість аргументів	Результат	Приклад використання
ABS(число) (англ. <i>absolute</i> – абсолютний)	1	Модуль (абсолютна величина) числа	ABS(C10)
ROUND(число; кількість розрядів) (англ. <i>round</i> – округлювати)	2	Округлене число Якщо кількість розрядів > 0, то число округлюється до цього розряду після коми Якщо кількість розрядів < 0, то число округлюється до цього розряду перед комою Якщо кількість розрядів = 0, то число округлюється до цілого	ROUND(C1;3)
SQRT(число) (англ. <i>square root</i> – квадратний корінь)	1	Арифметичний квадратний корінь із числа	SQRT(B10)
POWER(число; степінь) (англ. <i>power</i> – степінь)	2	Число, піднесене до степеня з показником степінь	POWER(C5;5)
PI()	0	Наближене значення числа $\pi = 3,14159265358979$	PI()

ЛОГІЧНІ ФУНКЦІЇ

Крім формул, що містять числа, знаки дій, функції їх опрацювання, в Excel можуть використовуватися логічні вирази, які містять знаки порівняння: > (більше), < (менше), = (дорівнює), <> (не дорівнює), >= (більше або дорівнює), <= (менше або дорівнює). Наприклад, **A2+15 = B4-1; SUM(A2:C10)>100** та ін.

Такі вирази, як ви вже знаєте, можуть приймати значення **True** (істина) або **False** (хиба, хибність).

В Excel використовуються і логічні функції.

Функцію, результат якої дорівнює True або False, називають *логічною*.

До логічних функцій в Excel належать функції **IF** (англ. *if* – якщо), **AND** (англ. *and* – і), **OR** (англ. *or* – або), **NOT** (англ. *not* – не) та ін.

Їх використовують у формулах тоді, коли табличний процесор повинен виконувати різні операції залежно від істинності або хибності певного логічного виразу. З такими задачами ви вже зустрічалися, зокрема у програмуванні. Наприклад, потрібно нарахувати працівнику премію в розмірі 20 % від окладу, якщо він перевиконав план, а якщо ні, то не нараховувати. Або якщо термін продажу товару в магазині закінчився, то його уцінюють на 50 %.

Загальний вигляд логічної функції **IF** такий:

IF(логічний_вираз;значення_якщо_істина;значення_якщо_хибність).

Значення цієї функції визначається так:

- обчислюється значення **логічного виразу (True чи False)**;
- якщо **логічний вираз** має значення **True**, то значення функції дорівнює значенню виразу **значення_якщо_істина**;
- якщо ж **логічний вираз** має значення **False**, то значення функції дорівнює значенню виразу **значення_якщо_хибність**.

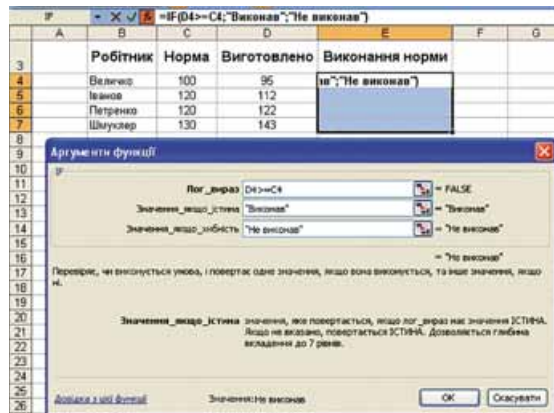
Наприклад, для функції **IF(A1+B1>100;C1*0,2;C1*0,1)** обчислення значення відбуватиметься таким чином. Спочатку Excel обчислить суму чисел з клітинок **A1** і **B1** та визначить значення логічного виразу **A1+B1>100**. Якщо це значення **True**, то результатом обчислень буде значення виразу **C1*0,2**. Якщо значення логічного виразу **A1+B1>100** дорівнюватиме **False**, то результат буде обчислюватися за виразом **C1*0,1**.

Ще один приклад задачі з використанням логічної функції **IF**: кожен з робітників бригади має щоденну норму й реальну кількість виготовлених деталей за день. З'ясувати для кожного робітника, чи виконав він свою норму. На малюнку 7.6 подано розв'язок цієї задачі.

Як бачите, правило обчислення логічної функції в табличному процесорі аналогічне до правила виконання повної команди розгалуження у програмуванні.

В Excel можна використовувати і скорочений запис логічної функції **IF**, який відповідає команді неповного розгалуження:

IF(логічний_вираз;значення_якщо_істина)



Мал. 7.6. Розв'язок задачі з використанням логічної функції **IF**



Розглянемо тепер логічні функції **AND**, **OR**, **NOT** (табл. 7.3).

Таблиця 7.3

Логічні функції **AND**, **OR**, **NOT**

Функція	Кількість аргументів	Результат	Приклад запису функції
AND (логіч1; [логіч2];...)	Від 1 до 255, усі, крім першого, необов'язкові	– True , якщо всі її аргументи мають значення True ; – False , якщо хоча б один аргумент має значення False	AND(A1>2;B1>10;B1<20;C1=5)
OR (логіч1; [логіч2];...)	Від 1 до 255, усі, крім першого, необов'язкові	– True , якщо хоча б один з аргументів має значення True ; – False , якщо всі аргументи мають значення False	OR(A1>2;B1>10;B1<20;C1=5)
NOT (логічне)	1	– True , якщо аргумент має значення False ; – False , якщо аргумент має значення True	NOT(F1>25)

Найчастіше ці функції використовуються як аргументи функції **IF**, а самостійно – дуже рідко.

У таблиці 7.4 представлено значення функцій **AND**, **OR**, **NOT** залежно від вмісту клітинок **A1** і **B1**.

Таблиця 7.4

Таблиця значень функцій **AND**, **OR**, **NOT**

A1	B1	AND(A1;B1)	OR(A1;B1)	NOT(A1)
True	True	True	True	False
True	False	False	True	False
False	True	False	True	True
False	False	False	False	True

Логічну функцію **AND** ще називають логічною функцією **I** (кон'юнкція), логічну функцію **OR** – логічною функцією **АБО** (диз'юнкція), а логічну функцію **NOT** – логічною функцією **НЕ** (заперечення).

Логічні функції **AND**, **OR**, **NOT** використовують для запису більш складних логічних виразів, ніж просто рівність або нерівність. Наприклад, у функції **IF** не можна записувати подвійну нерівність. Тому замість подвійної нерівності $10 < A3 < 20$ потрібно використовувати логічний вираз **AND(A3>10;A3<20)**. Або, наприклад, функцію задано так:

$$y = \begin{cases} 2x - 5, & \text{якщо } x < -2 \text{ або } x > 10 \\ \text{не існує,} & \text{при інших значеннях } x \end{cases}$$

і значення x знаходиться у клітинці **A5**, то формула для обчислення значення функції y записується так: **=IF(OR(A5<-2;A5>10);2*A5-5;"функція не визначена")**



Для тих, хто хоче знати більше

В **Excel 2007** у записі функцій можна використовувати вкладені функції (до 64-х рівнів вкладеності). Так, другий і третій аргументи функції **IF** можуть містити вкладені арифметичні операції та функції, у тому числі й функцію **IF**.

Наприклад, **IF(A1<0; -B1/A1; IF(A1>0;B1/A1; "Ділення неможливе!"))**.



Найважливіше в цьому пункті

Табличний процесор **Excel 2007** має бібліотеку вбудованих функцій, до якої входить більше ніж 300 різноманітних функцій. Усі вони для зручності розподілені на категорії: *математичні, статистичні, текстові, логічні, фінансові* та ін.

Приклади математичних функцій: модуль числа **ABS**, арифметичний квадратний корінь із числа **SQRT**, піднесення числа до степеня **POWER**, сума чисел **SUM**.

До логічних функцій в **Excel** належать функції **IF**, **AND**, **OR**, **NOT** та ін. Вони використовуються у формулах тоді, коли табличний процесор повинен виконувати різні операції залежно від істинності або хибності певного логічного виразу.

Загальний вигляд логічної функції **IF**:

IF(логічний_вираз;значення_якщо_істина;значення_якщо_хибність).

Значення цієї функції визначається так: обчислюється значення **логічного виразу**, якщо **логічний вираз** має значення **True**, то значення функції дорівнює значенню виразу **значення_якщо_істина**; якщо ж **логічний вираз** має значення **False**, то значення функції дорівнює значенню виразу **значення_якщо_хибність**.










Дайте відповіді на запитання

- 1°. Який загальний вигляд, кількість аргументів, результат виконання математичних функцій **ABS**, **ROUND**, **PI**, **POWER**?
- 2°. Який загальний вигляд, кількість аргументів, результат виконання математичних функцій **SQRT**, **SUM**?
- 3°. Який загальний вигляд, кількість аргументів, результат виконання статистичних функцій **AVERAGE**, **MAX**, **MIN**?
- 4°. Що таке логічні вирази? Які знаки вони можуть містити?
- 5°. Яких значень може набувати логічний вираз?
- 6°. Який загальний вигляд, кількість аргументів, результат використання логічної функції **IF**?
- 7°. Як визначається значення логічної функції **IF**?
- 8°. Який загальний вигляд, кількість аргументів, результат виконання логічних функцій **AND**, **OR**, **NOT**?



- 9°. Як визначається значення логічних функцій AND, OR, NOT?
- 10*. Що таке вкладеність функцій? Який рівень вкладеності максимальний?

**Виконайте завдання**

- 1°. Відкрийте табличний процесор Excel 2007. Створіть таблицю для обчислення довжин діагоналей 5 прямокутників за даними довжинами їх сторін. Установіть формат результатів обчислення з двома десятковими розрядами. Збережіть книгу у вашій папці у файлі з іменем завдання 7.2.1.xlsx.
- 2°. Відкрийте вказаний учителем файл (наприклад, Розділ 7\Пункт 7.2\зразок 7.2.2.xlsx). На аркуші Аркуш1 подано таблицю очок, які набрав кожен з учасників змагань із п'ятиборства в кожному виді змагань. Додайте до таблиці формули для обчислення суми очок і максимальної кількості очок, які набрав кожен з учасників, а також середньої кількості очок, які набрали учасники в кожному виді змагань. Середню кількість очок округліть до цілих. Збережіть книгу у вашій папці у файлі з іменем завдання 7.2.2.xlsx.
- 3°. У клітинці A5 міститься число 10, у клітинці B5 – число -7, а у клітинці C5 – число 0. Визначте результати таких функцій:
- а) $AND(A5>5;A5<20)$;  д) $NOT(B5<20)$;
- б) $OR(C5<10;C5>=20)$;  е) $AND(OR(B5>5;B5<=-5);NOT(B5<>10))$;
- в) $OR(C5<10;A5>20)$;  є) $OR(AND(A5>2;A5<=10);B5<0)$;
- г) $AND(A5>=-2;B5>0)$;  ж) $NOT(AND(A5<100;C5=0))$.
-  4°. Відкрийте вказаний учителем файл (наприклад, Розділ 7\Пункт 7.2\зразок 7.2.4.xlsx). На аркуші Аркуш1 подано кількість деталей, які зробив кожен з робітників протягом дня. У клітинці A1 міститься денна норма. Додайте до таблиці формули для визначення, виконав кожен з робітників норму чи ні. Результат указати у вигляді True або False. Збережіть книгу у вашій папці у файлі з іменем завдання 7.2.4.xlsx.
-  5*. Відкрийте вказаний учителем файл (наприклад, Розділ 7\Пункт 7.2\зразок 7.2.5.xlsx). На аркуші Аркуш1 подано результати виступів учасників олімпіади в кожному з двох турів. За умовами проведення олімпіади **Дипломом I ступеня** нагороджуються учасники, що набрали не менше ніж 85 % очок, **Дипломом II ступеня** – учасники, що набрали не менше ніж 75 %, але менше ніж 85 % очок, **Дипломом III ступеня** – учасники, що набрали не менше ніж 50 %, але менше ніж 75 % очок, усі інші одержують **Диплом учасника**. Додайте до таблиці формули для визначення, який **Диплом** буде вручено кожному з учасників. Збережіть книгу у вашій папці у файлі з іменем завдання 7.2.5.xlsx.
-  6*. Відкрийте вказаний учителем файл (наприклад, Розділ 7\Пункт 7.2\зразок 7.2.6.xlsx). На аркуші Аркуш1 подано таблицю значень

змінної x . Додайте до таблиці формули для визначення відповідних

$$\text{значень функції } y = \begin{cases} 2x - 5, & \text{якщо } x < -2 \text{ або } x > 10, \\ 3x + 1, & \text{якщо } 2 \leq x \leq 3, \\ x^5 - 4x^3 & \text{при інших значеннях } x. \end{cases}$$

- Збережіть книгу у вашій папці у файлі з іменем **завдання 7.2.6.xlsx**.
- 7*. Відкрийте табличний процесор **Excel 2007**. Складіть таблицю, за якою визначте, які умови вкладу в банк вигідніші клієнту: 24 % річних з нарахуванням відсотків щомісячно чи 26 % річних з нарахуванням відсотків кожного півріччя. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.2.7.xlsx**.



7.3. СТВОРЕННЯ ТА НАЛАГОДЖЕННЯ ДІАГРАМ РІЗНОГО ТИПУ



1. Що таке діаграми? Для чого їх використовують?
2. Які об'єкти діаграм в **Excel 2007** ви знаєте? Які властивості вони мають?
3. Як створити діаграму? Які операції редагування та форматування можна виконувати над діаграмою?

ДІАГРАМИ В EXCEL 2007

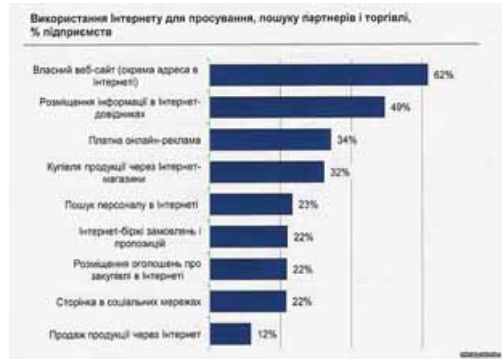
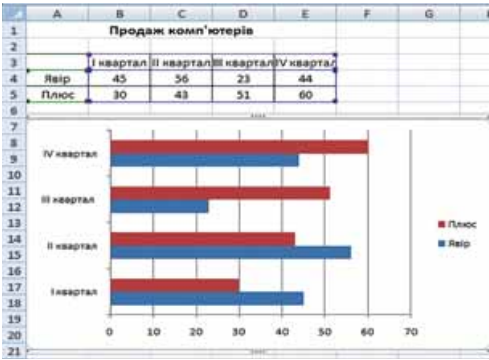
З курсу інформатики 7 класу ви вже знаєте, що для більш наочного представлення та полегшення розуміння числових даних використовують діаграми, на яких співвідношення між числами відображається з використанням геометричних фігур. Розміри цих фігур на діаграмах пропорційні числовим даним, які вони відображають.

Найчастіше використовують секторні та стовпчасті діаграми, з якими ви вже ознайомилися в 7 класі. **Секторні** діаграми призначені для відображення частки, яку складає кожне окреме число в їх загальній сумі. **Стовпчасту** діаграму доцільно створювати тоді, коли потрібно порівняти значення одного або кількох наборів числових даних.

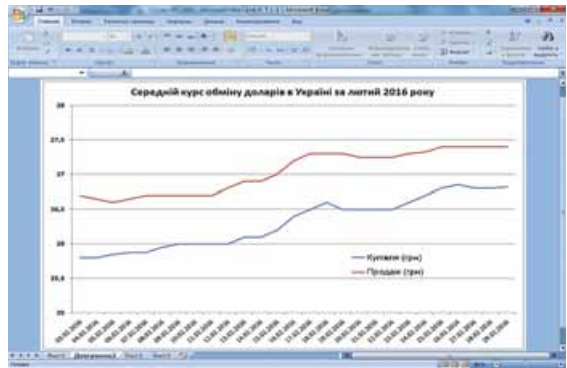
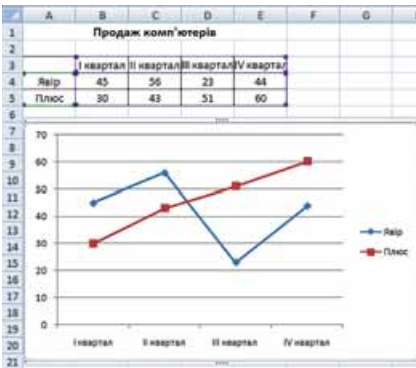
Крім цих типів діаграм, в **Excel 2007** можна побудувати діаграми й інших типів, наприклад: *лінійчату, гистограму, графік, точкову, бульбашкову, пелюсткову* тощо. Кожен із цих типів має кілька різновидів. Їх перелік можна переглянути, відкривши списки відповідних кнопок на вкладці **Вставлення** в групі **Діаграми**.

Розглянемо детальніше кілька з них.

