

ВИДАВНИЦТВО
РАНОК



Інтернет-
підтримка

О. О. Бондаренко, В. В. Ластовецький,
О. П. Пилипчук, Є. А. Шестопапов

8

ІНФОРМАТИКА



Класифікація персонал

НАСТІЛЬНІ



Десктоп — системний блок в окремому корпусі



Неттоп — портативний системний блок може кріпитися до монітора



Моноблок — системний блок вбудовано в корпус монітора

ПОРТАТИВНІ



Ноутбук — усі складові в одному корпусі



Нетбук — спрощений і зменшений ноутбук



Планшет — плаский ПК без клавіатури із сенсорним екраном

ьних комп'ютерів (ПК)

КИШЕНЬКОВІ



Комунікатор —
ПК із функцією
мобільного телефону



Кишеньковий —
мініатюрний ПК
з клавіатурою або без неї



Смартфон — мобільний
телефон із функціями
комп'ютера



Планшетний ноутбук —
ноутбук із сенсорним
екраном



В Україні створено потужні центри суперкомп'ютерних обчислень у Національному технічному університеті «Київський політехнічний інститут» та Інституті кібернетики ім. В. М. Глушкова НАН України.

О. О. Бондаренко, В. В. Ластовецький,
О. П. Пилипчук, Є. А. Шестопапов

ІНФОРМАТИКА

**Підручник для 8 класу
загальноосвітніх навчальних закладів**

Рекомендовано Міністерством освіти і науки України

Харків
Видавництво «Ранок»
2016

УДК [004:37.016](075.3)
ББК 32.973+74.263.2
Б 81

Рекомендовано Міністерством освіти і науки України
(наказ Міністерства освіти і науки України від 10.05.2016 р. № 491)

Видано за рахунок державних коштів. Продаж заборонено

Експерти, які здійснили експертизу даного 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»:

В. Г. Боднар, учитель комунального закладу «Плисківська загальноосвітня школа I–III ступенів» Погребищенської районної ради Вінницької області, учитель-методист;

С. В. Гурсьва, керівник районного методичного центру вчителів інформатики Ленінського району м. Запоріжжя, ст. учитель;

М. В. Липтева, доцент кафедри інформатики Харківського національного педагогічного університету імені Г. С. Сковороди, канд. пед. наук

Бондаренко О. О.

Б 81 Інформатика : підруч. для 8 кл. загальноосвіт. навч. закл. / О. О. Бондаренко, В. В. Ластовецький, О. П. Пилипчук, Є. А. Шестоपालов. — Х. : Вид-во «Ранок», 2016. — 256 с. : іл.

ISBN 978-617-09-2852-8

УДК [004:37.016](075.3)
ББК 32.973+74.263.2

Інтернет-підтримка
Для користування електронними матеріалами до підручника увійдіть на сайт interactive.ranok.com.ua



Служба технічної підтримки:
тел. (057) 719-48-65, (098) 037-54-68
(понеділок–п'ятниця з 10:00 до 18:00)
E-mail: interactive@ranok.com.ua

ISBN 978-617-09-2852-8

- © Бондаренко О. О., Ластовецький В. В., Пилипчук О. П., Шестоपालов Є. А., 2016
- © Хорошенко В. Д., ілюстрації, 2016
- © Юхтман О. С., художнє оформлення, 2016
- © ТОВ Видавництво «Ранок», 2016

ШАНОВНІ ВОСЬМИКЛАСНИКИ ТА ВОСЬМИКЛАСНИЦІ!

Ви тримаєте в руках підручник з інформатики, призначений саме для вас, учнів 8 класу. У 5–7 класах ви вже знайомилися з базовими поняттями інформатики та основами роботи з комп'ютером, навчилися створювати графічні, текстові й мультимедійні документи, працювати з електронними таблицями, шукати в Інтернеті інформаційні матеріали, складати прості алгоритми та створювати за ними програми.

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

Підручник складається з десяти розділів. На початку кожного розділу ви знайдете рубрику *«Повторюємо»*. Вона допоможе вам згадати відомості, які ви вивчали з даної теми у попередніх класах і які будуть необхідні для засвоєння нового матеріалу.

Розділ складається з параграфів і практичних робіт за темами розділу. Кожний параграф містить *теоретичні відомості* за темою уроку, *прикладі практичного застосування* отриманих знань, рубрики *«Питання для самоперевірки»*, *«Вправа»*, *«Комп'ютерне тестування»*.

«Питання для самоперевірки» допоможуть вам з'ясувати, чи зрозуміли ви вивчений матеріал, а також підготуватися до виконання вправ і практичних робіт за комп'ютером.

Рубрика *«Вправа»* складається з теоретичних і практичних завдань. Вправи з позначкою ► містять окремі завдання, вправи з позначкою ►► — завдання із покроковим описом виконання. Виконуючи ці завдання, ви здобудете необхідні практичні навички роботи з комп'ютером.

Оцінити свої знання, вміння та навички вам допоможе рубрика *«Комп'ютерне тестування»*. Завдання цієї рубрики можна виконати в онлайн-режимі на електронному освітньому ресурсі *«Інтерактивне навчання»* (www.interactive.ranok.com.ua).

У підручнику ви знайдете описи всіх *практичних робіт*, передбачених чинною програмою з інформатики. Щоб виконати кожен з них, ви маєте повторити матеріал, вивчений протягом попередніх уроків, — тоді ви зможете успішно застосувати свої знання, виконуючи завдання за комп'ютером.

У тексті підручника використані такі позначки:



Запам'ятайте



Розгляньте приклад



Зверніть увагу



Знайдіть відповідь у підручнику



Знайдіть відповідь в Інтернеті



Виконайте практичне завдання за комп'ютером



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



Виконайте завдання підвищеної складності

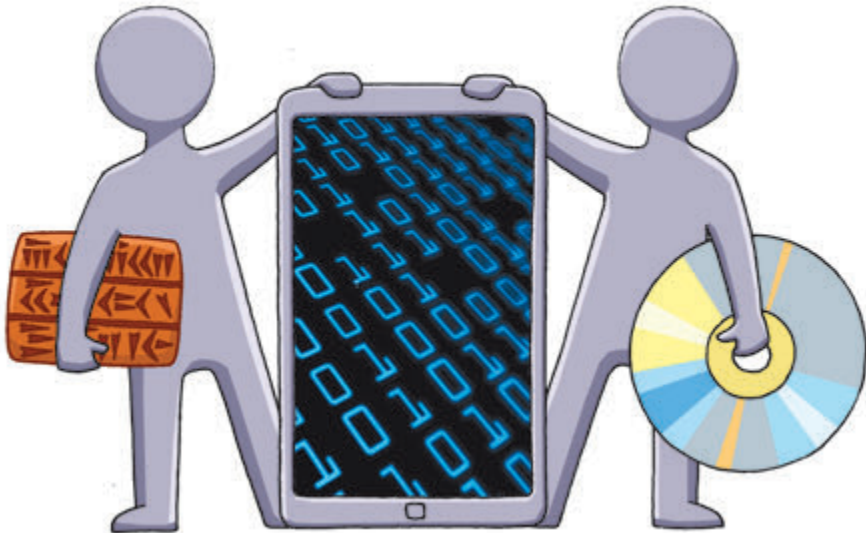


Виконайте завдання в парах

Бажаємо натхнення та успіхів!

РОЗДІЛ 1

КОДУВАННЯ ДАНИХ



§ 1. Кодування та декодування повідомлень

§ 2. Двійкове кодування

Практична робота 1. Розв'язування задач на визначення довжини двійкового коду текстових даних

ПОВТОРЮЄМО



З курсу 5 класу ви знаєте, що відомості про навколишній світ і процеси, які в ньому відбуваються, називають *інформацією*. Людина отримує інформацію з навколишнього середовища за допомогою повідомлень, які сприймає *органами чуття*. Тому за способом сприйняття повідомлення поділяють на зорові, смакові, нюхові, слухові та дотикові. За способом подання повідомлення поділяють на текстові, числові, графічні, звукові, відео та комбіновані.

Різноманітні пристрої теж можуть сприймати повідомлення — за допомогою датчиків або сенсорів.

Дії, які можна виконувати з інформацією, що міститься в повідомленні, називають *інформаційними процесами*. До них відносять зберігання, опрацювання, передавання, отримання, пошук, захист тощо.

1. Що таке інформація?
2. Хто чи що може бути приймачем інформації?
3. Як людина сприймає інформацію? Як пристрої сприймають інформацію?
4. Що таке повідомлення?
5. Які бувають повідомлення за способом сприйняття; за способом подання?
6. Що називають інформаційними процесами?



У цьому розділі ви дізнаєтеся про такі інформаційні процеси, як кодування і декодування.

§ 1. Кодування та декодування повідомлень

Упродовж життя людина сприймає безліч повідомлень, опрацьовує інформацію, зберігає отримані знання або передає їх комусь іншому. Як це відбувається?

Опрацювання даних як інформаційний процес

Ще в прадавні часи люди намагалися зберігати отриману з навколишнього середовища інформацію, зокрема, у вигляді наскельних малюнків (рис. 1.1). Зрозуміло, що передавати збережену таким чином інформацію від однієї людини до іншої було дуже незручно. Проте ми отримали дані про життя наших предків.



Рис. 1.1

Під опрацюванням даних розуміють такий інформаційний процес, у результаті якого ми отримуємо нові дані або дані, подані в інший спосіб.

Кодування та декодування повідомлень

Повідомлення передаються від джерела до приймача по каналах зв'язку.

» Під час розмови по телефону звукові сигнали (голос) перетворюються (кодуються) на електричні сигнали, які по мережі передаються до співрозмовника і знову перетворюються (декоднуються) на звукові сигнали (рис. 1.2).



Рис. 1.2

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



Кодування — це перетворення повідомлення в зручну для передавання, зберігання, опрацювання форму.

Код — це набір символів або сигналів і правил їх використання для кодування повідомлень.

Декодування — це перетворення закодованого повідомлення у форму, прийнятну для приймача.

Після винайдення писемності як способу кодування природної мови найпоширенішим став спосіб подання повідомлень за допомогою символів, наприклад ієрогліфів, літер (букв) тощо.



Мрія

Рис. 1.3

» Окремими ієрогліфами позначалися певні об'єкти або поняття (рис. 1.3), літерами — окремі звуки усного мовлення, із яких склалися слова, речення тощо.

Ті самі повідомлення можна кодувати різними способами. Для кодування повідомлень використовують також цифри, знаки, жести, світлові, звукові та електричні сигнали тощо.

Способи кодування звука [a]:

Aa

Український алфавіт



Семафорна азбука



Азбука Морзе

» Звук [a] можна закодувати і літерою «А», і певним жестом, і у вигляді короткого та довгого звукових сигналів, які на папері позначають крапкою і тире (рис. 1.4).

Ви можете вигадати власний код для передавання текстових повідомлень, наприклад, замінюючи літери іншими символами, малюнками або цифрами. Для декодування повідомлення потрібно знати, як воно закодоване.

Зверніть увагу, що той самий символ має різні значення залежно від галузі його використання.

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

Приймати, декодувати й опрацювати повідомлення можуть людина, створений нею прилад, тварина.

Рис. 1.4

» Навчений службовий собака вправно виконує команди, які закодовані людиною у вигляді жестів.

Сидіти!



Лежати!



Стояти!



Рис. 1.5

Способи кодування безперервно удосконалюються, адже вдало закодовані дані сприяють більш ефективному їх використанню у всіх галузях людської діяльності.

Питання для самоперевірки



1. Як первісні люди кодували повідомлення?
2. Що таке кодування; код; декодування?
3. Як повідомлення передається від джерела до приймача?
4. Як кодуються звуки природної мови?
5. Чи можна закодувати поняття?
6. Покажіть кілька способів кодування звука [а].

Вправа 1



» 1. Розгляньте наскельний малюнок (див. рис. 1.1) і визначте, яка, на ваш погляд, інформація в ньому міститься.



» 2. Знайдіть в Інтернеті азбуку Морзе і закодуйте слово «Інформатика».



» 3. Вигадайте власний код та за його допомогою запишіть будь-яку фразу в зошиті. Обміняйтеся зошитами з однокласником (однокласницею) та за допомогою коду (ключа) декодуйте написане ним (нею).



Комп'ютерне тестування



Виконайте тестове завдання 1 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 2. Двійкове кодування

Комп'ютер — універсальний пристрій для опрацювання даних різних типів. Але як він сприймає, розпізнає, зберігає дані? Розглянемо це на прикладі текстових даних.

Двійкове кодування

У комп'ютері дані подаються електричними або магнітними сигналами, що набувають тільки двох значень: 0 — вимкнено (нема струму, розмагнічено) або 1 — увімкнено (є струм, намагнічено).

Будь-які дані, що опрацьовує комп'ютер: числа, текст, малюнки, схеми, фотографії, музика, відео тощо — кодують за допомогою нуля й одиниці. Такий спосіб кодування називають **двійковим**. Саме двійкове кодування реалізоване в електронних пристроях.

Кодування символів

Для кодування даних у комп'ютері потрібні багатозначні двійкові коди. У таких кодах цифру 0 або 1 називають **бітом** (скор. від англ. *binary digit* — двійкова цифра).

Для кодування символів у комп'ютерах зазвичай використовують 8-бітовий двійковий код.



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

Існує $2^8 = 256$ різних комбінацій із 8 бітів. Цієї кількості варіантів достатньо для кодування великих і малих літер англійського й українського алфавітів, цифр, розділових знаків, а також графічних елементів.

Усі символи, які використовують у текстах, для зручності кодування (декодування) зводять у таблиці двійкових кодів.



Існує таблиця кодів ASCII (*American Standard Code for Information Interchange* — американський стандартний код для обміну

інформацією), яка містить 128 символів, зокрема літери англійського алфавіту.

Для кодування літер інших алфавітів цю таблицю доповнюють до 256 символів. Так, для літер українського алфавіту використовують таблицю кодів KOI8-U (*Код Обміну Інформації 8-бітовий Український*), фрагмент якої подано нижче.

Зазвичай у таблицях кодів поряд або замість двійкового коду записують десятковий. Так, маленька українська літера «а» у таблиці кодів KOI8-U має десятковий код 193, що відповідає двійковому: 11000001.

Фрагмент таблиці кодів KOI8-U

192	11000000	ю	198	11000110	ф	204	11001100	л	210	11010010	р
193	11000001	а	199	11000111	г	205	11001101	м	211	11010011	с
194	11000010	б	200	11001000	х	206	11001110	н	212	11010100	т
195	11000011	ц	201	11001001	и	207	11001111	о	213	11010101	у
196	11000100	д	202	11001010	й	208	11010000	п	214	11010110	ж
197	11000101	е	203	11001011	к	209	11010001	я	215	11010111	в

У сучасних комп'ютерних текстових документах розповсюджене також кодування символів 16-бітовим кодом Юнікод (*Unicode* — уніфіковане кодування), який містить алфавіти практично всіх мов світу.

Одиниці вимірювання довжини двійкового коду

Обсяг даних вимірюється довжиною двійкового коду.

При стандартному 8-бітовому кодуванні тексту кожний символ кодується одним байтом.

Розділові знаки і пробіли також є символами, що мають свої коди.

» Текст Андрій грає у футбол складається з 20 символів. Тому у випадку 8-бітового кодування символів цей текст має обсяг 20 байтів.

На практиці зручніше користуватися більшими одиницями вимірювання: кілобайт (Кбайт), мегабайт (Мбайт), гігабайт (Гбайт), терабайт (Тбайт) та ін. В інформатиці префікси *кіло-*, *мега-*, *гіга-*, *тера-* мають дещо інший зміст, ніж в інших науках:

$$1 \text{ Кбайт} = 2^{10} \text{ байтів} = 1024 \text{ байти} \approx 10^3 \text{ байтів}$$

$$1 \text{ Мбайт} = 2^{10} \text{ Кбайтів} = 1024 \text{ Кбайти} \approx 10^6 \text{ байтів}$$

$$1 \text{ Гбайт} = 2^{10} \text{ Мбайтів} = 1024 \text{ Мбайти} \approx 10^9 \text{ байтів}$$

$$1 \text{ Тбайт} = 2^{10} \text{ Гбайтів} = 1024 \text{ Гбайти} \approx 10^{12} \text{ байтів}$$

На початку ХХІ ст. було введено стандарт іменування одиниць вимірювання обсягів даних: кібібайт, мебібайт, гігібайт і т. д. Але введені назви практично не використовують.

» Нехай на сторінці міститься 56 рядків по 64 символи в кожному рядку. Обчислимо обсяг даних (довжину двійкового коду) на сторінці: $56 \cdot 64 = 3584$ (байти).

Щоб обчислити приблизний обсяг даних у книжці, потрібно кількість символів на одній сторінці помножити на кількість сторінок. Таким чином, обсяг книжки, яка має 256 подібних сторінок: $3584 \cdot 256 = 917\,504$ (байти) $= 896$ (Кбайтів) $= 0,875$ (Мбайта).

Не слід ототожнювати довжину двійкового коду текстового повідомлення з об'ємом інформації, який воно містить.

» Двійковий код літери «Р» має довжину 1 байт; ця ж літера, розташована на дорожньому знаку, несе змістовне повідомлення для водія, що поряд зі знаком облаштоване місце для стоянки.

Тепер ви розумієте, у яких одиницях і яким способом можна вимірювати обсяг текстових даних. З вимірюванням обсягів відео- та аудіоданих, якими оперує комп'ютер, ви ознайомитеся пізніше.

Питання для самоперевірки



1. Як кодуються дані в комп'ютері?
2. Чому в комп'ютері використовується двійкове кодування?
3. Яким чином кодуються літери тексту в комп'ютері?
4. Що таке біт; байт?
5. У яких одиницях вимірюють довжину двійкового коду?
6. Скільки байтів містить 1 кілобайт? 1 мегабайт?

Вправа 2



- 1. Знайдіть в Інтернеті таблицю кодів символів ASCII. Користуючись цією таблицею, розкодуйте повідомлення:
01001000 01100101 01101100 01101100 01101111 00100001.



- 2. Визначте, вміст скількох книжок (див. приклад на с. 12) можна зберегти в електронному вигляді на DVD-диску обсягом 4,7 Гбайта.



- 3. Визначте приблизний обсяг даних на сторінці підручника.



Комп'ютерне тестування



Виконайте тестове завдання 2 з автоматичною перевіркою на сайті interactive.ranok.com.ua.



Практична робота 1

Розв'язування задач на визначення довжини двійкового коду текстових даних

Завдання: обчислити довжину двійкового коду текстових даних.

Обладнання: ПК з операційною системою Windows; підручник; зошит.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- 1. Запишіть в зошиті своє ім'я та прізвище. Визначте довжину двійкового коду записаного тексту при однобайтовому кодуванні символів.
- 2. Запустіть на виконання програму Блокнот. Наберіть своє ім'я та прізвище через пробіл і збережіть файл із назвою Практична робота 1 у відповідній папці. За допомогою контекстного

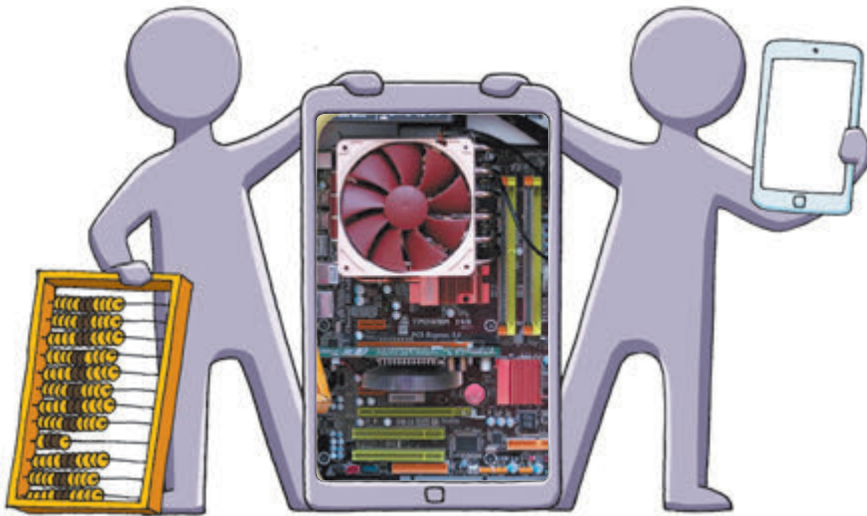
меню файла (команда Властивості) з'ясуйте розмір файла в байтах і порівняйте з даними, отриманими в п. 1.

- ▶ 3. За допомогою фрагмента таблиці кодів KOI8-U на с. 11 закодуйте слово «байт» і запишіть його в зошиті за допомогою двійкових і десяткових кодів.
- ▶ 4. Використовуючи програму Калькулятор, визначте приблизний обсяг пам'яті, необхідний для зберігання однієї сторінки вашого підручника з текстом (без малюнків) і всього підручника в електронному вигляді.
- ▶ 5. Обчисліть, скільки подібних підручників може поміститися на флеш-носії обсягом 8 Гбайтів.
- ▶ 6. Усі розрахунки, отримані у пп. 1–5, запишіть у текстовий документ Практична робота 1 і збережіть файл. Завершіть роботу за комп'ютером.

Зробіть висновок: як визначати довжину двійкового коду текстових даних.

РОЗДІЛ 2

АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРА



§ 3. Історія опрацювання інформаційних об'єктів

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

Практична робота 2. Конфігурація комп'ютера під потребу

§ 5. Програмне забезпечення комп'ютера

Практична робота 3. Архівування та розархівування даних

ПОВТОРЮЄМО



Ви вже знаєте, що основними складовими настільного комп'ютера є *системний блок*, *монітор*, *миша* та *клавіатура*. До комп'ютера також можуть приєднуватися й інші пристрої.

Усі складові комп'ютера поділяють на пристрої: *введення* — клавіатура, миша, мікрофон, сканер, камера тощо; *виведення* — монітор, принтер, навушники, гучномовці тощо; *опрацювання* — процесори; *зберігання* — внутрішня та зовнішня пам'ять комп'ютера (жорсткі магнітні диски, CD- і DVD-диски, флеш-пам'ять тощо).

Ви знаєте, що комп'ютер не працюватиме без програм. *Програма* — це послідовність команд, призначених для виконання комп'ютером. Найважливішим комплексом програм є операційна система. Саме вона забезпечує роботу і взаємодію всіх складових комп'ютера, а також комп'ютера і користувача.

1. Назвіть основні складові комп'ютера.
2. Які є пристрої зберігання інформації?
3. Які є пристрої введення інформації?
4. Назвіть пристрої виведення інформації.
5. Що таке комп'ютерна програма?
6. Яке призначення операційної системи?



У цьому розділі ви детальніше ознайомитеся з апаратним і програмним забезпеченням комп'ютера, а також зі способами взаємодії комп'ютера і користувача.

§ 3. Історія опрацювання інформаційних об'єктів

Історія людства нерозривно пов'язана з інформаційними процесами. Для полегшення опрацювання інформації (зокрема, таких інформаційних об'єктів, як звук, зображення, тексти, числа) люди здавна розробляли різноманітні засоби.

Етапи розвитку обчислювальних засобів

Розрізняють кілька етапів розвитку обчислювальних засобів.

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

На **механічному етапі** з розвитком промисловості й торгівлі було створено низку механічних пристроїв для обчислень. Біля витоків механічної обчислювальної техніки стояли відомі науковці, інженери і дослідники (рис. 3.1).



Рис. 3.1

Перший відомий ескіз механічного обчислювального пристрою близько 1500 року виконав італійський винахідник, інженер і художник Леонардо да Вінчі. Перший діючий пристрій для додавання та віднімання чисел («паскаліну») розробив і виготовив у 1642 році Блез Паскаль — згодом відомий французький учений. У 1673 році німецький математик і фізик Готфрід Вільгельм Лейбніц створив механічний калькулятор на основі двійкової системи числення, на якій ґрунтується робота сучасних комп'ютерів.

У 1833 році англійський математик Чарльз Беббідж розробив проєкт програмованої обчислювальної машини. Опис та програму для неї склала його учениця Ада Лавлейс — її й вважають першим у світі програмістом.

Значний внесок у подальший розвиток обчислювальних пристроїв зробили відомі вчені, інженери, дослідники з різних країн світу.

» У 1938 році німецький інженер Конрад Цузе побудував першу електромеханічну програмовану цифрову машину.

На електронному етапі були створені електронно-обчислювальні машини (комп'ютери), розвиток яких триває і в наш час.

Основні етапи розвитку комп'ютерів

Розглянемо історію розвитку комп'ютерів як універсального засобу для роботи з різними інформаційними об'єктами.

За елементною базою комп'ютери поділяють на кілька поколінь.



Рис. 3.2

Комп'ютери першого покоління почали виготовляти в 40-х роках ХХ ст. з використанням *електронних ламп* (рис. 3.2). Швидкодія комп'ютерів не перевищувала 20 тис. операцій за секунду.



Рис. 3.3

» Перший комп'ютер під назвою «ENIAC» було створено в США в 1945 році під керівництвом Джона Моучлі і Джона Преспера Еккерта. Комп'ютер масою 30 т виступав до 5 тис. операцій за секунду і займав цілу будівлю з кондиціонерами для охолодження.

Основоположником обчислювальної техніки в СРСР був академік Сергій Олексійович Лебедев (рис. 3.3). Під його керівництвом у Київському електротехнічному інституті у 1951 році було створено перший у СРСР і Європі комп'ютер з назвою «МЭСМ» (рос. — «*малая электронная счетная машина*»).



Рис. 3.4

Комп'ютери другого покоління з'явилися у 50-ті роки ХХ ст. з винайденням *транзисторів* (рис. 3.4). Розміри комп'ютерів зменшились у десятки разів, а їх швидкодія досягла 100 тис. операцій за секунду.

» В Інституті кібернетики Академії наук України у 60-х роках ХХ ст. було створено серію машин для інженерних розрахунків (скорочено МІР). Колектив учених очолював Віктор Михайлович Глушков (рис. 3.5).

Комп'ютери третього покоління з'явилися у 70-ті роки ХХ ст. з появою *інтегральних схем* (рис. 3.6). У таких схемах на одній кремнієвій кристалічній пластині площею до 1 см² розміщувалися тисячі елементів. Розміри комп'ютерів ще зменшилися (до кількох шаф), а їх швидкодія досягла мільйонів операцій за секунду.

Комп'ютери четвертого покоління виникли завдяки розробці *великих інтегральних схем* (рис. 3.7). Такі схеми на кристалічній пластині площею близько 1 см² містять мільйони елементів. Швидкодія комп'ютерів досягла сотень мільйонів операцій за секунду. Розміри комп'ютера знову зменшилися, він став настільним.

» Один із перших персональних комп'ютерів було створено у США в 1976 році засновниками фірми Apple Стивом Джобсом і Стивом Возняком (рис. 3.8).

Зменшення розмірів та збільшення продуктивності комп'ютерів сприяло їх широкому впровадженню у всі сфери життя. На початок 2016 року кількість комп'ютерів у світі сягнула двох мільярдів і продовжує стрімко зростати.



Рис. 3.5



Рис. 3.6

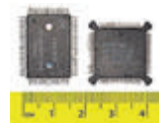


Рис. 3.7



Рис. 3.8

Види сучасних комп'ютерів

Сучасні комп'ютери розрізняють за їх обчислювальними можливостями.

- **Суперкомп'ютери** — багатопроцесорні системи для здійснення складних обчислень.

» В Україні створено потужні центри суперкомп'ютерних обчислень у Національному технічному університеті «Київський політехнічний інститут» та Інституті кібернетики ім. В. М. Глушкова НАН України.

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

- **Мікрокомп'ютери** призначені для реалізації простих задач керування, наприклад, побутовою технікою.
- **Персональні комп'ютери (ПК)** призначені для використання однією людиною, тобто для персонального користування. Персональні комп'ютери розрізняють за їх розмірами (див. форзац 1, рис. 3.9).

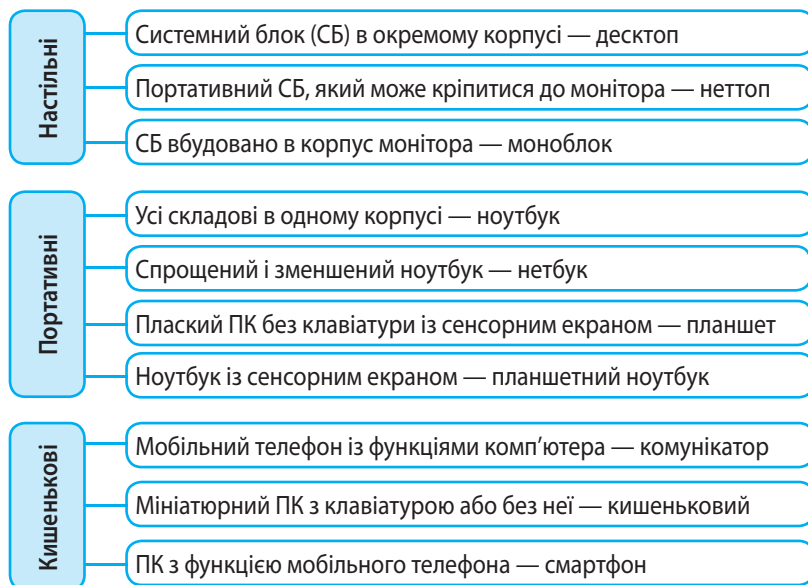


Рис. 3.9



Рис. 3.10

Виробництво комп'ютерів постійно удосконалюється, розширюється сфера застосування. Набувають поширення нові види комп'ютерів: роботизовані іграшки, роботи — помічники по господарству тощо (рис. 3.10).

Питання для самоперевірки



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

Вправа 3



1. Знайдіть у тексті параграфа дані про швидкодію комп'ютерів першого покоління і сучасного комп'ютера та порівняйте їх.



2. Запустіть текстовий процесор і створіть новий документ. Знайдіть в Інтернеті і скопіюйте в документ інформацію про внесок у створення перших комп'ютерів Джона Вінсента Атанасова і Михайла Пилиповича Кравчука.



3. Складіть невеликий за обсягом твір на тему «Комп'ютери майбутнього». Уведіть текст у створений документ. Збережіть у файл з іменем Вправа 3 у відповідній папці.

Комп'ютерне тестування



Виконайте тестове завдання 3 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

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

Більшість сучасних комп'ютерів побудовано за принципами, які обґрунтував американський учений Джон фон Нейман (рис. 4.1) ще у 1948 році. Відтоді змінився зовнішній вигляд комп'ютерів, з'явилися нові пристрої. Але загальна схема роботи комп'ютера не змінилася (рис. 4.2).



Рис. 4.1

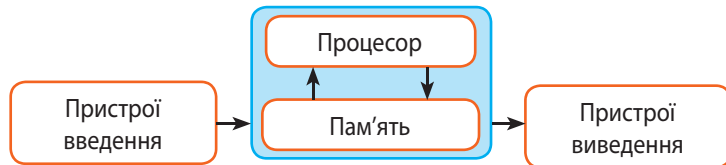


Рис. 4.2

Архітектура комп'ютера



Архітектура комп'ютера — це опис будови комп'ютера і принципів роботи його складових.

Основні вузли настільного персонального комп'ютера містяться в системному блоці.

Системна (материнська) плата — це найважливіший вузол комп'ютера, на якому розміщені *процесор, внутрішня пам'ять, системна шина, порти* (роз'єми, слоти).

Системна шина (магістраль) — набір провідників і пристроїв, призначених для обміну даними між усіма компонентами комп'ютера.

Розглянемо більш детальну схему обміну даними між пристроями комп'ютера (рис. 4.3).

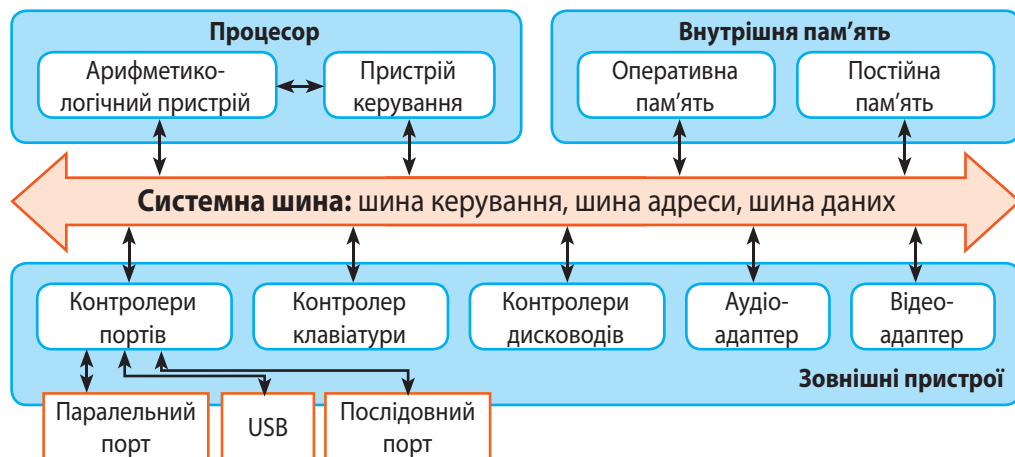


Рис. 4.3

Процесор

Процесор (центральний процесор) — це основна мікросхема комп'ютера, призначена для виконання елементарних арифметичних і логічних операцій та керування роботою комп'ютера.

Процесор складається з двох основних частин: *пристрою керування* та *арифметико-логічного пристрою*. Процесор має свою внутрішню пам'ять.

Конструктивно сучасний процесор (мікропроцесор) — велика інтегральна схема, вміщена у захисний корпус (рис. 4.4).

Швидкодія процесора — основний чинник, який впливає на продуктивність комп'ютера.



Рис. 4.4

Основні технічні характеристики процесора:

- *тактова частота* — кількість тактів за секунду; вимірюється в герцах (Гц); визначає швидкість процесора;
- *обсяг кеш-пам'яті* — обсяг надшвидкої пам'яті, яка використовується процесором для опрацювання часто вживаних даних;
- *розрядність* — кількість бітів даних, які процесор опрацьовує одночасно (розрядність сучасних процесорів — 64 біти);
- *кількість ядер* — кількість окремих обчислювальних ядер (окремих процесорів) в одній мікросхемі;
- *сокет* — тип роз'єму для процесора на системній платі.



Розглянемо запис стислої характеристики процесора:

Intel Core 2 Duo E8600 3.33 ГГц/6 Мб/LGA775.

Маємо: *Intel* — виробник; *Core 2 DUO* — тип — 64-розрядний двох'ядерний; *E8600* — модель; *3.33 ГГц* — тактова частота; *6 Мб* — обсяг кеш-пам'яті; *LGA775* — сокет.

Пам'ять комп'ютера

Пам'ять комп'ютера призначена для тимчасового або постійного зберігання даних. Пам'ять поділяють на *внутрішню* і *зовнішню*; *енергозалежну* і *енергонезалежну* (рис. 4.5).

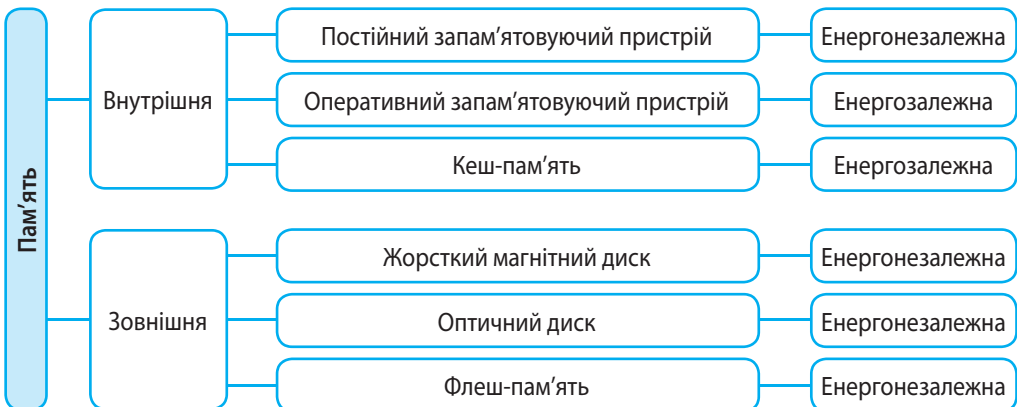


Рис. 4.5

Після вимкнення комп'ютера дані з енергозалежної пам'яті зникають, а в енергонезалежній пам'яті зберігаються постійно.



Внутрішня пам'ять — це швидкодіюча пам'ять, яка у вигляді мікросхем розміщена на системній платі комп'ютера.

Внутрішню пам'ять поділяють на *оперативну* і *постійну*.

Оперативна пам'ять (ОЗП — оперативний запам'ятовуючий пристрій, або RAM — Random Access Memory) призначена для зберігання даних і програм під час роботи комп'ютера.

Основною технічною характеристикою ОЗП є його обсяг.

Постійна пам'ять (ПЗП — постійний запам'ятовуючий пристрій, або ROM — Read-Only Memory) призначена для зберігання програм і даних, що забезпечують роботу комп'ютера після ввімкнення.




У постійну пам'ять записана базова система введення-виведення (BIOS — Basic Input-Output System) — програма для керування роботою пристроїв, завантаження операційної системи тощо.

Зовнішні запам'ятовуючі пристрої



Зовнішня пам'ять — пам'ять, призначена для довготривалого енергонезалежного зберігання даних.

Розглянемо деякі пристрої зовнішньої пам'яті (табл. 1):

Пристрій	Назва пристрою	Принцип дії
	Накопичувач на жорстких магнітних дисках (вінчестер)	Дані записуються і зчитуються спеціальними магнітними головками
	Оптичні CD- і DVD-накопичувачі	Дані записуються або зчитуються лазерним променем
	Флеш-накопичувач	Дані кодуються електричними зарядами

Пристрої введення-виведення даних

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



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

Існують різні адаптери: відеоадаптери для роботи з відеокамерами і моніторами, аудіоадаптери для роботи з аудіопристроями та ін.

Монітор — пристрій виведення, призначений для візуального відображення текстової, графічної та відеоінформації. За принципом дії розрізняють такі типи моніторів: на електронно-променевої трубки (CRT); на рідких кристалах (LCD); на органічних світлодіодах (OLED); «електронне чорнило» (e-ink); плазмові; проєкційні та ін.

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

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

За технологією друку розрізняють такі типи принтерів (табл. 2):

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

Існують також принтери з іншими технологіями друку.

Основні технічні характеристики принтера:

- *роздільна здатність* — найбільша кількість точок на дюйм (dpi — dots per inch);
- *швидкість друку* — кількість друкованих сторінок за хвилину;
- *формат паперу* — найбільший розмір паперу для друку;
- *параметри друку* — можливість друку без полів, двосторонній друк тощо;
- *кількість барвника* — на скільки сторінок вистачає барвника.

Дедалі більшого поширення набувають 3D-принтери, які методом пошарового нанесення речовини формують об'ємні предмети.

Мультимедійні пристрої

Комп'ютер опрацьовує повідомлення, подані різними способами (текстові, графічні, звукові, комбіновані тощо). Комбіновані повідомлення називають мультимедійними.

Мультимедіа — поєднання різних способів подання повідомлень та використання різних пристроїв для їх відтворення. Для введення в комп'ютер мультимедійних даних приєднують такі зовнішні пристрої,

як мікрофон, сканер, відеокамера тощо, для виведення — монітор, проєктор, навушники, принтер тощо. Для узгодження дій цих пристроїв і комп'ютера використовують відповідні адаптери.



Аудіоадаптер — пристрій для відтворення та запису звуку.

Відеоадаптер — пристрій для підтримки перетворення даних у відеосигнал для монітора або проєктора.

Аудіоадаптер (звукова карта) та відеоадаптер (відеокарта) можуть бути вбудовані (інтегровані) в системну плату або встановлені в роз'єм (слот) на ній.

! Відеоадаптер, монітор і спеціальні програми для забезпечення їх підключення утворюють **відеосистему** комп'ютера.

Основними характеристиками відеоадаптера є частота та обсяг відеопам'яті. Від цих характеристик залежить якість зображення на екрані монітора під час перегляду відео, комп'ютерних ігор тощо.

Конфігурація комп'ютера

Конфігурацію комп'ютера описує набір його складових та технічні характеристики цих складових.



Базовою конфігурацією комп'ютера називають обов'язковий набір його складових.

В описі конфігурації комп'ютера використовують позначення, що вказують на характеристики монітора, процесора, оперативної пам'яті, накопичувача на жорстких магнітних дисках, відеоадаптера, оптичного приводу, мережевої карти, операційної системи тощо.



Розглянемо запис, який використовується для стислої характеристики конфігурації комп'ютера: 19.5" (1920x1080) LED/AMD Athlon ІІх4 750К (3.4 ГГц)/RAM 8 Гб/HDD 1 Тб/nVidia GeForce GT730 2 Гб/DVD+RW/LAN/Windows 7.

Питання для самоперевірки



1. Що таке процесор?
2. Що визначає швидкодію комп'ютера?
3. Для чого потрібна оперативна пам'ять?
4. Які функції постійної пам'яті комп'ютера?
5. Яке призначення пристроїв зовнішньої пам'яті?
6. Що розуміють під конфігурацією комп'ютера?

Вправа 4



1. Знайти та проаналізувати технічні характеристики процесора та оперативної пам'яті (ОЗП) вашого комп'ютера.

1) Запустіть текстовий процесор та створіть новий документ. У контекстному меню об'єкта Комп'ютер відкрийте вкладку Властивості. Знайдіть, розшифруйте і введіть у документ характеристики процесора та ОЗП.



2) За каталогами інтернет-магазинів знайдіть описи процесорів. Скопіюйте в документ посилання на сторінку та запис стислої характеристики одного з процесорів. За цим записом визначте й запишіть його розрядність, кількість ядер, тактову частоту, обсяг кеш-пам'яті, сокет.

3) Знайдіть в Інтернеті опис таких характеристик ОЗП: пропускна здатність, тип, обсяг, частота. Скопіюйте знайдені дані у документ. Збережіть файл з іменем Вправа 4 у відповідній папці.



2. Скласти стислу характеристику конфігурації свого домашнього комп'ютера (див. приклад на с. 26).

Комп'ютерне тестування



Виконайте тестове завдання 4 з автоматичною перевіркою на сайті interactive.ranok.com.ua.



Практична робота 2

Конфігурація комп'ютера під потребу

Завдання: визначити конфігурацію комп'ютера під потребу комп'ютерної гри з такими системними вимогами: 30 Гб вільного місця на жорсткому диску; 8 Гб оперативної пам'яті; процесор i7 4790 3.6 ГГц або AMD FX-9 590 4.7 ГГц.

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

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.



- ▶ 1. За каталогами інтернет-магазинів доберіть складові для комплектування комп'ютера з конфігурацією, потрібною для гри.
- ▶ 2. Запустіть текстовий процесор. Створіть новий документ, уведіть назви й вартість таких складових комп'ютера: корпус системного блока з блоком живлення, системна плата, процесор, монітор, оперативна пам'ять, накопичувач на жорстких магнітних дисках (вінчестер), оптичний привід, відеоадаптер (якщо він не вбудований у системну плату), мережева карта, клавіатура, миша.
- ▶ 3. Визначте складові, від яких залежить відповідність конфігурації комп'ютера потребам гри. Скопіюйте їх назви, стислі технічні характеристики та посилання на сторінки.
- ▶ 4. Запишіть, за якими характеристиками ви вибирали процесор; системну плату.
- ▶ 5. Збережіть файл з іменем Практична робота 2 у відповідній папці. Завершіть роботу за комп'ютером.

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

§ 5. Програмне забезпечення комп'ютера

Ми ознайомилися з апаратною складовою комп'ютера. Та комп'ютер не може працювати без програм, тому розглянемо другу його складову — програмну.



Програмним забезпеченням називають сукупність програм, призначених для забезпечення роботи інформаційної системи.

Програмне забезпечення (ПЗ) поділяють на *системне, прикладне та службове*.

Системне програмне забезпечення

Системне програмне забезпечення (системні програми) призначене для керування роботою складових комп'ютера та обміну даними між ними. Серед системних програм особливе місце належить операційним системам.

Ви вже знаєте, що **операційна система (ОС)** — це комплект програм, призначених для керування ресурсами, обчислювальними процесами комп'ютера, а також для забезпечення взаємодії з користувачем. Під час увімкнення комп'ютера ОС завантажується автоматично.

Основні функції ОС:

- введення-виведення даних, запуск-зупинка програм тощо;
- забезпечення роботи пристроїв введення-виведення;
- завантаження програм в оперативну пам'ять та їх виконання;
- керування оперативною пам'яттю;
- керування доступом до даних на зовнішніх носіях;
- забезпечення інтерфейсу користувача;
- підтримка файлової системи.

Розглянемо типи ОС за основними ознаками (рис. 5.1).



Рис. 5.1

» На персональних комп'ютерах використовують ОС Windows, Linux, MacOS. Для мобільних пристроїв призначені ОС Android, Windows CE, iOS, PalmOS та ін.

Прикладне програмне забезпечення

Прикладне програмне забезпечення (прикладні програми) призначене для розв'язання задач, необхідних користувачу.

Прикладні програми поділяють за призначенням (табл. 1):

Прикладні програми	Призначення
Програми загального призначення	Різні галузі діяльності (наприклад, опрацювання текстів, малюнків, таблиць тощо)
Програми спеціального призначення	Певні галузі діяльності або конкретні підприємства (наприклад, створення відеоефектів тощо)
Інструментальні програмні засоби	Створення інших програм

Службове програмне забезпечення

Службове програмне забезпечення (службові програми) призначене для діагностування апаратної і програмної складових комп'ютера. До службових належать програми для архівування даних, форматування та перевірки цілісності диска тощо.



Форматування носія даних — процес розмітки області зберігання даних.

Форматування виконується перед першим використанням диска або у випадку виявлення серйозних збоїв у його роботі. Цей процес можна здійснити за допомогою контекстного меню носія. Після форматування дані на носії знищуються.

У ході форматування також можна перевірити цілісність носія.



Архівування даних — процес стиснення даних, що зберігаються у файлах.

У результаті архівування створюється **архів** — файл, який містить один або більше файлів зі стисненими даними.

Архівування даних дозволяє заощаджувати значний обсяг дискового простору, передавати файл за допомогою Інтернету.

Поширеним методом стиснення даних є знаходження фрагментів, що повторюються, та їх кодування. Такими фрагментами можуть бути пікселі растрового зображення, які мають однаковий колір, повторювані слова чи символи у тексті тощо.

Ступінь стиснення даних (відношення обсягу нестисненого файлу до обсягу його архіву) залежить від типу файлу та методу стиснення.

» У текстах українською мовою разом із розділовими знаками і цифрами використовується близько 60 символів. Для стиснення такого тексту замість 8-бітового кодування можна використати 6-бітове і без втрати даних заощадити 25 % пам'яті.



Розархівування — процес повернення даних архіву до початкового стану.

Операції архівування та розархівування здійснюються за допомогою службових програм — **архіваторів**. Серед найбільш поширених архіваторів можна назвати 7-Zip (безкоштовний), WinRAR і WinZIP (умовно-безкоштовні).

Програми-архіватори дозволяють створювати архіви різних типів: *звичайні, багатотомні, саморозпаковувані*.

Для **створення звичайного архіву** потрібно:

- 1) викликати контекстне меню файлу (групи файлів, папки);
- 2) вибрати для архіватора пункт меню Додати до архіву...;
- 3) зробити необхідні налаштування в діалоговому вікні Ім'я та параметри архіву;
- 4) клацнути кнопку ОК.

Архів створиться в одній папці з вихідним файлом і матиме ім'я, зазначене користувачем на кроці 3.

Якщо архівний файл має великий обсяг, який, наприклад, не дозволяє передати його через Інтернет або зберегти на носії, архів розділяють на кілька частин (томів). Такий архів називають **багатотомним**.

Архівний файл із розширенням .exe, який після запуску розпаковується автоматично, називають **саморозпаковуваним (SFX)**. Цей архів доцільно використовувати, якщо на комп'ютері отримувача не встановлений потрібний архіватор.

Крім операцій архівування та розархівування, сучасні архіватори дозволяють виконувати й інші дії: переглядати вміст архіву, додавати файли до архіву, вилучати окремі файли з архіву, перевіряти цілісність архівного файлу тощо.

Ліцензії на програмне забезпечення

Комп'ютерні програми є результатом праці багатьох людей, тому їх не можна використовувати без дозволу авторів (власників).



Ліцензія на програмне забезпечення — це угода, яка надає право на його використання.

Закон про авторське право забороняє виконувати над програмами дії (наприклад, модифікувати, копіювати, розповсюджувати тощо), які не дозволені ліцензією.

Розглянемо деякі види ліцензій на програмне забезпечення (табл. 2):

Вид ліцензії	Опис програми
Комерційна (commercial)	Власницькі програми (для використання їх необхідно купувати)
Некомерційна (non-commercial use)	Програми, безкоштовні для некомерційного використання (наприклад, спрощена версія комерційної програми)
Умовно-безкоштовна (shareware)	Програми, якими можна користуватися безкоштовно за певних умов (низка комерційних програм надається для безкоштовного ознайомлення протягом певного періоду)
Вільна (free software)	Програми, які дозволено копіювати, модифікувати, розповсюджувати

Однією з найпопулярніших ліцензій на вільне програмне забезпечення є ліцензія GNU GPL, вона надає максимальну кількість прав користувачам.

Інсталяція і деінсталяція програм

Перед початком використання встановленої на комп'ютері програми треба здійснити узгодження її дій з операційною системою.



Інсталяція — установлення програм на комп'ютері.

Під час інсталяції відбувається копіювання всіх або окремих складових програми на жорсткий диск, вносяться необхідні записи до системних файлів, створюються команди в головному меню тощо.



Деінсталяція — видалення встановлених програм з комп'ютера.

Під час деінсталяції також вносяться зміни до системних файлів, тому не можна просто вилучити файли (папки) програми з комп'ютера. Потрібно скористатися програмою деінсталяції конкретної програми або системною утилітою для видалення програм: Пуск → Панель керування → Програми та компоненти.

Інсталювати та деінсталювати програми може тільки користувач з правами адміністратора.

Сумісність програмного забезпечення

Операційні системи (ОС) можуть значно відрізнятися одна від одної, тому здебільшого програма, розроблена для однієї ОС (наприклад Windows 7), не працюватиме з іншою ОС (наприклад Linux). У таких випадках кажуть про *несумісність програми і ОС*.

» Для налаштування сумісності програми з різними версіями ОС Windows іноді достатньо вибрати в контекстному меню команду Властивості → Сумісність та потрібну версію ОС.

Перед тим як купити або встановити програму, слід обов'язково звернути увагу на вимоги до комп'ютера та ОС.

Питання для самоперевірки



1. Що таке програмне забезпечення комп'ютера?
2. Яке призначення прикладного програмного забезпечення?
3. Яке призначення службового програмного забезпечення?
4. Що таке архівування і розархівування даних?
5. У чому полягає форматування носіїв даних?
6. Що таке інсталяція та деінсталяція програм?

Вправа 5



► 1. Знайдіть в Інтернеті відповідь на питання, чи дозволяє ліцензія GNU GPL продавати копії програм. Створіть у текстовому процесорі документ, уведіть до нього адресу сайта, де було знайдено відповідь, скопіюйте відповідь. Збережіть файл з іменем Вправа 5 у відповідній папці.



► 2. В обліковому записі користувача з правами адміністратора встановіть на комп'ютер програму, запропоновану вчителем. Запустіть програму, продемонструйте вчителю та деінстальуйте її.



► 3. Складіть орієнтовний перелік програм, потрібних для роботи за комп'ютером учням; програмістам; фотохудожникам.



Комп'ютерне тестування



Виконайте тестове завдання 5 з автоматичною перевіркою на сайті interactive.ranok.com.ua.



Практична робота 3

Архівування та розархівування даних

Завдання: здійснити за допомогою програми-архіватора стиснення даних в архіви різних типів, виконати розархівування, порівняти ступінь стиснення даних.

Обладнання: комп'ютер зі встановленим архіватором, папка з файлами, підготовлена для архівування.

Хід роботи

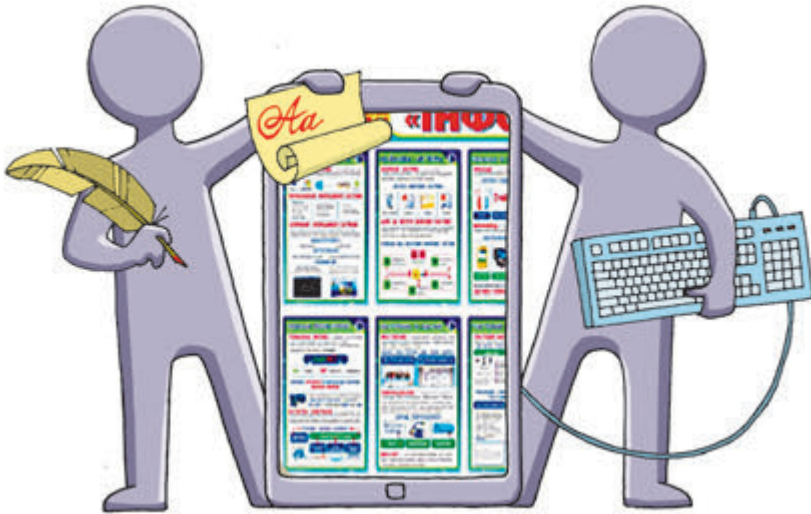
Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- ▶ **1.** Запустіть текстовий процесор. Визначте загальний обсяг папки з файлами, підготовленої вчителем. Створіть новий документ і введіть дані.
- ▶ **2.** Створіть звичайний архів папки. Визначте обсяг цього архіву та введіть дані в документ.
- ▶ **3.** Визначте ступінь стиснення архіву. Введіть дані в документ.
- ▶ **4.** Створіть багатотомний архів папки із трьох томів приблизно однакового розміру. Під час стиснення задайте розмір тому: втричі менший за обсяг звичайного архіву. Знайдіть загальний обсяг трьох томів архіву і введіть дані в документ.
- ▶ **5.** Створіть саморозпаковуваний архів папки. Визначте обсяг архіву й введіть дані в документ. Подвійним клацанням добудьте з архіву папку, змінивши її ім'я.
- ▶ **6.** Збережіть файл з іменем Практична робота 3 у відповідній папці. Завершіть роботу за комп'ютером.

Зробіть висновок: як за допомогою архіватора створити звичайний, багатотомний, саморозпаковуваний архіви, який тип архіву ви виберете для пересилання й чому.

РОЗДІЛ 3

ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ



- § 6. Списки в текстовому документі
- § 7. Таблиці в текстовому документі
- § 8. Опрацювання текстового документа, що містить різні об'єкти
- § 9. Оформлення документів. Структура складного текстового документа

Практична робота 4. Створення текстового документа, що містить об'єкти різних типів

Практична робота 5. Структура документа. Автоматизоване створення змісту та покажчиків

ПОВТОРЮЄМО



Ви вже маєте навички роботи з документом у середовищі текстового редактора Блокнот і текстового процесора Word. Знаєте, що документ може містити різні *текстові об'єкти*, основними з яких є символ і абзац.

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

Ви вмієте також виконувати операції над текстовими об'єктами: *редагувати* (видаляти, додавати, переміщувати, копіювати), *форматувати* (встановлювати шрифт, змінювати колір і накреслення символів, розташування абзаців тощо), вставляти графічні об'єкти в документ.

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



У цьому розділі ви ознайомитеся з такими об'єктами текстового документа, як таблиця, список, формула, недруковані знаки, посилання, колонтипули, а також навчитеся оформлювати складні документи.

§ 6. Списки в текстовому документі

Системи опрацювання текстових документів різняться за своїми можливостями. Текстові редактори дозволяють здійснювати просте форматування, текстові процесори — додавати до тексту графічні об'єкти, здійснювати складне форматування тощо.

Формати файлів текстових документів

Імена файлів текстових документів, створених у різних текстових редакторах, мають різні розширення, які вказують на формат, у якому зберігаються дані.



Розширення файла є вказівкою операційній системі, якою програмою опрацювати файл.

Розглянемо найпоширеніші формати файлів текстових документів та їх характеристики.

- Формат `.txt` (*Text* — текст) — зберігається текст із поділом на абзаци без форматування.
- Формати `.doc` (*Document* — документ) і `.docx` (*Document OpenXML*) — зберігаються текст, малюнки, вставлені об'єкти тощо; формати текстового процесора Word.
- Формат `.rtf` (*Rich Text Format* — формат збагаченого тексту) — зберігаються текст, малюнки, вставлені об'єкти; формат дозволяє опрацювати документ різними програмами.
- Формат `.pdf` (*Portable Document Format* — формат портативного документа) — документи, збережені у цьому форматі, не призначені для редагування, а їхній вигляд не залежить від пристрою виведення.

Створення списків

Упорядковану інформацію, наприклад різні переліки, інструкції, плани заходів тощо, у документі часто подають у вигляді списків.



Список — це послідовність абзацив, що мають спільну позначку (маркер або номер).

Програма Word надає можливість створювати списки (рис. 6.1).

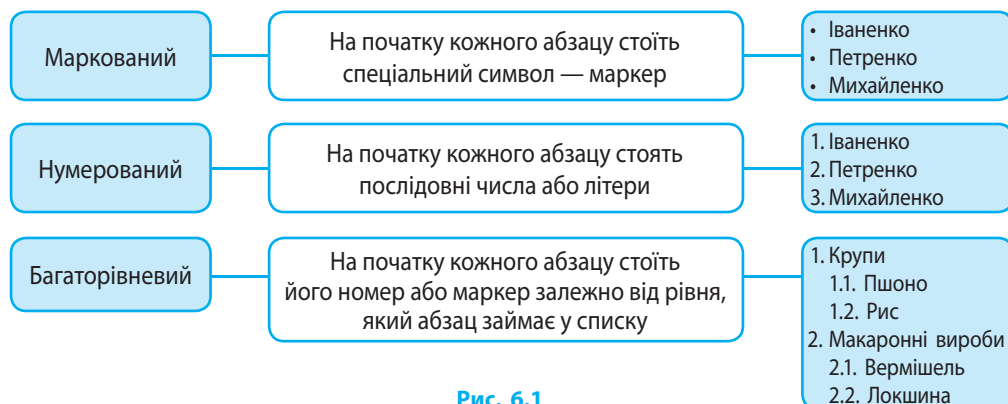


Рис. 6.1

Існує кілька способів створення списків.

Для перетворення введеного тексту на список потрібно:

- 1) виділити відповідні абзаци;
- 2) вибрати на вкладці Основне у групі Абзац значок Маркери (рис. 6.2, а) або Нумерація (рис. 6.2, б);
- 3) відкрити список кнопкою ▼ і вибрати значок маркера або формат номера;
- 4) клацнути кнопку ОК.



Рис. 6.2

Для створення списку під час уведення тексту потрібно спочатку вибрати значок маркера або формат номера — з'явиться позначка першого абзацу.

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

Для створення багаторівневих списків застосовують інструмент Багаторівневий список (рис. 6.2, в). Окрім номерів та літер багаторівнева нумерація може містити різні маркери, що позначають різні рівні списку. На рис. 6.3 перший, другий, третій рівні списку виділено червоним, зеленим і синім кольором відповідно.

- | | |
|----------------------|------------------|
| 1. Перший рівень | 1. Перший рівень |
| 1.1. Другий рівень | А. Другий рівень |
| 1.1.1. Третій рівень | • Третій рівень |
| 1.2. Другий рівень | Б. Другий рівень |
| 1.2.1. Третій рівень | • Третій рівень |
| 2. Перший рівень | 2. Перший рівень |
| 2.1. Другий рівень | А. Другий рівень |
| 2.2.1. Третій рівень | • Третій рівень |

Рис. 6.3

Програма Word у багаторівневому списку підтримує дев'ять рівнів нумерації. Для підвищення рівня елемента списку (наприклад від 1.1. до 1.) потрібно використовувати інструменти Зменшити відступ (див. рис. 6.2, з) або сполучення клавіш Shift + Tab, а для зниження рівня списку — інструмент Збільшити відступ (див. рис. 6.2, д) або клавішу Tab.

Редагування списків

У текстовому процесорі Word списки зручно редагувати.

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

Для відміни нумерації одного абзацу потрібно клацнути мишею в межах абзацу і вимкнути кнопку списку (див. рис. 6.2, а–в) або вилучити номер (маркер). Для цього на початку абзацу слід поставити курсор і натиснути клавішу Backspace. Решта абзаців списку автоматично перенумерується. Для відміни нумерації кількох абзаців ці абзаці потрібно виділити і вимкнути кнопку списку.

Якщо нумерація, утворена, наприклад, під час копіювання списку, не відповідає потребам, користувач може її змінити, вибравши у контекстному меню номера команду Почати заново або Продовжити нумерацію.

Форматування списків

Форматування списку передбачає зміну зовнішнього вигляду значків маркерів і форматів номерів, відстані між маркерами і текстом.

Для заміни наявного маркера на інший потрібно клацнути по маркеру, відкрити список маркування (рис. 6.4, а), вибрати потрібний маркер або, скориставшись командою Визначити новий маркер, установити маркер, відсутній у стандартній колекції.

Для встановлення нового формату номерів треба виділити потрібні абзаци списку або клацнути якийсь номер (якщо є потреба змінити всі номери певного рівня) і змінити формат, скориставшись командами Визначити новий формат номера (рис. 6.4, б) або Визначити новий стиль багаторівневого списку.

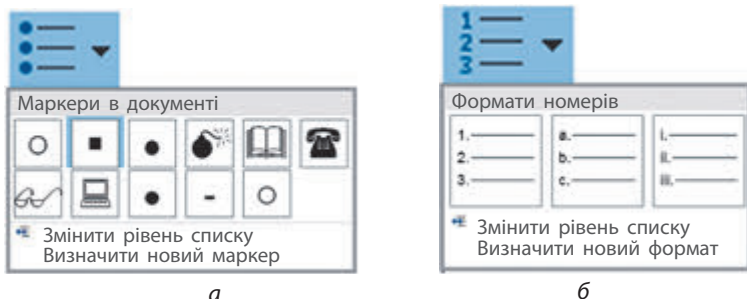


Рис. 6.4

» Якщо при визначенні нового формату номера додати слово *Розділ*, то елементи списку 1., 2., 3. будуть позначатися так: *Розділ 1., Розділ 2., Розділ 3.*

Питання для самоперевірки






1. Які списки можна створити у текстовому процесорі Word?
2. Назвіть способи створення списків.
3. Як перейти з одного рівня нумерації на інший?
4. Як додати до списку новий елемент?
5. Як у списку замінити один знак маркера на інший?
6. Який список називають багаторівневим?

Вправа 6



1. Запустіть текстовий процесор і створіть новий документ. Установіть параметри сторінки: розмір паперу — А5, поля — по 2 см, шрифт — *Arial*, розмір — 12.
 - 1) Створіть нумерований список за зразком.
 1. Січень
 2. Лютий
 3. Березень
 4. Квітень

2) Створіть маркований список за зразком.

-  Комп'ютери
-  Ноутбуки
-  Смартфони

3) Створіть багаторівневий список за зразком.

1. Перші страви
 - 1.1. Борщі
 - 1.1.1. Червоний
 - 1.2. Супи
 - 1.2.1. Гороховий
2. Другі страви
 - 2.1. М'ясні
 - 2.1.1. Вареники з м'ясом
 - 2.2. Рибні
 - 2.2.1. Судак у білому соусі

► 2. Збережіть файл з іменем Вправа 6 у відповідній папці.



Комп'ютерне тестування

Виконайте тестове завдання 6 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 7. Таблиці в текстовому документі

Для впорядкування і наочного подання даних у текстовому документі часто використовують таблиці. Дані, подані у вигляді таблиці, є більш зручними для сприйняття.

Створення таблиці

Таблиця в текстовому документі — це сукупність клітинок, які можуть містити текст, числа, графічні об'єкти тощо.

Розглянемо способи створення таблиці.

Для створення таблиці простої структури потрібно на вкладці Вставлення у групі Таблиці вибрати команду Таблиця, на схемі таблиці

виділити потрібну кількість рядків і стовпців та клацнути лівою кнопкою миші (рис. 7.1, а). Або вибрати команду Вставити таблицю та у вікні Вставлення таблиці задати кількість рядків і стовпців (рис. 7.1, б).

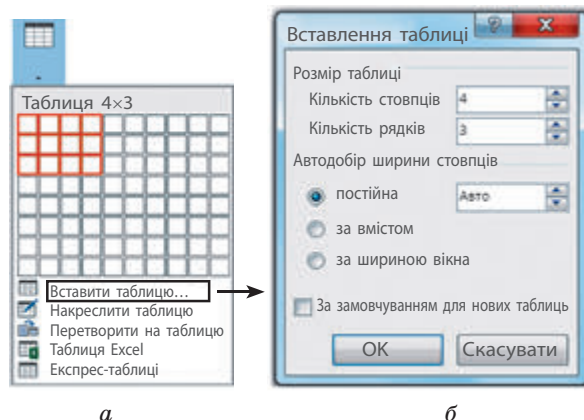






Рис. 7.1

Щоб **накреслити таблицю довільної структури**, потрібно вибрати команду Накреслити таблицю (рис. 7.1, а) і, коли вказівник набуде вигляду олівця, намалювати таблицю. Вилучити зайві межі можна за допомогою інструмента Гумка на вкладці Конструктор.

Для **перетворення виділеного тексту на таблицю** слід вибрати команду Перетворити на таблицю (рис. 7.1, а).

Перед початком роботи з таблицею її потрібно виділити, наприклад клацнувши маркер  над лівим верхнім кутом.

Об'єкти таблиці зручно виділяти за допомогою миші (див. таблицю):

Об'єкт	Опис дії
Клітинка	Установити вказівник біля лівої межі клітинки і клацнути, коли він набуде вигляду 
Діапазон клітинок	Протягнути вказівник із натиснутою лівою кнопкою миші від першої клітинки до діагонально протилежної у прямокутному діапазоні
Стовпець	Установити вказівник над стовпцем і клацнути, коли він набуде вигляду 
Рядок	Установити вказівник ліворуч від рядка і клацнути, коли він набуде вигляду 
Суміжні об'єкти	Протягнути вказівник із натиснутою лівою кнопкою миші по потрібних об'єктах
Несуміжні об'єкти	Утримуючи клавішу Ctrl, послідовно виділити мишею потрібні об'єкти

Виділені об'єкти таблиці (клітинки, рядки, стовпці) позначаються іншим кольором, після чого можна здійснювати редагування та форматування.

Щоб зняти виділення, треба клацнути у будь-якому місці сторінки.

Редагування та форматування таблиці

Редагування таблиці передбачає зміну вмісту клітинок, а також розмірів клітинок, стовпців і рядків.

Для **переміщення окремої межі таблиці** треба навести вказівник на потрібну межу до появи двосторонньої стрілки (рис. 7.2), натиснути ліву кнопку миші й перемістити межу, яка при цьому позначається штриховою лінією.

Якщо перетягнути вертикальну межу виділеної клітинки (або кількох клітинок), то зміняться розміри тільки цієї клітинки (або клітинок) і суміжної (або суміжних).

Для **зміни розміру одного стовпця або рядка** (без зміни розміру інших) слід перетягнути відповідний маркер на лінійці (рис. 7.3). Щоб ці маркери з'явилися, слід клацнути всередині таблиці.

Для **переміщення курсора по клітинках** потрібно клацнути клітинку і скористатися клавішами керування курсором або клавішею Tab.

Для **додавання нового рядка** потрібно встановити курсор у клітинку, що міститься в таблиці внизу справа, і натиснути клавішу Tab.

Для налаштування загального стилю таблиці та форматування її об'єктів, наприклад зміни напрямку тексту, об'єднання або розділення клітинок, використовують контекстне меню клітинки або вкладку Конструктор. Форматування вмісту клітинки таблиці здійснюють стандартними засобами Word.



Рис. 7.2

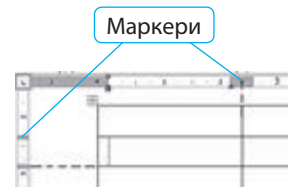


Рис. 7.3

Питання для самоперевірки



1. Назвіть способи вставлення таблиці в документ.
2. Як змінити розміри стовпця або рядка таблиці?
3. Які об'єкти таблиці можна виділити за допомогою миші?
4. Як розділити одну клітинку таблиці на кілька клітинок?
5. Як об'єднати кілька клітинок таблиці?
6. Як налаштувати загальний стиль таблиці?

Вправа 7



- ▶ Створити таблицю успішності учнів з кількох предметів.
- 1) Завантажте текстовий процесор Word. Установіть: розмір паперу — А5, поля — по 2 см, шрифт — *Arial*, розмір — 12.
 - 2) Створіть таблицю із семи стовпців і шести рядків. Об'єднайте клітинки, змініть висоту рядків за зразком:

№	Прізвище та ім'я	Предмети				
		Біологія	Гео-графія	Інформатика	Історія	Фізика

- 3) Установіть ширину стовпців, уведіть заголовки і відформатуйте за вже наведеним зразком.
- 4) Введіть у таблицю власні дані та дані ще про трьох учнів класу. Пронумеруйте рядки; заповніть таблицю довільними оцінками. Збережіть файл з іменем Вправа 7 у відповідній папці.

Комп'ютерне тестування



Виконайте тестове завдання 7 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 8. Опрацювання текстового документа, що містить різні об'єкти

Текстовий документ Word може містити текстові та графічні об'єкти. Розглянемо роботу з об'єктами детальніше.

Робота з колонками

Програма Word дозволяє розташувати текст у вигляді колонок. Зазвичай колонки використовують у газетних і журнальних статтях, бюлетенях тощо.

Для розміщення тексту в колонках потрібно:

- 1) виділити фрагмент тексту;
- 2) розгорнути на вкладці Розмітка сторінки в групі Параметри сторінки список Стовпці;
- 3) вибрати потрібну кількість стовпців.

Налаштувати властивості стовпців можна за допомогою команди Додаткові стовпці.

Робота із символами і формулами

Під час форматування текстових документів можна використовувати недруковані знаки та символи, яких немає на клавіатурі.

Для вставлення символів, відсутніх на клавіатурі (наприклад символ §), потрібно на вкладці Вставлення у групі Символи вибрати елемент Символ (рис. 8.1). Якщо необхідного символу в колекції немає, слід у вікні Символ вибрати Інші символи і вставити потрібний. У цьому вікні також можна визначити код символу або вставити символ за значенням коду.

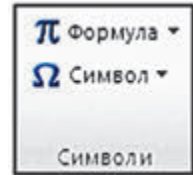


Рис. 8.1

Недруковані знаки належать до текстових об'єктів і позначають місця натискання клавіш Пробіл, Tab, Enter тощо. Недруковані знаки не виводяться під час друку на папері.

Щоб увімкнути або вимкнути режим, при якому відображаються ці знаки, необхідно на вкладці Основне в полі Абзац вибрати ¶.

» Після ввімкнення режиму відображення недрукованих знаків текст набуде такого вигляду:

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

Режим відображення недрукованих знаків дозволяє швидко усунути недоліки форматування за допомогою клавіш (див. таблицю):

Недрукований знак	Назва знака	Клавіші
·	Пробіл	Пробіл
¶	Кінець абзацу	Enter
↵	Розрив рядка	Shift + Enter
⏏	М'який перенос	Ctrl + «-»
°	Нерозривний пробіл	Ctrl + Shift + Пробіл

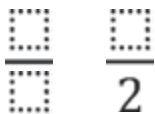


Рис. 8.2

Для **вставлення готової формули** в документ потрібно клацнути на вкладці Вставлення в групі Символи стрілку ▼ поряд із написом Формула і вибрати потрібну формулу.

Для **створення нової формули** потрібно вибрати Вставити нову формулу, на вкладці Конструктор вибрати потрібні структури (наприклад дріб) та/або символи і сконструювати власну формулу (рис. 8.2).

Графічні об'єкти в текстовому документі

Як відомо, у текстовий документ можна додавати графічні об'єкти — малюнки, фотографії, схеми тощо.

У документах Word використовують як графічні об'єкти, створені в інших програмах, так і створені засобами Word (організаційні діаграми, автофігури тощо).

Готовий малюнок можна вставити в текстовий документ за допомогою команд контекстного меню або через буфер обміну.

Для **вставлення малюнка** потрібно:

- 1) на вкладці Вставлення в групі Зображення вибрати Рисунок;
- 2) у діалоговому вікні вибрати малюнок;
- 3) клацнути кнопку Вставити.

Вставлений малюнок часто потребує редагування і форматування: зміни розмірів і орієнтації, визначення положення на сторінці, налаштування відображення меж, яскравості тощо.

Спочатку малюнок потрібно виділити, клацнувши його лівою кнопкою миші. Якщо малюнок виділити не вдається, то на вкладці Основне в групі Редагування слід розгорнути список Виділити, вибрати інструмент Вибір об'єктів і клацнути потрібний об'єкт.

Виділене зображення має рамку з маркерами, за допомогою яких малюнок можна зменшувати, збільшувати, стискати, обертати тощо. Так, змінити розміри можна перетягуванням межових маркерів: якщо перетягнути серединний маркер, то зміниться тільки висота або ширина, якщо кутовий — зміняться обидва розміри.

Для **переміщення малюнка** потрібно на цьому малюнку встановити вказівник та перетягнути з натиснутою лівою кнопкою миші або клацнути малюнок та скористатися клавішами зі стрілками на клавіатурі.

Малюнок може обтікатися текстом навколо рамки, згори і знизу тощо. Щоб задати спосіб обтікання малюнка текстом, потрібно клацнути (виділити) малюнок і вибрати спосіб обтікання на вкладці Формат у списку Обтікання текстом.

Над рамкою виділеного малюнка знаходиться маркер, за допомогою якого можна повертати малюнок навколо центра.

Щоб обрізати малюнок, треба вибрати команду Формат → Обтинання. Після цього переміщенням маркерів слід обрізати край малюнка та клацнути поза межами малюнка.