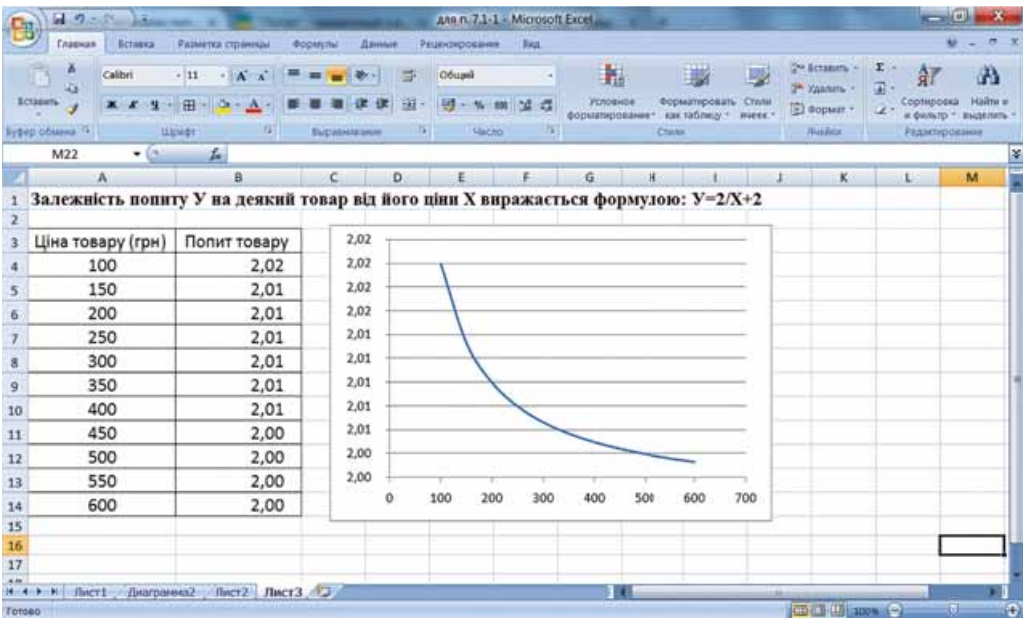
Лінійчата діаграма (в україномовній версії **Excel 2007** тип діаграми **Гістограма**) аналогічна до *стовпчастої* діаграми, тільки розташована вона горизонтально (мал. 7.7). Такі діаграми зручно використовувати, якщо горизонтальні елементи даних наочніші, ніж вертикальні, наприклад, якщо вздовж горизонтальної осі відкладаються зростаючі проміжки часу, температури тощо. Їх зручно використовувати ще й у тих випадках, коли підписи вздовж вертикальної осі довгі й легше сприймаються в горизонтальному положенні.



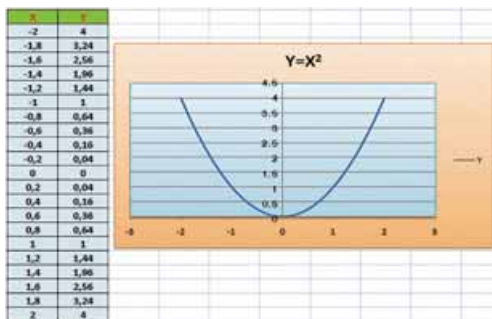
Мал. 7.7. Лінійчаті діаграми



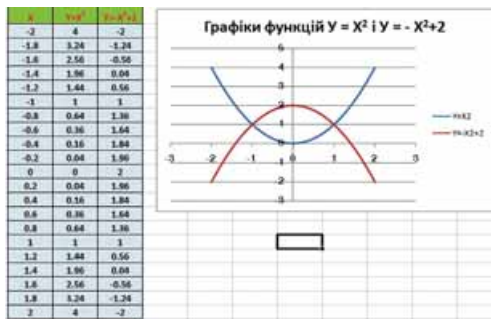
Мал. 7.8. Графік з маркерами даних і без них



Мал. 7.9. Точкова діаграма



Мал. 7.10. Графік функції



Мал. 7.11. Графіки двох функцій

Графік (в україномовній версії Excel 2007 тип діаграми **Лінійчата**). Графіки доцільно використовувати, якщо потрібно відобразити деякий процес у часі, динаміку зміни окремих даних, які не залежать одне від одного (мал. 7.8). Графік можна будувати з маркерами та без них (маркери відображають на діаграмі точку даних).

Точкові діаграми використовують для демонстрації взаємозв'язку між значеннями двох змінних, коли значення однієї величини залежить від значення іншої (мал. 7.9).

Точкові діаграми найчастіше використовують для побудови графіків функцій (мал. 7.10). На одній діаграмі можна також побудувати графіки двох функцій і використати їх для наближеного розв'язування рівнянь (мал. 7.11).

СТВОРЕННЯ ДІАГРАМ

Як ви знаєте із 7 класу, діаграми будуються на основі даних, поданих в електронній таблиці, і є динамічними – зі зміненням даних у таблиці діаграми автоматично змінюються. Створити діаграму можна, використовуючи елементи керування вкладки **Вставка** групи **Діаграми**. Для цього потрібно:

1. Виділити діапазон клітинок з даними, на основі яких будуватиметься діаграма.
2. Відкрити на вкладці **Вставлення** у групі **Діаграми** список кнопки відповідного типу діаграм.
3. Вибрати потрібний вид діаграми.

Зазначимо, що виділений діапазон клітинок може бути як зв'язним, так і незв'язним. Бажано, щоб до нього увійшли підписи рядків і стовпців.

У результаті виконання вказаного алгоритму на аркуші з електронною таблицею буде побудовано діаграму, значення властивостей об'єктів якої встановлюються за замовчуванням.

На діаграмі можна виділити такі об'єкти: область діаграми, область побудови діаграми, заголовок діаграми, елемент даних, ряд даних, легенда, головна горизонтальна вісь, головна вертикальна вісь, назви осей, підписи даних.



РЕДАГУВАННЯ ТА ФОРМАТУВАННЯ ДІАГРАМ

Редагування діаграми полягає у:

- змінненні діапазону даних, у тому числі обміні місцями відображення стовпців і рядків, додаванні нових рядків і стовпців даних до вже існуючих або видаленні рядків (стовпців) даних;
- вставленні або змінненні назв діаграми й осей, легенди, підписів даних та ін.;
- змінненні відображення осей і ліній сітки;
- виділенні та редагуванні окремих об'єктів діаграми;
- змінненні типу діаграми;
- змінненні місця розташування діаграми (переміщення на аркуші, розміщення на окремому аркуші)

та ін.

Форматування діаграми полягає у форматуванні діаграми в цілому або її окремих об'єктів і передбачає:

- зміннення розмірів діаграми;
- зміннення стилю оформлення діаграми;
- зміннення шрифту, розміру, кольору, заливки тощо окремих об'єктів діаграми;
- налаштування відображення осей та сітки на діаграмі тощо.

Для виконання операцій редагування та форматування використовують елементи керування вкладок **Конструктор**, **Макет**, **Формат** тимчасового розділу **Знаряддя для діаграм**.

Нагадаємо порядок виконання деяких з них:

- щоб змінити тип діаграми, потрібно вибрати її і виконати **Конструктор** ⇒ **Тип** ⇒ **Змінити тип діаграми**;
- щоб поміняти місцями відображення стовпців і рядків, потрібно вибрати діаграму і виконати **Конструктор** ⇒ **Дані** ⇒ **Перехід Рядок/Стовпець**;
- щоб змінити розміри діаграми, потрібно перетягнути один з маркерів зміни розмірів, що розташовані на її межі;
- щоб змінити місце розташування діаграми на аркуші, її слід перетягнути у потрібне місце.

Як ви вже знаєте із 7 класу, для форматування об'єкта діаграми його спочатку потрібно вибрати, а потім можна:

- вибрати один зі стилів оформлення виділеного об'єкта зі списку групи **Стилі фігур** вкладки **Макет**;
- вибрати один зі стилів оформлення текстового об'єкта зі списку групи **Стилі WordArt** вкладки **Макет**;
- вибрати один з видів заливки, оформлення меж, ефектів зі списків кнопок **Заливка фігури**, **Контур фігури**, **Ефекти для фігур** групи **Стилі фігур** вкладки **Макет**;
- виконати **Макет** ⇒ **Поточний виділений фрагмент** ⇒ **Форматувати виділений фрагмент** і на вкладках вікна встановити потрібні значення його властивостей;
- використати команди контекстного меню виділеного об'єкта.

Найважливіше в цьому пункті

Лінійчаті діаграми аналогічні до стовпчастих, і використовуються, якщо горизонтальні елементи даних наочніші, ніж вертикальні.

Графіки доцільно будувати, якщо потрібно відобразити неперервні дані, а також динаміку зміни незалежних між собою даних.

Точкові діаграми використовують для графічного відображення залежних між собою даних, найчастіше для побудови графіків функцій.

Створити діаграму можна, використовуючи елементи керування вкладки **Вставка** групи **Діаграми**.

Для виконання операцій редагування та форматування використовують елементи керування вкладок **Конструктор**, **Макет**, **Формат** тимчасового розділу **Знаряддя для діаграм**.

Дайте відповіді на запитання

- 1°. Які ви знаєте види діаграм в **Excel 2007**?
- 2°. Які об'єкти є на діаграмі в **Excel 2007**? Опишіть їх.
- 3°. Які властивості мають об'єкти діаграм в **Excel 2007**?
- 4°. Як створити лінійчатую діаграму? Яке її призначення?
- 5°. Як створити графік? Яке його призначення?
- 6°. Як створити точкову діаграму? Яке її призначення?
- 7°. Чим відрізняється використання діаграми **Графік** від використання точкової діаграми?
- 8*. Як можна використати точкові діаграми для наближеного розв'язування рівнянь?
- 9°. Які операції редагування діаграм ви знаєте? Як їх виконувати?
- 10°. Які операції форматування діаграм ви знаєте? Як їх виконувати?
- 11°. Як визначити тип діаграми, який потрібно застосувати?

Виконайте завдання

- 1°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.3\зразок 7.3.1.xlsx**). На аркуші **Аркуш1** подано таблицю опадів за останній місяць. Побудуйте за даними цієї таблиці секторну діаграму, лінійчатую діаграму та графік опадів. Яка з діаграм наочніше демонструє дані? Оформіть визначену діаграму (назва, підписи даних, формат тексту, заливка тощо) та розмістіть її на окремому аркуші. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.3.1.xlsx**.
- 2°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.3\зразок 7.3.2.xlsx**). На аркуші **Аркуш1** подано таблицю видобування газу в Україні різними компаніями за три останніх роки. За даними цієї таблиці побудуйте лінійчатую діаграму видобутку газу за останній рік різними компаніями та графік сумарного видобутку газу за останні три роки. Розмістіть діаграми на окремих аркушах. Оформіть діаграми (назва, підписи осей та даних, формат тексту, заливка тощо). Збережіть книгу у вашій папці у файлі з іменем **завдання 7.3.2.xlsx**.



3*. Знайдіть в Інтернеті дані щодо кількості населення у вашому населеному пункті (або в Україні) за останні 5 років. Створіть електронну таблицю за цими даними. Побудуйте за даними таблиці три діаграми: секторну, лінійчату й графік. Розмістіть їх на окремих аркушах та оформіть на власний розсуд. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.3.3.xlsx**.

4*. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.3\зразок 7.3.4.xlsx**). На аркуші **Аркуш1** подано таблицю розчинності деяких речовин залежно від температури. Побудуйте за даними цієї таблиці точкову діаграму з гладкими лініями та маркерами. Додайте заголовок, підписи осей та даних, установіть один зі стилів оформлення. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.3.4.xlsx**.

5*. Відкрийте табличний процесор **Excel 2007**. На аркуші **Аркуш1** побудуйте графік функції $y = x^3 - 3x$ на відрізку $[-3; 3]$ з кроком $0,2$. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.3.5.xlsx**.



6*. Відкрийте табличний процесор **Excel 2007**. На аркуші **Аркуш1** побудуйте графік функції $y = \frac{1}{x^2 + 1}$ на відрізку $[-3; 3]$ з кроком $0,2$.

Збережіть книгу у вашій папці у файлі з іменем **вправа 7.3.6.xlsx**.

7*. Відкрийте табличний процесор **Excel 2007**. На аркуші **Аркуш1** побудуйте графіки функцій $y = x^4 - 2x^2$ та $y = 1/4x - 0,5$ на відрізку $[-3; 3]$ з кроком $0,2$ і визначте наближені значення коренів рівняння $x^4 - 2x^2 = 1/4x - 0,5$ на цьому відрізку. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.3.7.xlsx**.



8*. Відкрийте табличний процесор **Excel 2007**. З'ясуйте, використовуючи **Довідку**, особливості призначення та побудови **Бульбашкової** та **Пелюсткової** діаграм. Знайдіть в Інтернеті щоденник погоди з даними по вашому обласному центру. Створіть за даними щоденника погоди електронну таблицю хмарності за попередній місяць: кількість ясних, малохмарних, хмарних та похмурих днів. Побудуйте за даними цієї таблиці бульбашкову та пелюсткову діаграми. Розмістіть діаграми поруч під таблицею. Оформіть їх на власний розсуд. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.3.8.xlsx**.

ПРАКТИЧНА РОБОТА 15

«Використання математичних, логічних та статистичних функцій табличного процесора. Побудова діаграм»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте вказаний учителем файл (наприклад, **Розділ7\Практична15\зразок пр15.xlsx**).
2. На аркуші **Аркуш1** подано таблицю показників лічильника електроенергії за кожен місяць минулого року. У клітинці **A1** знаходиться вартість $1 \text{ кВт} \cdot \text{год}$. Додайте до таблиці формули для обчислення що-



місячної плати за електроенергію, сумарної плати за рік, середніх щомісячних витрат електроенергії.

3. Побудуйте на окремому аркуші графік щомісячної плати за електроенергію. Додайте назву діаграми, підписи осей та значень. Відформатуйте діаграму на власний розсуд.
4. На аркуші **Аркуш2** подано прізвища учнів класу та їх зріст. Для кожного учня класу визначте, чи більший його зріст за середній зріст по класу.
5. Побудуйте на окремому аркуші лінійчату діаграму зросту учнів класу. Додайте назву діаграми, підписи осей та значень. Відформатуйте діаграму на власний розсуд.
6. На аркуші **Аркуш3** обчисліть таблицю значень функції $y = x^4 - 4x$ на відрізку $[-3; 3]$.
7. Побудуйте на окремому аркуші графік цієї функції на цьому відрізку, використовуючи точкову діаграму. Додайте назву діаграми, підписи осей та значень. Відформатуйте діаграму на власний розсуд.
8. Збережіть книгу в своїй папці у файлі з іменем **практична 15.xlsx**.



7.4. УПОРЯДКОВУВАННЯ ДАНИХ У ТАБЛИЦЯХ. АВТОМАТИЧНІ ТА РОЗШИРЕНІ ФІЛЬТРИ



1. Який порядок розташування символів у таблиці Unicode?
2. Які значення логічних функцій AND, OR залежно від значень аргументів?
3. Яке призначення символів * і ? у масках імен файлів?

УПОРЯДКОВУВАННЯ ДАНИХ У ТАБЛИЦІ

Дані в клітинках електронної таблиці можна впорядковувати (сортувати), тобто змінювати порядок їх розташування в рядках або стовпцях за деякими ознаками.

У табличному процесорі можна виконувати два види сортування: за зростанням або за спаданням. Порядок сортування залежить від типу даних. Упорядкування даних здійснюється посимвольно, *у порядку розміщення символів у кодовій таблиці Unicode*.

В Excel 2007 сортування за зростанням відбувається за такими правилами:

- якщо в таблиці знаходяться дані різних типів, то після сортування вони будуть розміщуватися у такому порядку: числові, дата/час, текстові, логічні;
- числа упорядковуються від найменшого значення до найбільшого;
- дати упорядковуються від найстаршої до наймолодшої;
- спочатку розташовуються тексти, написані латиницею, потім – кирилицею;

Уведені дані	Відсортовані за зростанням	Відсортовані за спаданням
35	-10	TRUE
школа	0,5	FALSE
-10	35	школа
	Lyceum	Сіренко
TRUE	Сіренко	Lyceum
	школа	35
0,5	FALSE	0,5
FALSE	TRUE	-10
Lyceum		
Сіренко		

Мал. 7.12. Приклади сортування даних



- спочатку розміщуються тексти, написані маленькими літерами, потім – великими;
- тексти упорядковуються так: спочатку порівнюються перші символи, у разі їх рівності – другі символи і т. д.;
- логічне значення **FALSE** розміщується перед значенням **TRUE**;
- порожні клітинки розташовуються останніми (хоча символ **Пропуск** у таблиці **Unicode** стоїть першим і має найменший код).

Під час сортування за спаданням порядок розташування буде зворотний. Приклади розміщення значень різних типів після сортування за зростанням і за спаданням показано на малюнку 7.12.

Для того щоб швидко виконати сортування рядків деякого діапазону за значеннями першого стовпця цього діапазону, слід:

1. Виділити потрібний діапазон клітинок.
2. Виконати *Основне* ⇒ *Редагування* ⇒ *Сортування й фільтр* ⇒ *Сортування від найменшого до найбільшого* або *Сортування від найбільшого до найменшого* (мал. 7.13).