Питання для самоперевірки



1. Як розмістити текст у кілька колонок?
2. Як увімкнути режим відображення недрукованих знаків?
3. Як вставити формулу в текстовий документ?
4. Як вставити малюнок у текстовий документ?
5. Які є способи обтікання малюнка текстом?
6. Як обрізати малюнок?

Вправа 8



▶ 1. Запустіть текстовий процесор і створіть новий документ. Знайдіть в Інтернеті невеликий вірш Т. Г. Шевченка та скопіюйте його в документ. Розташуйте текст у дві колонки.

▶ 2. Вставте в текстовий документ формулу $v_1 = \frac{s_1}{t}$.



▶ 3. Скопіюйте зображення екрана (сторінку з формулою) за допомогою клавіші PrintScreen у буфер обміну. Вставте малюнок із буфера обміну в документ, обріжте так, щоб залишилася лише формула. Задайте для малюнка обтікання Перед текстом, поверніть на довільний кут і розташуйте нижче від тексту. Збережіть файл з іменем Вправа 8 у відповідній папці.



Комп'ютерне тестування

Виконайте тестове завдання 8 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 9. Оформлення документів. Структура складного текстового документа

Форматування великих за обсягом і складних за структурою документів займає багато часу. Щоб прискорити процес форматування таких документів, зручно застосовувати стильове оформлення.

Стильове оформлення документа



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

До застосування стилю:

Перший у Європі комп'ютер був створений у Києві під керівництвом академіка С. О. Лебедева.

ОСНОВНЕ



Після застосування стилю
Виділена цитата

Перший у Європі комп'ютер був створений у Києві під керівництвом академіка С. О. Лебедева.

Рис. 9.1

Для задання стильового оформлення виділеного об'єкта потрібно вибрати певний стиль на вкладці Основне в групі Стилі (рис. 9.1).

Для редагування стилю потрібно:

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

Після цього всі об'єкти, що мали заданий стиль, набудуть нового вигляду.

Шаблони документів

Досить часто доводиться створювати документи, що мають схожу структуру. Для цього використовують шаблони.



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

У Word файли шаблонів мають розширення .dot (.dotx). Під час створення документа за замовчуванням відкривається шаблон Normal.dotx.

Користувачу надається можливість створити новий документ на основі цього шаблону або за допомогою команди Office → Створити відкрити вікно, у якому можна вибрати інший шаблон.

Якщо користувач вирішить зберегти як шаблон документ, оформлений на власний розсуд, то слід клацнути кнопку Office, зазначити місце збереження, назву і вибрати тип файлу — Шаблон Word.

Збережений шаблон, на відміну від файла документа, буде захищений від випадкових змін доти, доки користувач не збереже інший документ із таким самим іменем у форматі шаблону.

Розділи

Коли для сторінок, які містяться у різних частинах документа, потрібно встановити різні значення параметрів, документ слід поділити на окремі розділи, вставивши розриви (рис. 9.2).

Для **вставлення розриву розділу** потрібно:

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

Тепер у документі для окремих розділів можна встановити різні параметри поля; розмір і орієнтацію паперу; джерело паперу для принтера; межі сторінки тощо.

Розрив розділу зазначає місце в документі, до якого буде застосовано форматування. У разі видалення розриву розділу видалиться також форматування розділу, що містився перед розривом. Цей текст стане частиною наступного розділу та набуде його формату.

Для **видалення розриву розділу** потрібно:

- 1) клацнути в правому нижньому куті екрана кнопку Чернетка (рис. 9.3) — після цього розриви розділів будуть показані подвійними пунктирними лініями;
- 2) виділити розрив розділу;
- 3) натиснути клавішу Delete.

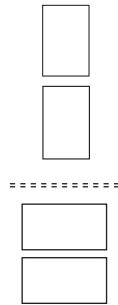


Рис. 9.2

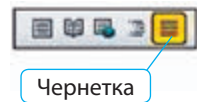


Рис. 9.3

Структура документа



Структурою документа називають ієрархічну схему розміщення складових документа.

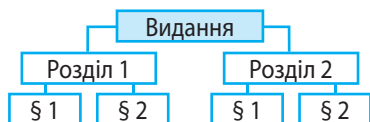


Рис. 9.4

Для організації документів великого обсягу, які складаються зі змісту, розділів і підрозділів різних рівнів (наприклад підручника), використовують **режим структури** (рис. 9.4).

Режим структури в програмі Word дозволяє подавати документ у вигляді, що відображає заголовки розділів, параграфів тощо. У цьому режимі можна змінювати рівень заголовків, копіювати і переміщувати параграфи та розділи шляхом переміщення відповідних заголовків.

Для **вмикання режиму структури** потрібно на вкладці **Вигляд** вибрати команду **Структура**.

Колонтитули

У документах зі складною структурою часто встановлюють колонтитули, у яких зазначають назву книги, розділу, номери сторінок тощо.



Колонтитули — повідомлення, які розміщуються у верхньому або нижньому полі сторінки документа і відображаються на всіх сторінках документа.

Для **встановлення колонтитулів** необхідно:

- 1) двічі клацнути в області верхнього (або нижнього) краю сторінки — відкриваються поля колонтитулів і вкладка **Конструктор** (рис. 9.5);
- 2) ввести текст у верхньому (нижньому) полі колонтитула.

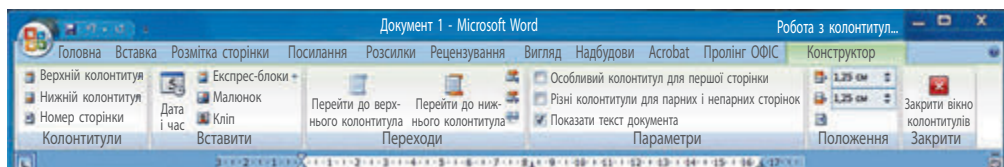


Рис. 9.5

Інструменти вкладки **Конструктор** дозволяють автоматично вставляти в колонтитули *номери сторінок, назву документа, дані про автора, поточну дату* тощо.

Для повернення до основного тексту документа на ньому слід двічі клацнути.

Для **видалення колонтитула** треба на ньому двічі клацнути, виділити його вміст і натиснути клавішу **Delete**.

Посилання

У програмі Word разом із підготовкою документів, які призначені для друку, можна розробляти й електронні документи, які, зокрема, містять посилання.



Посилання (гіперпосилання) — це частина документа (малюнок, текст тощо), клацанням по якій здійснюється перехід у певне місце документа або до іншого файлу, розміщеного на комп'ютері користувача, або в Інтернеті.

Для **переходу за посиланням** слід при натиснутій клавіші Ctrl клацнути на посиланні лівою кнопкою миші.

Для **створення посилання** потрібно:

- 1) вибрати фрагмент тексту або графічний об'єкт;
- 2) на вкладці Вставлення (група Посилання) вибрати Посилання;
- 3) вибрати Гіперпосилання (рис. 9.6) і зазначити, на який саме об'єкт створюється посилання (рис. 9.7).

Перед тим як створити посилання на якесь місце документа, в тексті документа потрібно створити закладку, на яку воно вказуватиме.

Для **створення закладки** потрібно:

- 1) виділити об'єкт або встановити курсор у потрібне місце документа;
- 2) вибрати команду Вставлення → Посилання → Закладка;
- 3) увести назву нової закладки, клацнути кнопку Додати у вікні зі списком закладок.



Рис. 9.6

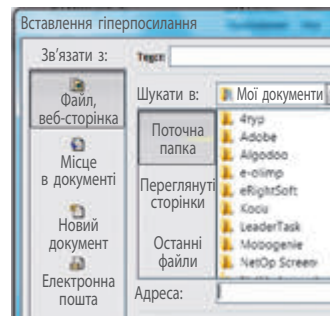


Рис. 9.7

Автоматизоване створення змісту та покажчика

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

Для **автоматизованого створення змісту** потрібно:

- 1) вибрати місце вставлення змісту (на початку або в кінці документа);
- 2) клацнути на вкладці Посилання у групі Зміст елемент Зміст;
- 3) вибрати стиль змісту та натиснути клавішу Enter.

Для **оновлення змісту** потрібно:

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

Для **видалення змісту** потрібно:

- 1) на вкладці Посилання клацнути Оновлення таблиці в групі Зміст;
- 2) вибрати команду Видалення змісту.

Для швидкого пошуку потрібного тексту зазвичай у кінці документа розміщують предметні покажчики.



Покажчиком (предметним покажчиком) називають список слів із зазначенням номерів сторінок, на яких вони згадуються.

Для **створення покажчика** потрібно:

- 1) виділити в тексті необхідний термін (слово або фразу);
- 2) вибрати команду Посилання → Покажчик → Позначити елемент;
- 3) налаштувати параметри у вікні Визначення елемента покажчика:
 - за потреби змінити або замінити текст у полі основний;
 - щоб елемент вказував на кілька термінів (наприклад, архівний файл; стиснений файл; архів), у полі додатковий вписати через двокрапку інші терміни;
 - натиснути кнопку Позначити або Позначити все (щоб знайти і позначити в документі всі терміни);
- 4) установити курсор у відповідне місце документа і вибрати команду Посилання → Покажчик → Покажчик.

Для **видалення елемента покажчика** потрібно відобразити недруковані знаки тексту (Основне → ¶) і видалити біля елемента покажчика позначку у фігурних дужках.

Щоб робота зі **складним текстовим документом** була ефективною, треба дотримуватися певної послідовності дій.

Розглянемо алгоритм опрацювання складних текстових документів.

- 1) **Набір і збереження** — введення тексту з клавіатури і збереження у вигляді файла на носії.
- 2) **Редагування документа** — виправлення помилок, додавання або вилучення фрагментів тексту, інших об'єктів, зміна їх розташування.
- 3) **Форматування документа** — надання документу необхідного вигляду: зміна властивостей об'єктів, налаштування стилів тощо.
- 4) **Підготовка документа до друку** — оформлення заголовків, розбивка на сторінки, їх нумерування, створення змісту.
- 5) **Друк документа** — виведення на папір всіх або окремих сторінок створеного документа.

У програмі Word під час роботи зі складним документом вікно текстового документа можна розділити на кілька областей на вкладках Вигляд → Розділити (група Вікно).

У кожній частині документ буде видимим. Особливо зручно використовувати ділення вікна під час копіювання (перенесення) фрагментів з однієї частини в іншу.

Питання для самоперевірки



1. Як змінити стиль абзацу?
2. Що таке шаблон документа?
3. Для чого використовують розрив розділу; режим структури документа?
4. Що таке колонтитул?
5. Як створити посилання?
6. Як автоматизувати створення змісту?
7. Як створити покажчик?

Вправа 9



- Створити й оформити текстовий документ із використанням різних стилів.
- 1) Запустіть текстовий процесор Word, створіть новий документ. Знайдіть в Інтернеті та скопіюйте в документ одну з дій п'єси Івана Карпенка-Карого «Сто тисяч». Для кожної яви встановіть різне стильове оформлення на власний вибір.
 - 2) Якщо назви твору немає, введіть її на початку документа і встановіть стиль Назва. У кінці документа введіть ім'я і прізвище автора твору, задайте стиль Строгий.
 - 3) Установіть верхній колонтитул із поточною датою. Додайте в нижньому колонтитулі номери сторінок. Зробіть назву твору посиланням на веб-сторінку із цим твором. Збережіть документ у форматі шаблону з іменем Вправа 9 у відповідній папці.



Комп'ютерне тестування

Виконайте тестове завдання 9 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

Практична робота 4



Створення текстового документа, що містить об'єкти різних типів

Завдання: створити текстовий документ (картку для уроку фізики) з використанням об'єктів різних типів.

Обладнання: комп'ютер зі встановленим текстовим процесором Word, з'єднаний із мережею Інтернет.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- ▶ 1. Запустіть текстовий процесор та створіть новий документ. Установіть: формат сторінки — А4, поля — по 2 см, орієнтація — книжна, інтервали до і після абзаців — 0 см, міжрядковий інтервал — одинарний, шрифт — *Times New Roman*, розмір — 14, накреслення — жирний курсив.
- ▶ 2. Введіть у першому абзаці номер параграфа — § 1, у другому абзаці — заголовок «Рівномірний прямолінійний рух», задайте вирівнювання — по центру.
- ▶ 3. Знайдіть в Інтернеті зображення гепарда, скопіюйте малюнок та вставте в документ під заголовком.
- ▶ 4. Задайте формат обтікання малюнка текстом Зверху і знизу та обріжте малюнок так, щоб залишилося лише зображення гепарда.
- ▶ 5. Пропорційно змініть розміри малюнка так, щоб за шириною він зайняв усю сторінку (від поля до поля).
- ▶ 6. Установіть для заголовка розмір шрифту — 31, відформатуйте текст так, щоб заголовок розмістився в одному рядку.
- ▶ 7. Знайдіть в Інтернеті дані про швидкості руху пішохода, гепарда і сокола та введіть у документ в окремих абзацах.
- ▶ 8. Створіть з отриманих даних нумерований список.
- ▶ 9. Уставте в документ під списком таблицю: п'ять рядків, три стовпці.
- ▶ 10. Об'єднайте клітинки верхнього рядка таблиці та введіть назву таблиці «Основні величини».
- ▶ 11. Заповніть таблицю за зразком, уведіть необхідні формули.

§1

Рівномірний прямолінійний рух



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

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

- ▶ 12. Оформте таблицю за зразком, розташуйте написи по центру, перемістіть межі таблиці.
- ▶ 13. Збережіть файл з іменем Практична робота 4 у відповідній папці. Завершіть роботу за комп'ютером.

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

Практична робота 5



Структура документа. Автоматизоване створення змісту та покажчиків

Завдання: створити текстовий документ, який складатиметься з декількох розділів, здійснити форматування його об'єктів, створити зміст в автоматичному режимі та покажчик.

Обладнання: комп'ютер зі встановленим текстовим процесором Word, з'єднаний із мережею Інтернет.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- ▶ 1. Запустіть текстовий процесор Word, створіть новий документ. Знайдіть в Інтернеті три-чотири вірші одного з українських поетів і скопіюйте в документ. Якщо текст міститься в таблиці — витягніть його. Для цього виділіть всю таблицю і виконайте: Конструктор → Дані → Перетворити на текст (розділювач — знак абзацу).
- ▶ 2. Відформатуйте текст так, щоб рядки (крім першого) не починалися з нового абзацу; видаліть порожні рядки; розмістіть назву вірша в окремому абзаці.
- ▶ 3. Введіть у першому абзаці назву збірки, наприклад «Вірші Т. Г. Шевченка».
- ▶ 4. Установіть для документа стиль Звичайний, для назв віршів — стиль Заголовок 1, для назви збірки — стиль Назва.
- ▶ 5. Знайдіть в Інтернеті портрет автора віршів і скопіюйте в документ після назви збірки. За потреби відформатуйте малюнок: обріжте, змініть розміри, обтікання, розташування, рамку тощо.
- ▶ 6. Створіть під портретом автора автоматизований зміст.
- ▶ 7. Переіменуйте створений Зміст на Зміст збірки.
- ▶ 8. Поділіть документ на розділи. Для цього після портрета й останнього рядка кожного вірша вставте розриви Поточна сторінка, а після змісту — розрив Наступна сторінка.
- ▶ 9. Установіть для всього документа міжрядковий інтервал — полуротний. Виконайте оновлення змісту.
- ▶ 10. Установіть у нижньому колонтитулі нумерацію сторінок.
- ▶ 11. Виберіть два-три слова з тих, що повторюються у віршах, і створіть покажчик.
- ▶ 12. Додайте до назви збірки посилання на сторінку, з якої копіювалися тексти віршів. Перевірте, чи працює посилання. Збережіть файл з іменем Практична робота 5 у відповідній папці. Завершіть роботу за комп'ютером.

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

РОЗДІЛ 4

ОПРАЦЮВАННЯ ОБ'ЄКТІВ МУЛЬТИМЕДІА



- § 10. Ознайомлення з форматами відео- та аудіофайлів
- § 11. Програмне забезпечення для опрацювання об'єктів мультимедіа
- § 12. Створення аудіо- та відеофрагментів
- § 13. Опрацювання, зберігання та розміщення відеокліпу в Інтернеті

Практична робота 6. Створення відеокліпу. Додавання відеоефектів, налаштування часових параметрів аудіо- та відеоряду

Практична робота 7. Розміщення аудіо- та відеоматеріалів в Інтернеті

ПОВТОРЮЄМО



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

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

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

Файли мультимедія опрацьовують за допомогою різних комп'ютерних програм і зберігають у спеціальних форматах. Ви вже навчилися користуватися деякими з них, наприклад програмами для створення комп'ютерних презентацій, мультимедійним програвачем, графічним редактором тощо.

1. Що таке мультимедія?
2. Що є об'єктами мультимедія?
3. Наведіть приклади пристроїв мультимедія.
4. Назвіть галузі застосування мультимедія.
5. Що називають графічним редактором?
6. Яка програма призначена для перегляду відео?



У цьому розділі ви навчитесь працювати з такими об'єктами мультимедія, як аудіо та відео, опануєте процес створення відеокліпу й способи розміщення матеріалів в Інтернеті.

§ 10. Ознайомлення з форматами відео- та аудіофайлів

Понад 90 % інформації з навколишнього середовища людина сприймає за допомогою зору і слуху, тому саме відео- й аудіоматеріали є найбільш інформативними.

Формати відеофайлів

Технологічно фільм — це набір послідовних зображень рухомих об'єктів, відзнятих з певною частотою. Під час швидкозмінного перегляду таких зображень у людини створюється враження безперервного руху об'єктів, зображених у кадрі.



Відеофайл — це набір статичних зображень, призначених для почергового виведення з певною частотою.

» У плівковому кіно для зберігання всіх кадрів фільму тривалістю одна година при частоті, наприклад, 24 кадри за секунду, потрібно витратити кілометри плівки. Для зберігання такої самої кількості кадрів нестисненого відео знадобляться сотні гігабайтів пам'яті.

Яким же чином повнометражний фільм уміщується на одному компакт-диску або флеш-накопичувачі? Завдяки сучасним технологіям відеофайли можна «стискати», що робить їх обсяг у сотні разів меншими, ніж у разі збереження фільму в покадровому режимі. Для стиснення і відтворення відео використовують спеціальні програми — **кодеки** (від слів *кодувати* — *декодувати*).

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

Розглянемо формати деяких відеофайлів.

AVI — медіаконтейнер, розроблений спочатку для ОС Windows. Відтворюється на смартфонах, комунікаторах тощо.

WMV — формат компанії Microsoft. Може містити аудіо- і відеодані. Відтворюється плеєрами MPlayer, Windows Media Player та ін.

MOV — формат компанії Apple для QuickTime-плеєра. Може містити відео, анімацію, графіку. Підтримує будь-які аудіо- і відеокодеки.

MPG — часто вживаний формат завдяки невеликим втратам під час стиснення та сумісності з багатьма програмами і пристроями.

MP4 — формат для зберігання відео, аудіо та деяких видів анімації. Підтримує багатоканальний звук. Найчастіше використовується в портативних пристроях і для передачі відеофайлів в Інтернеті.

ASF — формат компанії Microsoft. Використовується для передачі відео в Інтернеті. Підтримує практично всі відеокодеки.

MKV (Matroska) — формат за можливостями перевищує AVI, може зберігати інформацію про титри, навігаційні меню тощо. У контейнер MKV можна впакувати відео і аудіо найрізноманітніших форматів.

WebM — відкритий формат (не вимагає ліцензії) компанії Google для веб-медіа. Спрощений аналог MKV з функціями, потрібними для мережі. Підтримується більшістю сучасних веб-браузерів і медіаплеєрів.

Формати аудіофайлів




Людина отримала змогу зберігати і відтворювати звукові повідомлення, коли навчилася записувати звуковий «слід» на грамплатівках, магнітофонних стрічках тощо. Такий звукозапис називають **аналоговим**.



Цифровим звукозаписом називають подання звуку у вигляді двійкового коду.

Для зберігання оцифрованого звуку в аудіофайлах також використовують різні технології стиснення.

Розглянемо формати деяких аудіофайлів (див. таблицю):

Піктограма	Аудіофайл	Особливості
	WAV, AIFF	Без стиснення
	APE, FLAC	Стиснення без втрат якості
	MP3, Ogg	Стиснення з втратами якості за рахунок виключення частот, що не сприймаються людським вухом

Існує значно більше форматів відео- та аудіофайлів, ніж ми розглянули. Їхня кількість зростає, оскільки вони створюються для різних пристроїв і мають різне призначення. Крім того, технології опрацювання мультимедійних даних розвиваються, це також приводить до появи нових форматів файлів і «відмирання» застарілих.

Питання для самоперевірки



1. Що таке фільм?
2. Навіщо «стискають» цифрове відео?
3. Назвіть відомі вам формати відео.
4. Що таке цифровий звукозапис?
5. Назвіть відомі вам формати аудіофайлів.
6. Чому існує велика кількість відео- та аудіоформатів?

Вправа 10



▶ 1. Відтворіть на комп'ютері музичний аудіофайл. Запишіть у зошиті, які дані (окрім власне звукових) відображаються медіаплеером під час прослуховування.



▶ 2. Знайдіть в Інтернеті відеофільм О. Довженка «Земля» для завантаження. Запишіть у зошиті формат і розмір відеофайла; тривалість фільму.



▶ 3. Поясніть, чому саме цей формат відеофайла використовується для передавання відео в Інтернеті.

Комп'ютерне тестування



Виконайте тестове завдання 10 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 11. Програмне забезпечення для опрацювання об'єктів мультимедіа

Для роботи з об'єктами мультимедіа використовують різні програми. Одні з них призначені для імпортування або відтворення відео та/або аудіо, інші — для редагування мультимедійних даних, зміни форматів відео- та аудіофайлів тощо.

Програми для опрацювання аудіоданих

У користувачів часто виникає потреба в опрацюванні на комп'ютері звукових даних, отриманих із зовнішніх пристроїв, наприклад мікрофона, електрогітари, синтезатора тощо.



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

Для захоплення аудіо потрібно приєднати до комп'ютера аудіоприспирій, наприклад мікрофон, і запустити програму для запису звуку.

- » В ОС Windows є стандартна програма Звукозапис, яку можна запустити за допомогою команди Пуск → Всі програми → Стандартні → Звукозапис (рис. 11.1). Якщо під час записування звуку натиснути кнопку зупинки запису, відкриється вікно з пропозицією зберегти файл.

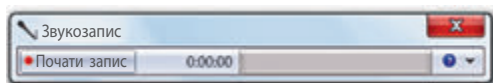


Рис. 11.1

Записаний звук зберігатиметься у форматі WAV. Процес звукозапису можна призупинити, а потім продовжити з того самого місця.

- » У деяких програмах, наприклад у стандартній програмі Звукозапис ОС Windows XP (рис. 11.2), тривалість записуваного фрагмента обмежена. До відкритого звукового файла фрагменти можна дописувати багато разів.

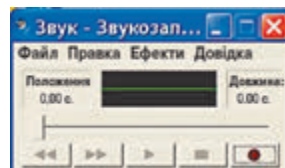


Рис. 11.2



Аудіоредактор — програма для опрацювання аудіофайлів.

За допомогою встановленого на комп'ютер аудіоредактора користувач має можливість записувати аудіо з будь-яких джерел, накладати спецефекти, обрізати аудіофайли, створювати різні звуки тощо.

- » Серед популярних аудіоредакторів можна назвати Audacity, Sony SoundForge, Adobe Audition та ін. Для відтворення аудіоданих (прослуховування звуку) використовують аудіоплеєри, наприклад Winamp, AIMP, JetAudio.

Програми для опрацювання відеоданих

З відеоданими, як і з аудіоданими, на комп'ютері можна виконувати певні дії: захоплення, опрацювання, зберігання. Для цього використовують відповідні пристрої та програмне забезпечення.



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

За джерелом і типом відеосигналу розрізняють захоплення відео з аналогового пристрою, з екрана монітора, із цифрових відеокамер.

Захоплення з аналогового пристрою — це запис відео з телевізора, аналогової відеокамери, касетного відеомагнітофона тощо. Для оцифрування аналогового відеосигналу використовують спеціальні пристрої — відеоадаптери.

Захоплення з екрана монітора — це запис усього, що відображається на екрані монітора або його частині.



Серед програм для захоплення відео з екрана монітора можна назвати Icecream Screen Recorder, CamStudio тощо. Крім захоплення з екрана монітора, вони дозволяють здійснювати захоплення звуку з мікрофона або колонок та зберігати відеофайли у форматах, які підтримуються сучасними медіаплеєрами.

Захоплення із цифрових відеокамер — це запис усього, що фіксує підключена до комп'ютера цифрова відеокамера.

Більшість цифрових відеопристроїв комплектується спеціальним програмним забезпеченням, зокрема і для захоплення відео. Але зазвичай відеоредактори, серед яких і Windows Movie Maker, мають функції захоплення відео з цифрових камер.



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



Серед популярних відеоредакторів можна назвати такі програми, як Lightworks, Pinnacle Studio, Sony Vegas Pro тощо (рис. 11.3). У Windows 7–10 можна встановити відеоредактор Кіностудія, завантаживши його з Інтернету.

Для відтворення аудіо та відео, а також для перегляду зображень, використовують медіаплеєри, одним із яких є вже знайомий вам Програваач Windows Media.



Рис. 11.3

Засоби перетворення аудіо- та відеоформатів

Різні пристрої та програми накладають різні вимоги до формату даних. Перетворити аудіо- або відеофайли з одного формату в інший допомагають спеціальні програми — конвертери.



Конвертер — програма, яка дозволяє перекодувати медіафайли в потрібний формат.



Щоб виконати конвертацію відео- або аудіофайла, наприклад програмою Convertilla, треба додати або перетягти потрібний файл у робоче поле конвертера, вибрати формат, до якого конвертується файл, і клацнути кнопку Конвертувати.

Питання для самоперевірки



1. Що таке захоплення аудіо?
2. Яке призначення аудіоредактора?
3. Що таке захоплення відео?
4. З яких пристроїв можна захоплювати відео?
5. Назвіть програму для перегляду відеофайлів.
6. Як називають програми для зміни формату відеофайлів?

Вправа 11



- ▶ Створити аудіофайл за допомогою стандартної програми за наведеним описом.
- 1) Створіть на робочому столі папку Медіа. Підключіть мікрофон (рожеве гніздо) та навушники (зелене гніздо) до комп'ютера. Запустіть стандартну програму звукозапису.
 - 2) Увімкніть кнопку Запис. Надиктуйте своє прізвище, ім'я, клас. Зупиніть запис і збережіть файл з іменем Вправа 11 у папці Медіа. Завершіть роботу програми, вимкніть мікрофон.
 - 3) Прослухайте запис за допомогою медіа- або аудіоплеєра.

Комп'ютерне тестування



Виконайте тестове завдання 11 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 12. Створення аудіо- та відеофрагментів

Знайомство з Windows Movie Maker

Windows Movie Maker — це стандартний відеоредактор деяких версій ОС Windows, призначений для створення, редагування і монтажу відео.

Розглянемо основні елементи інтерфейсу програми (рис. 12.1).

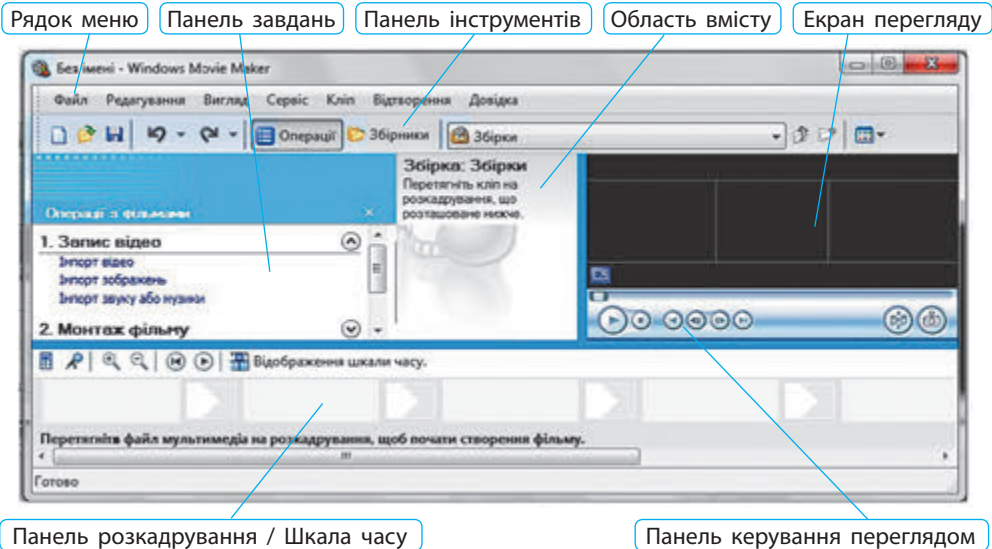


Рис. 12.1

Відображення панелей програми налаштовується за допомогою меню Вигляд. Основні дії з об'єктами виконуються на Панелі розкадрування та Шкалі часу.

Панель розкадрування використовується для перегляду послідовності фрагментів відеоряду, зміни їх порядку, а також для перегляду і встановлення відео ефектів або відеопереходів (рис. 12.2).



Рис. 12.2

Шкала часу дозволяє обрізати фрагменти, налаштовувати тривалість переходів між ними, змінювати час відтворення фрагментів тощо (рис. 12.3).

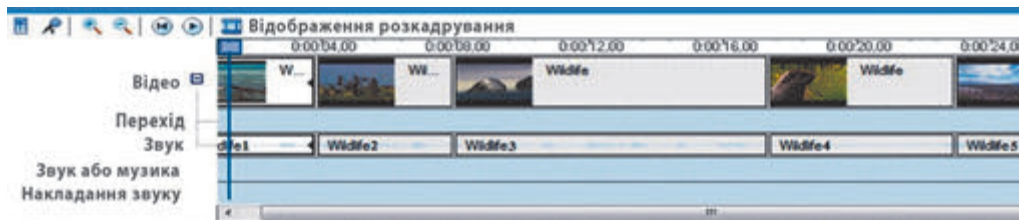


Рис. 12.3

Імпортування аудіо- та відеофрагментів

Процес створення відеокліпу за допомогою відеоредактора Windows Movie Maker складається з певної послідовності кроків.

Першим кроком роботи над створенням відеокліпу є формування добірки графічних зображень, відео- та аудіоматеріалів.



Процес додавання до проекту файлів із зовнішніх джерел називають **імпортом**.

Підготовлені файли потрібно імпортувати за допомогою відповідних команд меню Файл або перетягуванням в Область вмісту.

Під час захоплення відео з цифрової відеокамери (запис із відео-пристрою) одночасно відбувається збереження відеофайла у зазначеному користувачем місці та розміщення в Області вмісту.

Графічні, відео- та аудіофайли, які імпортуються у Windows Movie Maker, можна побачити у вікні Провідника.

Редагування аудіо- та відеофрагментів

Другим кроком роботи над створенням відеокліпу є редагування (розділення, обрізання тощо) відео- і аудіофрагментів. Для цього відеофайли потрібно перетягнути з Області вмісту на Панель розкадрування або Шкалу часу.

Windows Movie Maker дозволяє розділити довгий фрагмент на два коротші, об'єднати два фрагменти в один, обрізати початок або кінець фрагмента і навіть створити новий.

Для **розділення фрагмента** потрібно:

- 1) вибрати потрібний фрагмент;
- 2) установити місце розділення відтворенням або переміщенням маркера Пошук під вікном перегляду (рис. 12.4) або на Шкалі часу;
- 3) призупинити відтворення і натиснути кнопку Розділити.

Під час імпортування файл може розділитися на кілька фрагментів. Такі фрагменти можна об'єднати, якщо вони суміжні.

Суміжні фрагменти можна об'єднати на Панелі розкадрування або в Області вмісту (див. рис. 12.1). Для цього потрібно:

- 1) виділити фрагменти (утримуючи Ctrl);
- 2) у контекстному меню одного з фрагментів вибрати Об'єднати (рис. 12.5).

Для **автоматичного розділення фрагмента** потрібно:

- 1) вибрати фрагмент в Області вмісту;
- 2) вибрати команду Створити кліпи в меню Сервіс.

Для більшості форматів (крім WMV і AVI) операція розділення доступна тільки в ручному режимі.

Щоб **обрізати початок або кінець фрагмента**, потрібно перетягнути на Шкалі часу один із маркерів монтажу, що є на краях фрагмента, в потрібне місце, визначаючи його візуально у Вікні перегляду або за часовими параметрами на Шкалі часу (рис. 12.6).

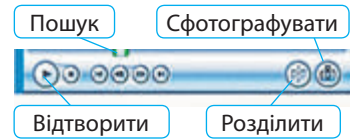


Рис. 12.4

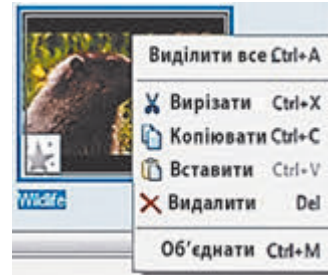


Рис. 12.5

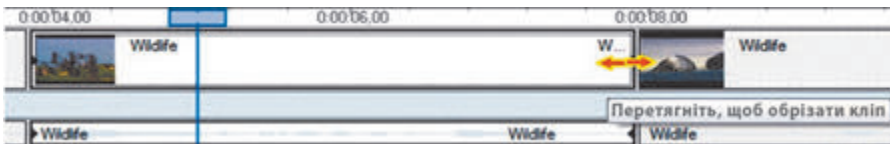


Рис. 12.6

Обрізаний фрагмент можна відновити перетягуванням маркерів монтажу.

Побудова аудіо- та відеоряду

Третім кроком роботи над створенням відеокліпу є побудова аудіо- та відеоряду.

Шкала часу Windows Movie Maker має дві звукові доріжки. Зазвичай відеофрагмент має звуковий супровід, що міститься на доріжці Звук.

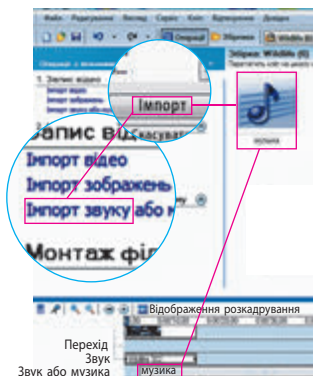


Рис. 12.7

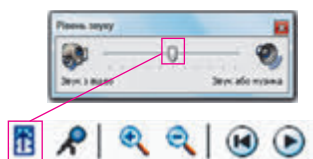


Рис. 12.8

Аудіофрагмент на доріжці Звук можна обрізати, переміщувати і видаляти тільки разом із відповідним відеофрагментом.

Імпортовані аудіофайли можна перетягувати з Області вмісту на доріжку Звук або музика. Якщо перетягнути й відеофрагмент, на доріжці залишиться лише звуковий супровід (рис. 12.7).

Редагування об'єктів на доріжці Звук або музика відбувається незалежно від відеофрагментів і не впливає на відео.

Відносний рівень гучності двох звукових доріжок можна регулювати кнопкою Встановити рівень звуку (рис. 12.8).

Windows Movie Maker дозволяє захоплювати звук із мікрофона. Для цього потрібно підключити мікрофон, натиснути кнопку Шкала часу коментаря, додати звуковий коментар, керуючись вказівками, що з'являться. Файл звукового коментаря зберігається в автоматично створеній папці або в папці за вказівкою користувача.

Відеоряд майбутнього фільму утворюють імпортовані, відредаговані, розміщені на Шкалі часу або Панелі розкадрування об'єкти (тексти, зображення, переходи, відеоефекти).

Текстові фрагменти (назви, титри тощо) часто розміщують на початку або наприкінці відеофільму, на певному відеофрагменті або зображенні, а також перед або після нього.

Час відтворення текстових фрагментів коригується маркерами на Шкалі часу, так само як і час відтворення переходів або відеоефектів. Текст можна редагувати, робити його анімованим, змінювати шрифт, розмір, накреслення і колір символів.

Імпорт зображення не відрізняється від імпорту аудіо чи відео.

Способи імпортування зображення:

- перетягнути графічний файл із місця розміщення в Область вмісту;
- вибрати команду Імпорт зображень (на Панелі завдань).

Потім слід перетягнути зображення з Області вмісту на Панель розкадрування або Шкалу часу.

Щоб зберегти зображення одного кадру, треба здійснити **захоплення кадру**. Для цього потрібно призупинити перегляд на певному

кадрі або знайти його маркером пошуку, клацнути кнопку Сфотографувати (див. рис. 12.4), у вікні Провідника зазначити параметри збереження графічного файлу, який потім автоматично з'явиться в Області вмісту.

Питання для самоперевірки



1. Що відображає Панель розкадрування?
2. Які можливості надає Шкала часу?
3. Як обрізати відеофрагмент?
4. Чи можна обрізаний відеофрагмент відновити?
5. Як створити власний звуковий супровід відеофільму?
6. Як здійснити захоплення кадру?

Вправа 12



- ▶▶ Створити відеокліп зі звуковим супроводом.
- 1) Знайдіть в Інтернеті й завантажте файл для звукового супроводу відеокліпу в папку, зазначену вчителем. Збережіть у тій самій папці відеофайли, підготовлені вчителем, або власні.
 - 2) Запустіть відеоредактор, імпортуйте до нього підготовлені аудіо-та відеофайли. Перегляньте відеофайли.
 - 3) Перетягніть імпортовані відеофайли на Панель розкадрування, а аудіофайл — на доріжку Звук або музика, що міститься на Шкалі часу. Збережіть проект (Файл → Зберегти проект) з іменем Вправа 12 у відповідній папці.

Комп'ютерне тестування



Виконайте тестове завдання 12 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 13. Опрацювання, зберігання та розміщення відеокліпу в Інтернеті

Щоб під час перегляду відео уникнути різкої зміни одного фрагмента іншим, використовують переходи; для налаштування особливого стилю показу відеофрагмента додають відеоєфекти.

Додавання до відеокліпу відеоефектів та налаштування переходів між його фрагментами

Додавання відеоефектів до кліпів дозволяє наочніше реалізувати задум автора, дає можливість прикрасити відеофрагменти (надати вигляду «старого кіно», прискорити його відтворення тощо).

Для додавання відеоефекту треба зі збірки в Області вмісту перетягнути цей ефект на потрібний відеофрагмент або зображення на Панелі розкадрування або Шкалі часу. Після цього відеоефект можна редагувати.

Коли до одного відеофрагмента додати кілька ефектів, то ці ефекти накладуться.

» Якщо до фрагмента двічі додати відеоефект прискорення у два рази, відео відтворюватиметься в 4 рази швидше від початкового.

Додавання переходів між відеофрагментами надає відеокліпу більш професійного та естетичного вигляду.

Для додавання переходів потрібно вибрати команду Перегляд відео-переходів на Панелі завдань, перетягнути потрібний перехід з Області вмісту на Панель розкадрування або Шкалу часу між відеофрагментами або статичними зображеннями.

Тривалість переходу визначається часом одночасного відображення двох відеофрагментів.

Для зміни тривалості переходу потрібно на Шкалі часу перетягнути один із відеофрагментів, збільшуючи або зменшуючи площу перекриття (перехід) з іншим відеофрагментом (рис. 13.1). При цьому, навіть якщо перехід між відеофрагментами не було додано, автоматично встановлюється Плавний перехід.

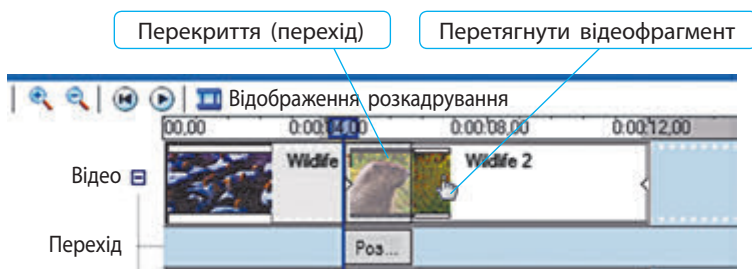


Рис 13.1

Налаштування часових параметрів аудіо- та відеоряду

Кожен відеокліп може містити титри, зображення, аудіо- та відеофрагменти, відеоефекти тощо. Налаштування часових параметрів кліпу полягає у встановленні часу відтворення цих складових, додаванні ефектів (наприклад прискорення) та синхронізації аудіо і відео.

Керування тривалістю відтворення фрагментів, переходів або ефектів здійснюється на Шкалі часу переміщенням маркерів монтажу відповідного об'єкта (рис. 13.2). При цьому для графічних зображень, переходів, відеоефектів і текстових написів змінюється час відтворення, а для відео- і аудіофрагментів виконується обрізання (або відновлення).

Щоб видалити з відеокліпу певний об'єкт, достатньо його клацнути і натиснути клавішу Delete.

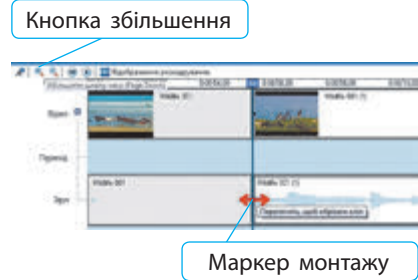


Рис. 13.2

Зберігання проекту відеокліпу

Якщо за відведений час створити відеокліп не вдається, потрібно зберегти незавершений проект.

Під час зберігання проекту створюється файл із розширенням .mswm, який містить інформацію про відеофайли, але не містить відеоданих. Тому, коли користувач відкриє файл проекту на іншому комп'ютері, то обов'язково виникнуть проблеми через відсутність потрібних аудіо- і відеофайлів, зображень тощо. Щоб уникнути такої ситуації, необхідно зберегти всі потрібні файли і файл проекту в окремій папці, і вже з неї здійснити імпортування у Windows Movie Maker.

Після завершення роботи кліп необхідно зберегти. Для цього можна скористатися командою Збереження на комп'ютері на Панелі завдань.

Сервіси розміщення аудіо- та відеофайлів в Інтернеті

Щоб отримати можливість переглянути відео в мережі Інтернет, відеофайл можна зберегти на «хмарному» диску (Yandex-диск, Google-диск тощо) і через пошту або соціальні мережі надіслати посилання на збережений файл тим, кому ви вважатимете за потрібне.

Існує й інший варіант розповсюдження відео — це публікація на спеціалізованих інтернет-сервісах або в соціальних мережах.

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

Для розміщення відеокліпу в Інтернеті потрібно:

- 1) увійти в акаунт YouTube (Google);
- 2) клацнути кнопку Додати відео;
- 3) зазначити тип доступу;
- 4) вибрати відео для завантаження;
- 5) ввести інформацію про відео;
- 6) клацнути Опублікувати.



Після розміщення відеокліпу в Інтернеті користувач отримує електронного листа з посиланням на фільм.

Питання для самоперевірки



1. Яке призначення відеоефектів; переходів між відеофрагментами?
2. Як додати переходи; відеоефекти?
3. Як обрізати аудіо- або відеофрагмент?
4. Як скоротити час відтворення зображення?
5. Що містить збережений файл проекту Windows Movie Maker?
6. Як опублікувати відеофільм в Інтернеті?

Вправа 13



- » Налаштувати часові параметри, додати відеопереходи між відеофрагментами.
- 1) Запустіть Windows Movie Maker і відкрийте проект з іменем Вправа 12, створений на минулому уроці. Перегляньте відеоряд фільму на екрані перегляду.
 - 2) Додайте переходи між відеофрагментами та відеоефекти.
 - 3) Перемістіть аудіофрагмент звукового супроводу так, щоб його відтворення починалося на початку фільму. Збережіть проект з іменем Вправа 13 у відповідній папці.

Комп'ютерне тестування



Виконайте тестове завдання 13 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

Практична робота 6



Створення відеокліпу. Додавання відеоефектів, налаштування часових параметрів аудіо- та відеоряду

Завдання: створити відеокліп із музичним супроводом і відеоефектами на основі раніше підготовленого проекту.

Обладнання: комп'ютер зі встановленим браузером, відеоредактором, з'єднаний із мережею Інтернет; навушники або звукові колонки; підготовлений раніше проект Вправа 13.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- ▶ 1. Запустіть відеоредактор. Відкрийте проект з іменем Вправа 13.
- ▶ 2. Перегляньте відеофрагменти, захопіть із кожного з них по одному кадру, який характеризує зміст фрагмента.
- ▶ 3. Перетягніть захоплені кадри на Шкалу часу або Панель розкадрування, розміщуючи їх перед відповідними відеофрагментами.
- ▶ 4. Установіть переходи між усіма об'єктами відеоряду.
- ▶ 5. Додайте відеоефекти до деяких або до всіх об'єктів відеоряду.
- ▶ 6. На одному із захоплених кадрів додайте текст зі стислим описом відповідного відеофрагмента.
- ▶ 7. Додайте анімовану назву на початку фільму.
- ▶ 8. Додайте в кінці відеофільму титри, у яких зазначте автора та дату створення фільму.
- ▶ 9. Вимкніть звучання доріжки Звук.
- ▶ 10. Синхронізуйте час відтворення аудіо- та відеоряду. Для цього обріжте звуковий супровід або змініть час відображення відеоряду, змінивши тривалість показу картинок, написів або обрізавши відеофрагмент.
- ▶ 11. Перегляньте відеокліп. Перевірте, щоб він не закінчувався на «півкадрі» або «півслові», за потреби відредагуйте.
- ▶ 12. Збережіть проект з іменем Практична робота 6 у відповідній папці. Завершіть роботу за комп'ютером.

Зробіть висновок: як створити відеокліп, додати відеоефекти, налаштувати часові параметри аудіо- та відеоряду, як синхронізувати аудіо- та відеоряд фільму.



Практична робота 7

Розміщення аудіо- та відеоматеріалів в Інтернеті

Завдання: опублікувати власний відеокліп у мережі Інтернет та надати до нього доступ глядачам.

Обладнання: комп'ютер зі встановленими браузером, відеоредактором, відеоконвертером, з'єднаний із мережею Інтернет; підготовлений проект Практична робота 6.

Хід роботи

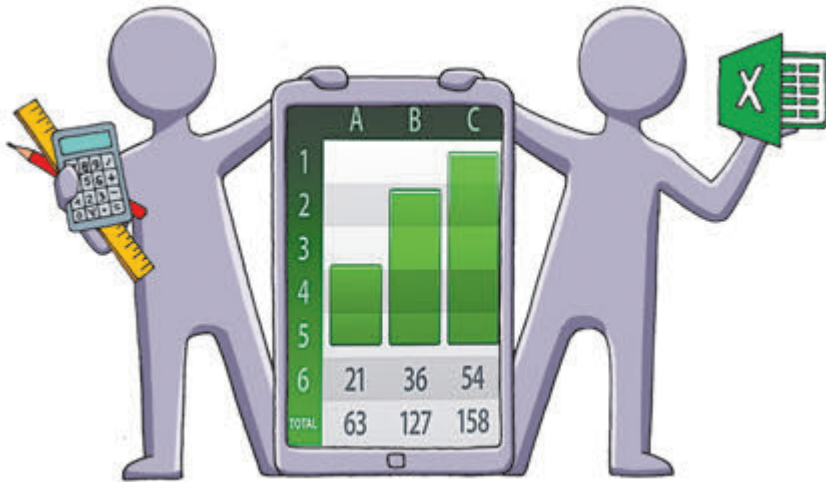
Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- ▶ 1. Підключіть навушники. Запустіть відеоредактор і відкрийте проект з іменем Практична робота 6. Перегляньте відеокліп.
- ▶ 2. Збережіть відеокліп з іменем Практична робота 7 у папці, зазначеній учителем.
- ▶ 3. Закрийте відеоредактор. Запустіть відеоконвертер за вказівкою вчителя.
- ▶ 4. Сконвертуйте файл Практична робота 7 у формат .mp4 і збережіть його у відповідній папці.
- ▶ 5. Закрийте відеоконвертер. Запустіть браузер і зайдіть у свою електронну пошту.
- ▶ 6. Зайдіть на свій «хмарний» диск.
- ▶ 7. Збережіть на «хмарному» диску свій відеокліп у форматі .mp4.
- ▶ 8. Відкрийте доступ до відеокліпу всім, хто має на нього посилання.
- ▶ 9. Надішліть електронного листа з посиланням на відеокліп на адресу, зазначену учителем.
- ▶ 10. Зареєструйтеся на сервісі YouTube (Google) та (або) увійдіть у свій акаунт.
- ▶ 11. Опублікуйте відеокліп на YouTube для загального доступу.
- ▶ 12. Надішліть електронного листа з посиланням на відеокліп в YouTube на адресу, зазначену учителем. Завершіть роботу за комп'ютером.

Зробіть висновок: як розмістити аудіо- та відеоматеріали в Інтернеті, який зі способів розміщення відеокліпу в Інтернеті є для вас більш прийнятним і чому.

РОЗДІЛ 5

ТЕХНОЛОГІЇ ОПРАЦЮВАННЯ ЧИСЛОВИХ ДАНИХ У СЕРЕДОВИЩІ ТАБЛИЧНОГО ПРОЦЕСОРА



§ 14. Типи посилань на комірки в табличному процесорі Excel
Практична робота 8. Розв'язування задач на обчислення

§ 15. Логічні функції

§ 16. Математичні та статистичні функції

§ 17. Параметри сторінки. Друкування таблиці

§ 18. Упорядкування даних. Умове форматування

Практична робота 9. Використання математичних, логічних та статистичних функцій табличного процесора.
Умове форматування

§ 19. Розширені фільтри. Проміжні підсумки

§ 20. Діаграми та графіки

Практична робота 10. Упорядкування даних у таблицях.
Автоматичні та розширені фільтри

ПОВТОРЮЄМО



Ви вже ознайомилися з табличним процесором Excel — прикладною програмою для опрацювання даних, поданих в електронних таблицях.

Документ Excel називається *робочою книгою*, вона складається з аркушів. Аркуш містить *електронну таблицю*, що складається із рядків і стовпців. На перетині рядків і стовпців електронної таблиці утворюються *комірки*. Кожна комірка має *адресу*, що складається із заголовка стовпця і номера рядка.

Комірки можуть містити дані різних типів: текстові, числові тощо, а також формули, призначені для виконання обчислень над даними. Будь-яка *формула* починається знаком «=» (дорівнює) і може містити певні значення, адреси комірок, вбудовані функції, знаки арифметичних і логічних операцій.

1. Що таке табличний процесор?
2. З чого складається адреса комірки?
3. Чи можуть збігатися адреси комірок?
4. Що можуть містити комірки електронної таблиці?
5. Для чого призначені формули?
6. З якого знака починається запис формули в табличному процесорі Excel?



У цьому розділі ви продовжите ознайомлення з функціями для опрацювання даних в електронних таблицях, розглянете методи опрацювання й упорядкування табличних даних.

§ 14. Типи посилань на комірки в табличному процесорі Excel

Під час розв'язування прикладних задач часто виникає потреба в опрацюванні даних, що містяться в кількох комірках. Так, щоб знайти найбільше або найменше число, розрахувати середнє арифметичне тощо, слід опрацювати множину чисел у певному діапазоні комірок.

Діапазони комірок



Посиланням називають адреси комірок і діапазонів комірок, використані у формулах.

Діапазон — це сукупність комірок, які можна опрацювати як єдине ціле.

Розрізняють зв'язний і незв'язний діапазони.

Зв'язним є діапазон, який можна виділити переміщенням вказівника при натиснутій лівій кнопці миші.

Для **виділення прямокутного діапазону** потрібно протягнути вказівник між кутовими діагонально протилежними комірками діапазону або клацнути одну кутову комірку діапазону та при натиснутій клавіші Shift — діагонально протилежну комірку.

Для **виділення діапазону стовпців або рядків** потрібно протягнути вказівник по заголовках стовпців або номерах рядків.

Окрема комірка також є зв'язним діапазоном, — він виділяється клацанням лівої кнопки миші.

Адреса **зв'язного діапазону** складається відповідно з адрес двох комірок: верхньої зліва і нижньої справа; номерів двох рядків (заголовків стовпців), розділених двокрапкою; адреси однієї комірки.

- A1:C5 — прямокутний діапазон із 15-ти комірок (рис. 14.1, а);
- 2:4 — діапазон з усіх комірок рядків 2, 3 і 4;
- A:B — діапазон з усіх комірок стовпців A і B;
- B7 — діапазон з однієї комірки (рис. 14.1, б).

	A	B	C	D
1				
2				
3				
4				
5				
6				
7				

а б

Рис. 14.1

Незв'язний діапазон складається зі зв'язних діапазонів. Адреса **незв'язного діапазону** складається з окремих адрес зв'язних діапазонів, розділених крапкою з комою.

» A1:C5;B7 — незв'язний діапазон із 16 комірок (рис 14.1, а, б).
 Для **виділення незв'язного діапазону** потрібно при натиснутій клавіші Ctrl послідовно виділити кілька зв'язних діапазонів.

Для **скасування всіх виділень** необхідно клацнути будь-яку комірку.

! Зв'язний діапазон можна копіювати і переміщувати, що не завжди є можливим для незв'язних діапазонів.

Типи посилань

Комірки, на які у формулах є посилання, називатимемо **адресними**. Розрізняють *відносні*, *абсолютні*, *мішані* посилання.

Пригадаємо, що адреса комірки складається із заголовка стовпця, який зазвичай позначається літерами, та номера рядка, що є числом.

Посилання, яке містить тільки адреси комірок, називають **відносним**.

При копіюванні формул відбувається модифікація відносних посилань так, що номери рядків (заголовки стовпців) змінюються на різницю номерів рядків (стовпців) вихідної й результуючої комірок з формулою.

» Якщо формулу =A4+B5 скопіювати до комірки, що міститься на два стовпці праворуч і три рядки вище від заданої, то в новій формулі всі заголовки стовпців збільшаться на 2, а номери рядків зменшаться на 3, і ми отримаємо формулу =C1+D2 (рис. 14.2).

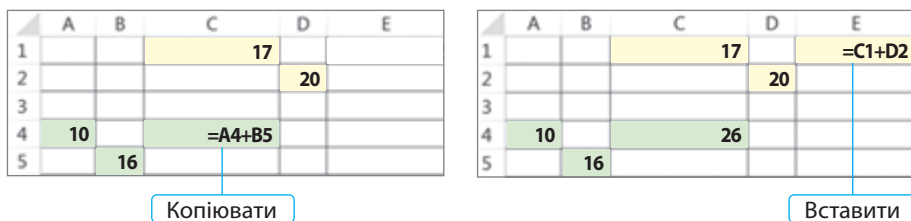


Рис. 14.2

! Якщо формула містить адресу комірки, у якій вона ж і записана, то виникає помилка «циклічне посилання».

Для **копіювання формули до суміжних комірок** можна виділити комірку з формулою та перетягнути маркер заповнення (рис. 14.3). Отримані таким чином формули будуть модифікованими.



Рис. 14.3

§ 14. Типи посилань на комірки в табличному процесорі Excel

Якщо у формулі є адреса, яка при копіюванні не повинна змінюватися, то використовують **абсолютне посилання**, що утворюється введенням знака «\$» перед заголовком стовпця і номером рядка.

➤ Якщо скопіювати до будь-якої комірки формулу $=\$A\1 , вона не змінить свого вигляду і матиме посилання на комірку з адресою A1 (рис. 14.4).

	A	B	C	D
1	7	$=\$A\1		
2	8			
3	9		$=\$A\1	
4				

Копіювати Вставити

Рис. 14.4

Якщо в посиланні на комірку заголовок стовпця і номер рядка є різнотипними (абсолютним і відносним), тобто знак «\$» міститься тільки перед заголовком стовпця або номером рядка, то таке посилання називають **мішаним**.

При копіюванні формул із мішаними посиланнями модифікуються тільки відносні заголовки стовпців або номери рядків, а абсолютні залишаються незмінними.

! Тип посилання буде змінюватися автоматично, якщо у формулі на посиланні встановити курсор і натиснути клавішу F4.

При копіюванні даних з адресних комірок до інших комірок посилання на них у формулах не модифікуються.

При переміщенні даних з адресних комірок до інших комірок усі посилання на них модифікуються залежно від їх нового місцезнаходження відповідно до типу посилань.

➤ При переміщенні даних з адресної комірки A1 до комірки C3 всі посилання A1; $\$A\1 ; A\$1; $\$A1$ відповідно модифікуються таким чином: C3; $\$C\3 ; C\$3; $\$C3$ (рис. 14.5).

Модифікуються також усі посилання при **переміщенні адресної комірки** внаслідок вставлення або видалення стовпців, рядків, комірок або діапазонів комірок, що містяться *вище* або *ліворуч* від адресної.

Видалення адресних комірок призводить до помилок у формулах, які містять посилання на видалені комірки (рис. 14.6).

	A	B	C
1			
2	$=A1$		
3			

Вирізати

	A	B	C
1			
2			$=C3$
3			

Вставити

Рис. 14.5

	A	B
1		
2	$=A1$	

Вирізати
Копіювати
Вставити
Видалити
Очистити вміст

	A	B
1		
2		###

Помилка — неприпустиме посилання на клітинку

Рис. 14.6

Питання для самоперевірки



1. Як виділити прямокутний діапазон комірок?
2. Скільки комірок містить діапазон A2:B3?
3. Як виділити діапазон стовпців або рядків?
4. Яке посилання називають відносним?
5. Як записують посилання: абсолютні, мішані?
6. Як модифікується формула =B5 при копіюванні в комірку на два стовпці ліворуч і на три рядки нижче від даної комірки?

Вправа 14



- Розрахувати масу продуктів для приготування борщу в шкільній їдальні як добуток норми їх закладки на кількість порцій.
- 1) Запустіть програму Excel. Заповніть таблицю за зразком.

	A	B	C	D	E	F	G
1		За нормою, г	85	172	155	57	
2	Класи	Порцій	М'ясо	Картопля	Капуста	Буряк	
3	8	117					
4	9	109					
5	10	107					
6	11	121					
7	Всього						

- 2) Уведіть у комірку C3 формулу для розрахунку маси м'яса, потрібного для приготування борщу для 8-х класів. Установіть числовий формат даних з одним десятковим знаком після коми.
- 3) Уведіть у комірки D3, E3, F3 формули для розрахунку маси овочів для 8-х класів. Перетягуванням маркера заповнення скопіюйте відповідні формули до діапазонів C4:C6, D4:D6, E4:E6, F4:F6.
- 4) У комірку B7 введіть формулу для розрахунку загальної кількості порцій борщу для всіх класів.
- 5) Перетягуванням маркера заповнення скопіюйте формулу з комірки B7 до діапазону комірок C7:F7. У комірку G2 введіть текст «Маса продуктів, кг», а в комірку G3 — формулу для розрахунку загальної маси всіх продуктів для 8-х класів.
- 6) Розрахуйте загальну масу продуктів у комірках G4:G6 для решти класів, у комірці G7 — для всіх класів.
- 7) Збережіть файл з іменем Вправа 14 у відповідній папці. Завершіть роботу за комп'ютером.

Комп'ютерне тестування



Виконайте тестове завдання 14 з автоматичною перевіркою на сайті interactive.ranok.com.ua.



Практична робота 8

Розв'язування задач на обчислення

Завдання: розрахувати зарплату працівників за місяць, використовуючи формули і посилання.

Обладнання: комп'ютер із табличним процесором Excel, з'єднаний із мережею Інтернет.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- **1.** Запустіть програму Excel. Створіть таблицю розрахунку зарплати за зразком (див. рисунок).

	A	B	C	D	E
1		Робочих днів у місяці			курс €
2		22			
3	Прізвище	Відпрацьованих днів	Оклад, грн	Зарплата, грн	Зарплата, євро
4	<i>Петренко</i>	20	5 900,00 ₴		
5	<i>Сидоренко</i>	13	4 880,00 ₴		
6	<i>Іванов</i>	12	6 175,00 ₴		
7	<i>Петрова</i>	17	6 266,00 ₴		
8					

- **2.** Знайдіть в Інтернеті курси валют, уведіть дані в комірку E2.
- **3.** У комірці D4 за формулою розрахуйте зарплату Петренка в гривнях. *Приклад розрахунку:* Оклад — це заробіток працівника за всі робочі дні в поточному місяці, зазначені в комірці B2. Петренко заробляє на день $5900/22 \approx 268,18$ (грн). Оскільки він відпрацював 20 днів, то його зарплата становитиме $268,18 * 20 \approx 5363,64$ (грн).
- **4.** Установіть у потрібних комірках відповідний грошовий формат даних із двома десятковими знаками після коми з позначенням грошової одиниці (грн ₴, євро €).
- **5.** Скопіюйте формулу для розрахунку зарплати Петренка у гривнях для решти працівників.
- **6.** Уведіть у комірку E4 формулу для розрахунку зарплати Петренка в євро.
- **7.** Скопіюйте отриману формулу у відповідні комірки для розрахунку зарплати кожного працівника в євро.
- **8.** Уведіть у комірку C8 формулу для розрахунку середнього окладу всіх працівників.

- 9. У комірках D8, E8 розрахуйте суму зарплати всіх працівників у гривнях і євро.
- 10. Збережіть файл з іменем Практична робота 8 у відповідній папці. Завершіть роботу за комп'ютером.

Зробіть висновок: від чого залежить правильність обчислень за скопійованими формулами; у яких випадках посилання у формулах повинні бути абсолютними і чому.

§ 15. Логічні функції

Класифікація функцій

У формулах Excel для опрацювання даних можна використовувати велику кількість вбудованих функцій, кожна з яких має унікальне ім'я. За категорією функції поділяються на *математичні, статистичні, логічні, фінансові* тощо.



Дані, які опрацьовує функція, називають **аргументами**.

Аргументами функції можуть бути інші функції, вирази, посилання тощо. Аргументи функції записують у дужках після імені (знака) функції (рис. 15.1).

Будь-яка функція в Excel характеризується не тільки іменем, а й *кількістю аргументів, типом аргументів, результатом*.



Розглянемо відому вам із 7 класу функцію для обчислення найбільшого значення (рис. 15.1):

- ім'я — MAX(МАКС);
- кількість аргументів — не менше одного;
- тип аргумента — числовий;
- результат — найбільше значення зі списку аргументів.

A			
1	=MAX B1:C2	4	1 2
			3 4

Ім'я: MAX(МАКС)
Аргументи: B1:C2
Результат: 4

Рис. 15.1

Якщо функція має кілька аргументів, їх розділяють символом «;».

35	52	70	87	105	122	140	157	175	8	3	6	9	5*	0.0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88*	3	6	9	6*	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87*	3	6	9	7*	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698	86*	3	6	9	8*	1401	1418	1435	1452	1469	1486
732	750	767	785	802	819	837	854	0.0872	85*	3	6	9	9*	1584	1601	1618	1635	1652	1669

§ 15. Логічні функції

Функції у формули вставляють інструментами вкладки Формули.

Функції також зручно вставляти у вікні Вставлення функції, яке відкривається при натисканні кнопки fx, що в рядку формул.

Потрібну функцію можна знайти за описом дії, категорією або іменем (рис. 15.2).

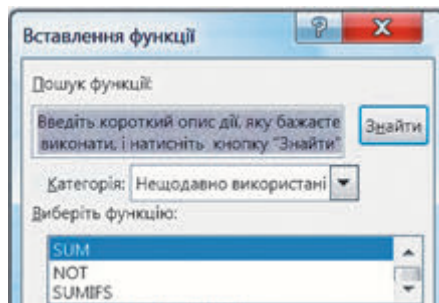


Рис. 15.2

Логічні функції

У 7 класі ви вже знайомилися з висловлюваннями, що можуть бути істинними або хибними.



Логічний вираз — це вираз, який набуває одного з двох значень: True(ИСТИНА) — істина або False(ЛОЖЬ) — хибність.

Прості логічні вирази складаються з двох виразів, пов'язаних операцією відношення:

- більше >
- менше <
- дорівнює =
- не менше (більше або дорівнює) >=
- не більше (менше або дорівнює) <=
- не дорівнює <>



$2 > 1$ — істина, $2 < 1$ — хибність.



Логічна функція — це функція, один або кілька аргументів якої — логічні вирази.

Розглянемо одну з основних логічних функцій IF(ЕСЛИ), призначену для обчислення різних виразів залежно від умови:

IF (Умова; Вираз1; Вираз2).

Вираз1 — обчислюється, якщо умова істинна.

Вираз2 — обчислюється, якщо умова хибна.



У відомості навчальних досягнень учнів біля оцінки потрібно ввести слово «Добре», якщо учень має оцінку, вищу від 6 балів, і слово «Задовільно», якщо учень має іншу оцінку.

Фрагмент алгоритму розв'язування цієї задачі подано у вигляді блок-схеми (рис. 15.3).

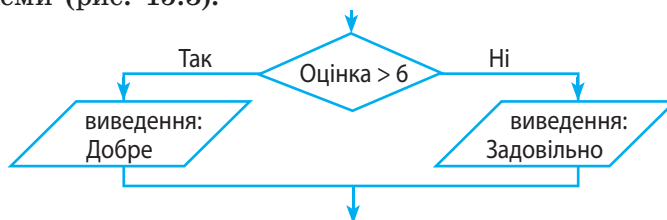


Рис. 15.3

У програмі Excel цей алгоритм реалізується за допомогою функції IF(ЕСЛИ): IF (B1>6; "Добре"; "Задовільно") (рис. 15.4).

	A	B	C	D	E
1	Петренко	5	=IF(B1>6; "Добре"; "Задовільно")		
2	Сидоренко	8	Добре		
3	Петрова	12	Добре		
4	Сидоров	4	Задовільно		

Рис. 15.4

У функціях Excel логічний вираз може містити адреси комірок та/або арифметичні вирази. У цьому випадку перевіряється, чи не дорівнює значення виразу нулю, і відповідно повертається значення логічного виразу False або True.

- 2 + 2 трактується як *істина* (тому що 2 + 2 ≠ 0);
- 2 - 2 трактується як *хибність* (тому що 2 - 2 = 0).

Іноді для розв'язання задачі потрібно комбінувати кілька умов.



Складеним логічним виразом називають вираз, у якому кілька простих логічних виразів пов'язані логічними операціями.

Основними логічними операціями є НЕ (заперечення), І (кон'юнкція або логічне множення), АБО (диз'юнкція, логічне додавання). В Excel вони реалізуються за допомогою функцій NOT(НЕ), AND(И), OR(ИЛИ).

- У відомості навчальних досягнень (див. рис. 15.4) біля оцінки потрібно ввести слово «Посередньо» для тих учнів, які мають оцінки, вищі від 6 балів, але нижчі за 10, для решти учнів — нічого не писати.

Скористаємося логічними функціями IF(ЕСЛИ) та AND(И).

35	52	70	87	105	122	140	157	0.0000	9	3	6	9	5*	0.0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88*	3	6	9	6*	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87*	3	6	9	7*	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698	86*	3	6	9	8*	1461	1478	1495	1512	1529	1546
732	750	767	785	802	819	837	854	0.0872	85*	3	6	9	9*	1713	1730	1747	1764	1781	1798

§ 15. Логічні функції

У даному випадку маємо: $=IF(AND(B1>6;B1<10); "Посередньо"; "")$.

Останній аргумент функції IF(ЕСЛИ) — це лапки без символів між ними, тобто порожній рядок.

Іноді буває доцільно використати вкладені логічні функції IF(ЕСЛИ).

➤ Нехай потрібно охарактеризувати рівень навчальних досягнень учнів за трьома умовними рівнями: «Початковий» (1–4 бали); «Середній» (5–8 балів); «Високий» (9–12 балів).

В Excel цю задачу можна розв'язати таким чином (рис. 15.5):

$=IF(B1<5;"Початковий"; IF(B1<9;"Середній";"Високий"))$

$=ЕСЛИ(B1<5;"Початковий"; ЕСЛИ(B1<9;"Середній";"Високий"))$

В Excel допускається вкладати одна в одну кілька функцій IF(ЕСЛИ).

	A	B	C	D	E	F	G	H	I
1	Петренко	5	$=IF(B1<5;"Початковий"; IF(B1<9;"Середній";"Високий"))$						
2	Сидоренко	8	Середній						
3	Петров	12	Високий						
4	Сидоров	4	Початковий						

Рис. 15.5

Розглянемо ще кілька логічних функцій (див. таблицю).

Ім'я (англ./рос.)	Опис	Кількість аргументів
IFERROR (ЕСЛИОШИБКА)	Якщо перший аргумент не є помилкою, то повертає його значення, інакше повертає значення другого аргумента	Два
FALSE(ЛОЖЬ)	Повертає значення «хибність»	Без аргументів
TRUE(ИСТИНА)	Повертає значення «істина»	Без аргументів

Питання для самоперевірки



1. Що таке функція?
2. Що таке аргумент функції?
3. Назвіть кілька категорій функцій в Excel.
4. Чи існують функції без аргументів в Excel?
5. Як позначають операції відношень?
6. Що є першим аргументом логічної функції IF(ЕСЛИ)?

18*	24	
924	941	958
1097	1115	1132
1271	1288	1305
1461	1478	1495

Вправа 15



- ▶ Створити в Excel таблицю навчальних досягнень учнів з інформатики за описом.
- 1) Запустіть програму Excel. Створіть і заповніть таблицю навчальних досягнень учнів вашої групи (класу) з інформатики: уведіть у стовпці А прізвища, у стовпці В — оцінки за семестр.
 - 2) Створіть формулу з використанням функції IF(ЕСЛИ) для запису в стовпці С рівнів навчальних досягнень учнів: «Початковий» (оцінка, менша за 5 балів) і «Високий» (для решти оцінок) за аналогією з прикладом на рис. 15.4.
 - 3) Створіть формулу для запису в стовпці D рівнів навчальних досягнень учнів: «Юніор» (1–4 бали), «Аматор» (5–8 балів), «Експерт» (9–12 балів).
 - 4) Збережіть файл з іменем Вправа 15 у відповідній папці. Завершіть роботу за комп'ютером.

Комп'ютерне тестування



Виконайте тестове завдання 15 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§16. Математичні та статистичні функції

Математичні функції використовують для виконання різних математичних обчислень. З повним переліком математичних функцій можна ознайомитись на вкладці Функції у групі Бібліотека функцій.

Математичні функції



Математичні функції в Excel — це функції, призначені для виконання обчислень над числами та/або даними комірок: округлення, перетворення чисел тощо.

35	52	70	87	95	122	140	157	175	8	3	6	9	5*	0.0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88*	3	6	9	6*	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87*	3	6	9	7*	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698	86*	3	6	9	8*	1461	1478	1495	1512	1529	1546
732	750	767	785	802	819	837	854	0.0872	85*	3	6	9	9*	1715	1732	1749	1766	1783	1800

§16. Математичні та статистичні функції

Розглянемо деякі математичні функції (див. таблицю).

Ім'я (англ./рос.)	Опис дії	Кількість аргументів
ROUND(ОКРУГЛ)	Округлення до вказаної кількості цифр після коми	Два
MOD(ОСТАТ)	Остача від ділення	Два
PRODUCT(ПРОИЗВЕД)	Добуток	Не менше одного
SUM(СУММ)	Сума	Не менше одного
SUMIF(СУММЕСЛИ)	Сума за умови, що...	Три

➤ Якщо знайти добуток числових даних у діапазоні комірок A1:B2 й округлити його до десятих, то формула матиме вигляд:
 =ROUND(PRODUCT(A1:B2);1) або =ОКРУГЛ(ПРОИЗВЕД(A1:B2);1)

Статистичні функції



Статистичні функції в Excel — це функції, призначені для аналізу значень діапазонів комірок.

За допомогою статистичних функцій можна знайти найбільше і найменше значення, розрахувати середнє значення тощо.

➤ Визначити, скільки учнів навчаються на 12 балів, можна за допомогою статистичної функції COUNTIF(СЧЕТЕСЛИ). Першим аргументом є діапазон комірок з оцінками, а другим аргументом — критерій вибору:

=COUNTIF(B3:B4;"=12") або =СЧЕТЕСЛИ(B3:B4;"=12")

Знак рівності й лапки в цій логічній умові можна пропустити:

=COUNTIF(B3:B4;12) або =СЧЕТЕСЛИ(B3:B4;12)

Критерієм вибору може бути не тільки число, але й дата, символ, слово тощо. За критерієм будується умова: чи містить поточна комірка зазначені дані. Наприклад, можна обчислити кількість пропущених учнем уроків (див. рисунок).

	A	B	C	D	E	F	G	H	I	
1		дата								
2	Прізвище	02.СЕН	09.СЕН	16.СЕН	23.СЕН	30.СЕН	07.ОКТ	кількість пропусків		
3	Петренко	6	7	8	8	8	8	=COUNTIF(B3:G3;"")		
4	Іванов	9			8		9			

Критерій вибору

Розглянемо деякі статистичні функції (див. таблицю).

Ім'я (англ./рос.)	Опис	Кількість аргументів
MIN(МИН)	Мінімальне значення	Не менше одного
MAX(МАКС)	Максимальне значення	Не менше одного
AVERAGE(СРЗНАЧ)	Середнє арифметичне	Не менше одного
COUNT(СЧЕТ)	Кількість комірок із числовими даними	Не менше одного
COUNTIF(СЧЕТЕСЛИ)	Кількість комірок, дані в яких задовольняють певній умові	Не менше одного
COUNTIFS (СЧЕТЕСЛИМН)	Кількість комірок, дані в яких задовольняють декільком умовам	Не менше одного
COUNTA(СЧЕТЗ)	Кількість непорожніх комірок	Не менше одного

Питання для самоперевірки



1. Яке призначення математичних функцій?
2. Наведіть приклади математичних функцій в Excel.
3. Для чого призначені статистичні функції?
4. Наведіть приклади статистичних функцій в Excel.
5. Наведіть приклад застосування однієї зі статистичних функцій.
6. Наведіть приклад статистичної функції, аргументом якої може бути логічний (умовний) вираз.

Вправа 16



- Проаналізувати в Excel успішність учнів вашої групи (класу) з інформатики за описом.
- 1) Запустіть програму Excel. Створіть таблицю зі стовпцями, що мають такі заголовки: А — «Прізвище», В, С, D — поточні оцінки, Е — «Підсумкова». Введіть у таблицю прізвища учнів та поточні оцінки.
 - 2) У стовпці Е за допомогою відповідних функцій обчисліть підсумкові оцінки учнів як середнє арифметичне поточних оцінок, округлене до цілого значення.
 - 3) У довільній комірці під прізвищами за допомогою функції COUNTIFS(СЧЕТЕСЛИМН) розрахуйте, скільки учнів мають підсумкові оцінки у 5–8 балів.