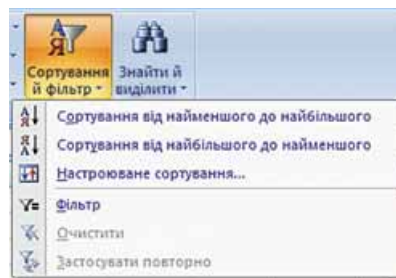
У результаті цих дій рядки виділеного діапазону змінюють своє розташування так, що значення в першому стовпці цього діапазону будуть впорядковані у вибраному порядку, а дані в інших стовпцях будуть переміщені разом з відповідним рядком.

Звертаємо вашу увагу:

1. Назви команд **Сортування від найменшого до найбільшого** і **Сортування від найбільшого до найменшого** автоматично змінюються залежно від типу даних, які підлягають сортуванню:
 - для текстових даних – **Сортування від А до Я** і **Сортування від Я до А**;
 - для дат і часу – **Сортування від найстаршого до наймолодшого** і **Сортування від наймолодшого до найстаршого**.
2. Якщо виконувати сортування, виділивши дані тільки в одному стовпці, відкривається вікно **Попередження сортування** (мал. 7.14), у якому пропонується розширити виділений діапазон клітинок або сортувати у межах виділення.

Сортування у виділеному діапазоні можна здійснювати й за значеннями кількох стовпців. Це відбувається так:

- спочатку рядки сортуються за значеннями в першому з указаних стовпців;
- потім сортування у кожному наступному з вибраних стовпців відбувається лише для тих рядків, у яких значення в усіх попередніх указаних для сортування стовпцях збігаються.

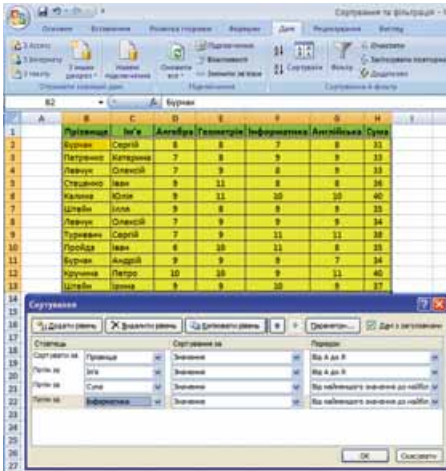


Мал. 7.13. Список команд кнопки **Сортування й фільтр**

2	11	0	4	4
4	4	8	2	3
1	5	9	3	2
7	3	7	1	1



Мал. 7.14. Вікно **Попередження сортування**



Мал. 7.15. Кнопка **Сортувати**, діапазон клітинок для сортування і вікно **Сортування**

Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Бурчак	Андрій	9	9	9	7	34
Бурчак	Сергій	8	8	7	8	31
Калінін	Юлія	9	11	10	10	40
Кручина	Ліля	9	10	9	11	40
Кручина	Петро	10	10	9	11	40
Левчук	Олександр	7	9	8	9	33
Левчук	Олександр	7	9	9	9	34
Петренко	Катерина	7	9	9	9	33
Поліщак	Іван	6	10	11	8	35
Степанко	Іван	9	11	8	8	36
Турчак	Сергій	7	9	11	11	38
Штейн	Ілля	9	8	9	9	35
Штейн	Григор	9	9	10	9	37

Мал. 7.16. Діапазон клітинок, відсортований за умовами, поданими на малюнку 7.15

ташовані в алфавітному порядку (від А до Я).

Сортування за значеннями у третьому з вибраних стовпців (**Сума**) відбувалося тільки в межах п'ятого і шостого рядків, у яких значення у перших з двох вибраних стовпців збіглися (Левчук Олександр). Сортування за значеннями у четвертому з вибраних стовпців не відбувалося, бо не виявилася хоча б двох рядків, у яких значення у перших з трьох вибраних для сортування стовпців (**Прізвище**, **Ім'я**, **Сума**) збіглися.

Щоб відсортувати рядки у виділеному діапазоні клітинок за значення м в кількох стовпцях, слід:

1. Виділити потрібний зв'язний діапазон клітинок (бажано, щоб до нього ввійшли підписи рядків, але до нього не повинні входити об'єднані клітинки).
2. Виконати **Дані** ⇒ **Сортування й фільтр** ⇒ **Сортувати**.
3. Установити або змінити значення параметрів сортування: у списку поля **Сортувати** за вибрати ім'я першого стовпця, за значеннями в якому потрібно відсортувати рядки, у списку поля **Сортування за** – **Значення**, а в списку поля **Порядок** – порядок сортування (мал. 7.15).

Кожен з указаних для сортування стовпців називають *рівнем сортування*.

Пояснимо таке сортування на конкретному прикладі. На малюнку 7.15 подано приклад несортованої таблиці річних оцінок учнів школи, а на малюнку 7.16 цю само таблицю відсортовано за значеннями в чотирьох стовпцях: **Прізвище** (від А до Я), **Ім'я** (від А до Я), **Сума** (від найменшого значення до найбільшого) та **Інформатика** (від найменшого значення до найбільшого).

Процес сортування відбувався так. Спочатку рядки переставлялися так, щоб дані в стовпці **Прізвище** були розташовані в алфавітному порядку (**від А до Я**). І якщо б у всіх учнів прізвища були різні, то більше ніякого сортування цієї таблиці не відбувалося б. Але таблиця містить три групи учнів з однаковими прізвищами: Бурчак, Левчук, Штейн.

Подальше сортування відбувається вже за значеннями у другому з вибраних стовпців (**Ім'я**) і тільки у межах кожної із цих трьох груп рядків, по кожній окремо. У результаті цього кроку сортування переставлялися (а може й ні) перший і другий рядки, п'ятий і шостий рядки, одинадцятий і дванадцятий рядки так, щоб імена були роз-



4. Визначити умови сортування за значеннями в іншому стовпці, вибравши кнопку **Додати рівень** або **Копіювати рівень**, і встановити в полях рядка **Потім за значення** параметрів сортування за значеннями у другому вибраному стовпці.
5. За потреби повторити команду 4 кілька разів.
6. Якщо у виділений діапазон увійшов рядок заголовка, але позначку прапорця **Мої дані мають заголовки** не встановлено, встановити її (тоді рядок заголовка не буде враховуватися під час сортування і залишиться на своєму місці).
7. Вибрати кнопку **ОК**.



Для тих, хто хоче знати більше

Сортувати можна не тільки рядки, а й стовпці. За замовчуванням в **Excel** встановлюється режим сортування рядків за значеннями в стовпцях. Для змінення цього режиму сортування потрібно у вікні **Сортування** вибрати кнопку **Параметри**, а потім – відповідний перемикач у групі **Орієнтація**. Там само можна визначити, чи враховувати реєстр під час сортування, тобто, чи розрізняти великі й малі літери.


АВТОМАТИЧНІ ФІЛЬТРИ

Фільтрування – це вибір значень у клітинках електронної таблиці, що відповідають певним умовам.

Після виконання фільтрування в таблиці відображаються лише ті рядки, дані в яких відповідають умовам **фільтрування**. Усі інші – тимчасово приховуються.

Щоб установити режим фільтрування даних, потрібно:

1. Установити табличний курсор у довільну клітинку таблиці або виділити потрібний діапазон клітинок.
2. Виконати **Основне** ⇒ **Редагування** ⇒ **Сортування й фільтр** ⇒ **Фільтр** (або **Дані** ⇒ **Сортування й фільтр** ⇒ **Фільтр**).

Після виконання цих дій біля правої межі кожної клітинки першого рядка таблиці або виділеного діапазону клітинок з'являться кнопки відкриття списку  команд фільтрування (мал. 7.17). У цих списках знаходяться команди сортування, фільтрування за кольором, а також числові та текстові фільтри.

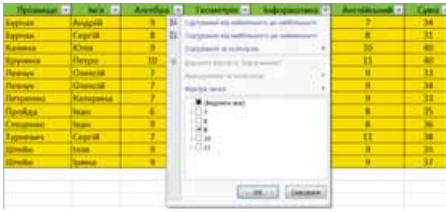
Розглянемо деякі приклади виконання фільтрування.

Найчастіше виконують фільтрування за значеннями в деякому стовпці даних. Наприклад, за поданою таблицею оцінок відобразити список учнів, які мають з **інформатики** оцінку **9**. Для цього потрібно:

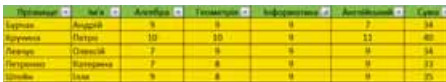
1. Відкрити список команд фільтрування у стовпці **Інформатика**.
2. Зняти позначку прапорця **Виділити все**.

Прізвище	Ін'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Бурмак	Андрій	9	9	9	7	34
Бурмак	Сергій	8	8	7	8	31
Малена	Юлія	9	11	10	10	40
Мухомов	Петро	10	10	9	11	40
Левчук	Олександр	7	9	8	9	33
Левчук	Олександр	7	9	9	9	34
Петрушко	Анатолій	7	8	9	9	33
Гришкін	Іван	8	10	11	8	37
Степаненко	Іван	9	12	8	8	37
Попович	Сергій	7	9	11	11	38
Штефан	Ірина	9	8	9	9	35
Штефан	Ірина	9	9	10	9	37

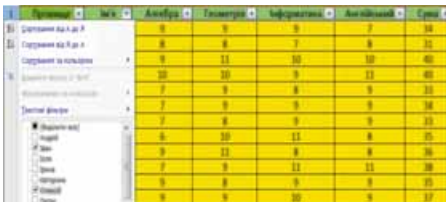
Мал. 7.17. Діапазон клітинок після виконання команди **Фільтр**



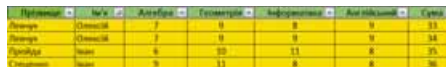
Мал. 7.18. Застосування фільтра за значенням у стовпці **Інформатика**



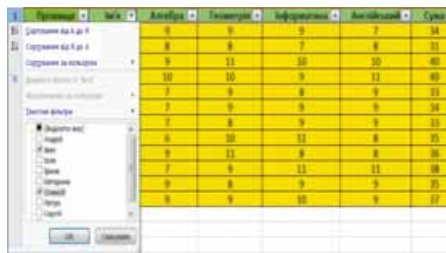
Мал. 7.19. Результат застосування фільтра на малюнку 7.18



Мал. 7.20. Фільтрування за двома значеннями стовпця **Ім'я**



Мал. 7.21. Результат застосування фільтра на малюнку 7.20



Мал. 7.22. Команди встановлення умов фільтрування числових даних

3. Установити позначку прапорця біля значення **9**.

4. Вибрати кнопку **ОК**.

Після виконання цих дій у таблиці будуть відображені лише ті рядки, у яких значення в потрібному стовпці дорівнюють вибраним. Виконання та результат описаного прикладу представлено на малюнках 7.18 та 7.19.

Зверніть увагу: вигляд кнопки фільтра у стовпці **Інформатика** на малюнку 7.19 змінився на такий . Наявність такої кнопки в клітинці означає наявність встановленого фільтра для даних у цьому стовпці таблиці.

Фільтрування можна здійснювати за кількома значеннями. Наприклад, потрібно відобразити дані про учнів з іменами, яких звать **Іван** або **Олексій**. Для цього потрібно:

1. Відкрити список команд фільтрування у стовпці **Ім'я**.
2. Зняти позначку прапорця **Виділити все**.
3. Установити позначку прапорця біля значень **Іван** та **Олексій** (мал. 7.20).
4. Вибрати кнопку **ОК**.

У результаті отримаємо таблицю, яку представлено на малюнку 7.21.

Якщо в стовпці знаходяться тільки числові дані, то за даними цього стовпця можна виконувати фільтрування з використанням **Фільтрів чисел**, застосовуючи команди **Дорівнює**, **Не дорівнює**, **Більше**, **Менше**, **Між** тощо (мал. 7.22).

Наприклад, потрібно відобразити дані про тих учнів, які мають сумарний бал за предметами ≥ 35 і ≤ 40 . Для цього слід виконати:

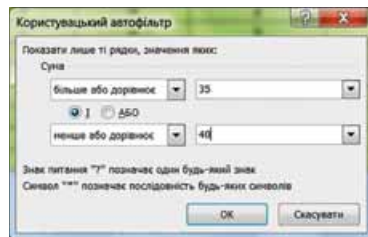
1. Відкрити список команд фільтрування у стовпці **Сума**.
2. Вибрати **Фільтри чисел** \Rightarrow **Між**.
3. Увести у відповідні поля вікна потрібні дані – **35** та **40** (мал. 7.23).
4. Вибрати кнопку **ОК**.



Застосувавши фільтр згідно з малюнком 7.23, отримаємо таблицю, яку подано на малюнку 7.24.

Команда **Перші 10** відкриває вікно **Фільтр перших 10 елементів** (мал. 7.25), у якому можна встановити умову фільтрування, за якою в таблиці відображатимуться тільки ті рядки таблиці, які містять у даному стовпці найбільших (найменших) 10 (або іншу встановлену кількість) значень.

Команда **Більше середнього (Менше середнього)** виконує фільтрування, за яким відображаються тільки ті рядки таблиці, які містять у даному стовпці значення, більші (менші) ніж середнє арифметичне значення в даному стовпці.



Мал. 7.23. Встановлення числового фільтру

Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Іванова	Юлія	9	11	10	10	40
Ніжченко	Петро	10	10	9	11	40
Бродько	Іван	6	10	11	8	35
Степанов	Іван	9	11	8	8	36
Турчин	Сергій	7	9	11	11	38
Швейн	Іван	9	8	9	9	35
Швейн	Ірина	9	9	10	9	37

Мал. 7.24. Результат застосування фільтра на малюнку 7.23

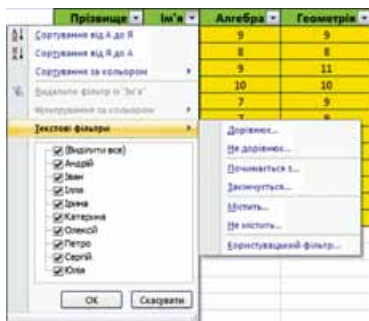


Мал. 7.25. Вікно **Фільтр перших 10 елементів**

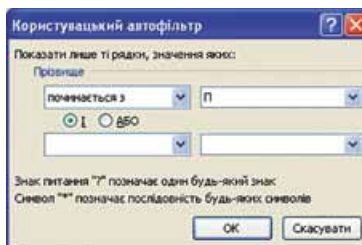
Для текстових даних застосовуються дещо інші команди фільтрування (мал. 7.26): **Дорівнює**, **Не дорівнює**, **Починається з**, **Закінчується**, **Містить** тощо.

Наприклад, потрібно відобразити дані про тих учнів, прізвище яких починається з літери **П**. Для цього слід виконати:

1. Відкрити список команд фільтрування у стовпці **Прізвище**.
2. Вибрати **Текстові фільтри** ⇒ **Починається з...**
3. Увести у відповідне поле літеру **П** (мал. 7.27).
4. Вибрати кнопку **ОК**.



Мал. 7.26. Команди встановлення умов фільтрування текстових даних



Мал. 7.27. Умова для фільтрації текстів

Фільтрування можна здійснювати послідовно для кількох стовпців таблиці. Тоді кожний наступний фільтр буде застосований уже тільки для тих рядків таблиці, які відображаються після застосування всіх поперед-

ніх фільтрів. Тобто загальна умова фільтрування – це умови фільтрування для кожного стовпця, об'єднані логічною операцією **I**. Наприклад, якщо провести фільтрування за даними стовпця **Сума** з умовою **більше 35**, а потім за даними стовпця **Інформатика** з умовою **більше 8**, то загальне фільтрування буде виконано за умовою (**сума балів більше 35**) **I** (**бал з інформатики більше 8**). Результат такого фільтрування наведено на малюнку 7.28.

Відмінити фільтрацію даних можна кількома способами:

- виконати **Основне** \Rightarrow **Редагування** \Rightarrow **Сортування й фільтр** \Rightarrow **Очистити**;
- виконати команду **Видалити фільтр** із у списку стовпця, за даними якого було виконане фільтрування;
- установити позначку прапорця **Виділити все** у списку стовпця, за даними якого було виконане фільтрування, після чого вибрати кнопку **ОК**.

Щоб узагалі відмінити режим, у якому можна проводити фільтрування, потрібно повторно виконати одну з команд, яка встановлює цей режим.

Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Туркевич	Сергій	7	9	11	11	38
Штейн	Ірина	9	9	10	9	37

Мал. 7.28. Результат послідовного фільтрування за значеннями стовпців **Сума** та **Інформатика**

РОЗШИРЕНІ ФІЛЬТРИ

Використання автофільтрів дає змогу виконати фільтрування не в усіх випадках. Так, наприклад, розглянутими способами не можна виконати фільтрування за умовою, яка є об'єднанням умов фільтрування двох стовпців логічною операцією **АБО**, наприклад (**сума балів більше 35**) **АБО** (**бал з інформатики більше 8**).