35	52	70	87	105	122	140	157	175	192	210	227	244	262	279	297	314	332	349	366	384	401	419	436	454	471	488	506	523	541	558	576	593	610	628	645	663	680	698	715	732	750	767	785	802	819	837	854	872	889	906	924	941	958																					
209	227	244	262	279	297	314	332	349	366	384	401	419	436	454	471	488	506	523	541	558	576	593	610	628	645	663	680	698	715	732	750	767	785	802	819	837	854	872	889	906	924	941	958	976	993	1011	1028	1046	1063	1081	1098	1116	1133	1151	1168	1186	1203	1221	1238	1256	1273	1291	1308	1326	1343	1361	1378	1396	1413	1431	1448	1466	1483	1501

§ 17. Параметри сторінки. Друкування таблиці

- У довільній комірці під прізвищами розрахуйте загальну кількість поточних оцінок усіх учнів.
- Збережіть файл з іменем Вправа 16 у відповідній папці. Завершіть роботу за комп'ютером.



Комп'ютерне тестування

Виконайте тестове завдання 16 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 17. Параметри сторінки. Друкування таблиці

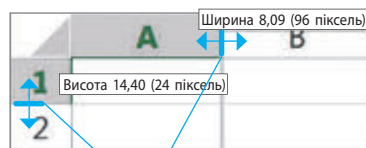
Перед тим, як надрукувати електронну таблицю, слід встановити параметри сторінки. У табличному процесорі Excel таке налаштування здійснюється, як у текстовому процесорі Word.

Встановлення параметрів сторінки

Для встановлення параметрів сторінки слід у вікні Параметри сторінки на вкладці Розмітка сторінки вибрати потрібний інструмент (наприклад поля) та одне із запропонованих значень або ввести потрібне значення з клавіатури.

Встановлення розмірів стовпців і рядків таблиці

Щоб змінити ширину стовпця (висоту рядка), достатньо перетягнути його праву (нижню) межу між заголовками. При цьому розміри сусідніх стовпців (рядків) не змінюються, а в інформаційному вікні в режимі перегляду Вигляд → Звичайний виводиться поточний розмір: у пікселях та пунктах — для рядків, у символах і пікселях — для стовпців (рис. 17.1).



Перетягування межі

Рис. 17.1

Для встановлення однакової ширини кількох стовпців (рис. 17.2, а) або висоти кількох рядків (рис. 17.2, б) їх потрібно виділити, наприклад, протягнувши вказівник по заголовках рядків (стовпців), а потім перетягнути одну з меж або вибрати в контекстному меню Ширина стовпця (Висота рядка) і встановити потрібне значення.

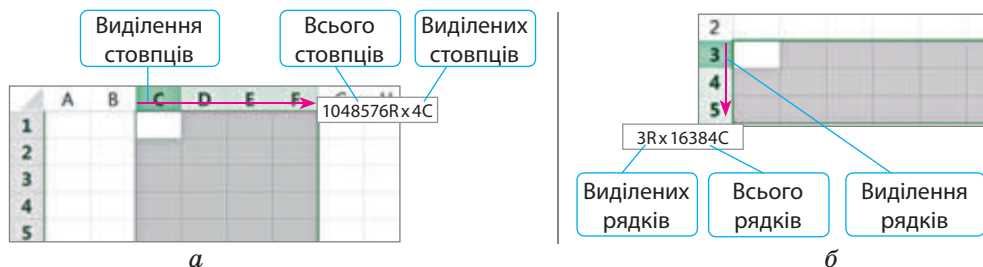


Рис. 17.2

Оптимізація таблиці

Якщо таблиця не вміщується на сторінці (рис. 17.3), можна здійснити її оптимізацію: змінити орієнтацію сторінки, розмір та орієнтацію тексту, розміри рядків і стовпців тощо (рис. 17.4).

A	B	C	D	E	F	G	H	I
Прізвище	Інформатика	Фізика	Хімія	Алгебра	Геометрія	Образотворче мистецтво	Астрономія	Географія
Петренко	12	10	8	9	11	9	10	12
Іваненко	7	8	9	8	7	8	9	8
Петров	5	6	5	7	6	5	6	7

Рис. 17.3

Прізвище	Інформатика	Фізика	Хімія	Алгебра	Геометрія	Образотворче мистецтво	Астрономія	Географія
Петренко	12	10	8	9	11	9	10	12
Іваненко	7	8	9	8	7	8	9	8
Петров	5	6	5	7	6	5	6	7

Рис. 17.4

Для оптимізації таблиці (рис. 17.3) необхідно:


- 1) виділити потрібні комірки верхнього рядка;
- 2) на вкладці Основне розгорнути параметри вирівнювання;
- 3) налаштувати у вікні Формат комірок вирівнювання й орієнтацію тексту;
- 4) змінити висоту верхнього рядка і ширину стовпців з оцінками.

35	52	70	87	105	122	140	157	175	192	210	227	244	262	279	297	314	332	349	366	384	401	419	436	454	471	488	506	523	541	558	576	593	610	628	645	663	680	698	715	733	750	767	785	802	819	837	854	872	889	906	924	941	958	976	993																			
209	227	244	262	279	297	314	332	349	366	384	401	419	436	454	471	488	506	523	541	558	576	593	610	628	645	663	680	698	715	733	750	767	785	802	819	837	854	872	889	906	924	941	958	976	993	1010	1028	1045	1063	1080	1097	1115	1132	1150	1167	1185	1202	1220	1237	1255	1272	1289	1307	1324	1342	1359	1377	1394	1412	1429	1447	1464	1482	1499

§ 17. Параметри сторінки. Друкування таблиці

Об'єднання комірок

Щоб зробити спільний заголовок для кількох стовпців таблиці, потрібно об'єднати кілька комірок в одну.

Для **об'єднання комірок** потрібно їх виділити і вибрати на вкладці Основне в групі Вирівнювання кнопку Об'єднати й помістити у центрі . У комірці залишаться тільки дані комірки, розташованої зверху зліва.

Для **скасування об'єднання комірок** потрібно повторно клацнути ту саму кнопку, при цьому дані об'єднаних комірок не відновлюються.

Встановлення меж комірок

Таблиця, яку ми бачимо на робочому аркуші, розбита на комірки умовно. Якщо її роздрукувати, то на папері відобразяться тільки дані в комірках, не буде жодної лінії або нумерації.

Для **встановлення меж комірок** необхідно виконати такі кроки:

- 1) виділити комірку або діапазон;
- 2) на вкладці Основне вибрати групу Вирівнювання;
- 3) у вікні Формат комірок на вкладці Межі вибрати потрібний формат меж: тип, колір, розташування (рис. 17.5, с. 92).

Друкування таблиці

Друк електронної таблиці Excel здійснюється подібно до друку текстового документа у Word. Під час друку на папері виводяться тільки ті комірки таблиці, де є вміст, межі або заливка.

Під час підготовки документа до друку зручно працювати в режимі Вигляд → Розмітка сторінки, оскільки при цьому розміри рядків і стовпців зазначаються в сантиметрах і відображається орієнтовний результат друку.

Для **підготовки електронної таблиці до друку** потрібно:

- 1) встановити параметри сторінки;
- 2) оптимізувати таблицю форматкуванням її об'єктів;
- 3) об'єднати діапазони комірок (за необхідності);
- 4) встановити межі комірок у таблиці.

До **друку документа** можна перейти за допомогою команди Файл → Друк або сполучення клавіш Ctrl + P.

18'	24	
924	111	958
1097	1115	1132
1271	1288	1305
	1461	1478

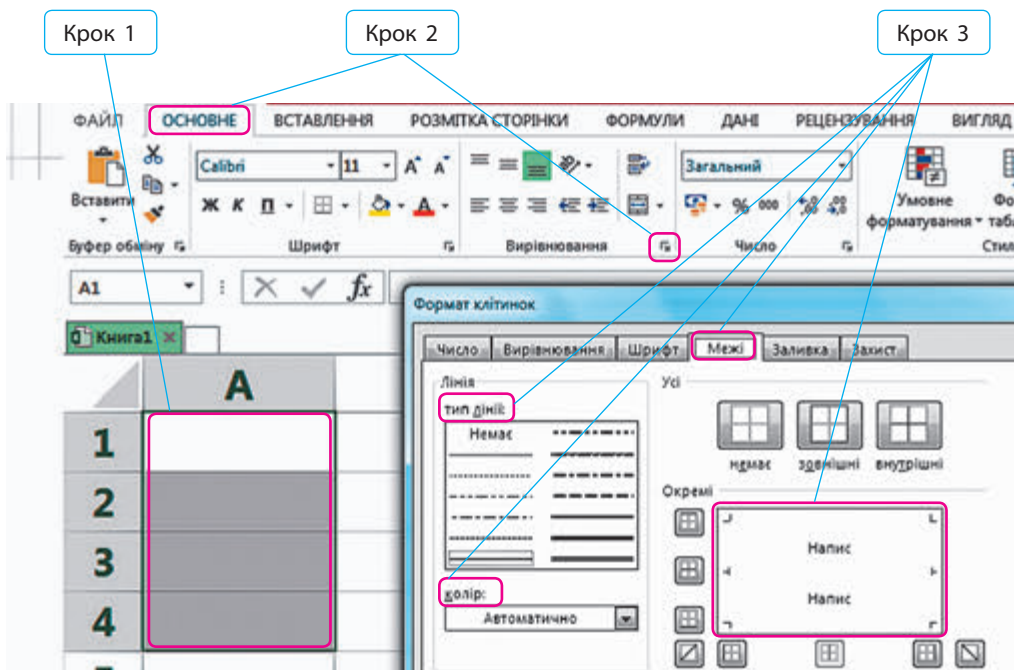


Рис. 17.5

Питання для самоперевірки



1. Як змінити розмір одного стовпця або рядка таблиці?
2. Як встановити однаковий розмір для кількох стовпців (рядків)?
3. Що передбачає оптимізація таблиці?
4. Як об'єднати кілька комірок таблиці?
5. Як налаштувати вигляд заливки або меж комірок таблиці?
6. Як підготувати електронну таблицю до друку?

Вправа 17



- ▶ Створити електронну таблицю за поданим описом.
- 1) Запустіть програму Excel. Установіть параметри сторінки: усі поля — по 1 см, розмір — А4 (21×29,7 см), орієнтація — книжкова. Установіть режим перегляду сторінки: Вигляд → → Розмітка сторінки.

35	52	70	87	105	122	140	157	175	192	210	227	244	262	279	297	314	332	349	366	384	401	419	436	454	471	488	506	523	541	558	576	593	610	628	645	663	680	698	715	733	750	767	785	802	819	837	854	872	889	906	924	941	958	976	993	1011	1028	1045	1063	1080	1097	1115	1132	1150	1167	1184	1202	1219	1236	1253	1271	1288	1305	1322	1340	1357	1374	1392	1409	1426	1443	1461	1478	1495	1513	1530	1547	1565	1582	1600	1617	1634	1652	1669	1686	1704	1721	1738	1756	1773	1790	1808	1825	1843	1860	1877	1895	1912	1929	1947	1964	1981	1999	2016	2033	2051	2068	2085	2103	2120	2137	2155	2172	2189	2207	2224	2241	2259	2276	2293	2311	2328	2345	2363	2380	2397	2415	2432	2449	2466	2484	2501	2518	2536	2553	2570	2588	2605	2622	2640	2657	2674	2692	2709	2726	2744	2761	2778	2796	2813	2830	2848	2865	2882	2900	2917	2934	2952	2969	2986	3004	3021	3038	3056	3073	3090	3108	3125	3142	3160	3177	3194	3212	3229	3246	3264	3281	3298	3316	3333	3350	3368	3385	3402	3420	3437	3454	3472	3489	3506	3524	3541	3558	3576	3593	3610	3628	3645	3662	3680	3697	3714	3732	3749	3766	3784	3801	3818	3836	3853	3870	3888	3905	3922	3940	3957	3974	3992	4009	4026	4044	4061	4078	4096	4113	4130	4148	4165	4182	4200	4217	4234	4252	4269	4286	4304	4321	4338	4356	4373	4390	4408	4425	4442	4460	4477	4494	4512	4529	4546	4564	4581	4598	4616	4633	4650	4668	4685	4702	4720	4737	4754	4772	4789	4806	4824	4841	4858	4876	4893	4910	4928	4945	4962	4980	4997	5014	5032	5049	5066	5084	5101	5118	5136	5153	5170	5188	5205	5222	5240	5257	5274	5292	5309	5326	5344	5361	5378	5396	5413	5430	5448	5465	5482	5500	5517	5534	5552	5569	5586	5604	5621	5638	5656	5673	5690	5708	5725	5742	5760	5777	5794	5812	5829	5846	5864	5881	5898	5916	5933	5950	5968	5985	6002	6020	6037	6054	6072	6089	6106	6124	6141	6158	6176	6193	6210	6228	6245	6262	6280	6297	6314	6332	6349	6366	6384	6401	6418	6436	6453	6470	6488	6505	6522	6540	6557	6574	6592	6609	6626	6644	6661	6678	6695	6713	6730	6747	6765	6782	6800	6817	6834	6852	6869	6886	6904	6921	6938	6956	6973	6990	7008	7025	7042	7060	7077	7094	7112	7129	7146	7164	7181	7198	7216	7233	7250	7268	7285	7302	7320	7337	7354	7372	7389	7406	7424	7441	7458	7476	7493	7510	7528	7545	7562	7580	7597	7614	7632	7649	7666	7684	7701	7718	7736	7753	7770	7788	7805	7822	7840	7857	7874	7892	7909	7926	7944	7961	7978	7996	8013	8030	8048	8065	8082	8100	8117	8134	8152	8169	8186	8204	8221	8238	8256	8273	8290	8308	8325	8342	8360	8377	8394	8412	8429	8446	8464	8481	8498	8516	8533	8550	8568	8585	8602	8620	8637	8654	8672	8689	8706	8724	8741	8758	8776	8793	8810	8828	8845	8862	8880	8897	8914	8932	8949	8966	8984	9001	9018	9036	9053	9070	9088	9105	9122	9140	9157	9174	9192	9209	9226	9244	9261	9278	9296	9313	9330	9348	9365	9382	9400	9417	9434	9452	9469	9486	9504	9521	9538	9556	9573	9590	9608	9625	9642	9660	9677	9694	9712	9729	9746	9764	9781	9798	9816	9833	9850	9868	9885	9902	9920	9937	9954	9972	9989	10006	10024	10041	10058	10076	10093	10110	10128	10145	10162	10180	10197	10214	10232	10249	10266	10284	10301	10318	10336	10353	10370	10388	10405	10422	10440	10457	10474	10492	10509	10526	10544	10561	10578	10596	10613	10630	10648	10665	10682	10700	10717	10734	10752	10769	10786	10804	10821	10838	10856	10873	10890	10908	10925	10942	10960	10977	10994	11012	11029	11046	11064	11081	11098	11116	11133	11150	11168	11185	11202	11220	11237	11254	11272	11289	11306	11324	11341	11358	11376	11393	11410	11428	11445	11462	11480	11497	11514	11532	11549	11566	11584	11601	11618	11636	11653	11670	11688	11705	11722	11740	11757	11774	11792	11809	11826	11844	11861	11878	11896	11913	11930	11948	11965	11982	12000	12017	12034	12052	12069	12086	12104	12121	12138	12156	12173	12190	12208	12225	12242	12260	12277	12294	12312	12329	12346	12364	12381	12398	12416	12433	12450	12468	12485	12502	12520	12537	12554	12572	12589	12606	12624	12641	12658	12676	12693	12710	12728	12745	12762	12780	12797	12814	12832	12849	12866	12884	12901	12918	12936	12953	12970	12988	13005	13022	13040	13057	13074	13092	13109	13126	13144	13161	13178	13196	13213	13230	13248	13265	13282	13300	13317	13334	13352	13369	13386	13404	13421	13438	13456	13473	13490	13508	13525	13542	13560	13577	13594	13612	13629	13646	13664	13681	13698	13716	13733	13750	13768	13785	13802	13820	13837	13854	13872	13889	13906	13924	13941	13958	13976	13993	14010	14028	14045	14062	14080	14097	14114	14132	14149	14166	14184	14201	14218	14236	14253	14270	14288	14305	14322	14340	14357	14374	14392	14409	14426	14444	14461	14478	14496	14513	14530	14548	14565	14582	14600	14617	14634	14652	14669	14686	14704	14721	14738	14756	14773	14790	14808	14825	14842	14860	14877	14894	14912	14929	14946	14964	14981	15000
----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

§ 17. Параметри сторінки. Друкування таблиці

- 2) Налаштуйте ширину стовпців: A — 0,5 см, B — 3,5 см. Іншим 27 стовпцям таблиці задайте ширину 0,5 см.
- 3) Установіть висоту першого рядка — 1,5 см. Іншим 48 рядкам (з 2-го по 49-й) установіть висоту 0,5 см, 4-му рядку — 1,5 см.
- 4) Об'єднайте комірки першого рядка над таблицею та потрібні комірки в таблиці. Верхня частина таблиці має виглядати, як показано на рисунку.



- 5) Виділіть діапазон комірок A2:AC49. Установіть межі (крім бічних) мінімальної товщини. У найбільшій комірці встановіть діагональну межу середньої товщини. Виділіть потрібні діапазони і встановіть межі різної товщини, як показано на рисунку.

- 6) Збережіть файл з іменем Вправа 17 у відповідній папці. Завершіть роботу за комп'ютером.



Комп'ютерне тестування

Виконайте тестове завдання 17 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

18°	24	
924	958	
1097	1115	1132
1271	1288	1305
	1461	1478

§ 18. Упорядкування даних. Умове форматування

Зазвичай дані електронної таблиці, отримані в результаті розрахунків або після введення з клавіатури, потребують упорядкування й систематизації.

Сортування даних



Сортування даних в Excel — це встановлення порядку розташування рядків за даними певного стовпця (стовпців).

Дані можна сортувати *за зростанням* (в алфавітному порядку) або *спаданням* (у порядку, зворотному до алфавітного).

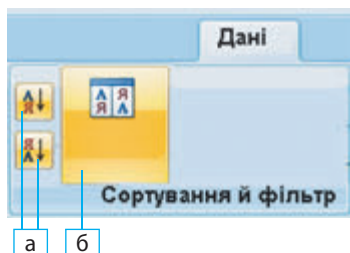


Рис. 18.1

Щоб виконати **сортування даних** за зростанням або спаданням, спочатку потрібно виділити зв'язний діапазон комірок із даними.

Для **простого сортування даних за зростанням або спаданням** потрібно на вкладці Дані в групі Сортування й фільтр натиснути кнопку відповідного інструмента (рис. 18.1, а).

При цьому рядки виділеного діапазону переставляються за даними першого стовпця, порожні комірки якого розташуються знизу незалежно від типу сортування. Якщо виділити один стовпець, то дані можуть відсортуватися тільки в ньому або в розширеному діапазоні за даними виділеного стовпця. Який саме діапазон сортувати при такому виділенні, визначає користувач.

Для **складного сортування даних за кількома умовами** слід на вкладці Дані натиснути кнопку відкриття діалогового вікна Сортування в групі Сортування й фільтр (рис. 18.1, б), у якому зазначити умови сортування: за даними яких стовпців, за якими параметрами, у якому порядку тощо.

При налаштуванні складного сортування за даними кількох стовпців дані верхнього рядка виділеного діапазону вважаються заголовками і сортуванню не підлягають.

35	52	70	87	105	122	140	157	175	192	210	227	244	262	279	297	314	332	349	367	384	401	419	436	454	471	488	506	523	541	558	576	593	610	628	645	663	680	698	715	732	750	767	785	802	819	837	854	872	889	906	924	941	958
----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----

§ 18. Упорядкування даних. Умовне форматування

Порядок одночасного сортування за кількома стовпцями задає користувач у вікні Сортування. При цьому за кожним наступним стовпцем сортуватимуться лише ті рядки, в яких у попередньому відсортованому стовпці дані збігалися (рис. 18.2).

Не відсортовано Початкова таблиця			Відсортовано за ім'ям Проміжний етап			Відсортовано за прізвищем Проміжний етап			Відсортовано за оцінкою Кінцева таблиця		
Прізвище	Ім'я	Оцінка	Прізвище	Ім'я	Оцінка	Прізвище	Ім'я	Оцінка	Прізвище	Ім'я	Оцінка
Бабич	Віктор	11	Бабич	Віктор	11	Бабич	Віктор	11	Бабич	Віктор	11
Бойко	Тарас	12	Валеев	Олег	10	Валеев	Олег	10	Валеев	Олег	10
Зленко	Тарас	10	Бойко	Тарас	12	Бойко	Тарас	12	Бойко	Тарас	9
Валеев	Олег	10	Зленко	Тарас	10	Бойко	Тарас	9	Бойко	Тарас	12
Бойко	Тарас	9	Бойко	Тарас	9	Зленко	Тарас	10	Зленко	Тарас	10

Рис. 18.2

Фільтрування даних



Фільтрування даних в Excel — це відображення в таблиці тільки тих рядків, дані яких відповідають певним умовам (умовам фільтрування).

При фільтруванні решта рядків таблиці (рядки, що не відповідають умовам фільтрування) приховуються і можуть бути відображені за інших умов фільтрування.

Верхній рядок виділеного діапазону не фільтрується і завжди є видимим, тому в нього зазвичай вписують заголовки стовпців.

Для ввімкнення режиму фільтрування потрібно активувати кнопку Фільтр (рис. 18.3). При цьому для поточної комірки у верхньому рядку зв'язаного з нею діапазону з'являться значки розгортання фільтра ▼.

Значки розгортання фільтра

A	B	C	D	E	F
Іванов	7	10			
Тесленко	9	8	7		
Сидоров			9	12	
Лешченко					11



Рис. 18.3

18'	24	
924	958	
1097	1132	
1271	1305	
1461	1478	

Зазвичай для фільтрування виділяють потрібну кількість стовпців таблиці разом із заголовками. Тоді при ввімкненні режиму фільтрування значки розгортання фільтра з'являться у всіх комірках рядка заголовків.

Якщо колір шрифту або заливки відрізняється від чорного і білого, можна встановити Фільтр за кольором.

Для стовпців із числовими даними активним буде пункт Фільтри чисел.

Для налаштування діапазону, відсутнього в переліку меню Фільтри чисел, наприклад *більше п'яти та менше дев'яти*, слід вибрати Користувацький автофільтр.

Для стовпців з текстовими даними активним буде пункт Текстові фільтри (рис. 18.4).

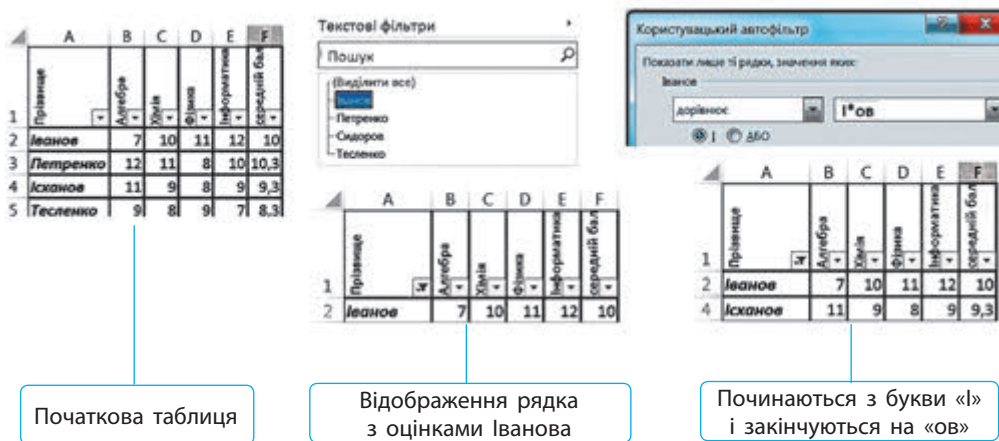


Рис. 18.4

Після встановлення нового фільтра в тому самому стовпці «старе» правило фільтрування анулюється.

Фільтри можна застосовувати до кількох стовпців одночасно. При цьому кожен фільтр застосовується до рядків, які залишилися видимими після попереднього фільтрування.

Умове форматування

Щоб комірки з певними даними вирізнялися серед інших комірок, застосовують умове форматування.

35	52	70	87	105	122	140	157	175	8	3	6	9	54	0.0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88	3	6	9	6	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87	3	6	9	7	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	697	85	3	6	9	7	1461	1478	1495	1512	1529	1546
732	750	767	785	802	819	837	854	871	84	3	6	9	7	1713	1730	1747	1764	1781	1798

§ 18. Упорядкування даних. Умовне форматування



Умовне форматування комірок — автоматична зміна формату комірки на заданий, якщо значення даних в ній відповідають певній умові.

Для **умовного форматування** треба виділити діапазон комірок, вибрати команду Основне → Стилі → Умовне форматування і вибрати потрібне.

Умовне форматування надає можливість оформити комірки таблиці за стандартними правилами і стилями або ж встановити свої правила і стилі оформлення комірок, дані в яких відповідають деякій умові (команда Інші правила).

До комірки можна застосовувати кілька правил умовного форматування. Якщо ж такі правила суперечать одне одному, то «чинним» буде правило, встановлене пізніше.



Якщо для того самого діапазону встановити червону заливку комірок зі значеннями, більшими за 5, а потім встановити зелену заливку комірок зі значеннями, меншими за 7, то в підсумку комірки зі значенням 6 зафарбуються зеленим кольором, оскільки відповідне правило встановлено останнім (рис. 18.5).

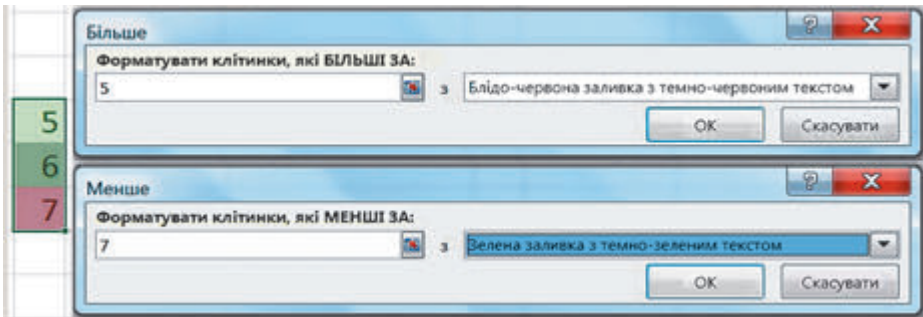


Рис. 18.5

Питання для самоперевірки



1. У чому полягає сортування даних?
2. Як відбувається сортування за даними кількох стовпців?
3. У чому полягає фільтрування даних?
4. Що відбувається під час фільтрування даних таблиці за кількома стовпцями?
5. Що таке умовне форматування?
6. Що відбувається, коли правила умовного форматування даних суперечать одне одному?

18	24	
924	941	958
1097	1115	1132
1271	1288	1305
1461	1478	1495

Вправа 18



- ▶ Створити таблицю, здійснити умовне форматування та впорядкування даних за описом.
- Запустіть програму Excel. Створіть таблицю за зразком (див. рисунок).
 - Налаштуйте умовне форматування комірок діапазону B3:E7:
 - 1-ше місце — червона заливка;
 - 2-ге місце — зелена заливка;
 - 3-тє місце — жовта заливка (колір шрифту — на власний вибір).
 - Налаштуйте умовне форматування комірок для непризових місць: шрифт — напівжирний, колір шрифту — синій, без заливки комірок.
 - Упорядкуйте дані рядків таблиці так, щоб навчальні предмети розташовувалися в алфавітному порядку.
 - Додайте до таблиці фільтри для відображення лише тих предметів, з яких учні школи у 2015 році посіли перше місце.
 - Збережіть файл з іменем Вправа 18 у відповідній папці. Завершіть роботу за комп'ютером.

	A	B	C	D	E
1	Досягнення (місце)				
2	Предмет	2013	2014	2015	2016
3	Математика	2	4	2	3
4	Фізика	4	4	3	3
5	Хімія	3	2	1	2
6	Біологія	4	2	7	5
7	Інформатика	3	1	1	3

Комп'ютерне тестування



Виконайте тестове завдання 18 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

Практична робота 9

Використання математичних, логічних та статистичних функцій табличного процесора.
Умове форматування



Завдання: створити таблицю успішності учнів, взявши за зразок шкільний журнал, з автоматичним обчисленням підсумкових оцінок та рівня навчальних досягнень учнів.

Обладнання: комп'ютер зі встановленим табличним процесором Excel.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- ▶ 1. Запустіть програму Excel. Установіть параметри сторінки: усі поля — по 1 см, колонтитули — 0 см, орієнтація — книжкова, розмір паперу — А4.
- ▶ 2. Створіть таблицю за зразком (див. рисунок). Об'єднайте потрібні комірки, змініть розміри відповідних стовпців та рядків. Установіть для таблиці межі середньої товщини.
- ▶ 3. У відповідних комірках встановіть потрібну орієнтацію тексту та формати даних: для оцінок — числовий без десяткових знаків, для написів — текстовий, для дат — короткий формат дати.
- ▶ 4. Впишіть прізвища учнів і їх оцінки, а також інші текстові написи.
- ▶ 5. Відсортуйте прізвища учнів в алфавітному порядку.
- ▶ 6. Уведіть нумерацію і дати в режимі автозаповнення.
- ▶ 7. Обчисліть відповідною функцією підсумкову оцінку кожного учня як округлене середнє арифметичне його оцінок.
- ▶ 8. Функцією COUNTIF(СЧЕТЕСЛИ) розрахуйте кількість відсутніх у кожній із зазначених днів.
- ▶ 9. Функцією COUNTIF за підсумковими оцінками розрахуйте кількість учнів, оцінки яких відповідають початковому (менше за 4 бали) і високому (більше від 9 балів) рівням.
- ▶ 10. Функцією COUNTIFS(СЧЕТЕСЛИМН) за підсумковими оцінками учнів розрахуйте кількість тих, хто має достатній (від 7 до 9 балів) та середній (від 4 до 6 балів) рівні.
- ▶ 11. Налаштуйте умовне форматування комірок з оцінками залежно від рівня досягнень учнів: високий — червоний колір тексту, середній — зелений, достатній — синій, початковий — чорний (без заливки комірок кольором).

	A	B	C	D	E	F	G	H	I	J	K	L	M
1			Дата										
2	№ п/п	Прізвище	9.2	16.2	23.2	2.3	9.3	16.3	23.3	30.3	6.4	13.4	Підсумкова
3	1	Іванов	6	8	4	4	5	н	н	н	н	5	
4	2	Петренко	12	10	12	9	н	н	н	н	н		
5	3	Сидоров	10	9	н	8	н	н	н	н	н		
6	4	Тесленко	11	10	5	8	н	н	н	н	н		
7	5	Айсман	5	7	8	6	н	н	н	н	н		
8	6	Бойко				12	12	12	12	н	н		
9	7	Грицюк		9	8	9	10	11	н	н	н		
10	8	Буренко	н	12				12	9	н	н		
11	9	Вітчар	н			8	8	н	8	8	н		
12	Кількість відсутніх												
13	Початковий рівень												
14	Достатній рівень												
15	Середній рівень												
16	Високий рівень												

- 12. Збережіть файл з іменем Практична робота 9 у відповідній папці. Закрийте програму Excel. Завершіть роботу за комп'ютером.

Зробіть висновок: які функції використовують для розрахунків; яка з використовуваних функцій має найбільше аргументів.

§ 19. Розширені фільтри. Проміжні підсумки

Розширені фільтри

Автоматичні фільтри, використання яких розглядалось у § 18, придатні не для всіх випадків.

► Нехай потрібно отримати дані про успішність учнів, які мають оцінки більші від 9 балів з алгебри або інформатики. Застосування фільтрів з умовою (> 9) до стовпців Алгебра, Інформатика дозволить побачити лише рядок Петренко, тоді як заданим критеріям відповідають рядки: Іванов, Петренко, Сидоров (рис. 19.1).

	A	B	C	D	E	F
1	Прізвище	Алгебра	Хімія	Фізика	Інформатика	середній бал
2	Іванов	7	10	11	12	10
3	Петренко	12	11	8	10	10,3
4	Сидоров	11	9	8	9	9,3
5	Тесленко	9	8	9	7	8,3

Рис. 19.1

Щоб застосувати складні умови, використовують розширені фільтри. Для застосування розширеного фільтра необхідно:

- 1) скопіювати в окреме місце заголовки стовпців, за якими відбуватиметься фільтрування (наприклад, Алгебра, Інформатика);
- 2) у комірки під скопійованими заголовками ввести критерії фільтрування (наприклад, >9);
- 3) вибрати Дані → Сортування й фільтр → Додатково;
- 4) установити у вікні Розширений фільтр необхідні прапорці і вказати діапазони (рис. 19.2).

При цьому критерії, об'єднані логічною операцією АБО, вказуються в різних рядках, а об'єднані логічною операцією І — в одному рядку.

Щоб копіювати результати фільтрування в інше місце, достатньо у відповідному полі вказати адресу комірки, розташованої у верхньому лівому куті нової таблиці.

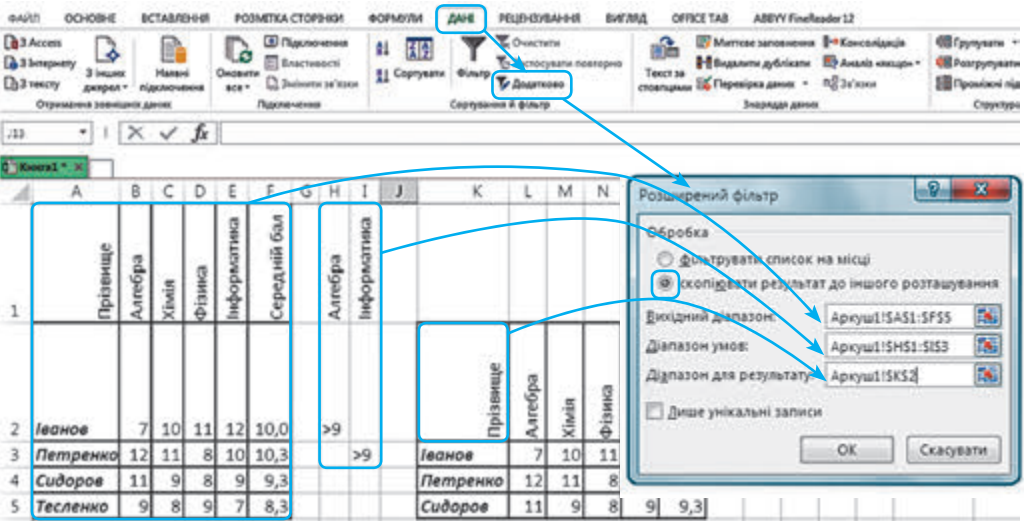


Рис. 19.2

Проміжні підсумки

У таблицях Excel зручно аналізувати дані. Одним зі способів є застосування *проміжних підсумків*, коли підрахунок проводиться не за всіма даними стовпця, а за їх частинами.

➤ Нехай потрібно обчислити за допомогою проміжних підсумків, скільки учнів навчаються з інформатики на 10, 11, 12 балів (рис. 19.3).

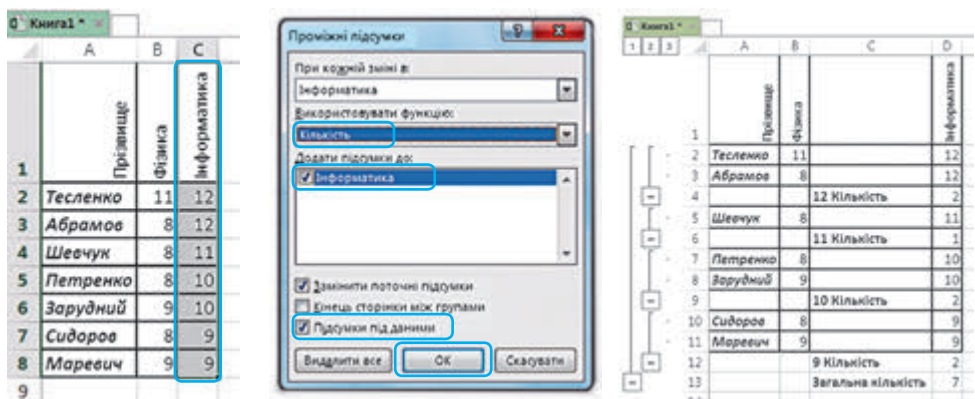


Рис. 19.3

Щоб налаштувати **проміжні підсумки**, потрібно відсортувати таблицю за стовпцем, за даними якого здійснюється обчислення (у нашому випадку — стовпець Інформатика); виділити комірки із даними і заголовками (рис. 19.3, а); виконати команду Дані → Структура → Проміжний підсумок; установити критерії у вікні Проміжні підсумки (рис. 19.3, б). За допомогою кнопок, що з'являються ліворуч від таблиці, дані можна *приховати*, залишивши результати проміжних підсумків (рис. 19.3, в).

Питання для самоперевірки



1. У якому випадку не можна застосувати автоматичний фільтр?
2. У якому випадку доцільно застосовувати розширений фільтр?
3. Опишіть порядок застосування розширеного фільтра.
4. Як у розширеному фільтрі записуються умови (критерії), об'єднані логічною операцією І?
5. Коли доцільно використовувати проміжні підсумки?

Вправа 19



▶ Створити таблиці із застосуванням розширеного фільтра та проміжних підсумків за описом.

- 1) Запустіть програму Excel. Створіть таблицю за зразком.
- 2) Застосуванням розширеного фільтра створіть нову таблицю, яка міститиме дані про вантажні автомобілі або такі, обсяг продажів яких перевищив 1 000 000.
- 3) Для початкової таблиці, використовуючи проміжні підсумки, обчисліть сумарні обсяги продажів автомобілів кожного типу.
- 4) Приховайте дані початкової таблиці, залишивши проміжні підсумки продажів автомобілів за типами.
- 5) Збережіть файл з іменем Вправа 19 у відповідній папці. Завершіть роботу за комп'ютером.

	A	B	C
1	Тип	Марка	Обсяг продажів
2	легкові	ЗАЗ	700 000
3	вантажні	МАЗ	500 000
4	легкові	Мерседес	3 000 000
5	вантажні	Мерседес	10 000 000
6	вантажні	Форд	750 000
7	тягачі	Форд	1 000 000
8	легкові	Форд	15 000 000
9	легкові	Хонда	1 500 000

Комп'ютерне тестування



Виконайте тестове завдання 19 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

35	52	70	87	105	122	140	157	175	8	3	6	9	5*	0.0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	88*	3	6	9	6*	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87*	3	6	9	7*	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698	86*	3	6	9	8*	1401	1418	1435	1452	1469	1486
732	750	767	785	802	819	837	854	0.0872	85*	3	6	9	9*	1584	1601	1618	1635	1652	1669

§ 20. Діаграми і графіки

§ 20. Діаграми і графіки

Діаграми

В Excel включені спеціальні засоби, названі діловою графікою, які дозволяють зобразити табличні дані в графічному вигляді.

В Excel існує велика кількість стандартних діаграм, серед яких виділяють три основні різновиди: кругову, стовпчасту, графік (рис. 20.1).

Кругова діаграма (рис. 20.2, а) наочно показує співвідношення між частинами в цілому. Для побудови кругової діаграми використовують числові дані, розміщені в *одному рядку або стовпчику*.

Стовпчаста діаграма (рис. 20.2, б) ілюструє співвідношення між числовими даними, розміщеними в *кількох рядках і стовпцях*.

Графік (рис. 20.2, в) використовують для зображення змін показників *протягом певного часу*.

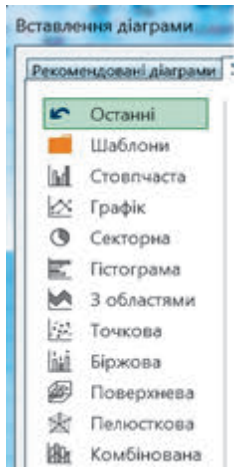
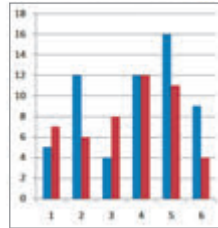


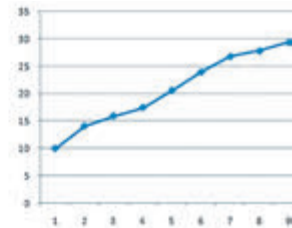
Рис. 20.1



а



б



в

Рис. 20.2

Після побудови діаграми зміна даних таблиці приводить до зміни відповідних об'єктів діаграми.

Побудова діаграми

Для створення діаграми необхідно:

- 1) заповнити (створити) таблицю даних;
- 2) виділити потрібні дані разом із назвами рядків (стовпців);
- 3) вибрати тип діаграми (Вставка → Діаграма).

18*	24	
	103	958
924	1115	1132
1097	1288	1305
1271	1461	1478

Пригадаємо, які об'єкти містить діаграма (рис. 20.3).

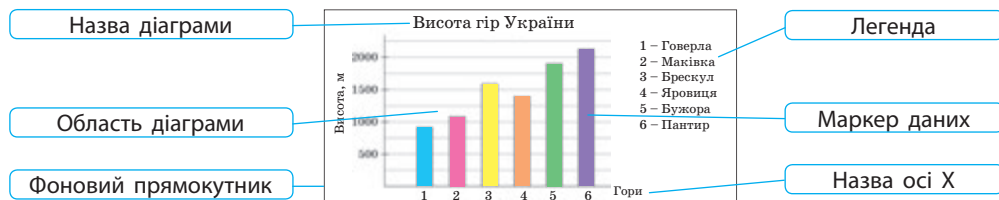


Рис. 20.3

За допомогою стрічки інструментів або кнопок швидкого налаштування можна змінити тип діаграми тощо (див. таблицю).

Зображення	Призначення кнопки швидкого налаштування
	Елементи
	Стилі
	Фільтри

Діаграма може бути розміщена як на тому самому аркуші, так і на окремому аркуші. Щоб **перемістити діаграму на інше місце аркуша**, слід перетягнути фоновий прямокутник із натиснутою лівою кнопкою миші. Щоб **розмістити діаграму на окремому аркуші**, слід вибрати команду Перемістити діаграму, що є в контекстному меню.

Щоб **змінити розміри** діаграми або її об'єктів, треба виділити (клацнути) потрібний об'єкт і перетягти відповідні маркери зміни розміру. Сектори кругової діаграми, наприклад, можна роз'єднати, перетягуючи їх мишею. За допомогою контекстного меню можна змінити колір заливки секторів, їхні контур, розмір, рельєф тощо.

Графіки

Розглянемо хід побудови графіка функції в Excel.



Побудувати графік функції $y = \frac{2}{x}$ при $1 \leq x \leq 19$ із кроком $h = 1,5$.

Запустимо Excel, створимо таблицю значень функції на заданому відрізку. Для цього:

1. Впишемо в комірках A1, A2, A3 літери h, x, y.
2. У комірку B1 уведемо значення кроку зміни координати x ($h = 1,5$).
3. У комірку B2 введемо перше значення із діапазону координати x ($x = 1$).

35	52	70	87	105	122	140	157	175	85	3	6	9	5 ⁴	0,0872	889	906	924	941	958
209	227	244	262	279	297	314	332	349	86 ⁴	3	6	9	6 ⁴	1045	1063	1080	1097	1115	1132
384	401	419	436	454	471	488	506	523	87 ⁴	3	6	9	7 ⁴	1219	1236	1253	1271	1288	1305
558	576	593	610	628	645	663	680	698	88 ⁴	3	6	9	8 ⁴	1419	1436	1453	1471	1488	1505
732	750	767	785	802	819	837	854	0,0872	89 ⁴	3	6	9	9 ⁴	1645	1663	1680	1697	1715	1732

§ 20. Діаграми і графіки

- У комірку C2 введемо формулу, яка додаватиме до поточної координати x стале значення кроку ($=B2+\$B\1).
- Скопіюємо формулу з комірки C2 в решту комірок діапазону координати x та отримаємо кінцеве значення діапазону — 19.
- У комірку B3 введемо формулу для обчислення значення функції ($=2/B2$).
- Установимо для діапазону комірок B3:N3 числовий формат із двома десятковими знаками.
- Скопіюємо формулу з комірки B3 у діапазон C3:N3.
- В отриманій таблиці аргументів (x) і значень (y) даної функції виділимо значення координат (разом із літерами x та y).
- Виберемо на вкладці Вставлення у групі Діаграми тип діаграми — точкова (рис. 20.4).

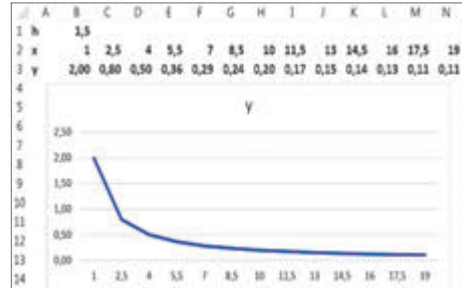


Рис. 20.4

Питання для самоперевірки



- Які типи діаграм можна створити в Excel?
- Коли доцільно застосовувати кругову діаграму?
- Коли доцільно застосовувати стовпчасту діаграму?
- Які основні кроки побудови діаграм?
- Як змінити розміри діаграми або її об'єктів?

Вправа 20



►► Побудувати графік функції $y = \frac{a}{x}$ ($a = \text{const}$).

- Запустіть програму Excel. Для змінної a відведіть окрему комірку та встановіть довільне значення.
- Установіть довільний крок зміни координати x .
- Створіть таблицю значень для вибраного проміжку.
- Змініть значення a та проаналізуйте зміну графіка.
- Збережіть файл з іменем Вправа 20 у відповідній папці. Завершіть роботу за комп'ютером.

Комп'ютерне тестування



Виконайте тестове завдання 20 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

18 ⁴	24	
924	111	958
1097	1115	1132
1271	1288	1305
	1461	1478



Практична робота 10

Упорядкування даних у таблицях. Автоматичні та розширені фільтри

Завдання: створити таблицю за зразком, упорядкувати дані за зазначеними умовами і правилами.

Обладнання: комп'ютер зі встановленим табличним процесором Excel.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

- ▶ 1. Запустіть програму Excel. Створіть таблицю на шести аркушах за зразком (див. рисунок).
- ▶ 2. Відсортуйте дані в таблиці (Аркуш2) за зростанням ціни.
- ▶ 3. Відсортуйте рядки таблиці (Аркуш3), щоб марки автомобілів були розташовані в алфавітному порядку, а кількість — у порядку спадання.
- ▶ 4. Увімкніть автоматичні фільтри для всіх стовпців таблиці (Аркуш4) та налаштуйте фільтр відповідного стовпця на відображення рядків тільки з марками автомобілів «Форд» і «Мерседес».
- ▶ 5. Налаштуйте у таблиці (Аркуш5) розширений фільтр відображення даних для автомобілів, ціна яких менша за 123 000 або кількість яких не перевищує 150.
- ▶ 6. Установіть умовне форматування даних таблиці (Аркуш6) так, щоб напис «Мерседес» відображався червоним кольором, а «Форд» — зеленим.
- ▶ 7. Збережіть файл з іменем Практична робота 10 у відповідній папці. Завершіть роботу за комп'ютером.

Ціна, грн	Марка автомобіля	Кількість, шт.
120 000	МАЗ	100
150 000	Форд	125
110 000	Форд	150
110 000	Мерседес	200
100 000	Хонда	225
120 500	Мерседес	250

Зробіть висновок: як можна впорядкувати дані в таблиці; які фільтри можна застосувати; для чого використовують умовне форматування.

РОЗДІЛ 6

ОСНОВИ ПОДІЙНО- ТА ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ



§ 21. Знайомство із середовищем програмування

§ 22. Створення найпростішого проекту

§ 23. Запис програми

§ 24. Основні компоненти програми

§ 25. Властивості і методи елементів керування

§ 26. Налаштування програмного коду

Практична робота 11. Створення об'єктно-орієнтованої програми, що відображає вікно повідомлення

Практична робота 12. Створення програми з кнопками і написами

ПОВТОРЮЄМО

У 5–7 класах ви знайомились із алгоритмами і складали програми в навчальному програмному середовищі.

Алгоритм — це чітко визначена послідовність команд для розв'язування певної задачі. Алгоритми можна подавати словесним і графічним способами, а також у вигляді програми, записаної певною мовою програмування.

Мова програмування — це штучна мова, яка являє собою систему позначень і правил для запису алгоритмів у формі, придатній для їх виконання на комп'ютері.

Для зручної розробки програм існують спеціальні засоби створення — системи програмування.



1. Що таке програма?
2. Що таке програмування?
3. Що таке мова програмування?
4. З якими мовами програмування ви вже знайомі?
5. Що таке система програмування?



Опрацювавши цей розділ, ви навчитесь створювати програми з цікавим інтерфейсом, використовуючи широкі можливості середовища з візуальним конструктором інтерфейсу користувача.

§ 21. Знайомство із середовищем програмування

Матеріал підручника орієнтовано на роботу із середовищем програмування Lazarus*, в основу якого покладено принципи об'єктно-орієнтованого програмування. Об'єктно-орієнтоване програмування — це метод програмування, що ґрунтується на поданні програми у вигляді сукупності об'єктів, що взаємодіють. Кожен із об'єктів характеризується станом, що описується за допомогою властивостей, та поведінкою, що визначається певними діями. Під час створення програми використовуються готові компоненти, властивості та дії яких налаштовуються відповідно до потреби.

Середовище програмування

В основі середовища програмування Lazarus лежить мова програмування Object Pascal, яка бере свій початок від класичної мови Pascal (Паскаль). Одне з поширених середовищ програмування Delphi теж ґрунтується на цій мові. Мова програмування Pascal була розроблена в 1971 році професором Інституту інформатики швейцарської вищої політехнічної школи Нікlausом Віртом (нар. 1934) і названа на честь французького математика, фізика й філософа Блеза Паскаля (1623–1662).

Середовище програмування Lazarus — це комплекс програмних засобів, до якого входять: текстовий редактор, компілятор, налагоджувач. Середовище програмування має графічний інтерфейс, який містить набір меню, панелей і вікон для розробки програм.

Під час роботи в середовищі програмування Lazarus програміст описує певну сукупність об'єктів та їхню взаємодію. Кожен об'єкт має набір властивостей, може виконувати певні дії та реагувати на деякі події. Дії, які може виконати об'єкт, називають його **методами**.

Автоматизація програмування досягається завдяки можливості створювати на формі об'єкт, вибираючи відповідний компонент на Палітрі компонентів, і змінювати його властивості, не вносячи вручну змін до програмного коду. Програміст має змогу змінювати значення *властивостей* даного об'єкта і викликати різні його *методи*.

* Офіційний сайт: www.lazarus-ide.org.

Таким чином, технологія роботи в середовищі Lazarus базується на ідеях об'єктно-орієнтованого, подійного програмування та візуального конструювання інтерфейсу користувача.

Інтерфейс середовища програмування Lazarus

Для **відкриття середовища** Lazarus треба виконати команди:

Пуск → Програми → Lazarus → Lazarus.

Інтерфейс середовища програмування Lazarus містить чотири основні вікна (рис. 21.1).

Головне вікно, на якому розміщено три панелі: меню, панель швидкого доступу, палітра компонентів.

Вікно форми — заготовка вікна майбутньої програми. *Форма* — це головний компонент програми, контейнер для всіх інших її компонентів.

Вікно Інспектор об'єктів — вікно, в якому налаштовують властивості розміщених на формі об'єктів.

Вікно Редактор тексту — вікно текстового редактора, призначене для введення та редагування програмного коду.

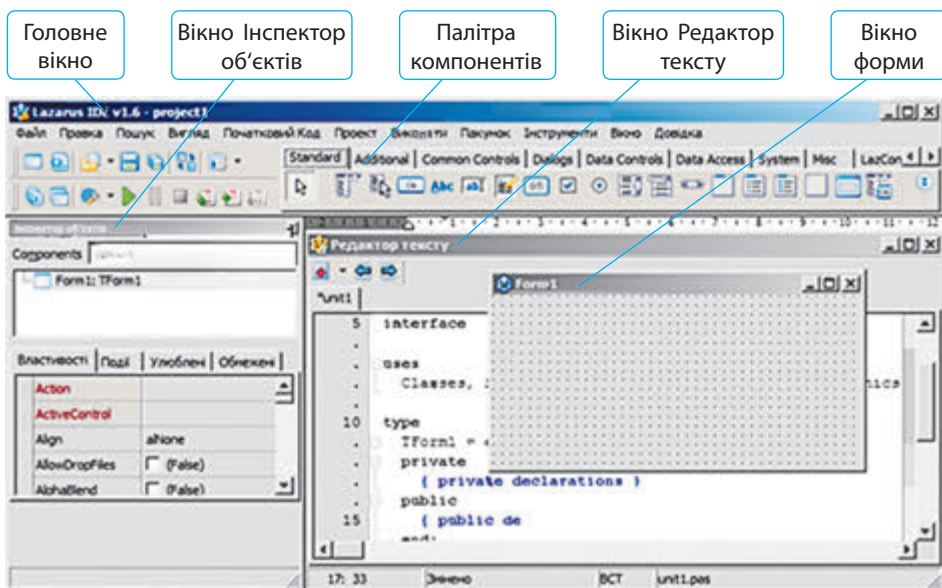


Рис. 21.1

Меню головного вікна програми складається з таких основних пунктів: Файл, Правка, Пошук та ін., в яких команди згруповано за призначенням. Меню (рис. 21.2) використовують для виконання широкого кола завдань: відкриття і закриття файлів, налаштування середовища програмування та ін. На Панелі швидкого доступу розташовано значки найчастіше вживаних команд меню. Палітра компонентів містить вкладки Standard, Additional, Dialogs та ін., на яких згруповано компоненти за їх призначенням. За допомогою Палітри компонентів вибирають потрібні компоненти програми для розміщення їх на формі.

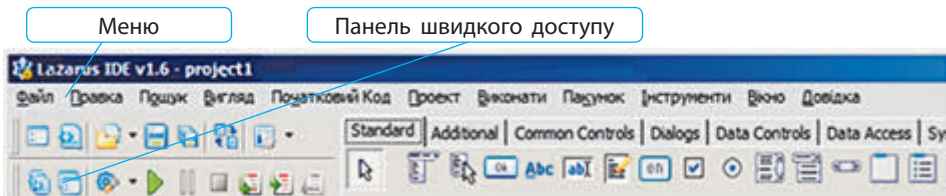


Рис. 21.2

Знайомство з об'єктами програми

Об'єктами програми є форма і компоненти, що на ній містяться. Для додавання компонента на форму треба:

- 1) клацнути мишею значок компонента на Палітрі компонентів;
- 2) клацнути мишею в будь-якому місці на формі.

Вибраний компонент з'явиться на формі, після чого його потрібно налаштувати.

» Ознайомитися з компонентом Label. Додамо на форму компонент Label **Abc** (Напис) із Палітри компонентів. На формі з'явиться об'єкт Label1, який можна пересувати по формі та змінювати його розміри за допомогою маркерів. Змінювати вигляд і поведінку об'єкта можна за допомогою вікна Інспектор об'єктів. Це вікно має вкладки Властивості і Події, вміст яких відрізняється для різних об'єктів (рис. 21.3).

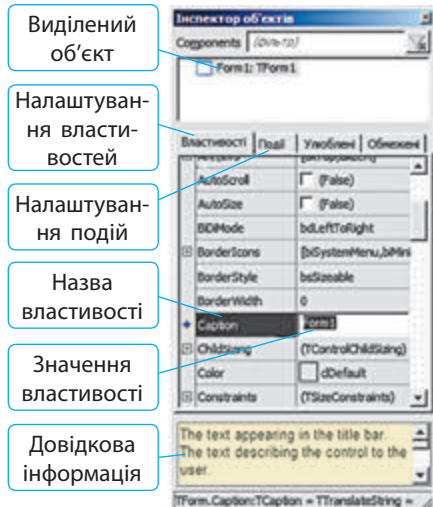
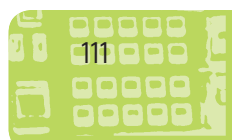


Рис. 21.3



Для об'єкта Label1 на вкладці Властивості ви можете визначити розташування і розмір, змінюючи значення властивостей Left (*Відстань від лівого краю форми*), Top (*Відстань від верхнього краю форми*), Height (*Висота*) і Width (*Ширина*). На вкладці Події міститься список подій, на які може реагувати даний об'єкт.

Під час розробки програми у середовищі програмування Lazarus на форму, що міститься у вікні форми, додають компоненти майбутньої програми з Палітри компонентів. Кожен із компонентів налаштовують як окремий об'єкт програми.

Питання для самоперевірки



1. Назвіть основні вікна інтерфейсу середовища програмування Lazarus.
2. Для чого призначена Палітра компонентів?
3. Для чого призначене вікно Інспектор об'єктів?
4. Як додати компонент на форму?
5. Як змінити розміщення об'єкта на формі?
6. Як змінити властивості об'єкта на формі?

Вправа 21



- Ознайомитися з послідовністю дій для налаштування властивостей форми.
- 1) Відкрийте середовище Lazarus, вибравши команди: Пуск → Програми → Lazarus → Lazarus.
 - 2) Виділіть форму: клацніть на ній або виберіть об'єкт Form1 у вікні Інспектор об'єктів.
 - 3) Змініть розміри і положення форми за допомогою миші.
 - 4) Перейдіть до вікна Інспектор об'єктів. Задайте значення властивостей Width і Height по 500 пікселів.
 - 5) У вікні Інспектор об'єктів змініть заголовок форми. Для цього в рядку властивості Caption (*Заголовок*) замініть назву об'єкта Form1 на «Вправа 21».
 - 6) Змініть колір форми за допомогою властивості Color.

Комп'ютерне тестування



Виконайте тестове завдання 21 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 22. Створення найпростішого проекту

Поняття проекту

Для розв'язання конкретних задач користувача існує багато прикладних програм (застосунків). Прикладна програма, як правило, складається з багатьох файлів, кожен з яких має певне призначення. Набір файлів у певному середовищі програмування, необхідних для створення прикладної програми, називають **проектом**.

Основні файли проекту

Файли проекту	Ярлик	Тип файла	Призначення
Головний файл проекту (Lazarus Project Information)		.lpi	Підключає всі використовувані програмні модулі й містить оператори для запуску програми
Файл основної програми		.lpr	Містить код основної програми, підключає зовнішні бібліотеки, всі необхідні модулі (створюється разом з файлом .lpi)
Файли опису форм		.lfm	Зберігають значення властивостей форм
Файли програмних модулів		.pas	Містять коди програмних модулів

У середовищі програмування Lazarus під час компіляції з файлів проекту формується єдиний виконуваний файл і створюється новий застосунок, призначений для розв'язання певної задачі.

Створення проекту

Для створення проекту треба виконати команду: меню Проект → Новий проект... → Проект → Програма (рис. 22.1). На екрані з'явиться вікно форми та вікно Редактор тексту.

На формі можна розмістити різні об'єкти: написи (Label), кнопки (Button), текстові поля (Edit) та інші



Рис. 22.1

й налаштувати їх для керування програмою. Створені таким чином об'єкти називають також **елементами керування**.

Розглянемо докладніше процес створення елемента керування Label. Цей елемент керування застосовується для виведення заголовків, підписів інших елементів керування тощо.

Для додавання компонента Label на форму треба:

- 1) вибрати компонент Label із вкладки Standard на Палітрі компонентів (рис. 22.2) та клацнути на ньому мишею;
- 2) клацнути мишею в будь-якому місці на формі. На формі створений об'єкт містить напис Label1 (рис. 22.3).

На вкладці Standard



Рис. 22.2

На формі



Рис. 22.3

Для зміни значень властивостей створеного об'єкта Label1 треба перейти до вікна Інспектор об'єктів. На вкладці Властивості показано всі властивості елемента, в даному випадку властивості напису (рис. 22.4). Можна змінити значення властивостей Caption (*Заголовок*), Color (*Колір фону*), Font (*Параметри шрифту*). З тим, як налаштувати події для елемента керування, ви ознайомитеся в § 24.

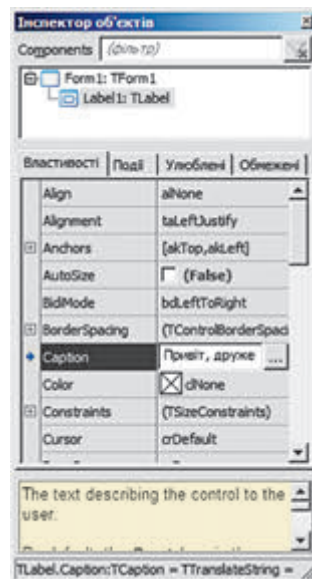


Рис. 22.4

Компіляція проекту та його виконання

Для створення єдиного виконуваного файлу проекту потрібно здійснити компіляцію.




Компілятор — це спеціальна програма, яка перетворює вихідну програму, написану мовою програмування високого рівня, на програму, написану машинною мовою, «зрозумілою» комп'ютеру.


Для компіляції проекту в середовищі програмування Lazarus потрібно вибрати команду меню Виконати → Компілювати. Компілятор послідовно опрацьовує файли проекту й буде з них **виконуваний файл**.

Під час компіляції текст програми перевіряється на відсутність помилок, перекладається на машинну мову, після чого встановлюються зв'язки між файлами проекту та формується ехе-файл (виконуваний файл).

Якщо під час компіляції виявляється, що текст програми має синтаксичні помилки, то в нижній частині вікна Редактор тексту виводяться повідомлення про помилки, а також попередження і підказки.

Якщо в програмі немає синтаксичних помилок — створюється файл, готовий до виконання. Назва такого файла збігається з назвою файла проекту та має розширення ехе. Середовище Lazarus поміщає виконуваний файл у папку, де міститься файл проекту.

Запустити проект на виконання можна різними способами: за допомогою команди меню Виконати → Виконати; клацнувши кнопку  на Панелі швидкого доступу (рис. 22.5) або натиснувши клавішу F9.

Зупинити виконання проекту можна: клацнувши кнопку Закрити  у вікні програми; за допомогою кнопки Завершити (див. рис. 22.5) на панелі швидкого доступу або натиснувши сполучення клавіш Ctrl + F2.

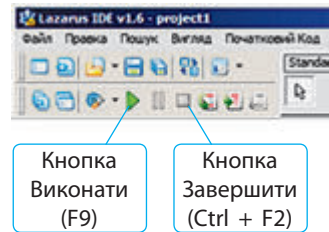



Рис. 22.5

Збереження проекту

При першому збереженні проекту треба виконати такі дії:

- 1) вибрати команду: меню Файл → Зберегти всі (або клацнути кнопку Зберегти всі  на Панелі швидкого доступу);
- 2) у діалоговому вікні Зберегти проект project1(*.lpi), що відкрилося, створити нову папку для проекту;
- 3) зберегти у цій папці файл програмного модуля (Unit1.pas) і файл проекту (project1.lpi).

Середовище програмування пропонує для збереження стандартну назву проекту, але ви можете її змінити.

Питання для самоперевірки



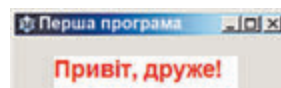
1. З яких файлів складається проект?
2. Вкажіть послідовність додавання компонента на форму і зміну його властивостей.
3. Як виконати компіляцію проекту?
4. Якими способами можна запустити проект на виконання?

5. Якими способами можна зупинити виконання проекту?
6. Зазначте порядок дій для збереження проекту.

Вправа 22



- ▶ Створити проект за поданим алгоритмом.
- 1) Створіть новий проект, виконавши команду: меню Проект → Новий проект... → Проект → Програма.
- 2) У вікні Інспектор об'єктів змініть заголовок форми (значення властивості Caption) на «Перша програма».
- 3) Розмістіть на формі компонент Label. Змініть значення властивості Caption на «Привіт, друже!» (див. рисунок).
- 4) Змініть значення властивостей Color і Font на власний розсуд.
- 5) Запустіть проект на виконання. Вікно, що відкрилося, має всі атрибути, властиві вікнам програм Windows: заголовок, меню, кнопки Згорнути, Розгорнути, Закрити; є можливість змінювати розміри вікна.
- 6) Зупиніть виконання проекту. Збережіть проект, виконавши команду: меню Файл → Зберегти всі. У діалоговому вікні Зберегти проект project1(*.lpi), що відкрилося, створіть нову папку для вашого проекту. Назвіть папку Вправа 22. Збережіть у папці файли Unit1.pas і project1.lpi, змінивши імена файлів на власний розсуд.



Комп'ютерне тестування



Виконайте тестове завдання 22 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 23. Запис програми

Складові мови програмування

Основними складовими будь-якої мови програмування є алфавіт, синтаксис і семантика.



Алфавіт мови програмування — це набір символів, які можуть використовуватися під час створення програм.

До **алфавіту мови** Object Pascal входять такі символи:

- 1) великі і малі літери латинського алфавіту: A...Z, a...z;
- 2) цифри: 0...9;
- 3) знаки арифметичних операцій, розділові знаки, спеціальні символи:
+ - * / \ = <> () . , ; : @ \$ # _ ;
- 4) складені символи: <=, >=, <>, :=.

У мові Object Pascal є ключові (службові) слова, що мають фіксований зміст: If, For, while, array і т. д.



Синтаксис мови програмування — сукупність правил побудови мовних конструкцій (команд, ідентифікаторів тощо) у певній мові програмування.

Семантика мови програмування — сукупність правил використання мовних конструкцій, записаних мовою програмування.

Подійно-орієнтоване програмування

Керування виконанням програми у середовищі програмування Lazarus здійснюється за допомогою подій: коли відбувається певна подія (*event*), то виконується процедура обробки цієї події (*event procedure*).

Подія — це будь-який вплив на елемент керування в активному вікні, зокрема, при клацанні кнопки миші чи натисканні клавіші клавіатури. У Lazarus кожна подія має назву. Наприклад, клацання мишею — це подія OnClick. Для кожного об'єкта існує набір стандартних подій, що пов'язані з даним об'єктом. Щоб програма виконувала певні дії у відповідь на подію, яка відбулася, програміст повинен написати процедуру обробки відповідної події.

Редактор коду

Програмний код форми міститься у вікні Редактор тексту. У процесі редагування програми виникає необхідність перемикання між вікном Редактор тексту і вікном форми. Зробити це можна, клацнувши кнопку Перемкнути форму/модуль (F12) (рис. 23.1) або натиснувши клавішу F12.

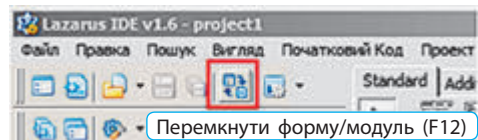


Рис. 23.1

Проект у Lazarus (рис. 23.2) складається з набору програмних одиниць — **модулів** (unit). Головний модуль програми створюється автоматично і містить інструкції, з яких починається виконання проекту.

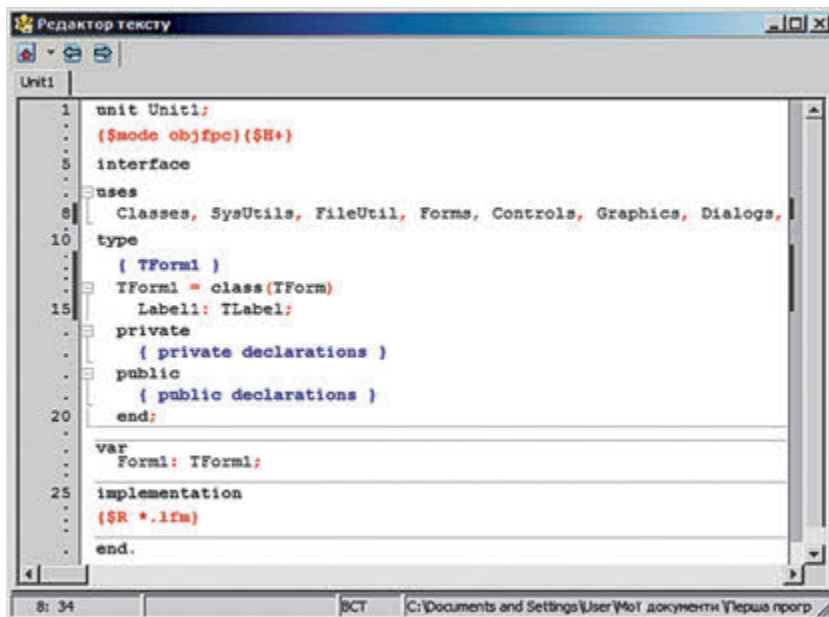


Рис. 23.2

Текстовий редактор, вбудований в середовище Lazarus, схожий на звичайні текстові редактори. Але для зручності рядки тексту програми пронумеровані. Також передбачено виділення окремих складових тексту програми певними кольорами і накресленням:

- усі ключові слова (**unit**, **uses**, **var**, **end** та ін.) автоматично виділяються напівжирним;
- розділові знаки стають червоними;
- коментарі виділяються синім кольором;
- рядки з помилками виділяються жовтогарячими смугами.

У вікні Редактор тексту містяться ключові слова, які утворюють структуру програми. Текст модуля починається із ключового слова **unit**, далі слідує назва модуля: **unit Unit1**;


Кожен модуль складається з двох блоків: інтерфейсного (**interface**) і реалізації (**implementation**). Інтерфейсний блок містить опис об'єктів програми, а блок реалізації — програмний код, який закінчується ключовим словом **end**.

Створення процедури обробки події

Для створення процедури обробки події для об'єкта потрібно:

- 1) виділити об'єкт;
- 2) у вікні Інспектор об'єктів перейти на вкладку Події;
- 3) двічі клацнути мишею в рядку з назвою потрібної події.
Заголовок створюваної процедури обробки події має формат:
procedure НазваОб'єктаНазваПодії(Список параметрів);

» Створити процедуру обробки події OnClick для командної кнопки.

Додамо на форму компонент Button  (Кнопка) із вкладки Standard (рис. 23.3).

Автоматично в частині описів з'явиться:
Button1: TButton;

Виділимо на формі командну кнопку. У вікні Інспектор об'єктів перейдемо на вкладку Події. Двічі клацнемо в рядку з назвою події OnClick (рис. 23.4).