Виконати такі види фільтрування в табличному процесорі **Excel 2007** можна за допомогою **розширених фільтрів**. Для цього потрібно:

1. Скопіювати або ввести у вільні клітинки електронної таблиці назви тих стовпців, за даними яких буде здійснюватися фільтрування (наприклад, на малюнку 7.29 назви стовпців *Сума, Інформатика*).
2. Увести в клітинки під назвами стовпців умови фільтрування. Якщо ці умови повинні об'єднуватися логічною операцією **I**, то вони мають розташовуватися в одному рядку. Якщо логічною операцією **АБО** – то у різних (на малюнку 7.29 умови $>35, >8$).
3. Виконати **Дані** \Rightarrow **Сортування й фільтр** \Rightarrow **Додатково**.
4. У вікні **Розширений фільтр**:
 1. Вибрати один з перемикачів для визначення області розташування результату фільтрування (на малюнку 7.29 – *фільтрувати список на місці*).
 2. Увести в поле **Вхідний діапазон** діапазон клітинок, дані у яких повинні фільтруватися (на малюнку 7.29 – *B2:H14*).



Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Бурчак	Андрій	9	9	9	7	34
Бурчак	Сергій	8	8	7	8	31
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Левчук	Олексій	7	9	8	9	33
Левчук	Олексій	7	9	9	9	34
Петренко	Катерина	7	8	9	9	33
Пройда	Іван	6	10	11	8	35
Стеценко	Іван	9	11	8	8	36
Туркевич	Сергій	7	9	11	11	38
Штейн	Ілля	9	8	9	9	35
Штейн	Ірина	9	9	10	9	37

Розширений фільтр

Обробка

фільтрувати список на місці

скопіювати результат до іншого розташування

Вихідний діапазон:

Діапазон умов:

Діапазон для результату:

Лише унікальні записи

OK Скасувати

Сума	Інформатика
>35	>8

Мал. 7.29. Умови фільтрування та вікно **Розширений фільтр**

3. Увести в поле **Діапазон умов** діапазон клітинок, у яких розташовані назви стовпців й умови фільтрування.
4. Якщо був вибраний перемикач **скопіювати результат до іншого розташування**, ввести в поле **Діапазон для результатів** діапазон клітинок, де має розміститися результат фільтрування.
5. Вибрати кнопку **ОК**.

На малюнку 7.30 представлено результат фільтрування, виконаного за умовами, поданими на малюнку 7.29.

Прізвище	Ім'я	Алгебра	Геометрія	Інформатика	Англійська	Сума
Бурчак	Андрій	9	9	9	7	34
Калина	Юлія	9	11	10	10	40
Кручина	Петро	10	10	9	11	40
Левчук	Олексій	7	9	9	9	34
Петренко	Катерина	7	8	9	9	33
Пройда	Іван	6	10	11	8	35
Стеценко	Іван	9	11	8	8	36
Туркевич	Сергій	7	9	11	11	38
Штейн	Ілля	9	8	9	9	35
Штейн	Ірина	9	9	10	9	37

Мал. 7.30. Результат фільтрування за умовами, поданими на малюнку 7.29

Найважливіше в цьому пункті

Дані в клітинках електронної таблиці можна сортувати, тобто змінювати порядок їх розташування в рядках або стовпцях за деякими ознаками. Сортування може проводитися за зростанням або за спаданням.

Для того щоб швидко виконати сортування виділеного зв'язного діапазону за значеннями першого стовпця цього діапазону, потрібно виконати **Основне** ⇒ **Редагування** ⇒ **Сортування й фільтр** ⇒ **Сортування від найменшого до найбільшого** або **Сортування від найбільшого до найменшого**.

Сортування у виділеному діапазоні можна здійснювати і за значеннями кількох стовпців. Для цього слід виконати **Дані** ⇒ **Сортування й фільтр** ⇒ **Сортувати** і встановити параметри та умови сортування. При цьому спочатку рядки сортуються за значеннями у першому з указаних стовпців, а потім у кожному наступному з вибраних стовпців відбувається лише для тих рядків, у яких значення в усіх попередніх указаних для сортування стовпцях збігаються.

Фільтрування – це вибір значень у клітинках електронної таблиці, що відповідають певним умовам. Після виконання фільтрування в таблиці відображаються лише ті рядки, дані в яких відповідають умовам фільтрації. Усі інші – тимчасово приховуються.

Для вибору режиму фільтрування потрібно встановити табличний курсор у довільну клітинку таблиці або виділити діапазон клітинок і виконати **Основне** ⇒ **Редагування** ⇒ **Сортування й фільтр** ⇒ **Фільтр** або **Дані** ⇒ **Сортування й фільтр** ⇒ **Фільтр**.

Для встановлення умов фільтрації можна використовувати фільтри за значенням, числові та текстові фільтри.

Фільтрування можна здійснювати послідовно для кількох стовпців таблиці. Тоді кожний наступний фільтр буде застосований вже тільки для тих рядків таблиці, які відображаються після застосування всіх попередніх фільтрів.

Для відміни фільтрації потрібно виконати **Дані** ⇒ **Сортування й фільтр** ⇒ **Очистити**.

Для виконання фільтрування за умовою, яка є об'єднанням умов фільтрування двох стовпців логічною операцією **АБО**, використовуються розширені фільтри. Для цього у вільних клітинках таблиці написати назви стовпців та умови фільтрування. Потім виконати **Дані** ⇒ **Сортування й фільтр** ⇒ **Додатково** та встановити параметри фільтрування у вікні **Розширений фільтр**.



Дайте відповіді на запитання

- 1°. У чому полягає сортування значень в Excel 2007?
- 2°. Які правила сортування даних за зростанням у Excel 2007?
- 3°. У який спосіб можна провести швидке сортування?
- 4°. Коли команда **Сортування від найменшого до найбільшого** змінюється на команду **Сортування від А до Я**?
- 5°. Як здійснюється сортування даних в таблицях за значеннями кількох стовпців?



- 6°. Що таке фільтрування?
- 7°. Які види фільтрування можна здійснювати для числових даних? для текстових даних?
- 8°. Для чого використовується розширений фільтр?
- 9°. Як задати умови до розширеного фільтра?
- 10*. У чому відмінності автоматичного та розширеного фільтра?

**Виконайте завдання**

- 1°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.4\зразок 7.4.1.xlsx**). На аркуші **Аркуш1** подано таблицю вильоту літаків з аеропортів міста Києва. Відсортуйте таблицю за зростанням часу вильоту літаків. На аркуші **Аркуш2** установіть режим фільтрування. Відобразіть дані лише про літаки, які вилітають з аеропорту «Бориспіль». Збережіть книгу у вашій папці у файлі з іменем **вправа 7.4.1.xlsx**.
- 2°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.4\зразок 7.4.2.xlsx**). На аркуші **Аркуш1** подано таблицю результатів участі спортсменів у змаганнях з триборства: фехтування (очки), біг (час), стрибки у висоту (м). Відсортуйте таблицю за значеннями в трьох стовпцях: **Очки** (за спаданням), **Час** (за зростанням), **Висота** (за спаданням). На аркуші **Аркуш2** установіть режим фільтрування. Відобразіть дані про учасників, які набрали у фехтуванні не менше ніж 40 очок і стрибнули у висоту від 2,20 до 2,25 м. Збережіть книгу у вашій папці у файлі з іменем **вправа 7.4.2.xlsx**.
-  3°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.4\зразок 7.4.3.xlsx**). На аркуші **Аркуш1** подано таблицю вильоту літаків з аеропортів міста Києва. Відсортуйте таблицю так, щоб дані про літаки, що вилітають з одного аеропорту, йшли підряд, дані про аеропорт призначення розміщувались у порядку від Я до А, а час вильоту збільшувався. На аркуші **Аркуш2** установіть режим фільтрування. Відобразіть дані про літаки, які вилітають до Варшави з 9 до 15 години. Збережіть книгу у вашій папці у файлі з іменем **вправа 7.4.3.xlsx**.
- 4°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.4\зразок 7.4.xlsx**). На аркуші **Аркуш1** подано таблицю з даними про жорсткі магнітні диски. Задайте умову розширеної фільтрації для добору даних про ЖМД, у яких або ємність більша ніж 500 Гбайт, або ціна менша ніж 1000 грн. Збережіть книгу у вашій папці у файлі з іменем **вправа 7.4.4.xlsx**.
-  5*. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.4\зразок 7.4.5.xlsx**). На аркуші **Аркуш1** подано таблицю з даними про TFT-монітори. Задайте умову розширеної фільтрації для добору даних про монітори, у яких або діагональ більша ніж 19", або ціна менша ніж 3000 грн. Збережіть книгу у вашій папці у файлі з іменем **вправа 7.4.5.xlsx**.

7.5. ПРОМІЖНІ ПІДСУМКИ. УМОВНЕ ФОРМАТУВАННЯ



1. Що таке сортування даних таблиці? Якими способами можна виконати сортування?
2. Що таке фільтрування даних таблиці? Якими способами можна його виконати?
3. Що таке розширене фільтрування? Наведіть приклад його використання.

ПРОМІЖНІ ПІДСУМКИ

Часто в одному або кількох стовпцях таблиці містяться дані, що повторюються. Це можуть бути, наприклад, гуртки, які відвідують учні, назви фірм-постачальників, товарів, пунктів призначення руху потягів та ін.

На малюнку 7.31 подано таблицю з даними про надходження та продаж книжок у деякому книжковому магазині.

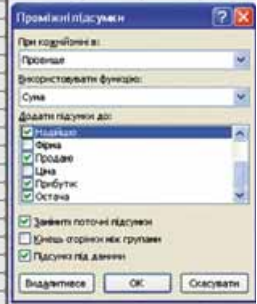
Як ми бачимо, у трьох стовпцях цієї таблиці (**Прізвище**, **Жанр** і **Фірма**) є дані, що багаторазово повторюються. Аналізуючи дані цієї таблиці, важливо з'ясувати, скільки екземплярів книжок кожного з авторів надійшло та скільки з них продано, скільки продано книжок жанру проза і скільки – поезія, скільки книжок надійшло від кожної з фірм-постачальників і скільки з них продано та ін. Такий аналіз дасть змогу власникам магазину планувати, який із жанрів (проза чи поезія) має більший попит, які

Прізвище	Ім'я	Назва	Жанр	Надійшло	Фірма	Продано	Ціна	Прибуток	Остача
Бедрик	Юрій	Свято небуття	Вірші	5	Орфей	4	15,5	62	1
Андрухович	Юрій	Середмістя	Вірші	8	Селена	5	25	125	3
Андрухович	Юрій	Дванадцять обручів	Проза	7	Орфей	3	30,5	91,5	4
Бердник	Олень	Камертон Дажбога	Проза	10	Селена	10	21,7	217	0
Бердник	Олень	Вогнесміх	Проза	12	Тріумф	8	19,8	158,4	4
Бураго	Дмитро	Ехо мертвого озера	Вірші	3	Орфей	1	15	15	2
Вороний	Микола	Ліричні поезії	Вірші	7	Орфей	5	26	130	2
Вінграновський	Микола	Первинка	Вірші	21	Орфей	16	30	480	5
Вінграновський	Микола	Київ	Вірші	15	Орфей	12	27	324	3
Забужко	Оксана	Сестро, сестро	Проза	10	Селена	8	32	256	2
Забужко	Оксана	Інопланетянка	Проза	28	Тріумф	12	35	420	16
Забужко	Оксана	Друга спроба	Вірші	25	Орфей	20	31	620	5
Забіла	Наталя	Ясоччина книжка	Проза	15	Орфей	12	12,5	150	3
Забіла	Наталя	Веселим малюкам	Вірші	12	Орфей	10	8,7	87	2
Кушнір	Віктор	Повернення	Вірші	11	Орфей	4	17	68	7
Павличко	Дмитро	Не зрадь	Вірші	15	Селена	4	21	84	11
Павличко	Дмитро	Сонети	Вірші	24	Селена	11	25	275	13
Павличко	Дмитро	Три строфи	Вірші	27	Тріумф	17	25	425	10
Симоненко	Василь	Вибране	Вірші	30	Орфей	25	20	500	5
Гончар	Олень	Собор	Проза	40	Орфей	28	27,6	772,8	12
Щербак	Юрій	Чорнобиль	Проза	12	Тріумф	3	19,4	58,2	9
Хорунжий	Юрій	Вірую	Проза	14	Селена	2	32	64	12
Хорунжий	Юрій	Таємна грамота	Проза	9	Селена	1	30	30	8
Стельмах	Михайло	Чотири броди	Проза	23	Селена	14	24	336	9
Старицький	Михайло	Історичні романи	Проза	14	Тріумф	5	27,8	139	9

Мал. 7.31. Дані про книжки, що надійшли до магазину від фірм-постачальників, і про продаж цих книжок



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Прізвище	Ім'я	Назва	Жанр	Надійшло	Фірма	Продано	Ціна	Прибуток	Остача				
2	Андрухович	Юрій	Дванадцять обрубків	Проза	7	Орфей	3	30,5	91,5	4				
3	Андрухович	Юрій	Середніста	Вірші	8	Селена	5	25	125	3				
4	Бердник	Юрій	Свято небуття	Вірші	5	Орфей	4	15,5	62	1				
5	Бердник	Олеся	Камертон Дажбога	Проза	10	Селена	10	21,7	217	0				
6	Бердник	Олеся	Вогнеснік	Проза	12	Тріумф	8	19,8	158,4	4				
7	Бураго	Дмитро	Ехо мертвого озера	Вірші	3	Орфей	1	15	15	2				
8	Вігнановський	Микола	Первинка	Вірші	21	Орфей	16	30	480	5				
9	Вігнановський	Микола	Київ	Вірші	15	Орфей	12	27	324	3				
10	Вороний	Микола	Ліричні поезії	Вірші	7	Орфей	5	26	130	2				
11	Гончар	Олеся	Собор	Проза	40	Орфей	28	27,6	772,8	12				
12	Забіла	Наталія	Ясочена книжка	Проза	15	Орфей	12	12,5	150	3				
13	Забіла	Наталія	Веселим малюкам	Вірші	12	Орфей	10	8,7	87	2				
14	Забужко	Оксана	Друга спроба	Вірші	25	Орфей	20	31	620	5				
15	Забужко	Оксана	Сестро, сестро	Проза	10	Селена	8	32	256	2				
16	Забужко	Оксана	Інопланетянка	Проза	28	Тріумф	12	35	420	16				
17	Кущеніс	Віктор	Повернення	Вірші	11	Орфей	4	17	68	7				
18	Павличко	Дмитро	Не зрадь	Вірші	15	Селена	4	21	84	11				
19	Павличко	Дмитро	Сонети	Вірші	24	Селена	11	25	275	13				
20	Павличко	Дмитро	Три строфи	Вірші	27	Тріумф	17	25	425	10				
21	Симоненко	Василь	Вибране	Вірші	30	Орфей	25	20	500	5				
22	Старицький	Михайло	Історичні романи	Проза	14	Тріумф	5	27,8	139	9				
23	Стецьмак	Михайло	Чотири броди	Проза	23	Селена	14	24	336	9				
24	Хорунжий	Юрій	Вірую	Проза	14	Селена	2	32	64	12				
25	Хорунжий	Юрій	Тасма грамота	Проза	9	Селена	1	30	30	8				
26	Щербак	Юрій	Чорнобиль	Проза	12	Тріумф	3	19,4	58,2	9				



Мал. 7.32. Таблиця, відсортована за значеннями в стовпці **Прізвище**, і вікно **Проміжні підсумки**


з авторів більш популярні, а які – менше, яка з фірм постачає книжки, що користуються більшим попитом, та ін.

Щоб одержати відповіді на ці та інші запитання, потрібно об'єднати в групи дані, що належать до кожного з повторюваних даних (наприклад, назва фірми або прізвище автора), і потім уже проводити потрібні обчислення (наприклад, знаходження суми, середнього, найбільшого тощо) уже всередині кожної групи окремо.

Одним зі способів виконати такі операції є знаходження **проміжних підсумків**.

Щоб знайти проміжні підсумки, потрібно:

1. Виділити клітинки, для значень яких будуть знаходитися проміжні підсумки.
2. Виконати сортування виділеного діапазону клітинок за даними в тому стовпці, за яким потрібно згрупувати дані.
3. Виконати **Дані** ⇒ **Структура** ⇒ **Проміжні підсумки**.
4. Вибрати у вікні **Проміжні підсумки** (мал. 7.32) у списку поля **Використовувати функцію** ту функцію, яка буде опрацьовувати дані (*Сума, Кількість, Середнє, Максимум* тощо).
5. Установити у вікні **Проміжні підсумки** в списку поля **Додати підсумки** до позначки прапорців поруч з назвами тих стовпців, для яких будуть обчислені проміжні підсумки.
6. Вибрати кнопку **ОК**.

На малюнку 7.33 подано результат виконання цього алгоритму для таблиці, розглянутої вище. Дані в таблиці об'єднані в групи за значеннями в стовпці **Прізвище** і для кожної групи утворився рядок **Підсумок**, у кожному з яких підраховані суми в стовпцях **Надійшло**, **Продано**, **Прибуток** і **Остача**. Кожна з груп повністю розкрита, про що свідчать кнопки  ліворуч від кожної з них. Якщо вибрати одну із цих кнопок, то відповідна група рядків згорнеться, відобразатиметься тільки рядок **Підсумок**, а кнопка

	A	B	C	D	E	F	G	H	I	J
1	Прізвище	Ім'я	Назва	Жанр	Надійшло	Фірма	Продано	Ціна	Прибуток	Остача
2	Андрухович	Юрій	Дванадцять обручів	Проза	7	Орфей	3	30,5	91,5	4
3	Андрухович	Юрій	Середмістя	Вірші	8	Селена	5	25	125	3
4	Андрухович Підсумок				15		8		216,5	7
5	Бедрик	Юрій	Свято небуття	Вірші	5	Орфей	4	15,5	62	1
6	Бедрик Підсумок				5		4		62	1
7	Бердник	Олесь	Камертон Дажбога	Проза	10	Селена	10	21,7	217	0
8	Бердник	Олесь	Вогнесміх	Проза	12	Тріумф	8	19,8	158,4	4
9	Бердник Підсумок				22		18		375,4	4
10	Бураго	Дмитро	Ехо мертвого озера	Вірші	3	Орфей	1	15	15	2
11	Бураго Підсумок				3		1		15	2
12	Вінграновський	Микола	Первинка	Вірші	21	Орфей	16	30	480	5
13	Вінграновський	Микола	Київ	Вірші	15	Орфей	12	27	324	3
14	Вінграновський Підсумок				36		28		804	8
15	Вороний	Микола	Ліричні поезії	Вірші	7	Орфей	5	26	130	2
16	Вороний Підсумок				7		5		130	2
17	Гончар	Олесь	Собор	Проза	40	Орфей	28	27,6	772,8	12
18	Гончар Підсумок				40		28		772,8	12
19	Забіла	Наталя	Ясоччина книжка	Проза	15	Орфей	12	12,5	150	3
20	Забіла	Наталя	Веселим малюкам	Вірші	12	Орфей	10	8,7	87	2
21	Забіла Підсумок				27		22		237	5
22	Забужко	Оксана	Друга спроба	Вірші	25	Орфей	20	31	620	5
23	Забужко	Оксана	Сестро, сестро	Проза	10	Селена	8	32	256	2
24	Забужко	Оксана	Інопланетянка	Проза	28	Тріумф	12	35	420	16
25	Забужко Підсумок				63		40		1296	23
26	Кушнір	Віктор	Повернення	Вірші	11	Орфей	4	17	68	7
27	Кушнір Підсумок				11		4		68	7
28	Павличко	Дмитро	Не зрадь	Вірші	15	Селена	4	21	84	11
29	Павличко	Дмитро	Сонети	Вірші	24	Селена	11	25	275	13
30	Павличко	Дмитро	Три строфи	Вірші	27	Тріумф	17	25	425	10
31	Павличко Підсумок				66		32		784	34

Мал. 7.33. Таблиця з проміжними підсумками за значеннями стовпця **Прізвище**

ліворуч від групи набуде вигляду **+**. У будь-який момент можна вибрати цю кнопку і знову розгорнути відповідну групу.

Над кнопками згортання/розгортання груп знаходяться три кнопки **1**, **2** і **3** швидкого згортання/розгортання рівнів згрупованої таблиці: перший рівень – це вся таблиця, другий – групи, третій – члени груп.

УМОВНЕ ФОРМАТУВАННЯ

Ще одним способом вибрати в таблиці значення, які задовольняють певні умови, є так зване **умовне форматування**.

Умовне форматування автоматично змінює формат клітинки на заданий, якщо виконується задана умова.

Наприклад, в електронному журналі класу зручно позначати різними кольорами дані про учнів, які мають оцінки низького, середнього, достатнього та високого рівнів. Тобто клітинки, значення яких знаходяться в діапазоні від 1 до 3, виділити червоним кольором, оцінки від 4 до 6 – зеленим, від 7 до 9 – жовтим, від 10 до 12 – синім кольором.

Звертаємо вашу увагу: на відміну від фільтрування, умовне форматування не приховує клітинки, значення в яких не задовольняють заданої



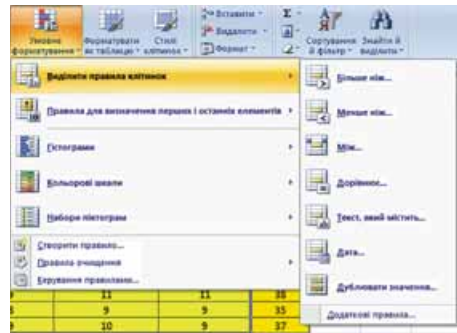
умови, а лише виділяє в таблиці певним чином ті клітинки, значення в яких задовольняють задану умову.

В Excel 2007 для умовного форматування існує п'ять способів виділення клітинок (мал. 7.34):

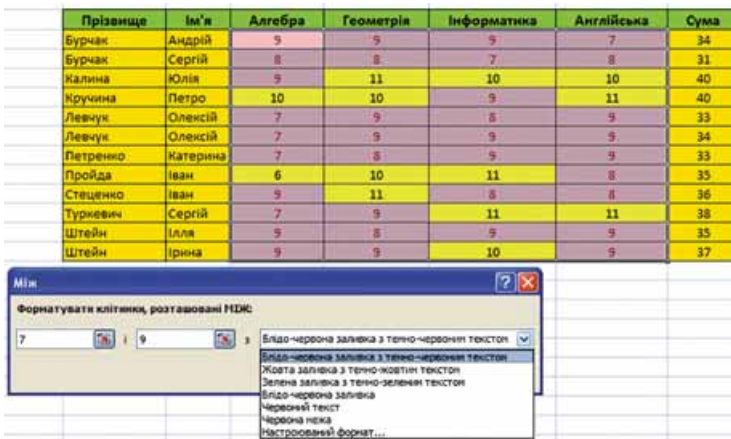
- Виділити правила клітинок.
- Правила для визначення перших і останніх елементів.
- Гістограми.
- Кольорові шкали.
- Набори піктограм.

Для встановлення перших двох видів умовного форматування, а саме встановлення **правил форматування**, слід:

1. Виділити потрібний діапазон клітинок.
2. Виконати **Основне** ⇒ **Стилі** ⇒ **Умове форматування**.
3. Вибрати у списках кнопки **Умове форматування** потрібний спосіб (мал. 7.34).
4. Указати у списку вибраного способу потрібне правило форматування.
5. Задати у вікні, що відкриється, умову форматування та формат, який буде встановлено, якщо умова виконуватиметься (мал. 7.35). Формат можна вибрати з наведених або, вибравши команду **Настроюваний формат**, установити власний у вікні **Формат клітинок**.
6. Вибрати кнопку **ОК**.



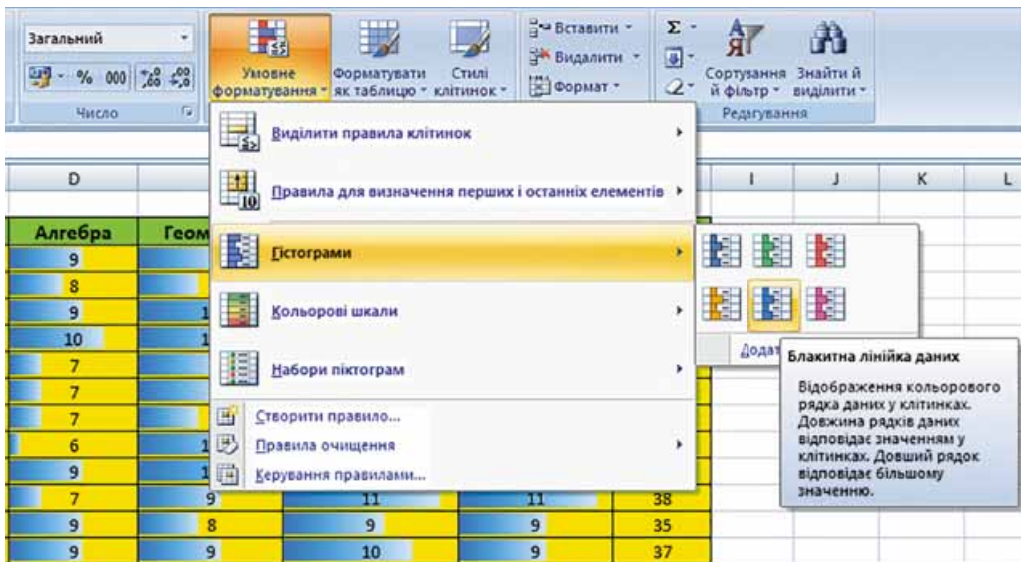
Мал. 7.34. Кнопка **Умове форматування**, список типів правил, а також список правил типу **Виділити правила клітинок**



Мал. 7.35. Виділений діапазон клітинок, вікно **Між** з умовою і списком стандартних форматів

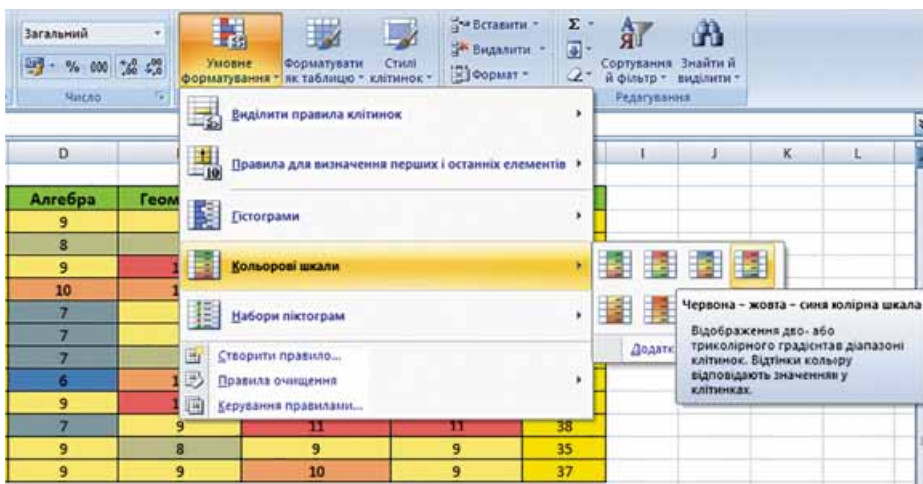
На малюнку 7.35 подано, як приклад, вікно **Між**, у якому встановлено умову форматування **Між 7 і 9**, зі списком стандартних форматів, а також попередній перегляд вибраного умовного форматування клітинок таблиці.

Установлення іншого виду умовного форматування типу **Гістограми** приводить до вставлення в клітинки виділеного діапазону гістограм, довжина кольорового рядка яких пропорційна значенню в клітинці (мал. 7.36).



Мал. 7.36. Умовне форматування зі вставленням у клітинки гістограм

Установлення умовного форматування типу **Кольорові шкали** приводить до встановлення заливки клітинок виділеного діапазону так, що клітинки з однаковими значеннями мають один і той самий колір заливки (мал. 7.37).



Мал. 7.37. Умовне форматування кольоровими шкалами

Можна також вибрати вид умовного форматування зі списку **Набори піктограм**. Під час такого форматування в клітинках виділеного діапазону з'являтимуться піктограми з вибраного набору. Поява конкретної піктограми з набору в клітинці означає, що значення в цій клітинці задовольняє умову, встановлену для кожної піктограми набору.



Звертаємо вашу увагу: до одного діапазону клітинок може бути застосовано кілька правил, заданих послідовно.

Для видалення умовного форматування потрібно виконати *Умовне форматування* ⇒ *Правила очищення* ⇒ *Очистити правила у виділених клітинках (Очистити правила у всьому аркуші)*.



Дайте відповіді на запитання

- 1°. Що таке проміжні підсумки? Коли їх доцільно створювати?
- 2°. Як знайти проміжні підсумки в **Excel 2007**?
- 3°. Які функції можна обчислювати в проміжних підсумках?
- 4°. Що таке умовне форматування? Для чого воно виконується?
- 5°. Які є способи умовного форматування?
- 6°. Як виконати умовне форматування?
- 7°. Як очистити умовне форматування на всьому аркуші?



Виконайте завдання

- 1°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.5\зразок 7.5.1.xlsx**). На аркуші **Аркуш1** подано таблицю зі значеннями властивостей вінчестерів. Обчисліть проміжні підсумки за значеннями в стовпці **Фірма**. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.5.1.xlsx**.
- 2°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.5\зразок 7.5.2.xlsx**). На аркуші **Аркуш1** подано таблицю зі значеннями властивостей моніторів. Утворіть проміжні підсумки за значеннями в стовпці **Діагональ**. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.5.2.xlsx**.
- 3°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.5\зразок 7.5.3.xlsx**). На аркуші **Аркуш1** подано таблицю з даними про TFT-монітори. Зафарбуйте в рожевий колір усі клітинки, у яких діагональ монітора менша ніж 21". Збережіть книгу у вашій папці у файлі з іменем **завдання 7.5.3.xlsx**.
- 4°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.5\зразок 7.5.4.xlsx**). На аркуші **Аркуш1** подано таблицю з даними про жорсткі магнітні диски. Установіть синій колір клітинок, у яких ємність кеш-пам'яті більша ніж 8 Мбайт. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.5.4.xlsx**.
- 5°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.5\зразок 7.5.5.xlsx**). На аркуші **Аркуш1** подано таблицю з даними про температуру за останній місяць в м. Києві. Установіть умовне форматування **Гістограми (фіолетова лінійка даних)** для всіх числових даних таблиці. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.5.4.xlsx**.
- 6°. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Пункт 7.5\зразок 7.5.6.xlsx**). Знайдіть в Інтернеті дані про кількість населення у вашому обласному центрі за останні 5 років. На аркуші **Аркуш1** створіть електронну таблицю за цими даними. Установіть



умовне форматування з використанням **Наборів піктограм (4 світлофори)** для даних таблиці. Збережіть книгу у вашій папці у файлі з іменем **завдання 7.5.6.xlsx**.

ПРАКТИЧНА РОБОТА № 16

«Упорядкування даних у таблицях. Автоматичні та розширені фільтри. Умовне форматування»

Увага! Під час роботи з комп'ютером дотримуйтеся правил безпеки та санітарно-гігієнічних норм.

1. Відкрийте вказаний учителем файл (наприклад, **Розділ 7\Практична 16\зразок пр16.xlsx**).
2. На аркуші **Аркуш1** виконайте сортування таблиці за значеннями в стовпці **Прізвище**.
3. На аркуші **Аркуш2** виконайте сортування таблиці за значеннями в кількох стовпцях в указаному порядку: **Прізвище** (за зростанням), **Ім'я** (за зростанням), **Рік народження** (за спаданням).
4. На аркуші **Аркуш3** відобразіть дані про учнів, зріст яких вище за 1 м 80 см.
5. На аркуші **Аркуш4** відобразіть дані про учнів, які відвідують гуртки малювання або співу.
6. На аркуші **Аркуш5** знайдіть проміжні підсумки за значеннями у стовпці **Вага учнів**, які знаходять середню та максимальну вагу.
7. На аркуші **Аркуш6** установіть умовне форматування, щоб клітинки, в яких вага учнів більше ніж 80 кг, зафарбовувалися червоним кольором, а клітинки, у яких вага учнів менше ніж 75 кг, – синім.
8. Збережіть книгу у вашій папці у файлі з іменем **практична 16.xlsx**.

Розділ 8. Розв'язування компетентнісних задач, виконання навчальних проектів

У цьому розділі ви дізнаєтеся про:

етапи розв'язування задач з різних галузей людської діяльності з використанням інформаційно-комунікаційних технологій

стратегію пошуку інформаційних матеріалів

дії на кожному з етапів розв'язування компетентнісних задач з використанням інформаційно-комунікаційних технологій

виконання навчальних проектів з інформатики

8.1. РОЗВ'ЯЗУВАННЯ КОМПЕТЕНТІСНИХ ЗАДАЧ



1. Що таке компетентісна задача?
2. Який алгоритм розв'язування задач з різних галузей людської діяльності?
3. Які засоби можуть бути використані для подання розв'язку компетентісних задач?

ЕТАПИ РОЗВ'ЯЗУВАННЯ КОМПЕТЕНТІСНИХ ЗАДАЧ

У 7 класі ви почали розв'язувати задачі, які називають компетентісними. Нагадаємо, що **компетентісними** називають задачі з різних галузей діяльності людини, які потребують від людини вміння використовувати набуті знання на практиці. Їх розв'язок полягає у вирішенні деякої життєвої проблеми із застосуванням знань, умінь та навичок, які ви отримали на уроках з різних предметів. Значна частина таких задач не обмежується предметною областю одного навчального предмета, а є міжпредметними.

Ви вже знаєте етапи розв'язування компетентісних задач (мал. 8.1).

Розглянемо детальніше названі етапи.

Виконати змістовий аналіз формулювання задачі означає виділити у формулюванні деталі завдання та явно вказані початкові дані, визначити потребу в інших даних, потрібних для розв'язання задачі, окреслити, що буде вважатися кінцевим результатом.



Мал. 8.1. Етапи розв'язування компетентісних задач



Здійснити пошук інформаційних матеріалів, виконавши два завдання:

- **Розробити стратегію пошуку інформаційних матеріалів**, визначивши:
 - мету та завдання пошуку;
 - джерела, у яких буде здійснюватися пошук;
 - ключові слова;
 - критерії оцінювання знайдених матеріалів.
- **Здійснити пошук інформаційних матеріалів**, зокрема в інтернет-ресурсах, відповідно до розробленої стратегії, зіставити результати пошуку з метою, оцінити за критеріями.

Стратегія (давньогр. στρατηγία – мистецтво полководця) – загальний, недеталізований план певної діяльності, який охоплює тривалий період, спосіб досягнення складної мети.