Після цього буде автоматично сформовано заготовку процедури обробки події OnClick для командної кнопки Button1:

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
end;
```

Водночас в частині описів з'явиться заголовок обробника події:

```
procedure Button1Click(Sender: TObject);
```

Запустимо проект на виконання. Але клацання командної кнопки не викличе ніяких дій, оскільки в процедурі Button1Click між ключовими словами **begin** і **end** не написано жодної команди.

Створимо програмний код, за допомогою якого при клацанні кнопки буде змінюватися колір форми. Запрограмуємо зміну значення властивості Color об'єкта Form1. Для цього у програмному коді між ключовими словами **begin** і **end** запишемо команду:

```
Form1.Color := clYellow;
```

Запустимо проект на виконання. Перевіримо дію кнопки.



Рис. 23.3



Рис. 23.4

Програмний код, який пише програміст у середовищі програмування Lazarus, повинен забезпечувати реакцію на події. Процедуру, яка виконується, коли відбувається певна подія, називають **обробником події**. Якщо в програмі немає обробника для певної події, то відповідно немає і реакції програми на цю подію.

Питання для самоперевірки



1. Поясніть, що таке подія. Наведіть приклади подій.
2. Чому описану в параграфі технологію називають подійно-орієнтованим програмуванням?
3. З якого ключового слова починається блок реалізації процедур обробки подій?
4. Що таке процедура обробки події?
5. Як створити процедуру обробки події OnClick для командної кнопки?

Вправа 23



- Скласти програму для зміни напису на формі при клацанні командної кнопки.
- 1) Відкрийте проект, збережений у папці Вправа 22:
 - виконайте команду: меню Проект → Відкрити проект;
 - у діалоговому вікні Відкрити проект відкрийте папку Вправа 22;
 - знайдіть у папці файл проекту project1.lpi і відкрийте його.
 - 2) Додайте на форму компонент Button із вкладки Standard на Панелі компонентів.
 - 3) Для створеної кнопки (Button1) змініть значення властивості Caption на «Змінити напис».
 - 4) Створіть процедуру обробки події OnClick для кнопки. Перейдіть на вкладку Події і двічі клацніть у рядку з назвою події OnClick.
 - 5) У створеній процедурі **procedure** TForm1.Button1Click(Sender: TObject); між ключовими словами **begin** і **end** запишіть команду:


```
Label1.Caption := 'Я програмую у Lazarus';
```

 Запустіть проект на виконання.
 - 6) Додайте на форму ще одну кнопку Button2 і змініть її заголовок на «Змінити колір». Створіть процедуру обробки події OnClick для кнопки Button2, в якій запрограмуйте зміну кольору форми. Збережіть проект у папці Вправа 23.

Комп'ютерне тестування



Виконайте тестове завдання 23 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 24. Основні компоненти програми


Властивості форми

Вивчення середовища програмування Lazarus ви почали зі знайомства з формою. Як і будь-який об'єкт, форма має властивості. Властивостями форми є заголовок, висота, ширина, місце розташування, колір фону тощо. У середовищі програмування Lazarus для форми встановлені типові значення властивостей. Але ці значення можна змінити під час проектування форми у вікні Інспектор об'єктів або передбачити їх зміну під час виконання програми за допомогою операторів, записаних у тексті програми.

Вибравши форму, у вікні Інспектор об'єктів побачимо її властивості.

Основні властивості форми

Властивість	Що визначає
Name	Назва (ім'я) — ідентифікатор, що використовується для звернення до форми у тілі програми (за замовчуванням першій формі нового проекту призначається назва Form1)
Caption	Текст заголовка вікна форми
Width, Height	Ширина і висота форми (у пікселях)
Position	Позиція вікна форми на екрані при виконанні програми
Color	Колір фону форми (базові кольори подано в списку властивості Color іменованими константами)
Hint	Текст підказки, що з'являється при наведенні вказівника миші на об'єкт
ShowHint	Дозвіл виведення підказки

Праворуч від значень деяких властивостей міститься кнопка . Це означає, що для задавання значення властивості можна скористатися додатковим діалоговим вікном.

У програмах бувають корисними спливаючі підказки, які з'являються в разі наведення вказівника миші на певний об'єкт.

Перелік об'єктів, розміщених на формі, виводиться у вікні Інспектор об'єктів. Для **вибору об'єкта** потрібно клацнути його або вибрати його назву у вікні Інспектор об'єктів (рис. 24.1).

Для **створення підказки**, що з'являється при наведенні вказівника миші на об'єкт, треба:

- 1) виділити цей об'єкт;
- 2) ввести текст підказки в рядок властивості Hint (Підказка);
- 3) увімкнути виведення підказки, встановивши значення True для властивості ShowHint (Показати підказку).

» Створити підказку, яка з'являтиметься при наведенні вказівника миші на форму.

Виділимо об'єкт Form1 у вікні Інспектор об'єктів (рис. 24.2).

Введемо текст «Це форма» у рядок властивості Hint (див. рис. 24.2).

Для властивості ShowHint встановимо значення True.

У режимі виконання проекту при наведенні вказівника миші на форму з'явиться підказка (рис. 24.3).

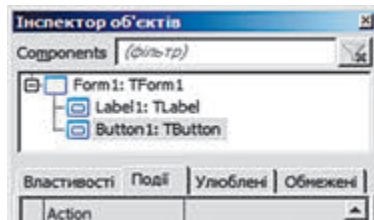


Рис. 24.1



Рис. 24.2



Рис. 24.3

Події для форми

Форма, як будь-який об'єкт програми, має свої властивості, методи і події.

Усі назви процедур обробки подій для форми мають формат: `FormНазваПодія`

Розглянемо події для форми `OnCreate` і `OnClick`. Подія `OnCreate` відбувається під час створення форми. В обробнику події `OnCreate` встановлюються початкові значення різних властивостей об'єкта.

Ви вже знаєте, що подія `OnClick` відбувається, коли користувач клацає об'єкт мишею.

Процедура обробки події OnClick у вікні Редактор тексту має вигляд:
procedure TForm1.FormClick(Sender: TObject);
begin
end;

Між ключовими словами **begin** і **end** потрібно записати оператори програмного коду, які будуть визначати реакцію форми на клацання мишею.

Якщо написати назву об'єкта, у даному випадку Form1, і поставити крапку, розкриється список властивостей і методів для об'єкта Form1. Середовище програмування Lazarus пропонує вибрати зі списку потрібну властивість або метод.

Виберемо властивість Color. Натиснемо літеру С — у вікні списку стають видимими властивості і методи, назви яких починаються з цієї літери. Виділимо рядок Color (рис. 24.4) і натиснемо клавішу Enter. У результаті отримаємо:

Form1.Color

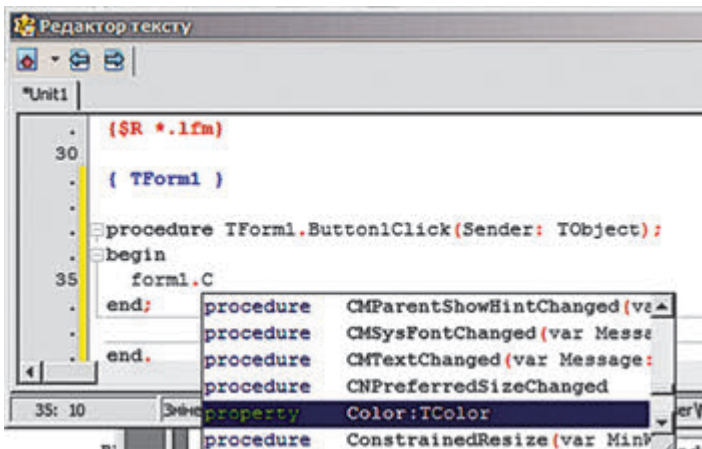


Рис. 24.4

Встановимо для властивості Color жовтий колір:

```
Form1.Color := clYellow; // clYellow — константа, що позначає жовтий колір
```

Тепер, якщо під час роботи проекту клацнути мишею на формі — вона набуде жовтого кольору.

Використання вікон повідомлень

Вікно повідомлення ShowMessage (*Показати повідомлення*) використовується, щоб надати користувачеві певну інформацію. Процедура

ShowMessage виводить на екран інформаційне вікно із заданим повідомленням та кнопкою ОК. Заголовок вікна містить назву виконуваного файлу.

Синтаксис виклику процедури:

ShowMessage(Текст);

Тут Текст — змінна або константа, що містить текст повідомлення.

Для того щоб продовжити роботу з програмою, користувач повинен клацнути кнопку ОК. Поки користувач не клацне кнопку ОК, подальша робота програми стає неможливою, тобто не можна «дістатися» вікна, розташованого позаду.

» Вивести у вікні програми повідомлення (рис. 24.5):

ShowMessage('Hello, World');

Вікна повідомлень можна використовувати для виведення результатів виконання певного етапу програми. Використання вікон повідомлень дозволяє програмісту уникнути додавання на форму зайвих елементів керування.



Рис. 24.5

Питання для самоперевірки



1. У якому вікні середовища програмування Lazarus виводиться перелік об'єктів, розміщених на формі?
2. Як створити підказку, що з'являється при наведенні вказівника миші на об'єкт?
3. Як змінити заголовок форми?
4. Як створити обробник події OnClick для форми?
5. Який оператор призначений для зміни кольору фону?
6. Наведіть приклади, коли доцільно використовувати вікно повідомлення ShowMessage.

Вправа 24



- Скласти процедуру для налаштування кольору та заголовку форми під час запуску проекту на виконання.
- 1) Створіть новий проект.
 - 2) Двічі клацніть на формі. У вікні Редактор тексту відкриється заготовка обробника події OnCreate.


```
procedure TForm1.FormCreate(Sender: TObject);
begin
end;
```

- 3) Налаштуйте червоний колір для фону форми. Між ключовими словами **begin** і **end** запишіть оператор, що задає значення властивості `Color` форми:
`Form1.Color := clRed;`
- 4) Запишіть оператор зміни заголовка форми:
`Form1.Caption := 'My program';`
- 5) Запустіть проект на виконання. За потреби виправте помилки. Збережіть проект у папці Вправа 24.
- 6) Налаштуйте для форми виведення підказки «Це форма».



Комп'ютерне тестування

Виконайте тестове завдання 24 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

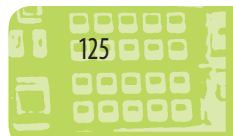
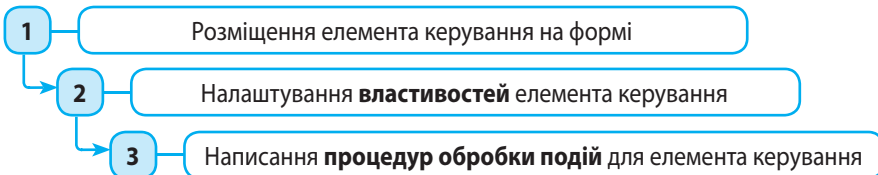


§ 25. Властивості і методи елементів керування

Етапи створення елемента керування

Як ви вже знаєте, елементи керування — це візуальні компоненти, кожен із яких має певний набір властивостей і методів. Елементи керування — це «будівельні блоки»: ви берете їх з Палітри компонентів і переносите на форму для створення графічного інтерфейсу користувача програми. Графічний інтерфейс дозволяє організувати діалог людини і комп'ютера.

Процес створення елемента керування поділяється на три етапи:



Властивості, методи і події

Властивості характеризують стан елемента керування, **методи** — дії, які він може виконати, а **події** — зовнішній вплив на елемент керування, на який цей елемент може реагувати.



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

Властивості — це атрибути елемента керування, які визначають, як цей елемент виглядає на екрані (наприклад, ширина і висота елемента керування, його відображення тощо). Початкові значення властивостей елемента керування встановлюються на етапі розробки інтерфейсу у вікні Інспектор об'єктів.

Деякі властивості є спільними для багатьох елементів керування: Name (*Назва*); Enabled (*Доступність*); Height (*Висота*); Width (*Ширина*); Hint (*Підказка*); Left (*Відстань від елемента керування до лівого краю форми*); Top (*Відстань від елемента керування до верхнього краю форми*); Visible (*Видимість*).

Значення властивостей елемента керування можна змінювати різними способами.

Змінити розміри кнопки (рис. 25.1) можна шляхом перетягування маркерів виділення. Того самого результату можна досягти, задавши нові значення властивостей Height і Width у вікні Інспектор об'єктів.

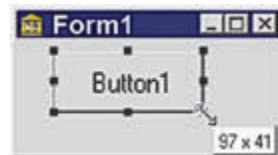


Рис. 25.1

Метод — це фрагмент програмного коду, пов'язаний із певним елементом керування. Запрограмувати виклик методу об'єкта можна лише у програмному кодї, тому він відбувається виключно під час виконання проекту.

Деякі з методів характерні для багатьох компонентів, наприклад методи setFocus (*Надати фокус*), Show (*Показати*), Hide (*Сховати*).

Процедура setFocus передає фокус введення текстовому полю: Edit2.SetFocus;

Середовище програмування Lazarus дозволяє легко змінювати значення властивостей елемента керування як у режимі розробки, так і в режимі виконання проекту.

Зміна значень властивостей елементів керування у режимі виконання проекту

Для зміни значень властивостей елементів керування у режимі виконання проекту потрібно, щоб програмний код містив необхідні оператори у такому форматі:

```
НазваКомпонента.НазваВластивості := ЗначенняВластивості;
```

» Присвоїти властивості Width кнопки значення 40 пікселів:
`Button1.Width := 40;`

Слід пам'ятати, що тип властивості та її значення повинні збігатися.

Питання для самоперевірки



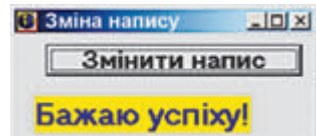
1. Що таке графічний інтерфейс програми?
2. Назвіть етапи створення елемента керування.
3. Що таке властивості елемента керування?
4. Що таке методи елемента керування?
5. Як змінити значення властивості елемента керування під час розробки проекту?
6. Як змінити значення властивості елемента керування під час виконання проекту?

Вправа 25



» Скласти програму для зміни властивостей елемента керування Label у режимі виконання проекту.

- 1) Створіть новий проект. Змініть заголовок форми на «Зміна напису». Розмістіть на формі компоненти Button і Label згідно з рисунком.
- 2) Виділіть створену кнопку (Button1) і у вікні Інспектор об'єктів змініть значення властивості Font на власний розсуд, значення властивості Caption — на «Змінити напис».
- 3) Змініть для напису Label1 значення властивостей Alignment, AutoSize, Color. Властивості Caption надайте значення «Я програмую у Lazarus».
- 4) Створіть процедуру обробки події OnClick для кнопки Змінити напис.



- 5) Запишіть оператори для зміни значень властивостей:
- ```
Label1.Caption := 'Бажаю успіху!';
Label1.Color := clYellow;
Label1.Font.Color := clBlue;
```

! Властивості `Caption` можна надавати тільки текстових значень, які у програмному коді беруться в одинарні лапки. При встановленні значення властивості `Caption` у вікні Інспектор об'єктів текст пишуть без лапок.

- 6) Запустіть проект на виконання. Перевірте дію кнопки. Збережіть проект у папці Вправа 25.



### Комп'ютерне тестування

Виконайте тестове завдання 25 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 26. Налаштування програмного коду

Текст програми, уведений із клавіатури, часто містить помилки, які програміст повинен усунути. Помилки, що можуть виникнути у процесі розробки програми, поділяють на:

- синтаксичні помилки (англ. *compile error*);
- помилки часу виконання (англ. *run-time error*);
- логічні (алгоритмічні) помилки.

### Синтаксичні помилки в програмному коді

Компілятор створює виконуваний файл лише в тому випадку, коли текст не містить помилок. Більшість помилок, які виявляються під час компіляції програми, є синтаксичними. Це помилки в написанні або розміщенні ключових слів та інших елементів коду. При виявленні помилок курсор встановлюється біля першого оператора з помилкою. Рядок, що містить помилку, виділяється жовтогарячою смугою. У нижній частині екрана з'являється текстове вікно, що містить відомості про цю помилку: ім'я файлу, в якому знайдено



помилку, номер рядка з помилкою і тип помилки. Для швидкого переходу до певної помилки необхідно двічі клацнути мишею на рядку з її описом.

- » Типова помилка `Syntax error, ";" expected` (*Синтаксична помилка, ";" очікувалось*) — після інструкції не поставлено крапку з комою (рис. 26.1).

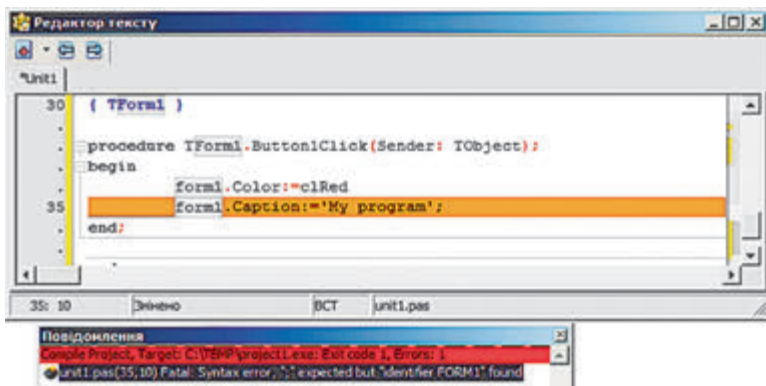


Рис. 26.1

Компілятор при знаходженні синтаксичної помилки зупиняє процес компіляції, виводить повідомлення про знайдену помилку. Програмісту потрібно виправити цю помилку і виконати повторну компіляцію.

## Помилки виконання

Деякі помилки виявляються під час виконання програми, коли програма намагається опрацювати неприпустимі дані. Такі помилки називають **помилками часу виконання**.

- » Розглянемо фрагмент програми:

```
X := a/b;
Edit1.Text := FloatToStr(X);
```

Якщо таку програму запустити із середовища програмування Lazarus, то вона скомпілюється і працюватиме правильно при будь-яких значеннях змінних  $a$  і  $b$ , крім  $b=0$ . Але в момент опрацювання зупинить свою роботу, вивівши на екран відповідне повідомлення (рис. 26.2).

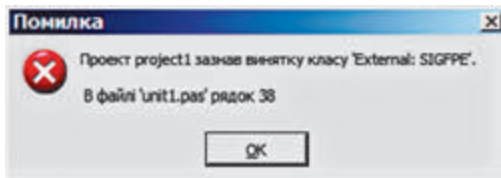


Рис. 26.2

Виконання програми буде зупинено на рядку, що містить помилку (рис. 26.3). Якщо навести на змінну вказівник миші, то з'явиться підказка з поточним значенням змінної.

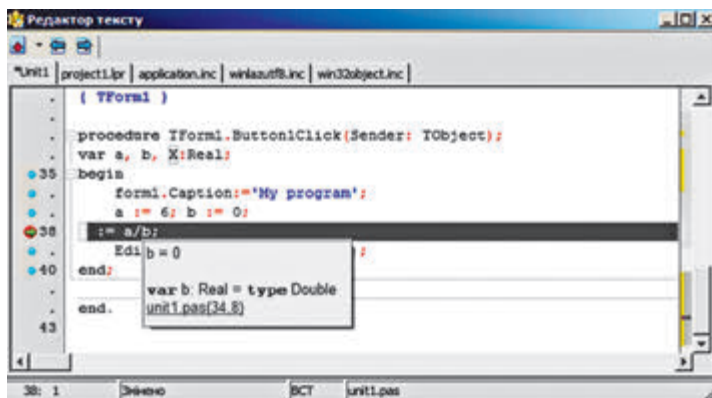


Рис. 26.3



Процес усунення помилок називають **налагодженням програмного коду**.

Метою налагодження є виявлення помилки, яка виникає на етапі виконання програми і приводить до неправильних результатів її роботи.

Під час налагодження програми слід уважно аналізувати повідомлення про помилку, яку відображає налагоджувач, щоб швидко визначити, яку допущено помилку та як її виправити.

Для керування налагодженням програми призначені кнопки для покрокового виконання програми Вступити і Переступити (рис. 26.4). Клацаючи кнопки покрокового виконання, можна по рядках виконувати програму, контролюючи зміну значень змінних і правильність обчислень. Якщо

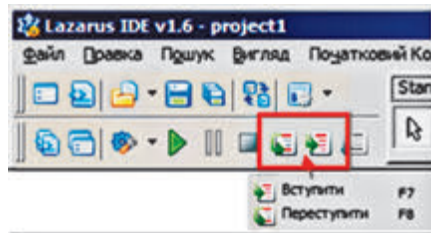


Рис. 26.4

команда виклику процедури є поточною, то при клацанні кнопки Вступити здійсниться перехід до покрокового виконання цієї процедури. Після завершення покрокового виконання процедури здійсниться перехід до наступного оператора програми.

Кнопка Переступити надає можливість пропустити покрокове виконання процедури й одразу перейти до наступного оператора програми.

Щоб перервати режим покрокового виконання програми, потрібно натиснути сполучення клавіш Ctrl + F2.

## Логічні помилки

Найскладніше виявити та виправити помилки, викликані неправильною логікою програми, тобто помилки, яких припустилися під час розробки алгоритму. Такі помилки називають **логічними** (алгоритмічними). У деяких випадках логічна помилка може привести до помилки виконання (див. приклад на с. 129–130).

Навіть якщо програма успішно скомпільована, результат її виконання може виявитися хибним.

» Якщо ви, бажаючи збільшити число  $a$  на 1, замість  $a := a + 1$  випадково напишете  $a := a + 2$ , то компілятор не виявить помилки, але результат роботи програми буде неправильним.

Одним зі способів виявлення логічної помилки є виконання програми в покроковому режимі з аналізом значень змінних на кожному кроці.

Для опису алгоритму програми та її складових використовують коментарі, які можна включати до тексту програми.



**Коментар** — це текст, який ігнорується компілятором і включається до тексту програми з метою полегшення розуміння програмного коду.

Щоб компілятор відрізняв команди від коментарів, у мові Object Pascal коментарі потрібно брати у фігурні дужки:

```
S := Edit3.Text; { введення дати народження }
```

Якщо текст коментаря короткий і не займає більше одного рядка, то можна перед коментарем замість дужок поставити дві похилі риски:

```
S := Edit3.Text; // введення дати народження
```

Вдало сформульовані коментарі значно спрощують розуміння алгоритму програми.

## Питання для самоперевірки



1. Які помилки називають синтаксичними?
2. Які помилки належать до помилок періоду виконання?
3. Які помилки називають логічними?
4. Визначте помилку в наведеному фрагменті програмного коду:  
 $x := 5; y := 1; z := x / (y - 1);$
5. У результаті роботи програми одержано відповідь: «1,5 землекопи». Визначте, до якого виду належить помилка.
6. Поясніть, як виконується програма в покроковому режимі.

## Вправа 26



- Виправити помилки в поданому програмному коді, ознайомитися з режимом покрокового виконання програми.
- 1) Створіть новий проект. Розмістіть на формі компонент Button. Створіть процедуру обробки події OnClick для командної кнопки (Button1).
  - 2) До коду процедури додайте такі команди:  
`Label1.Caption := 'Бажаю успіху!';`  
`Label1.Color := clYellow;`  
`Label1.Font.Color := clBlue;`
  - 3) Запустіть проект на виконання. Рядок `Label1.Caption := 'Бажаю успіху!';` виділено жовтогарячою смугою, у вікні помилок міститься повідомлення про помилку:  
`[Error] Unit1.pas(28): Undeclared identifier: 'Label1'`  
 Помилка виникла тому, що об'єкта Label1 на формі не існує.
  - 4) Додайте на форму компонент Label.
  - 5) Запустіть проект на виконання в покроковому режимі, натиснувши клавішу F8. Починає виконуватися код проекту:  

```
begin
 Application.Initialize;
 Application.CreateForm(TForm1, Form1);
 Application.Run;
end.
```

 Продовжуйте натискати клавішу F8, поки не з'явиться вікно форми. Клацніть командну кнопку і продовжте покрокове виконання.
  - 6) Перервіть режим покрокового виконання сполученням клавіш Ctrl+F2. Збережіть проект у папці Вправа 26.



## Комп'ютерне тестування

Виконайте тестове завдання 26 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## Практична робота 11



### Створення об'єктно-орієнтованої програми, що відображає вікно повідомлення

**Завдання:** створити програму, в ході виконання якої виводиться вікно повідомлення.

**Обладнання:** ПК із встановленим середовищем програмування Lazarus.

### Хід роботи

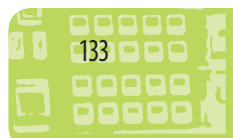
*Під час роботи з комп'ютером дотримуйтеся правил безпеки.*

#### I. Розміщення елементів керування на формі

- ▶ 1. Відкрийте новий проект.
- ▶ 2. Додайте на форму компонент Label, у вікні Інспектор об'єктів змініть значення його властивостей:
  - Caption — «Слідкуйте за ходом виконання програми»;
  - Color — clSkyBlue;
  - AutoSize — False (*заборона автоматично змінювати розмір напису по горизонталі*);
  - WordWrap — True (*дозвіл виводити текст у декілька рядків*);
  - Font — шрифт: *Comic Sans MS*, розмір: 14; колір: білий.
- ▶ 3. Додайте на форму компонент Button. Змініть значення властивості Caption створеної кнопки (Button1) на «Виконати програму».

#### II. Написання процедур обробки подій

- ▶ 4. Створіть процедуру обробки події OnClick для командної кнопки. Запишіть оператори, що змінюють колір фону форми на червоний і виводять на екран вікно повідомлення про виконану дію:



```

procedure TForm1.Button1Click(Sender: TObject);
begin
 Form1.Color := clRed;
 ShowMessage('Змінився колір форми');
end;

```

- ▶ 5. Змініть вигляд вікна повідомлення (рис. 1). Надайте значення властивості Title для об'єкта Application:

```

Application.Title := 'My program';
ShowMessage('Змінився колір форми');

```

- ▶ 6. Додайте до програмного коду оператори:
 

```

Label1.Color := clGreen;
ShowMessage('Змінився колір напису на зелений');

```

- ▶ 7. Розділіть текст повідомлення на два рядки: вставте службові символи повернення каретки (#13), тобто встановлення курсора на початку рядка, і перенесення рядка (#10). ShowMessage('Змінився колір '+#13#10+' напису на зелений');  
Запустіть проект на виконання. Проаналізуйте зміни у вигляді вікна повідомлення (рис. 2).

- ▶ 8. Встановіть значення ширини форми 500 пікселів:
 

```

Form1.Width := 500;

```

- ▶ 9. Змініть заголовок форми на «Практична робота 11».
- ▶ 10. Додайте оператор, який змінює колір фону форми на жовтий.
- ▶ 11. Запрограмуйте виведення вікна повідомлення після кожної виконаної дії.

### III. Тестування проекту

- ▶ 12. Після запуску проекту перевірте дію командної кнопки і переконайтеся, що кожна виконана дія супроводжується відповідним повідомленням. Збережіть проект у папці Практична робота 11. Завершіть роботу за комп'ютером.

**Зробіть висновок:** як за допомогою програмного коду змінювати властивості об'єктів під час виконання програми.

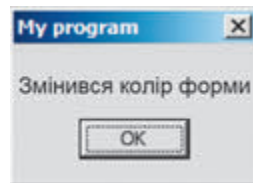


Рис. 1

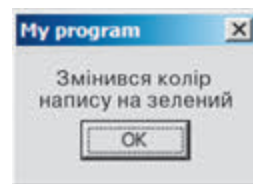


Рис. 2





## Практична робота 12

### Створення програми з кнопками і написами

**Завдання:** створити проект із різними кнопками і написами.

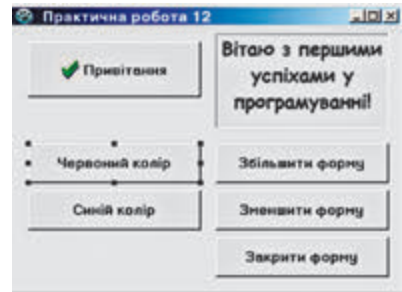
**Обладнання:** ПК із встановленим середовищем програмування Lazarus.

#### Хід роботи

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

#### I. Розміщення елементів керування на формі

- ▶ 1. Відкрийте новий проект. Змініть заголовок форми на «Практична робота 12».
- ▶ 2. Додайте на форму компонент StaticText (вкладка Additional на Палітрі компонентів), призначений для виведення повідомлення. У вікні Інспектор об'єктів змініть значення властивостей компонента:
  - Caption — «Вітаю з першими успіхами у програмуванні!»;
  - BorderStyle — sbsSunken (*стиль межі — втоплене вікно*);
  - AutoSize — False;
  - Visible — False.
- ▶ 3. Додайте на форму компонент BitBtn (вкладка Additional на Палітрі компонентів). Змініть вигляд кнопки, задавши у вікні Інспектор об'єктів властивості Kind значення bkAll. Змініть параметри шрифту, надайте властивості Caption значення «Привітання».
- ▶ 4. Додайте на форму п'ять компонентів Button. Змініть значення властивості Caption доданих командних кнопок згідно з рисунком.



#### II. Написання процедур обробки подій

- ▶ 5. Створіть процедуру обробки події OnClick для компонента BitBtn і в створеній процедурі BitBtn1Click запишіть оператор, який зробить видимим об'єкт StaticText1:
 

```
StaticText1.Visible := True;
```

- ▶ **6.** Створіть процедуру обробки події OnClick для кнопки Червоний колір. Запрограмуйте зміну кольору фону форми на червоний.  
`Form1.Color := clRed;`
- ▶ **7.** Створіть аналогічну процедуру обробки події для кнопки Синій колір.
- ▶ **8.** Створіть процедуру обробки події OnClick для кнопки Збільшити форму. Запрограмуйте збільшення висоти і ширини форми на 10 пікселів:  
`Form1.Width := Form1.Width + 10;`  
`Form1.Height := Form1.Height + 10;`
- ▶ **9.** Створіть аналогічну процедуру обробки події для кнопки Зменшити форму.
- ▶ **10.** Створіть для кнопки Закрити форму процедуру обробки події OnMouseMove. У створеній процедурі Button5MouseMove запишіть оператори:  
`Button5.Left := 100 + Random(250);`  
`Button5.Top := 30 + Random(250);`  
Значенням функції Random(250) є випадкове число в діапазоні від 0 до 250.
- ▶ **11.** Запустіть проект на виконання. Спробуйте клацнути кнопку Закрити форму (кнопка повинна «втікати» від вказівника миші).

### III. Тестування проекту

- ▶ **12.** Після запуску проекту перевірте роботу всіх командних кнопок. Збережіть проект у папці Практична робота 12. Завершіть роботу за комп'ютером.

**Зробіть висновок:** як створювати кнопки і написи.

# РОЗДІЛ 7

## АЛГОРИТМИ РОБОТИ З ОБ'ЄКТАМИ ТА ВЕЛИЧИНАМИ



§ 27. Величини та їхні властивості

§ 28. Елементи керування для введення даних

Практична робота 13. Складання та виконання лінійних алгоритмів опрацювання величин у навчальному середовищі програмування

§ 29. Величини цілих типів

§ 30. Величини дійсних типів

§ 31. Величини логічного типу

§ 32. Величини символьного типу

§ 33. Величини рядкового типу

§ 34. Перетворення величин одного типу на інший

Практична робота 14. Налаштування готової програми

## ПОВТОРЮЄМО



Комп'ютер як виконавець опрацьовує дані за певною програмою. Окремий інформаційний об'єкт, наприклад число, символ, називають *величиною*. З курсу фізики та математики ви вже знайомі з такими величинами, як швидкість руху, відстань та ін., знаєте, що вони бувають сталими і змінними. Величина, що характеризує стан деякого об'єкта, є базовим поняттям математичних і природничих наук. Усі дії в програмуванні виконуються з певними величинами.

Будь-яка величина, яку опрацьовує програма, займає певне місце в пам'яті комп'ютера. *Значення величини* — це дані, що зберігаються в цих комірках пам'яті.

1. Під час вивчення яких навчальних предметів ви зустрічалися з поняттям величини?
2. Наведіть приклади величин.
3. Які бувають величини?
4. Як комп'ютер опрацьовує величини?



У цьому розділі ви дізнаєтеся про алгоритми роботи з об'єктами, типи величин та алгоритми їх опрацювання.

## § 27. Величини та їхні властивості



Окремий інформаційний об'єкт (число, символ, рядок, таблиця та ін.) називають **величиною**.

Основними характеристиками величин є назва, вид, тип і значення.

**Назва (ім'я, ідентифікатор) величини** — це набір символів, що обирають для позначення констант, змінних, типів даних тощо. Назва може складатися з латинських літер, цифр, знака підкреслення, а починатися тільки з літери або знака підкреслення. При цьому регістр символів значення не має.

**Вид величини** визначає спосіб використання величини в програмі. Величина може бути константою (тобто постійною) або змінною. *Константи* — це величини, значення яких не можуть змінюватися в ході виконання програми. *Змінні* — величини, значення яких можуть змінюватися в ході виконання програми.

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

Стандартними типами даних є цілий (Integer), дійсний (Real), рядковий (String), логічний (Boolean) і символний (Char) типи.

*Порядкові* типи даних — такі типи, у яких значення впорядковані (пронумеровані) і для кожного зі значень можна вказати попереднє і наступне значення. До порядкових належать цілі, символні і логічні типи даних.

### Оголошення змінних

У мові Object Pascal кожна змінна перед використанням повинна бути оголошена. Під час оголошення (опису) змінної вказується її тип. Змінні оголошуються в програмі в блоці опису змінних, який починається словом **var**:

**var** Назва: Тип;

У модулі перед ключовим словом **implementation** є блок опису глобальних змінних:

**var** Form1: TForm1;

Змінні, оголошені в цьому блоці, називають **глобальними**, бо вони можуть використовуватися в будь-якій процедурі обробки подій програмного коду і зберігають свої значення доти, доки виконується код форми.

Блок опису локальних змінних за потреби можна створити в кожній процедурі. Цей блок розміщується між заголовком даної процедури і ключовим словом **begin**. Змінні, оголошені таким чином, є **локальними**. Вони стають недійсними і є недоступними за межами даної процедури.

» Якщо потрібно описати декілька змінних одного типу, то їх записують через кому:  
**var** A, B, C: Integer;

## Константи

У мові Object Pascal є два види констант: звичайні та іменовані.



**Звичайна константа** — це ціле або дробове число, рядок символів або окремий символ, логічне значення.

**Іменована константа** — це ідентифікатор, який в тексті програми використовується замість значення сталої величини.

Іменовану константу, як і змінну, перед використанням потрібно оголосити. Блок оголошення констант починається словом **const**:

```
const Назва = Значення;
```

» Оголошення константи *g*:  
**const** g = 9.81;

## Оператор присвоєння

Змінна може набувати різних значень у процесі виконання програми, наприклад внаслідок виконання оператора присвоєння.

Синтаксис оператора присвоєння:

```
Назва := Вираз;
```

Спочатку обчислюється значення виразу, яке надається змінній.

» **var** S: String; Age: Integer;  
**begin**  
     S := 'Мій текст';  
     Label1.Caption := S;  
     Age := 12 + 2;  
**end**;



» Знайти помилку в даному фрагменті програми:

```
var A, B, C: Integer;
begin
 A := 5;
 B := 10;
 C := A + B;
 Label1.Caption := C;
end;
```

Оскільки властивість `Caption` має рядковий тип (`String`), а змінні `A`, `B`, `C` — цілий тип (`Integer`), то потрібно подати числове значення змінної `C` у вигляді тексту. Для цього існує вбудована функція `IntToStr`. Рядок у тексті програми, що раніше викликав помилку, повинен виглядати так:

```
Label1.Caption := IntToStr(C);
```

### Питання для самоперевірки



1. Назвіть і охарактеризуйте види величин.
2. Назвіть основні характеристики величини.
3. Визначте, які з поданих послідовностей символів не можуть бути ідентифікаторами та чому:  
`sum`; `W1`; `Pryklad 1`; `(SUM)`; `A-4`; `Pryklad_1`; `var`; `9A`.
4. Наведіть приклади стандартних типів даних у мові `Object Pascal`.
5. Як у програмі оголосити, що змінна `A` — цілого типу, `B` — логічного типу, `C` — рядкового типу?

### Вправа 27



- » Скласти програму для додавання двох цілих чисел.
- 1) Створіть новий проект. Розмістіть на формі кнопку `Button1`, змініть заголовок створеної кнопки на «Обчислити».
  - 2) Розмістіть на формі компонент `Label`.
  - 3) Створіть процедуру обробки події `OnClick` для кнопки `Обчислити`. У вікні програмного коду між заголовком процедури і ключовим словом **begin** опишіть змінні `A`, `B`, `C` цілого типу:
 

```
procedure TForm1.Button1Click (Sender: TObject);
var A, B, C: Integer;
begin
```
  - 4) У програмному кодї процедури `Button1Click` запишіть оператори для знаходження суми двох чисел і виведення результату до напису `Label1`:

```
begin
 A := 5;
 B := 10;
 C := A + B;
 Label1.Caption := IntToStr(C);
end;
```

Запустіть проект на виконання, перевірте дію кнопки. Збережіть проект у папці Вправа 27.



- 5) Додайте на форму напис Label2. Доповніть програмний код операторами обчислення добутку чисел A і B та виведення результату до напису Label2. Перевірте дію кнопки.
- 6) Додайте на форму кнопку Button2. Змініть заголовок створеної кнопки на «Обчислити 2». Створіть процедуру обробки події OnClick для цієї кнопки та опишіть змінні x і y цілого типу. Додайте у програмний код процедури такі оператори:
 

```
x := 5; y := 6; x := y + x;
```

 Виведіть нові значення змінних x і y до написів Label1 і Label2. Перевірте дію кнопки.

### Комп'ютерне тестування



Виконайте тестове завдання 27 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 28. Елементи керування для введення даних

Більшість задач із програмування передбачають введення даних користувачем, після чого над ними виконуються деякі операції, а результат опрацювання виводиться у заданому вигляді:

введення → опрацювання → виведення.

## Введення і виведення текстових даних

Для введення даних користувачем і виведення результатів роботи програми використовують компонент Edit (рис. 28.1), що являє собою однорядкове текстове поле. Основна властивість компонента Edit — це властивість Text рядкового типу (String). Початкове значення властивості Text можна задавати у вікні Інспектор об'єктів (рис. 28.2). Її значення можна задати під час виконання програми, передбачивши в коді програми відповідний оператор присвоєння.

На вкладці Standard:



На формі:



Рис. 28.1

» Вміст текстового поля можна присвоювати змінній рядкового типу:

```
var S: String;
begin
 Edit1.Text := 'Текст, що виводиться';
 S := Edit1.Text;
end;
```

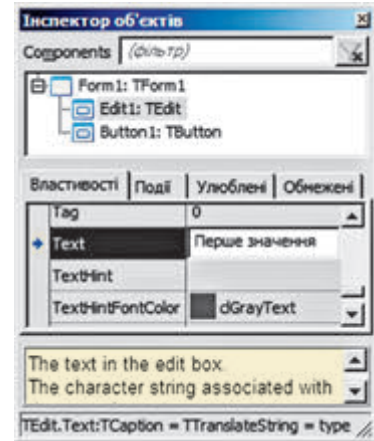


Рис. 28.2

## Поняття про перетворення типів

Іноді під час розробки програми виникає необхідність перетворення одного типу даних на інший. Наприклад, для виведення числа в текстове поле потрібно подати це число як рядок символів.


Для розв'язання цієї проблеми в мові Object Pascal існують вбудовані функції перетворення типів:

| Функція    | Призначення                      |
|------------|----------------------------------|
| IntToStr   | Перетворює ціле число на рядок   |
| StrToInt   | Перетворює рядок на ціле число   |
| FloatToStr | Перетворює дійсне число на рядок |
| StrToFloat | Перетворює рядок на дійсне число |

» Знайти суму двох цілих чисел, що вводяться в текстових полях Edit1 і Edit2. Значення суми вивести в текстове поле Edit3.