Побудувати інформаційну модель задачі означає скласти план та опис розв'язування задачі, визначити послідовність дій, що приведуть до виконання поставленого завдання.

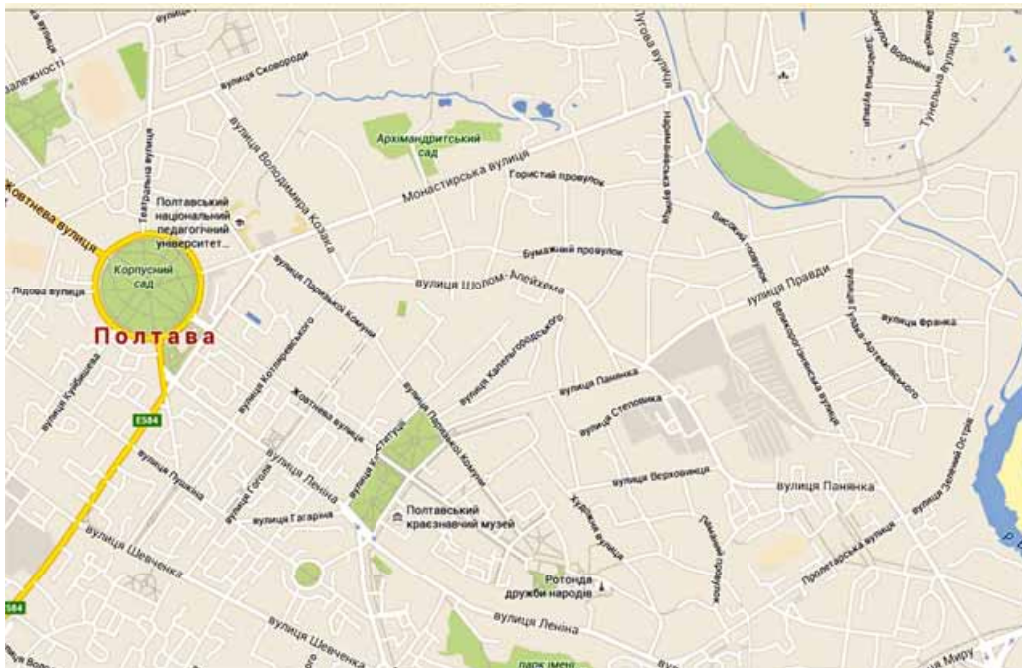
Вибрати засоби опрацювання даних залежно від поставленого завдання та наявних матеріалів. Засобами може бути текстовий процесор, графічний редактор, редактор презентацій, табличний процесор, навчальне середовище виконання алгоритмів, редактор мультимедійних даних та ін. У багатьох задачах може бути використано кілька засобів.

Здійснення опрацювання даних передбачає застосування вибраних програмних засобів для реалізації плану розв'язування задачі, описаного в інформаційній моделі задачі. На цьому етапі виконується порівняння та зіставлення відомостей, одержаних із різних джерел, виключення невідповідних та несуттєвих відомостей, структурування даних, отримання нових даних на основі відомих, формулювання висновків.

Подати результати розв'язування задачі відповідно до аудиторії означає розробити підсумковий документ, у якому відображається хід і результати розв'язування задачі, та презентувати його визначеній аудиторії. При цьому суттєвим є врахування призначення підсумкового документа, стислий та логічний виклад узагальнених даних, обґрунтування висновків.

ПРИКЛАД РОЗВ'ЯЗУВАННЯ КОМПЕТЕНТІСНОЇ ЗАДАЧІ

Згадаємо етапи розв'язування компетентісних задач на прикладі такої задачі: *Ваш клас збирається у період навчально-виробничої практики поїхати на одноденну екскурсію в один з обласних центрів України. Мета екскурсії – ознайомитися з історичними пам'ятками, пов'язаними з подіями, що вивчалися у курсі історії України у 8 класі. Щоб вибрати місто для поїздки, домовилися з'ясувати, які пам'ятки вивченого періоду історії України, можна побачити в кожному місті, та розробити маршрути пішохідних екскурсій. Маршрут може бути доповнений відвідуванням пам'яток, пов'язаних з визначними особистостями, які відомі з курсу історії України або української літератури. Кожен учень класу збирає відомості про одне з міст. Після обговорення в класі знайдених відомостей буде обрано місто проведення екскурсії. Вам дісталось збирати відомості про місто Полтаву. Для зручності виконання завдання вам надано файл з картою Полтави (мал. 8.2).*



Мал. 8.2. Карта Полтави

1. Під час змістового аналізу формулювання задачі потрібно з'ясувати:

- а) Що необхідно отримати як розв'язок задачі?
 - Прокласти на карті маршрут екскурсії містом Полтавою.
 - Забезпечити тривалість екскурсії на прокладеному маршруті, доцільну для одноденної поїздки.
 - Презентувати історичні пам'ятки на маршруті.
- б) Які дані, потрібні для розв'язання задачі, є в умові?
 - Місця, вибрані для екскурсії, повинні стосуватися подій, що вивчались у курсі історії України у 8 класі (*Україна в XVI–XVIII ст.*), та історичних осіб, відомих з історії України або української літератури.
 - Тривалість пішохідної екскурсії повинна бути в межах одного дня.
- в) Які дані потрібно знайти або пригадати з раніше вивченого для розв'язання задачі?
 - Які події з історії України в XVI–XVIII ст. та історичні особистості пов'язані з Полтавою? (*Полтавська битва 1709 р. – за даними з підручника історії України для 8 класу; Г.Сковорода, Т.Шевченко, М.Чурай, І.Котляревський – за даними підручників історії України та української літератури*).
 - Які пам'ятки в сучасній Полтаві пов'язані з історичними подіями та особистостями?
 - Які пам'ятки, пов'язані з історичними подіями та особистостями, можна відвідати впродовж одноденної пішохідної екскурсії?



2. Пошук інформаційних матеріалів потрібно виконати для відшукування відомостей про пам'ятки та їх розташування на території Полтави.

а) Визначимо стратегію пошуку:

- **Мета пошуку:** отримати опис та дані про розташування в Полтаві історичних пам'яток, пов'язаних з подіями в Україні XVI–XVIII ст. та історичними особистостями.
- **Завдання пошуку:** знайти тексти з описом пам'яток, їхні адреси, фотографії, карти з позначеними місцями розташування пам'яток.
- **Джерела пошуку** списку історичних пам'яток: друквані енциклопедії, довідники, путівники, Інтернет. Відомості про історичні пам'ятки можуть бути отримані як з офіційних сайтів міста, його музеїв, галерей, так і з сайтів енциклопедій, туристичних фірм, особистих сайтів та блогів користувачів, що відвідували Полтаву.
- **Ключові слова** для пошуку відомостей про історичні пам'ятки:
 - Полтава XVI–XVIII ст.;
 - історичні місця Полтави;
 - історичні особистості Полтави;
 - літературна Полтавщина;
 - екскурсія в Полтаву;
 - пам'ятки Полтавської битви;
 - музей Полтавської битви.
- **Критерії оцінювання** знайдених матеріалів:
 - матеріали повинні бути актуальними, не застарілими;
 - повинна існувати можливість перевірити знайдені дані за даними з інших джерел;
 - зображення повинні бути якісними.

б) У результаті пошуку та оцінювання знайдених ресурсів були отримані такі відомості:

- У Полтаві з подіями XVI–XVIII ст. пов'язані такі пам'ятки:
 - Державний історико-культурний заповідник «Поле Полтавської битви» (battle.poltava.org);
 - Музей історії Полтавської битви (tourism.poltava.ua/muzei/Muzej_istorii_Poltavskoi_bitvi);
 - історичні пам'ятники на території міста Полтави за межами заповідника «Поле Полтавської битви» (histpol.pl.ua/ru/poltava-istoricheskie-ocherki/poltavskaya-bitva-i-ego-pamyatniki).
- З історичними особистостями на території міста Полтави пов'язані такі пам'ятки:
 - меморіальний музей-садиба І. Котляревського;
 - пам'ятники Т. Шевченку, М. Чурай, І. Котляревському.

3. Після аналізу знайдених матеріалів можна скласти словесну інформаційну модель задачі:

Більшість історичних пам'яток Полтави, пов'язаних з подіями XVI–XVIII ст., розміщено на території Державного історико-культур-



Мал. 8.3. Приклад презентації з описом історичних пам'яток Полтави

ного заповідника «Поле Полтавської битви» та поруч з Музеєм історії Полтавської битви на північній околиці міста. Це десять гранітних обелісків на місці колишніх редутів, пам'ятник шведам від росіян, шведам від шведів, Сампсоніївська церква та ін. Екскурсію заповідником можна замовити в Музеї історії Полтавської битви.

У центрі міста може бути проведена пішохідна екскурсія, маршрут якої розпочинається з монумента Слави, проходить повз Пам'ятник загиблим українським козакам, Спаську церкву, пам'ятник Т. Шевченку, Музей-садину І. Котляревського та завершується біля Білої альтанки, звідки відкривається панорама на Хрестовоздвиженський монастир.

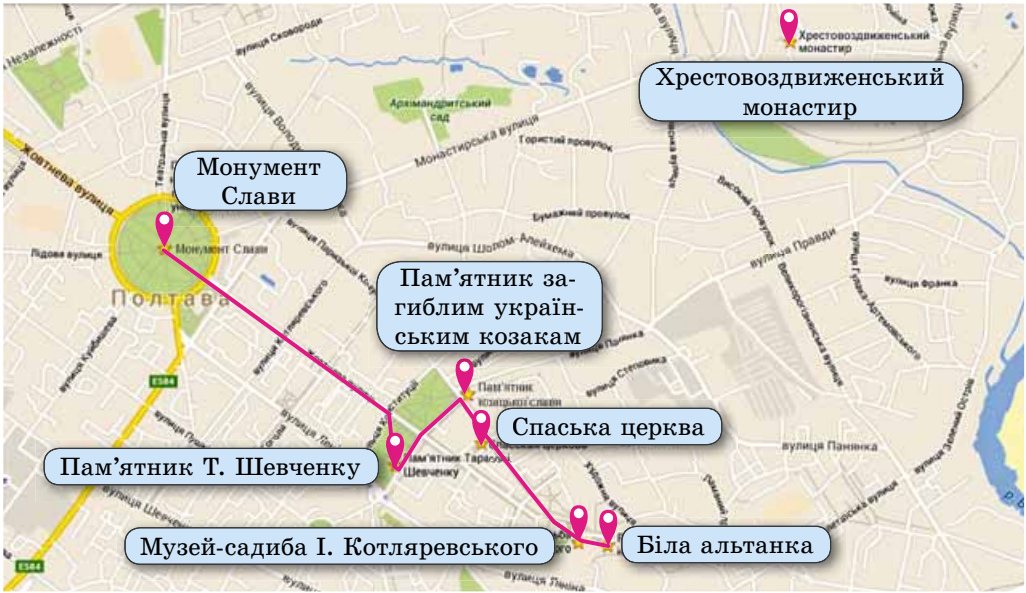
Відстань від першої до останньої точки маршруту складає близько 2 км, що відповідає умові пішохідної одноденної екскурсії.

4. Для подання результатів розв'язання задачі потрібно **визначити, які засоби для цього може бути використано.**

Прокласти маршрут пішохідної екскурсії на карті можна в середовищі графічного редактора. Для подання опису пам'ятних місць на маршруті може бути використано редактор презентацій, текстовий процесор або редактор відео.

5. **Опрацювання даних**, тобто створення підсумкових документів, буде виконуватися залежно від обраних програм.

6. Один з варіантів подання результатів розв'язування задачі наведено на малюнках 8.3 та 8.4.



Мал. 8.4. Приклад карти з маршрутом пам'ятними місцями Полтави

Найважливіше в цьому пункті

Під час розв'язування задач слід дотримуватися таких етапів:

1. Змістовий аналіз формулювання задачі.
2. Пошук необхідних матеріалів і відомостей.
3. Побудова інформаційної моделі задачі.
4. Добір засобів опрацювання даних.
5. Опрацювання даних.
6. Подання результатів розв'язування задачі.

Розробка **стратегії пошуку** інформаційних матеріалів полягає у визначенні:

- мети та завдань пошуку;
- джерел, у яких буде здійснюватися пошук;
- ключових слів;
- критеріїв оцінювання знайдених матеріалів.

Дайте відповіді на запитання

- 1°. Що таке компетентнісна задача?
- 2°. Які етапи розв'язування компетентнісних задач?
- 3°. Які дії здійснюються на етапі змістового аналізу формулювання задачі?
- 4°. На підставі чого визначають, які матеріали та відомості нам потрібні для розв'язання задачі?
- 5°. У чому полягає розробка стратегії пошуку інформаційних матеріалів?
- 6°. У яких формах може бути подано інформаційну модель задачі?



- 7°. Які дії здійснюються на етапі опрацювання даних?
- 8°. Від чого залежить вид подання результатів задачі?
- 9°. Які засоби опрацювання даних з використанням комп'ютерів ви вивчили у 8 класі?



Виконайте завдання

- 1°. Побудуйте схему маршруту екскурсії та опис історичних пам'яток Полтави за зразком на малюнку 8.3 (текстові та фотоматеріали містяться в папці **Розділ 8\Пункт 8.1\Завдання 8.1.1**) з використанням:

а°) графічного редактора, наприклад **Paint**, та редактора презентацій, наприклад **PowerPoint**;



б°) текстового процесора, наприклад **Word**, для зображення схеми маршруту екскурсії та редактора відео для створення відеосюжету про історичні пам'ятки;

в*) середовища виконання алгоритмів, наприклад **Scratch**, для створення анімації руху вказівника вздовж маршруту екскурсії та текстового процесора, наприклад **Word**, для опису пам'яток.

- 2°. Побудуйте схему маршруту одноденної пішохідної екскурсії та опис історичних пам'яток, пов'язаних з подіями з історії України XVI–XVIII ст. та визначними історичними особистостями у вашому обласному центрі. Розрахунок довжини маршруту виконайте в табличному процесорі. Інші засоби опрацювання даних виберіть самостійно.



- 3*. Побудуйте схему маршруту одноденної пішохідної екскурсії та опис історичних пам'яток, пов'язаних із подіями з історії України XVI–XVIII ст. та визначними історичними особистостями в будь-якому обласному центрі України за вашим вибором. Розрахуйте вартість екскурсії, включаючи проїзд, харчування, замовлення платних екскурсій. Розрахунки вартості виконайте в табличному процесорі, інші засоби опрацювання даних виберіть самостійно.

- 4°. Знайдіть відомості та створіть пам'ятку для уроків трудового навчання: у текстовому процесорі – як обирати режим прання одягу залежно від позначень, наведених на одязі; у редакторі презентацій – як обирати режим прасування одягу залежно від тканини, з якої виготовлено одяг, та позначень, наведених на одязі.

- 5°. Знайдіть відомості та створіть презентацію для уроку географії, у якій розкажіть про особливості розселення населення в різних областях України. Діаграми розселення виконайте в табличному процесорі.

- 6°. Знайдіть рецепти та фотографії страв, які готують у різних країнах до свята Великодня. Створіть у текстовому процесорі добірку святкових страв для використання на уроках трудового навчання. Складіть програму, наприклад у середовищі **Lazarus**, для обчислення вартості традиційного українського великоднього кошика.



- 7°. Знайдіть рецепти великодньої паски. Створіть відеоурок з приготування паски за одним з рецептів. У табличному процесорі обчис-



літь вартість продуктів для приготування паски за вибраним рецептом.

- 8*. Ви плануєте класом піти в похід до лісу. Знайдіть відомості про способи орієнтування в лісі без компаса та прийоми виживання у складних погодних умовах. Створіть у текстовому процесорі пам'ятку зі знайденими відомостями. У табличному процесорі виконайте розрахунки кількості питної води, потрібної для походу залежно від його тривалості та кількості учасників, і вартості покупки потрібної кількості води залежно від ємності упаковок.
- 9*. Ваша родина вирішила зробити ремонт у вашій кімнаті. Вам запропонували вибрати шпалери, розрахувати кількість рулонів, що потрібна для ремонту, їх вартість та подати батькам ваше бачення вигляду кімнати й отримані розрахунки. Засоби для розв'язання задачі та подання результатів батькам виберіть самостійно.



8.2. НАВЧАЛЬНІ ПРОЕКТИ



1. На уроках з яких предметів ви виконували навчальні проекти?
2. Які етапи реалізації проектів?
3. У яких формах може бути подано результати виконання проектів?

ЕТАПИ РЕАЛІЗАЦІЇ ПРОЕКТІВ

Як ви знаєте, проектна діяльність учнів полягає в самостійному пошуку та опрацюванні відомостей, потрібних для розв'язування деякої проблемної задачі, підготовці звіту про виконану роботу та захисті результатів пошукової роботи.

Згадаємо етапи реалізації проектів:

- аналіз формулювання задачі (визначення мети та завдань проекту);
- створення інформаційної моделі (розробка плану виконання проекту);
- пошук потрібних матеріалів і відомостей;
- вибір засобів опрацювання даних;
- опрацювання матеріалів, аналіз результатів;
- вибір засобів подання результатів навчального проекту;
- оформлення результатів проекту;
- захист проекту.

Над виконанням завдань проекту можна працювати поодиноці, але час-то навчальні проекти учні виконують групами.

Для створення інформаційних моделей в ході проекту, пошуку та опрацювання відомостей, підготовки звіту за результатами роботи, спілкування учасників може бути використано різні інформаційні технології.

Результати, отримані різними групами учнів, зручно розміщувати на сторінках сайта або блогу в Інтернеті.

Навчальні проекти ви виконували під час вивчення різних шкільних предметів. Проекти можуть стосуватися одного навчального предмета або

кількох. У цьому році вам пропонується виконати індивідуальні та групові навчальні проекти з дослідження предметної області навчального курсу «Інформатика».

НАВЧАЛЬНИЙ ПРОЕКТ З ІНФОРМАТИКИ

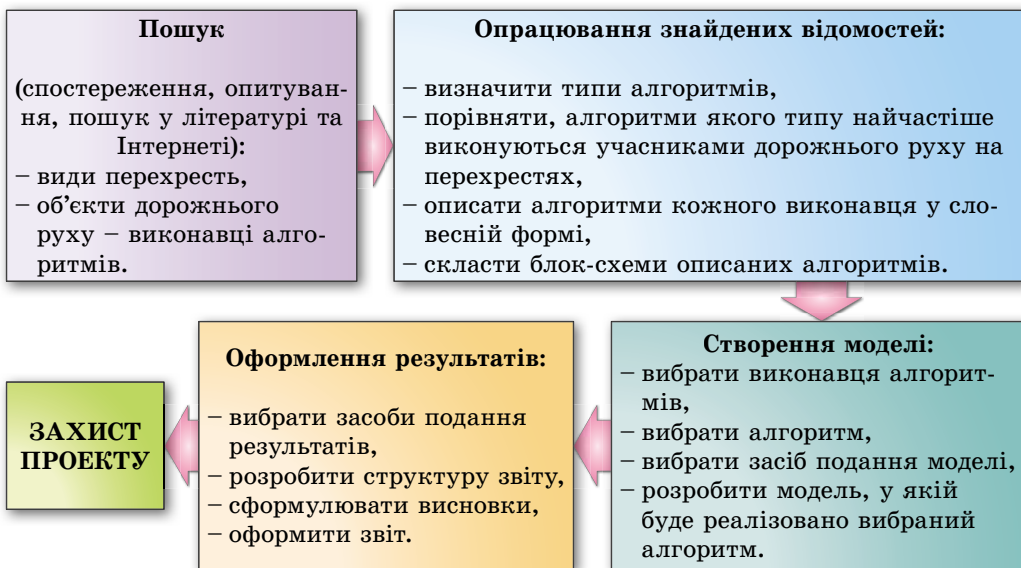
Розглянемо етапи роботи над проектом на прикладі проекту «Алгоритми навколо нас». Проект стосується розділу 6 «Алгоритми роботи з об'єктами та величинами».

Мета проекту: працюючи у групах, знайти приклади алгоритмів у власному оточенні та створити комп'ютерні моделі окремих виконавців алгоритмів.

Одна з груп може розглянути тему «Перехрестя алгоритмів». Для цієї групи можуть бути визначені такі **завдання**:

- визначити види перехресть, які зустрічаються на дорогах, та об'єкти на перехрестях, які є виконавцями алгоритмів;
- описати алгоритми, які виконує кожен з виконавців;
- вибрати одного з виконавців та його алгоритм, розробити комп'ютерну модель реалізації вибраного алгоритму;
- оформити результати роботи у проекті.

На основі завдань розробимо **план виконання проекту**. У плані конкретизуємо дії, пов'язані з виконанням кожного завдання (мал. 8.5).



Мал. 8.5. План виконання проекту

Пошук потрібних відомостей може здійснюватися різними методами: спостереження, опитування, пошук в Інтернеті та у друкованих джерелах. У нашому проекті результатом пошуку будуть відомості (описи, зображення, схеми) про види перехресть та учасників дорожнього руху на перехрестях, алгоритми їхніх дій.

Для обміну знайденими матеріалами між учасниками проекту та керівником може бути використано засоби електронної пошти.



Результатом **опрацювання знайдених відомостей** може стати схема або таблиця, у якій структуровано всі знайдені матеріали. Наприклад, у текстовому процесорі може бути створено таку таблицю (табл. 8.1).

Таблиця 8.1

Перехрестя алгоритмів

Перехрестя Учасники руху	Перехрестя з регульованим переходом			Перехрестя з нерегульованим переходом
	Рух регулюється світлофором	Рух регулюється семафором зі шлагбаумом	Рух регулюється світлофором та семафором зі шлагбаумом	
Пішохід	Алгоритм з розгалуженням	Алгоритм з розгалуженням	Алгоритм з розгалуженням	Алгоритм з розгалуженням
Водій авто-транспорту	Алгоритм з розгалуженням	Алгоритм з розгалуженням	Алгоритм з розгалуженням	Алгоритм з розгалуженням
Водій залізничного транспорту	—	Алгоритм з розгалуженням	Алгоритм з розгалуженням	—
Світлофор	Алгоритм з повторенням	—	Алгоритм з повторенням	—
Семафор	—	Алгоритм з розгалуженням та повторенням	Алгоритм з розгалуженням та повторенням	—
Шлагбаум	—	Алгоритм з розгалуженням	Алгоритм з розгалуженням	—

Таблиця дає підстави вважати, що на різних перехрестях учасники дорожнього руху найчастіше виконують алгоритми з розгалуженням. Це можна пояснити вимогами Правил дорожнього руху, які гарантують безпеку для кожного учасника руху.

Кожен учасник проекту може вибрати окремого виконавця алгоритмів на перехресті, описати й змодельовати алгоритм його діяльності. Розглянемо детальніше дії виконавця *світлофор*.

Алгоритм дії одного з видів світлофорів – це алгоритм з повторенням, який починає виконуватися після ввімкнення:

1. Увімкнути лампу червоного кольору на 30 секунд.
2. Вимкнути лампу червоного кольору.
3. Увімкнути лампу жовтого кольору на 15 секунд.
4. Вимкнути лампу жовтого кольору.
5. Увімкнути лампу зеленого кольору на 30 секунд.
6. Вимкнути лампу зеленого кольору.
7. Увімкнути лампу жовтого кольору на 15 секунд.



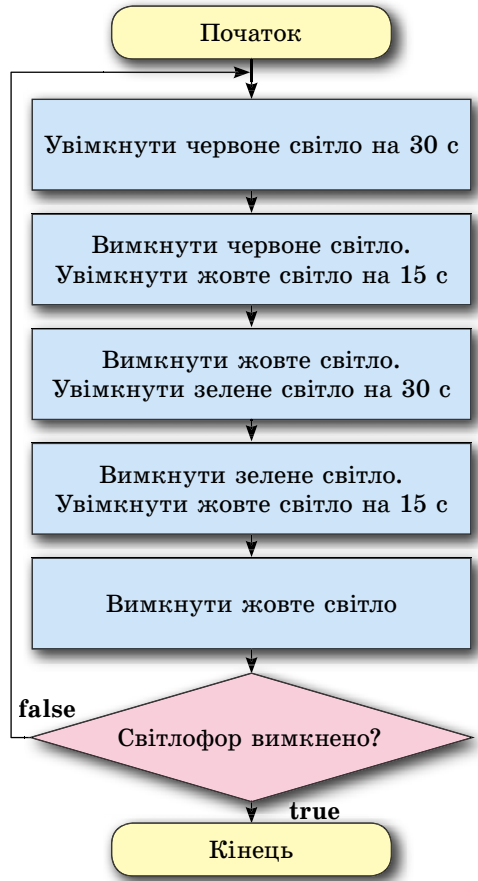
- 8. Вимкнути лампу жовтого кольору.
- 9. Повторювати дії 1–8, доки світлофор увімкнено.

Блок-схему цього алгоритму наведено на малюнку 8.6.

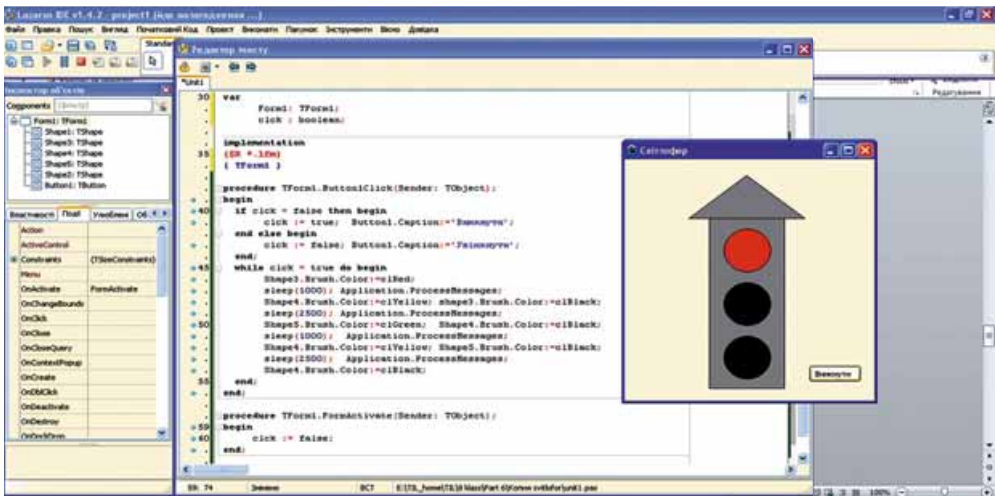
Комп'ютерну модель світлофора створимо в середовищі Lazarus. Текст програми та вікно виконання проекту моделювання роботи світлофора наведені на малюнку 8.7.

Для подання результатів роботи у проєкті можна розробити презентацію або текстовий документ, розмістити отримані матеріали на сайті чи блогі або вибрати інші засоби. У презентації важливо відобразити весь хід вашої роботи: завдання, що були поставлені, методи пошуку та опрацювання даних, які ви використали, отримані результати та висновки. Приклад презентації з результатами наведено на малюнку 8.8.

Завершальним етапом роботи у проєкті є його захист. Під час захисту учасники проєкту звітують про результати реалізації проєкту. Планом виступу на захисті буде презентація, яку ви створили. Важливо не тільки подати опис проєкту та хід його виконання, але й відобразити участь кожного із членів групи у проєкті. Отже, у виступі мають бути названі:



Мал. 8.6. Алгоритм роботи світлофора



Мал. 8.7. Текст програми та вікно виконання проєкту моделювання роботи світлофора



- назва і цілі проекту;
- засоби, методи і шляхи, що були обрані для реалізації проекту;
- з якими труднощами і проблемами зустрілися учасники проекту в ході його реалізації;
- результати проекту;
- висновки, зроблені на основі результатів проекту;
- наскільки вдалося досягти мети проекту.

ПЕРЕХРЕСТЯ АЛГОРИТМІВ

Виконали учні 8 класу

Наше завдання

- Визначити види перехрестя, які зустрічаються на дорогах, та обстежити перехрестя, які є виконавцями алгоритмів.
- Опинити алгоритми, які виконують кожен з виконавців.
- Порівняти алгоритми якого типу найчастіше використовують учасники дорожнього руху на перехрестях.
- Вибрати виконавця та один з алгоритмів, розробити комп'ютерну модель реалізації вибраного алгоритму.

Хід виконання

- У нашому населеному пункті ми побачили перехрестя, на яких рух:
 - не регулюється
 - регулюється світлофором
- Ми дізналися, що існують перехрестя, на яких рух регулюється:
 - світлофором з сигналізацією
 - регулювальником

Виконавці алгоритмів на перехресті

- Виконавцями алгоритмів є люди:
 - пішоходи
 - водії автомобілів
 - машинисти потягів
 - регулювальники руху
- Виконавцями алгоритмів є автоматичні пристрої:
 - світлофори
 - сигналізація

Види алгоритмів на перехресті

Тип перехрестя	Види алгоритмів	Види виконавців
Перехрестя з сигналізацією	Регулювальник	Регулювальник
Перехрестя з сигналізацією та світлофором	Регулювальник	Регулювальник
Перехрестя з світлофором	Світлофор	Світлофор
Перехрестя з регулювальником	Регулювальник	Регулювальник
Перехрестя з регулювальником та світлофором	Регулювальник	Регулювальник
Перехрестя з регулювальником та світлофором та сигналізацією	Регулювальник	Регулювальник

Алгоритм роботи світлофора

1. Увімкнути лампу червоного кольору на 30 секунд
2. Увімкнути лампу червоного кольору на 18 секунд
3. Увімкнути лампу жовтого кольору на 15 секунд
4. Увімкнути лампу жовтого кольору
5. Увімкнути лампу зеленого кольору на 30 секунд
6. Увімкнути лампу зеленого кольору
7. Увімкнути лампу жовтого кольору на 15 секунд
8. Увімкнути лампу жовтого кольору
9. Повторити дії 1-8, доки світлофор увімкнено.

Блок-схема алгоритму

Наша модель роботи світлофора

Функція процедури увімкнення світлофора

```

if color = black then begin
  color := true;
  Button1.Caption := "Ввімкнути";
  and else begin
    color := false;
    Button1.Caption := "Вимкнути";
  end;
while color = true do begin
  ShowForm1.Color := red;
  ShowForm1.Color := yellow;
  ShowForm1.Color := black;
  ShowForm1.Color := red;
  ShowForm1.Color := yellow;
  ShowForm1.Color := black;
  ShowForm1.Color := red;
  ShowForm1.Color := yellow;
  ShowForm1.Color := black;
end;
    
```

Висновки

- На перехрестях виконавцями алгоритмів є люди та автоматичні пристрої
- Більша частина алгоритмів на перехрестях, які використовують:
 - люди – алгоритми з регулюванням
 - автоматичні пристрої – алгоритми з поворотниками
- Виконувати алгоритми на перехрестях потрібно, щоб забезпечити безпеку руху
- Алгоритми роботи світлофора є алгоритмом з поворотниками

Використані джерела:
<http://ukrainska.wikipedia.org/wiki/Світлофор>
<http://ukrainska.wikipedia.org/wiki/Світлофор>

Мал. 8.8. Приклад презентації з результатами роботи у проекті



Дайте відповіді на запитання

- 1°. Які етапи реалізації проектів?
- 2°. Для чого складають план проекту?
- 3°. У яких джерелах та якими методами можна виконувати збір відомостей для проекту?
- 4°. Що може бути результатом пошуку матеріалів та відомостей для проекту?
- 5°. У яких формах може бути подано результати виконання проекту?
- 6*. Що впливає на добір комп'ютерних програм для опрацювання матеріалів проекту?
- 7°. Що зазвичай відображається у презентації за результатами участі у проекті?



Виконайте завдання

Розробіть план навчального проекту, виберіть один з об'єктів дослідження відповідно до мети, засоби опрацювання даних і форму подання результатів та виконайте проект з інформатики:

1. Тема: **Пишаємось, що українські.** Мета: дослідити історію розвитку комп'ютерної техніки в Україні та визначити, які досягнення в цій галузі має Україна та українці.
2. Тема: **Суд над комп'ютером.** Мета: визначити, у чому полягає шкідливий вплив комп'ютера на людину, та підготувати рекомендації для зменшення небезпеки.
3. Тема: **Мій напарник – комп'ютер.** Мета: проаналізувати види та характеристики сучасних комп'ютерів, що є на ринку вашого регіону, та визначити характеристики комп'ютерів, потрібні для виконання завдань у різних галузях людської діяльності.
4. Тема: **Професія – фахівець у галузі інформаційних технологій.** Мета: скласти довідник професій, які можна набути в галузі інформаційних технологій, з'ясувати, які навички повинні опанувати представники цих професій.
5. Тема: **Веб 2.0 у школі.** Мета: з'ясувати сутність поняття Веб 2.0, виконати класифікацію сервісів, що належать до Веб 2.0, та навести приклади сервісів, які можна використати в навчанні.
6. Тема: **Інформаційні технології під мікроскопом.** Мета: узагальнити знання про інформаційні технології опрацювання даних різних видів (графічних, текстових, числових, мультимедійних), скласти схеми з відомостями про інформаційні технології, указавши основні об'єкти опрацювання, галузі застосування, типи програмних засобів, історію розробки.



СЛОВНИЧОК

А

Абсолютне посилання – посилання на клітинку, яке не модифікується під час копіювання формули, с. 230.

Архіватори – програми, що використовуються для виконання операцій над архівами даних, с. 63.

Архівний файл (архів) – файл, що містить у стисненому або нестисненому стані файли і папки, с. 62.

Архівування – процес створення копій даних з використанням спеціальних програм, що можуть використовувати стиснення даних, с. 62.

Архітектура комп'ютера – модель, що описує взаємодію пристроїв та програм для забезпечення інформаційних процесів у комп'ютері, с. 25.

Б

Байт – послідовність із восьми бітів, с. 17.

Біт – цифра 0 або 1 у двійковому коді повідомлення, с. 17.

В

Відеоредактори – програми для редагування відеоданих, с. 138.

Відеостудії – програми, що використовуються в професійних цілях для редагування відеоданих, с. 138.

Відносне посилання – посилання на клітинку, яке модифікується під час копіювання формули, с. 230.

Властивості компонентів:

1. **Name** (англ. *name* – ім'я) – її значення визначає ім'я форми як об'єкта, с. 157.
2. **Caption** (англ. *caption* – заголовок) – її значення визначає текст у рядку заголовка форми, с. 157.
3. **Color** (амер. *color* – колір) – її значення визначає колір фону форми, с. 157.
4. **Height** (англ. *height* – висота) – її значення визначає висоту вікна в пікселях (піксель – це точка графічного зображення на екрані), с. 157.
5. **Width** (англ. *width* – ширина) – її значення визначає ширину вікна в пікселях, с. 157.
6. **Top** (англ. *top* – верх) – її значення визначає відстань (у пікселях) від верхньої межі вікна до верхньої межі екрана, с. 157.
7. **Left** (англ. *left* – лівий) – її значення визначає відстань (у пікселях) від лівої межі вікна до лівої межі екрана, с. 157.
8. **Enabled** (англ. *enabled* – доступний) – її значення визначає, чи доступний об'єкт для операцій над ним. Ця властивість може набувати лише одного з двох значень: **True** (англ. *true* – істина, правда) – об'єкт доступний або **False** (англ. *false* – хиба, неправда) – об'єкт недоступний, с. 171.
9. **Visible** (англ. *visible* – видимий) – її значення визначає, чи видимий даний об'єкт у вікні. Ця властивість може набувати теж лише одного з двох значень: **True** – об'єкт видимий або **False** – об'єкт невидимий, с. 171.
10. **Font** (англ. *font* – шрифт) – її значення визначає значення властивостей шрифту, яким буде виводитися текст на об'єкті. Ця властивість є комплексною, тому що складається з кількох властивостей шрифту: **Color, Height, Name, Size** та інших, с. 171.

Внутрішня пам'ять – складова пам'яті комп'ютера, що призначена для забезпечення роботи процесора. Складається внутрішня пам'ять з оперативної, постійної і кеш-пам'яті, с. 26.

Г

Грабер (англ. *grabber* – той, хто захоплює, хапуга, грабіжник) – див. **рекордер**, с. 133.

Графік – діаграма, яка використовується для відображення деякого процесу в часі, динаміки зміни окремих даних, які не залежать одне від одного, с. 244.

Графічний планшет – пристрій введення графічних даних. Використовується властивість робочої поверхні планшета реагувати на дотик стилуса. Користувач створює малюнок, дотикаючись стилусом до поверхні графічного планшета, с. 33.

Д

Двійкове кодування – це кодування повідомлень з використанням сигналів лише двох видів, с. 17.

Двійковий код повідомлення – це повідомлення, отримане в результаті двійкового кодування початкового повідомлення, с. 17.

Декодування повідомлення – це процес отримання початкового повідомлення із закодованого, с. 8.

Довжина двійкового коду повідомлення (обсяг даних) – це кількість байтів у двійковому коді цього повідомлення, с. 17.

З

Зміст документа – перелік назв структурних частин документа із зазначенням номерів відповідних сторінок, с. 118.

Змінна логічного типу – змінна, яка може набувати одного з двох значень **true** або **false**. Логічний тип змінних позначається **boolean**, с. 196.

Зовнішня пам'ять – складова пам'яті комп'ютера, що призначена для тривалого зберігання даних. До її складу входять пристрої на магнітних та оптичних дисках, флеш-накопичувачі тощо, с. 27.

І

Інсталяція – операція встановлення ПЗ у комп'ютер, с. 54.

К

Кирилиця – алфавіт для групи слов'янських мов, до яких належать українська, російська, білоруська, болгарська та інші мови, с. 11.

Клавіатура – пристрій введення, що призначений для введення символічних даних і команд, с. 32.

Кліп (англ. *clip* – стискувати, обрізати) – невеликий за розміром (тривалістю) кіно-або відеофільм, с. 138.

Кодеки – програми, що містять алгоритми кодування і декодування мультимедійних даних, с. 132.

Кодування повідомлення – це процес заміни однієї послідовності сигналів, якою подане повідомлення, іншою послідовністю сигналів, с. 7.

Колонтитули – службові повідомлення, які розміщують на полях сторінки і повторюються на інших сторінках документа, с. 109.

Команда неповного розгалуження в мові програмування **Object Pascal** має такий вигляд (с. 201):

```
If <логічний вираз>
Then begin
    <послідовність команд>
end;
```




Команда повного розгалуження в мові програмування **Object Pascal** має такий вигляд (с. 201):

```
If <логічний вираз>
Then begin
    <послідовність команд 1>
end
Else begin
    <послідовність команд 2>
end;
```

Команда циклу з лічильником в мові програмування **Object Pascal** має такий вигляд (с. 213):

```
for <ім'я змінної> := <вираз1> to <вираз2> do
begin
    <команди тіла циклу>
end;
```

Команда циклу з передумовою в мові програмування **Object Pascal** має такий вигляд (с. 217):

```
While <логічний вираз>
Then begin
    <команди тіла циклу>
end;
```

Коментар – текст, який ігнорується компілятором і включається до тексту проекту з метою полегшення його розуміння і пошуку логічних помилок. Коментар потрібно взяти у фігурні дужки {}, або в круглі дужки із зірочкою (* *), або відокремити від команд двома похилими рисками //, с. 189.

Комерційна ліцензія – ліцензія на програмне забезпечення, що передбачає оплату користувачем вартості використання програми на одному чи кількох зазначених у ліцензії комп'ютерах. Розрізняють «коробкову», OEM та корпоративну версії комерційної ліцензії, с. 52.

Комп'ютерна програма – це алгоритм опрацювання даних, записаний спеціальною мовою та призначений для виконання комп'ютером. У процесі своєї роботи програма опрацьовує дані, с. 154.

Конвертор – програма для зміни форматів файлів, с. 135.

Кратні одиниці вимірювання довжини двійкового коду повідомлення:

1 Кбайт (кілобайт) = 2^{10} байтів = 1024 байти

1 Мбайт (мегабайт) = 2^{10} Кбайт = 2^{20} байтів = 1 048 576 байтів

1 Гбайт (гігабайт) = 2^{10} Мбайт = 2^{20} Кбайт = 2^{30} байтів

1 Тбайт (терабайт) = 2^{10} Гбайт = 2^{20} Мбайт = 2^{30} Кбайт = 2^{40} байтів, с. 18.

Л

Лінійчата діаграма – діаграма, яка використовується для відображення кількох рядів даних, які не залежать один від одного, с. 242.

Лінія – графічний об'єкт, який характеризується властивостями: форма, товщина, тип, шаблон, колір, с. 91.

Ліцензія вільного використання програм або **freeware** (англ. *freeware* – вільний товар) – ліцензія, що передбачає вільне використання програм без виплат винагороди автору, але не передбачає можливості внесення змін у програму, с. 52.

Ліцензія з відкритим кодом або **free** (англ. *free* – вільний) **software** чи **libre** (ісп. *libre* – вільний) **software** – ліцензія на програмне забезпечення, що передбачає не тільки їх безкоштовне використання, але і право на модифікацію, внесення змін у програми, с. 53.

Логічна функція (IF, AND, OR, NOT) – функція, результат якої дорівнює True або False, с. 238.

Логічні вирази – це математичні вирази, які містять знаки порівняння, с. 237.

М

Маркери графічних об'єктів – спеціальні позначки у вигляді маленьких кругів, квадратів, ромбів, які розташовані навколо об'єкта і призначені для зміни об'єкта, с. 94.

Мішане посилання – посилання на клітинку, у якому під час копіювання формули модифікується або номер стовпця, або номер рядка, с. 231.

Монітор – основний пристрій для виведення даних у персональних комп'ютерах. Монітори поділяють за системою створення зображення на **монітори на рідких кристалах** або **LCD-монітори** (англ. *Liquid Crystal Display* – рідкокристалічний дисплей), **плазмові**, **OLED** (англ. *Organic Light Emitting Diode* – органічний світлодіод), на **електронних чорнилах** – **e-ink** (англ. *Electronic ink* – електронні чорнила) тощо, с. 34.

Музичні редактори – програми для редагування звукових (музичних) даних, с. 138.

Музичні студії – програми, що використовуються в професійних цілях для редагування звукових (музичних) даних, с. 138.

О

Опрацювання даних – це процес отримання нових даних з наявних, с. 6.

П

Плотер – пристрій виведення даних на папір, плівку або інші матеріали, що мають розмір A2 і більше, с. 35.

Подкаст (англ. *ipod* – медіаплеєр корпорації Apple; *broad casting* – ефірне мовлення) – мультимедійний файл, що розповсюджується з використанням Інтернету і, як правило, містить думки авторів з різноманітних питань: політичних, економічних, технічних, літературних, музичних тощо, с. 149.

Подкастер – той, хто створює подкасти, с. 149.

Посилання – спеціальний запис у документі, який вказує на якийсь об'єкт того або іншого документа, с. 118.

Прикладне програмне забезпечення – сукупність програм, призначених для реалізації конкретних задач опрацювання даних, які користувач розв'язує в ході своєї діяльності, с. 51.

Предметний (алфавітний) покажчик – алфавітний список термінів, які використані в документі із зазначенням номерів сторінок, де вони трапляються, с. 118.

Принтер – пристрій виведення даних на папір, плівку або інші матеріали, що мають розмір, як правило, не більше ніж A3, с. 35.

Пробна або trial (англ. *trial* – випробування) **ліцензія** – вид ліцензії, яка надається для пробного використання програми протягом певного часу (зазвичай близько місяця) або на певну кількість запусків програм, с. 52.

Програмне забезпечення (ПЗ) – сукупність програм для комп'ютерів, с. 51.

Проміжні підсумки – спосіб виконання обчислень в електронній таблиці з попереднім об'єднанням повторюваних даних у групи, с. 259.

Процесор – пристрій, який забезпечує опрацювання даних у комп'ютері. Він містить **пристрій керування**, який забезпечує виконання команд комп'ютерної програми, та **арифметично-логічний пристрій**, який і здійснює операції над даними, с. 25.



Р

Рекордер (англ. *record* – запис) – програма для запису (захоплення) аудіо- і відеоданих, с. 133.

С

Сенсорний екран – пристрій введення і виведення даних. Для введення даних використовується властивість поверхні екрана реагувати на дотик пальця або стилуса, с. 33.

Системи програмування – сукупність програм, призначених для створення нових програм з використанням мов програмування, с. 51.

Системне програмне забезпечення – сукупність програм, призначених для керування роботою складових комп'ютера та обміном даними між ними, діагностування та усунення недоліків у роботі комп'ютера, автоматизації процесу опрацювання даних, організації обміну даними між користувачем і комп'ютером. До складу системного програмного забезпечення належать операційні системи та різноманітні службові програми, с. 51.

Сканер – пристрій для введення графічних даних. За принципом дії виділяють планшетні, з протяжним механізмом, ручні та скан-камери, с. 33.

Службове програмне забезпечення – це програми, що призначені для діагностування апаратної і програмної складових комп'ютера, розширення можливостей ОС, с. 51.

Сортування – змінення порядку розташування даних у рядках або стовпцях таблиці за деякими ознаками, с. 248.

Списки – особливий вид форматування абзаців текстового документа, які використовують для переліку об'єктів, опису порядку дій тощо, с. 73.

Стиль – набір значень властивостей певного типу об'єктів, який має власне ім'я, с. 111.

Стиснення даних – це процес перекодування даних з метою зменшення розмірів файлів, с. 60.

Стратегія пошуку інформаційних матеріалів – визначення мети та завдань пошуку, джерел, у яких буде здійснюватися пошук, ключових слів, критеріїв оцінювання знайдених матеріалів, с. 267.

Структура документа – ієрархічна схема розміщення складових частин документа, с. 108.

Т

Таблиця кодів символів – таблиця, у якій для кожного символу, що може бути використаний у текстовому повідомленні, ставиться у відповідність деяке число, с. 10.

Таблиця символів – службова програма операційної системи **Windows**, що призначена для перегляду, пошуку та вставлення в документи символів із різних шрифтів, поданих з використанням різних таблиць кодів символів, с. 13.

Табуляція – засіб, який дає змогу розміщувати об'єкти в рядку текстового документа в строго визначених місцях (позиціях табуляції), с. 79.

Тачпед (англ. *touch* – дотик, *pad* – площадка, майданчик, подушечка) – пристрій введення, що використовується в ноутбуках і нетбуках, для керування роботою комп'ютера, с. 32.

ТВ-тюнер – пристрій, що забезпечує перегляд телевізійних передач на комп'ютері, с. 34.

Тестовий набір вхідних даних – набір значень вхідних даних, для яких можна заздалегідь визначити правильний результат виконання проекту, після чого порівняти його з результатом виконання проекту для цих значень вхідних даних, с. 188.

Тестування проекту – процес пошуку логічних помилок у проекті з використанням тестових наборів вхідних даних, с. 188.

Тип змінної визначає:

- яких значень може набувати ця змінна;
- які операції над нею можна виконувати;
- який обсяг оперативної пам'яті буде виділено для зберігання значення цієї змінної, с. 184.

Точка зупинки – рядок у проекті, перед виконанням якого виконання проекту припиняється, с. 190.

Точкова діаграма – діаграма, яка використовується для відображення взаємозв'язку між значеннями двох змінних, коли значення однієї величини залежить від значення іншої, с. 243.

У

Умове форматування – вид форматування даних, який автоматично змінює формат клітинки електронної таблиці на заданий, якщо виконується задана умова, с. 260.

Ф

Фігура – графічний примітив, який утворюється замкненими лініями, с. 91.

Фільтрування – це вибір значень у клітинках електронної таблиці, що відповідають певним умовам, с. 251.

Форма – об'єкт, на якому можна розмістити різні компоненти (елементи керування), зокрема кнопки, поля, написи, меню, смуги прокручування та ін. Під час виконання проекту форма відкривається як вікно прикладної програми, на якому будуть відображатися всі об'єкти, розміщені на формі, с. 157.

Ш

Шаблон – відформатований певним чином документ-заготовка, який зберігається в окремому файлі і використовується як зразок для створення нових документів певного типу, с. 121.

Ю

Юнікод (англ. *Unicode* – уніфіковане кодування) – таблиця кодів символів, що складається із 17 наборів по 65 536 значень кодів у кожному та дає можливість закодувати 1 114 112 різних символів с. 11.

А

ABS – математична функція для обчислення модуля числа, с. 237.

ASCII – таблиця кодів символів, що містить коди 128 символів (літер англійського алфавіту, цифр, розділових знаків, спеціальних символів), с. 10.

F

FLAC (англ. *Free Lossless Audio Codec* – вільний аудіокодек без втрат) – формат файлів, що використовує стиснення аудіоданих без втрат і розповсюджується за ліцензією вільного програмного забезпечення, с. 131.

Flash відео (розширення імені файлу flv) – формат файлів, що використовуються зазвичай для розповсюдження відео в Інтернеті. Характеризується високим ступенем стиснення даних, с. 132.

К

KO18-U – таблиця кодів символів, що включає без змін усі коди таблиці ASCII, а також містить коди літер кирилиці, с. 11.

**М**

MIDI (англ. *Musical Instrument Digital Interface* – цифровий інтерфейс музичних інструментів) – формат файлів, що містять команди для відтворення звуку спеціальним пристроєм або програмою – синтезатором, с. 131.

MP3, точніше **MPEG-1/2/2.5Layer 3** (англ. *Motion Picture Experts Group* – експертна група з питань рухомих зображень, **Layer 3** – третій рівень) – формат аудіофайлів із частковою втратою даних, що використовується для передавання звукових даних у глобальних мережах, у файлообмінних системах тощо, с. 131.

MP4 або **MPEG-4 Part 14** – файли цього формату зазвичай використовуються для цифрового телебачення, зберігання відеофільмів тощо, с. 131.

Q

QuickTime (англ. *Quick Time* – швидкий час) – формат файлів, що базується на технології збереження і відтворення відео- та звукових даних, яку було розроблено корпорацією Apple, с. 132.

P

PI – математична функція для обчислення наближеного значення числа π , с. 237.

POWER – математична функція для обчислення числа, піднесеного до степеня, с. 237.

R

ROUND – математична функція для обчислення округлення числа, с. 237.

S

ShowMessage ('<текст>'); – команда для відкриття вікна повідомлень і виведення на ньому текстового повідомлення, с. 167.

SQRT – математична функція для обчислення арифметичного квадратного кореня із числа, с. 237.

Y

YouTube (англ. *You Tube* – твоя труба (слухавка)) – сервіс для розміщення відеофайлів в Інтернеті, с. 146.

W

WAV або **WAVE** (англ. *waveform audio format* – аудіоформат типу хвиля) – формат звукових файлів, що використовується як основний для нестиснутих аудіоданих в операційних системах сімейства **Windows**, с. 131.

Windows-1251 – таблиця кодів символів, що включає без змін усі коди таблиці **ASCII**, а також містить коди літер кирилиці. Є стандартом для кодування літер кирилиці в операційних системах **Windows**, с. 11.

WMA/WMV (англ. *Windows Media Audio/Video*) – формат файлів для зберігання і відтворення аудіо/відеоданих, розроблений корпорацією Microsoft і стандартно використовується в ОС **Windows**, с. 132.