```
var A, B, C: Integer;
begin
 A := StrToInt(Edit1.Text);
 B := StrToInt(Edit2.Text);
 C := A + B;
 Edit3.Text := IntToStr(C);
end;
```

## Компонент ComboBox

Компонент ComboBox  (*Комбінований список*) міститься на вкладці Standard Палітри компонентів. Він є поєднанням списку рядків із однорядковим текстовим полем Edit. Кожен рядок списку ComboBox зберігає рядок даних типу String.

При цьому список рядків елемента керування ComboBox спочатку прихований і розкривається у разі клацання мишею кнопки розкриття, яка знаходиться праворуч у полі введення (рис. 28.3).

Для додавання рядків на етапі розробки інтерфейсу потрібно:

- 1) на вкладці Властивості у вікні Інспектор об'єктів у рядку Items клацнути кнопку ;
- 2) ввести потрібні дані у вікні Редактор рядків (рис. 28.4);
- 3) клацнути кнопку ОК.

» Вивести у текстове поле номер вибраного рядка списку ComboBox (рис. 28.5):

```
var S: String; N: Integer;
begin
 N := ComboBox1.ItemIndex; { якщо
 вибрано рядок «Венера» }
 Edit1.Text := IntToStr(N); // N = 1
end;
```

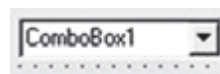


Рис. 28.3

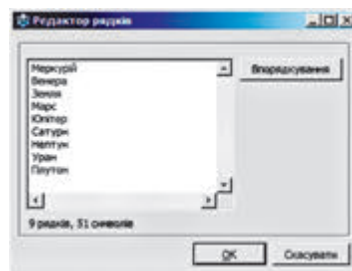


Рис. 28.4

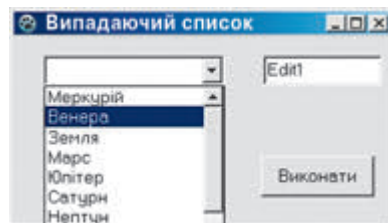


Рис. 28.5

» Вивести у текстове поле вміст вибраного рядка списку ComboBox:  
**var** S: String; N: Integer;  
**begin**  
     N := ComboBox1.ItemIndex;  
     S := ComboBox1.Items[N];  
     Edit1.Text := S;  
**end;**

Для того щоб під час виконання програми можна було вводити значення в текстове поле ComboBox (властивість Text) і додавати введені значення до списку ComboBox, процедура обробника подій повинна містити виклик методу Add за допомогою команди:

```
ComboBox1.Items.Add(ComboBox1.Text);
```

## Функція InputBox

Для введення даних можна використовувати функцію InputBox. Ця функція виводить вікно діалогу, в якому користувач повинен ввести дані й клацнути кнопку ОК (підтвердження дії) або Cancel (скасування дії).

Синтаксис функції InputBox:

Value := InputBox (Заголовок, Підказка, Значення);

де Value — змінна рядкового типу, яка отримує значення функції;

Заголовок — заголовок вікна функції (необов'язковий параметр);

Підказка — повідомлення, яке буде виведене в діалоговому вікні функції;

Значення — рядок тексту, який буде відображений у полі введення і виділений синім кольором.

У разі відмови користувача від введення тексту натисненням клавіші Esc або Cancel функція InputBox поверне це значення.

» Увести значення за допомогою функції InputBox (рис. 28.6).  
**var** myname: String;  
**begin**  
     myname := InputBox('Name', 'Ваше ім'я?',  
     'Не задано');  
**end;**



Рис. 28.6

За допомогою функції `InputDialog` можна організувати діалог користувача з програмою, не перевантажуючи форму зайвими елементами керування для введення даних.

### Питання для самоперевірки



1. Назвіть основні властивості компонента `Edit`.
2. Запишіть оператор присвоєння змінній цілого типу значення, що міститься у текстовому полі `Edit1`.
3. Запишіть оператор виведення значення змінної цілого типу у текстове поле `Edit2`.
4. Значенням якої властивості є елементи списку `ComboBox`?
5. Яка властивість зберігає номер рядка, вибраного елемента в списку `ComboBox`?
6. Поясніть призначення функцій та їх параметрів в операторі:  
`age := StrToInt(InputBox('Вік', 'Скільки тобі років?', '0'));`

### Вправа 28



- Скласти програму, що реалізує діалог користувача з програмою за поданим планом:
- 1) При клацанні кнопки отримується запит про ім'я.
  - 2) Після відповіді відкривається друге вікно із запитом про вік (рис. 1).
  - 3) Після відповіді у вікні виводиться повідомлення з привітанням і зазначенням класу, в якому навчається користувач (рис. 2).
- 1) Створіть новий проект. Розмістіть на формі кнопку `Button1` і змініть заголовок створеної кнопки на «Виконати».
  - 2) Створіть процедуру обробки події `OnClick` для кнопки `Виконати`.
  - 3) Опишіть змінні:  
`var myname: String; age, clas: Integer;`

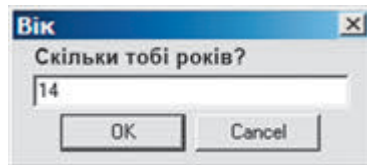


Рис. 1

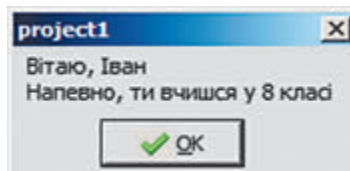


Рис. 2



- 4) У програмному кодї процедури Button1Click запишіть оператори: **begin**

```
myname := InputBox('Name', 'Твоє ім'я?', 'Не задано');
age := StrToInt(InputBox('Вік', 'Скільки тобі років?', '0'));
clas := age-6;
ShowMessage('Вітаю, '+myname+#10#13+ 'Напевно, ти вчишся у ' +
 IntToStr(clas)+ ' класі ');
```

**end;**

- 5) Запустіть проект на виконання. Збережіть проект у папці Вправа 28.



- 6) Доповніть проект виведенням ще одного повідомлення: «Вгадав клас навчання?», відповідь на яке «Так» або «Ні». Перевірте роботу програми.



### Комп'ютерне тестування

Виконайте тестове завдання 28 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## Практична робота 13

### Складання та виконання лінійних алгоритмів опрацювання величин у навчальному середовищі програмування



**Завдання:** створити проект для розв'язання задачі.

*Два потяги виїхали одночасно назустріч один одному. Потрібно знайти, через який час вони зустрінуться, якщо задано значення відстані між ними в момент початку руху та швидкості руху кожного потяга.*

**Обладнання:** ПК із встановленим середовищем програмування Lazarus.

#### Хід роботи

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

#### I. Розміщення елементів керування на формі

- 1. Створіть новий проект. Розмістіть на формі чотири компоненти Edit, компонент Button і чотири компоненти Label.

Змініть значення властивості Caption згідно з рисунком, задайте значення властивості Font на власний розсуд.

## II. Написання процедур обробки подій

- ▶ 2. Створіть процедуру обробки події OnClick для кнопки Обчислити. Виконання процедури має привести до виведення шуканого часу в текстове поле Edit4, якщо значення швидкостей руху потягів і відстані між ними введено у відповідні поля Edit1, Edit2, Edit3.

Орієнтовний програмний код:

```
var v1, v2, s, t: Real;
begin
 v1 := StrToFloat(Edit1.Text);
 v2 := StrToFloat(Edit2.Text);
 s := StrToFloat(Edit3.Text);
 t := S/(v1 + v2);
 Edit4.Text := FloatToStr(t);
end;
```

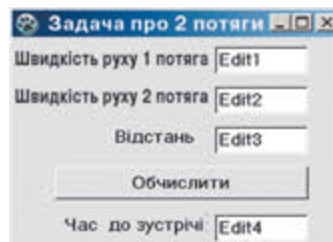
- ▶ 3. Отримайте результати для таких початкових даних: відстань — 1000 км, швидкість руху — 120 і 130 км/год відповідно.
- ▶ 4. Внесіть зміни до інтерфейсу програми та програмного коду, щоб проект реалізовував розв'язання іншої задачі.

*Два потяги виїхали одночасно з однієї станції у протилежних напрямках. Потрібно знайти відстань, на якій вони опиняться один від одного через заданий час, якщо дано швидкості руху кожного потяга.*

## III. Тестування проекту

- ▶ 5. Створіть три тестові набори вхідних даних для тестування проекту. Обчисліть очікувані результати за складеними тестами.
- ▶ 6. Перевірте роботу програми на створених тестових даних. Збережіть проект у папці Практична робота 13. Завершіть роботу за комп'ютером.

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



## § 29. Величини цілих типів

### Цілі типи даних

Дані цілих типів — це цілі числа, які можуть використовуватися в арифметичних виразах і займають у пам'яті комп'ютера від 1 до 8 байтів. Характеристики деяких цілих типів:

| Тип даних | Обсяг пам'яті (байти) | Діапазон значень        |
|-----------|-----------------------|-------------------------|
| LongInt   | 4                     | -2147483648..2147483647 |
| Integer   | 2                     | -32768..32767           |
| Byte      | 1                     | 0..255                  |
| ShortInt  | 1                     | -128..127               |

Під час запису в тексті програми арифметичних виразів потрібно дотримуватися певних правил. Для даних цілих типів визначені такі арифметичні операції: +, -, \*, div, mod. Результат виконання цих операцій також має цілий тип. З цілими числами можна виконувати операцію ділення (/), але результатом її виконання буде дійсне число.

Розглянемо детальніше операції div і mod.

| Назва операції                             | Позначення | Приклад         | Результат |
|--------------------------------------------|------------|-----------------|-----------|
| Цілочисельне ділення                       | div        | Res := 11 div 3 | Res = 3   |
| Остача від ділення<br>(ділення за модулем) | mod        | Res := 11 mod 3 | Res = 2   |

### Стандартні функції

У мові Object Pascal є низка вбудованих (стандартних) функцій. Їх використовують у виразах.

Розглянемо деякі стандартні функції, які можна застосовувати до даних цілого типу.

| Функція | Призначення                      | Тип результату | Приклад                           |
|---------|----------------------------------|----------------|-----------------------------------|
| Abs(x)  | Обчислює $ x $                   | Integer        | Abs(-5) = 5                       |
| Sqr(x)  | Обчислює $x^2$                   | Integer        | Sqr(5) = 25                       |
| Sqrt(x) | Обчислює $\sqrt{x}$              | Real           | Sqrt(64) = 8                      |
| Odd(x)  | Перевіряє аргумент на непарність | Boolean        | Odd (5) = True<br>Odd (4) = False |
| Succ(x) | Виводить наступне ціле число     | Integer        | Succ (5) = 6                      |
| Pred(x) | Виводить попереднє ціле число    | Integer        | Pred (5) = 4                      |

» Обчислити суму цифр введеного трицифрового числа A (рис. 29.1).

a1, a2, a3 — змінні для збереження відповідних цифр числа; S — сума цифр.

**var** A, a1, a2, a3, S: Integer;

**begin**

A := StrToInt(Edit1.Text);

a1 := A div 100;

a2 := A div 10 mod 10;

a3 := A mod 10;

S := a1 + a2 + a3;

Edit4.Text := IntToStr(S);

**end;**

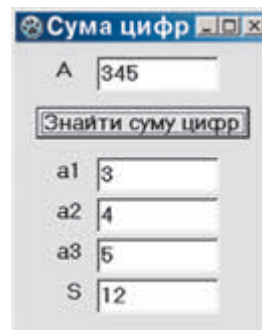


Рис. 29.1

Результат виконання операцій над даними цілих типів (наприклад, Integer і Byte) має тип операнда, який має більший діапазон значень.

Вихід за межі діапазону (переповнення пам'яті, виділеної під змінну) може привести до неправильних результатів обчислення.

» Обчислити вираз  $10a$  (рис. 29.2):

**var** x, a: Byte; y: Integer;

**begin**

a := 255;

x := 10 \* a; { переповнення пам'яті, оскільки  $0 \leq x \leq 255$  }

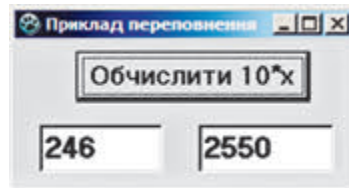


Рис. 29.2

```

y := 10 * a; // немає переповнення пам'яті, оскільки $-2^{32} \leq y \leq 2^{32} - 1$
Edit1.Text := IntToStr(x);
Edit2.Text := IntToStr(y);
end;

```

## Генератор випадкових чисел

Випадкові числа часто застосовують у програмуванні для моделювання результатів якогось експерименту за різних умов, при створенні ігрових або тестових програм тощо. Для отримання при кожному запуску програми різних випадкових чисел, необхідно виконати процедуру Randomize (*запуск генератора випадкових чисел*). Випадкові цілі числа генеруються за допомогою функції Random:

```
Random(діапазон)
```

Тут діапазон — параметр, що вказує верхню межу діапазону, з якого вибирається випадкове число.

» Отримати випадкове число  $x$  у діапазоні від 0 до 10.

```

var x: Integer;
begin
 Randomize;
 x := Random(10);
 Edit1.Text := IntToStr(x);
end;

```


## Питання для самоперевірки



- Для яких значень аргумента  $x$  функції Abs( $x$ ) і Sqr( $x$ ) повертають результат цілого типу?
- Обчисліть значення виразів:  
а) Odd(6); б) Odd(3); в) Succ(10); г) Pred(7); д) Sqr(2); е) Sqrt(16).
- Обчисліть значення виразів:  
а)  $123 \div 100$ ; б)  $123 \bmod 10$ ; в)  $123 \div 10 \bmod 10$ .
- Запишіть оператори присвоєння, які реалізують такі дії:  
а) змінній S присвоїти суму значень змінних A і B; б) подвоїти значення змінної A; в) змінну A збільшити на 10.
- Запишіть вираз, результатом якого є ціле випадкове число у діапазоні від 40 до 100.

## Вправа 29



- ▶ Скласти програму «Стрибаюча кнопка».
- 1) Створіть новий проект. Розмістіть на формі компоненти Label і Button. Змініть заголовок кнопки Button1 на «Спіймай мене!».
  - 2) Створіть процедуру обробки події OnClick для кнопки Спіймай мене!, що буде рахувати кількість клацань кнопки та виводити результат за допомогою компонента Label.  
Опишіть змінну K для збереження кількості клацань кнопки:  
`var K: Integer;`  
У програмний код процедури запишіть оператори:  
`K := K + 1;`  
`Label1.Caption := IntToStr (K);`
  - 3) Додайте на форму компонент Timer  (вкладка System на Палітрі компонентів). Властивість таймера Interval залиште без змін.
  - 4) Створіть процедуру обробки події OnTimer для об'єкта Timer. У програмному коді процедури запишіть оператори, що визначають випадковий рух кнопки (властивості Left і Top).  
Кнопка не має «вистрибнути» за межі форми, тому потрібно налаштувати властивості Left і Top відповідно до розмірів форми (ClientWidth і ClientHeight) та кнопки (Width і Height).  
`procedure TForm1.Timer1Timer(Sender: TObject);`  
`begin`  
    Randomize;  
    Button1.Left := Random(ClientWidth – Button1.Width);  
    Button1.Top := Random(ClientHeight – Button1.Height)  
`end;`
  - 5) Запустіть проект на виконання. Чи вдається «спіймати» кнопку? Якщо ні, зупиніть виконання і збільште значення властивості таймера Interval. Збережіть проект у папці Вправа 29.
  - 6) Додайте на форму кнопку Button2, в програмному коді якої запрограмуйте збільшення значення Interval:  
`Timer1.Interval := Timer1.Interval + 100;`  
Додайте на форму кнопку Button3, призначену для зменшення значення Interval. Перевірте дію кнопок.



## Комп'ютерне тестування



Виконайте тестове завдання 29 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 30. Величини дійсних типів

### Дійсні типи даних

Даними дійсних типів можуть бути будь-які дійсні числа із певного діапазону. Вони займають у пам'яті комп'ютера від 4 до 10 байтів. Характеристики деяких дійсних типів:

| Тип даних | Обсяг пам'яті (байти) | Кількість точних цифр | Діапазон значень                              |
|-----------|-----------------------|-----------------------|-----------------------------------------------|
| Extended  | 10                    | 19–20                 | $3,4 \cdot 10^{-4932} .. 1,1 \cdot 10^{4932}$ |
| Double    | 8                     | 15–16                 | $5 \cdot 10^{-324} .. 1,7 \cdot 10^{308}$     |
| Real      | 6                     | 11–12                 | $2,9 \cdot 10^{-39} .. 1,7 \cdot 10^{38}$     |

У таблиці вказано діапазони додатних значень, проте допустимими є від'ємні значення з такими самими діапазонами модулів, а також число 0.

Для даних дійсних типів визначено арифметичні операції: +, −, \*, /. Результат виконання цих операцій також має дійсний тип.

Для опрацювання даних дійсного типу в мові Object Pascal застосовують такі стандартні функції:

| Функція  | Призначення                      | Тип результату | Приклад                             |
|----------|----------------------------------|----------------|-------------------------------------|
| Frac(x)  | Виводить дробову частину числа x | Real           | Frac(1.234) = 0.234                 |
| Int(x)   | Виводить цілу частину числа x    | Real           | Int(1.234) = 1                      |
| Round(x) | Округлює число                   | Int64          | Round(1.234) = 1<br>Round(1.56) = 2 |
| Trunc(x) | Виводить цілу частину числа x    | Int64          | Trunc(1.234) = 1<br>Trunc(1.56) = 1 |
| Abs(x)   | Обчислює $ x $                   | Real           | Abs(-5.3) = 5.3                     |
| Cos(x)   | Обчислює $\cos x$                | Real           | Cos(Pi) = -1                        |
| Sin(x)   | Обчислює $\sin x$                | Real           | Sin(Pi/2) = 1                       |
| Sqr(x)   | Обчислює $x^2$                   | Real           | Sqr(5) = 25                         |
| Sqrt(x)  | Обчислює $\sqrt{x}$              | Real           | Sqrt(255) =<br>= 15.9687194226713   |



Аргумент функцій  $\text{Sin}(x)$  і  $\text{Cos}(x)$  задають у радіанах. У разі необхідності переведення значення градусів у радіани слід використовувати формулу:

$$\text{радіани} = \text{градуси} \cdot \pi / 180.$$

Якщо дробова частина аргумента функції  $\text{Round}(x)$  дорівнює 0,5, то округлення виконується до найближчого парного числа.

» Значення функцій  $\text{Round}(x)$  і  $\text{Trunc}(x)$  є цілими, хоча аргумент є дійсним.

```
var x, y: Integer; a: Real;
```

```
begin
```

```
 a := 4.65;
```

```
 x := Round(a); Edit1.Text := IntToStr(x); // x=5
```

```
 y := Trunc(a); Edit2.Text := IntToStr(y); // y=4
```

```
end;
```

Значення виразу, в якому використовуються змінні цілого і дійсного типів, має дійсний тип. При спробі присвоєння змінній дійсного типу значення цілого типу отримуємо дійсне значення.

При спробі змінній цілого типу присвоїти значення дійсного типу отримуємо повідомлення про помилку сумісності типів.

» Присвоїти змінній цілого типу  $x$  значення  $\text{Sqrt}(25)$  (рис. 30.1).

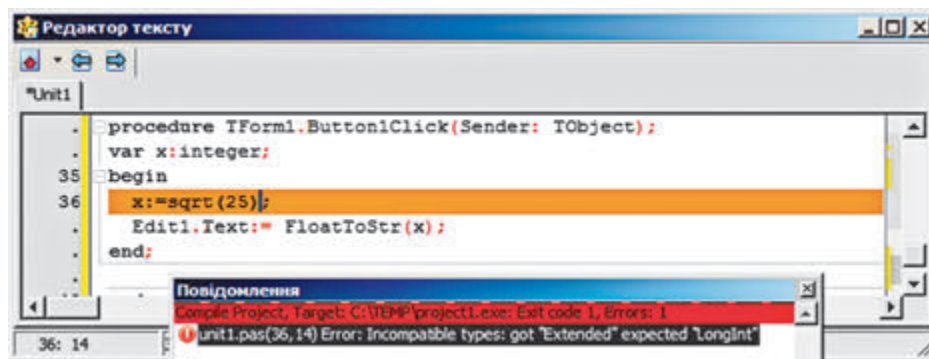


Рис. 30.1

## Правила запису арифметичних виразів

Арифметичні вирази складаються зі змінних, констант, функцій, знаків операцій і круглих дужок за математичними правилами.

Під час запису арифметичних виразів у мові Object Pascal слід дотримуватися таких правил:

| Правила запису арифметичних виразів                                                                                          | Математичний запис | Запис мовою Object Pascal |
|------------------------------------------------------------------------------------------------------------------------------|--------------------|---------------------------|
| 1. Вираз записують у вигляді рядка символів.<br>Порядок виконання операцій одного пріоритету регулюється за допомогою дужок. | $\frac{2x-5}{3+x}$ | $(2*x-5)/(3+x)$           |
|                                                                                                                              | $2ab$              | $2*a*b$                   |
| 2. Не можна опускати знак операції множення.                                                                                 | $v_0^2$            | $Sqr(v0)$                 |
| 3. Аргументи функції записують у круглих дужках                                                                              | $b+\sqrt{a}$       | $b+Sqrt(a)$               |

Обчислення виразів виконуються за певними правилами. У першу чергу обчислюється значення функції. Якщо аргумент функції є виразом, то спочатку обчислюється значення аргумента. Далі виконуються арифметичні операції зліва направо з урахуванням пріоритету та розташування дужок. Спочатку виконуються операції вищого пріоритету: \*, /, div, mod, а потім — нижчого пріоритету: +, -.

» Обчислити значення виразу  $\text{Sin}(2*a)$ .  
У виразі  $\text{Sin}(2*a)$  спочатку обчислюється значення аргумента  $2*a$ , а потім — значення функції  $\text{Sin}$ .

» Обчислити значення виразів  $24/2*3$  і  $24/(2*3)$ .  
У результаті виконання операторів присвоєння  
 $A := 24/2*3$ ;  $B := 24/(2*3)$ ;  
змінна A набуде значення 36, змінна B — значення 4.

### Питання для самоперевірки



- Визначте тип (цілий чи дійсний) виразів:  
а)  $20/4$ ; б)  $\text{Sqr}(4)$ ; в)  $\text{Sqrt}(16)$ ; г)  $\text{Trunc}(-3.14)$ .
- Які функції мають різні типи аргумента та результату?
- Обчисліть значення виразів:  
а)  $\text{Trunc}(6.9)$ ;  $\text{Round}(6.9)$ ;  $\text{Int}(6.9)$ ; б)  $\text{Trunc}(-1.8)$ ;  $\text{Round}(-1.8)$ ;  $\text{Int}(-1.8)$ .
- Поясніть правила запису арифметичних виразів.
- Запишіть наведені формули за правилами мови Object Pascal:  
а)  $\frac{x-5}{2x}$ ; б)  $x^5$ ; в)  $2a+10^5d-4\frac{3}{7}c^5+\frac{2}{3}$ ; г)  $\sqrt{|x-1|}$ .
- Обчисліть значення виразу  $\text{Succ}(\text{Round}(5/2))-\text{Pred}(3)$ .

## Вправа 30



▶ Дано функцію  $y = \sqrt{|x-1| + \sin x}$ . Скласти програму для знаходження значення функції при заданому значенні  $x$ .

- 1) Створіть новий проект. Змініть заголовок форми на «Обчислення значення функції». Розмістіть на формі кнопку Button1 і змініть заголовок створеної кнопки на «Обчислити Y».
- 2) Розмістіть на формі три компоненти Label, змініть їх заголовки згідно з рисунком.
- 3) Розмістіть на формі три текстові поля Edit згідно з рисунком.

4) Створіть процедуру обробки події OnClick для кнопки Обчислити Y. У програмному коді створеної процедури запишіть оператори:

```
var A: Integer; X, Y: Real;
```

```
begin
```

```
 X := StrToFloat(Edit1.Text);
```

```
 Y := Sqrt (Abs(X-1)+Sin(X));
```

```
 Edit2.Text := FloatToStr(Y);
```

```
 A := Round(Y);
```

```
 Edit3.Text := IntToStr(A);
```

```
end;
```

- 5) Запустіть проект на виконання. На рисунку подано результат роботи програми при  $x=12,7409$ .

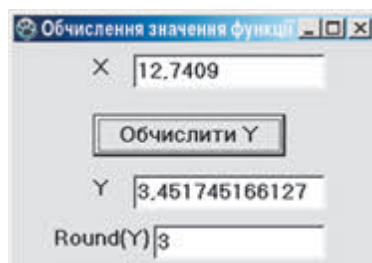
! У програмному коді цілу і дробову частини дійсного числа відокремлюють крапкою «.». Роздільник при введенні дробового числа в поле Edit1 залежить від налаштувань системи (в українському варіанті — кома «,»).

- 6) Перевірте роботу програми для різних значень  $x$ . Збережіть проект у папці Вправа 30.



- 7) Змініть програмний код для знаходження значення функції

$y = 1 + \frac{1}{x} + \frac{1}{x^2}$ . Обчисліть значення цієї функції для  $x=-0,387$ .



## Комп'ютерне тестування



Виконайте тестове завдання 30 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 31. Величини логічного типу

### Прості умови

Простий логічний вираз (проста умова) утворюється за допомогою операції відношення:

| Знак операції | Зміст               | Приклад | Результат |
|---------------|---------------------|---------|-----------|
| =             | Дорівнює            | 8 = 9   | False     |
| >             | Більше              | 8 > 9   | False     |
| <             | Менше               | 8 < 9   | True      |
| >=            | Більше або дорівнює | 5 >= 5  | True      |
| <=            | Менше або дорівнює  | 5 <= 2  | False     |
| <>            | Не дорівнює         | 2 <> 5  | True      |

Логічні вирази набувають значення True або False. У мові Object Pascal для величини, значенням якої є True або False, використовують логічний тип (Boolean).

» Обчислити значення логічних виразів:

```
var X, Y: Integer; A: Boolean;
```

```
begin
```

```
 X := 5; Y := 2;
```

```
 A := X > Y; // A = True, тому виконується умова X > Y
```

```
 A := X < Y; // A = False
```

```
 A := X - 4.5 < Y * 2; // A = True
```

```
end;
```

### Складені умови

Складений вираз (складена умова) — кілька простих виразів, з'єднаних логічними операціями and (*і*), or (*або*), xor (виключне *або*), not (*ні*).

Таблиці істинності

| A     | B     | A and B | A or B | A xor B |
|-------|-------|---------|--------|---------|
| False | False | False   | False  | False   |
| False | True  | False   | True   | True    |
| True  | False | False   | True   | True    |
| True  | True  | True    | True   | False   |

| A     | not A |
|-------|-------|
| False | True  |
| True  | False |

Логічні операції, операції відношення й арифметичні операції можуть зустрічатися в одному виразі. Обчислення таких виразів виконуються зліва направо з урахуванням пріоритету операцій:

- 1) not;
- 2) and, \*, /, div, mod;
- 3) or, xor, +, -;
- 4) операції порівняння.

» Розглянемо приклади складених логічних виразів:

- a) not ( $A \leq 3$ ) — те саме, що  $A > 3$ .
- б)  $(A > 10)$  and  $(A < 18)$  — набуває значення True, якщо значення  $A$  належить проміжку від 10 до 18.
- в)  $(A \leq 10)$  or  $(A \geq 18)$  — набуває значення True, якщо значення  $A$  не належить проміжку від 10 до 18.

» Визначити, чи належить точка з координатою  $x$  відріжку  $[-5; 5]$  (рис. 31.1). Точка належить відріжку, якщо справджується нерівність  $-5 \leq x \leq 5$ . У програмуванні таку подвійну нерівність записують як складену умову:

$(x \geq -5)$  and  $(x \leq 5)$ .

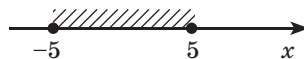


Рис. 31.1

» Обчислити значення логічних виразів:

var X, Y, Z: Integer; A: Boolean;

begin

X := 1; Y := 2; Z := 3;

A := (X < Y) and (Y < Z); // A = True

A := (X > Y) or (Y > Z); // A = False

A := (X > Y) xor (Y < Z); // A = True

end;

## Перетворення типів

При виведенні логічних значень треба перетворювати логічне значення на рядок символів, і навпаки, при введенні перетворювати рядок символів на відповідне логічне значення. Для цього в мові Object Pascal існують вбудовані функції перетворення типів:

- StrToBool(S) — перетворення рядка символів S на логічне значення;
- BoolToStr(b,True) — перетворення логічного значення на рядок символів.

Якщо другий аргумент функції BoolToStr дорівнює True, то отримуємо значення функції у вигляді рядка 'True' або 'False'. Якщо другий аргумент замінити на False, то значенням функції для істинного  $b$  буде рядок '-1', а для хибного — '0'.

### Питання для самоперевірки



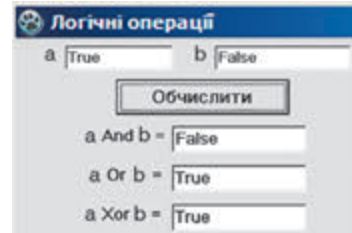
1. Яких значень може набувати змінна логічного типу?
2. Які операції можна виконувати над змінними логічного типу?
3. Змінні логічного типу  $A$  і  $B$  отримують значення в результаті виконання операторів:  $A := X > 3$ ;  $B := X < 0$ ;  
За яких значень  $X$  вираз  $A$  or  $B$  набуває значення True?  
За яких значень  $X$  вираз  $A$  and  $B$  має значення False?
4. Запишіть у вигляді логічних виразів подані умови:  
а)  $3 \leq x \leq 10$ ; б)  $-5 \leq x \leq 5$  за умови, що  $x \neq 0$ .
5. Обчисліть значення логічних виразів:  
а)  $(A \leq B)$  and  $(A = B - 1)$ , якщо  $A = 2$ ,  $B = 4$ ;  
б)  $(A \leq B)$  or  $(A = B - 1)$ , якщо  $A = 2$ ,  $B = 4$ .

### Вправа 31



►► За введеними значеннями змінних  $a$ ,  $b$ : Boolean обчислити значення логічних виразів  $a$  and  $b$ ,  $a$  or  $b$ ,  $a$  xor  $b$ .

- 1) Створіть новий проект. Змініть заголовок форми на «Логічні операції». Розмістіть на формі кнопку Button1 і змініть заголовок створеної кнопки на «Обчислити».
- 2) Розмістіть на формі по п'ять компонентів Label і Edit згідно з рисунком.
- 3) Створіть процедуру обробки події OnClick для кнопки Обчислити. У програмному коді створеної процедури запишіть оператори:



```
var a, b, v1, v2, v3: Boolean;
begin
```

```
 a := StrToBool(Edit1.Text);
 b := StrToBool(Edit2.Text);
 v1 := a and b; v2 := a or b; v3 := a xor b;
 Edit3.Text := BoolToStr(v1,True);
 Edit4.Text := BoolToStr(v2,True);
 Edit5.Text := BoolToStr(v3,True);
```

```
end;
```

- 4) Запустіть проект на виконання. Перевірте результат роботи програми при  $a = \text{True}$ ,  $b = \text{False}$ .
- 5) Перевірте роботу програми для різних значень  $a$  і  $b$ . Збережіть проект у папці Вправа 31.

### Комп'ютерне тестування



Виконайте тестове завдання 31 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)

## § 32. Величини символного типу

Для величини, значенням якої є окремий символ, використовують символний тип (Char). Як вам уже відомо, кожен символ має свій номер, який називають його кодом.

Таким чином, значенням змінної символного типу може бути будь-який символ із таблиці кодів: літера; цифра; розділовий знак, а також службові символи, які не відображаються на екрані.

У мові Object Pascal визначені два символні типи, один з яких забезпечує кодування символів у системі ASCII, а другий — у системі Юнікод.

Значення величин символного типу (Char) у тексті програми записують в одинарних лапках, наприклад 'A', '3', '!', ' ' (пробіл), або використовують його код, додаючи префікс #. Зокрема, так зручно позначати службові символи. Наприклад, символ повернення каретки — #13, символ нового рядка — #10.

» Присвоїти значення змінним символного типу:

```
var Letter, Symbol: Ch, Char;
begin
 Letter := 'a';
 Symbol := '7';
 Ch := #65;
end;
```

Величини символного типу можна порівнювати. Порівняння символів зводиться до порівняння їхніх кодів, оскільки кожному символу відповідає певне число.



» Порівняти величини символного типу:

```
var b: Boolean;
begin
 b := 'A' < 'B'; // True
end;
```

» Вивести на форму таблицю кодів для великих літер латиниці. Під час клацання командної кнопки в поле Label1 буде виводитися частина таблиці кодів (від 65 до 90), які відповідають великим літерам латиниці (рис. 32.1).

```
var c: Char; n: Byte; st: String;
begin
 st := "";
 For n := 65 to 90 do
 st := st + IntToStr(n) + ' ' + Chr(n) + #13;
 Label1.Caption := st;
end;
```



Рис. 32.1

До величин символного типу в мові Object Pascal застосовують такі вбудовані функції:

| Функція           | Призначення                      | Тип результату | Приклад                              |
|-------------------|----------------------------------|----------------|--------------------------------------|
| Chr(x)            | Повертає символ за його кодом    | Char           | S := Chr (65);<br>S = 'A'            |
| Ord(c)            | Повертає числовий код символу c  | Byte           | K := Ord ('Z');<br>K = 90            |
| Pred(c)           | Повертає символ, попередній до c | Char           | S := Pred ('b');<br>S = 'a'          |
| Succ(c)           | Повертає символ, наступний за c  | Char           | S := Succ ('b');<br>S = 'c'          |
| AnsiUpperCase(c)* | Перетворює малу літеру на велику | Char           | S := AnsiUpperCase ('b');<br>S = 'B' |

### Питання для самоперевірки



1. Яких значень можуть набувати змінні символного типу?
2. Які функції можна виконувати із символними даними?
3. Яка функція повертає символ за його кодом?

\* Великі літери і символи, які не є літерами, функція AnsiUpperCase не змінює.

4. Яка функція повертає числовий код символу?
5. Як порівнюються значення символьного типу?
6. Знайдіть значення виразів:
  - а)  $\text{Pred}('n')$ ; б)  $\text{Ord}('z') - \text{Ord}('y')$ ; в)  $\text{Ord}(\text{Chr}(13))$ .

### Вправа 32



► Для символу, що вводиться в текстове поле Edit1, визначити його код, попередній і наступний символи. Для малих літер латиниці вивести відповідні великі літери.

- 1) Створіть новий проект. Змініть заголовок форми на «Стандартні функції для роботи із символами». Розмістіть на формі кнопку Button1, змініть заголовок створеної кнопки на «Виконати».
- 2) Розмістіть на формі п'ять компонентів Label згідно з рисунком.
- 3) Розмістіть на формі п'ять компонентів Edit згідно з рисунком.
- 4) Створіть процедуру обробки події OnClick для кнопки Виконати. У програмному коді створеної процедури запишіть оператори:

```
var c: Char; st: String;
```

```
begin
```

```
 st := Edit1.Text;
```

```
 c := st[1];
```

```
 Edit2.Text := IntToStr(Ord(c));
```

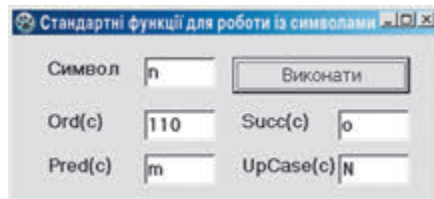
```
 Edit3.Text := Pred(c);
```

```
 Edit4.Text := Succ(c);
```

```
 Edit5.Text := AnsiUpperCase(c);
```

```
end;
```

- 5) Запустіть проект на виконання. Перевірте результат роботи програми для літери n.
- 6) Перевірте роботу програми для інших символів. Збережіть проект у папці Вправа 32.



### Комп'ютерне тестування



Виконайте тестове завдання 32 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua)

## § 33. Величини рядкового типу

**Рядок** — це послідовність символів. Для зберігання рядків призначений рядковий тип даних (String), який широко використовується в мові Object Pascal для опрацювання текстових даних.

Рядкові константи у тексті програми беруть в одинарні лапки. Опис рядкової змінної має вигляд:

```
var S: String;
```

Кількість символів у рядку може досягати  $2^8 - 1 = 255$  символів.

Довжину рядкової величини можна обмежити, якщо в її описі вказати максимально допустиму кількість символів:

```
var S: String[10];
```

Звернутися до окремого символу рядкової змінної можна за його номером у рядку:  $S[i]$  —  $i$ -й символ рядка  $S$ .

» Перетворити текст «to learn» на текст «to reach», виконуючи операції над символами заданого рядка.

```
var x: String;
```

```
begin X := 'to learn';
```

```
 X[4] := X[7]; X[7] := 'c'; X[8] := 'h';
```

```
end;
```



**Конкатенацією** двох або більше рядків називають їх злиття в один рядок. Цю операцію позначають символом '+

Операція конкатенації дозволяє об'єднувати в один рядок змінні й константи рядкового типу (String) і символного типу (Char).

» Виконати конкатенацію рядків.

```
var X: String;
```

```
begin
```

```
 X := 'a' + 'Ab' + 'Bc'; // отримуємо X = 'aAbBc';
```

```
end;
```

### Стандартні функції для обробки рядків

Для роботи з величинами рядкового типу у мові Object Pascal існують стандартні процедури і функції. Розглянемо деякі з них.

- 1) Функція `Length(S: String)` повертає довжину рядка (кількість символів).

```

>> var S: String; L: Integer;
 begin
 S := 'Hello!';
 L := Length(S); // L = 6
 end;

```

- 2) Функція `Pos(SubS, S: String)` повертає позицію першого входження підрядка `SubS` в рядок `S`. Нумерація символів починається з одиниці. У разі відсутності підрядка в рядку функція повертає 0.

```

>> var S, SubS: String; P: Integer;
 begin
 S := 'Hi! How do you do?';
 SubS := 'do';
 P := Pos(SubS, S); // P = 9
 end;

```

- 3) Функція `Copy(S: String; n, Length: Integer)` повертає підрядок рядка `S` завдовжки `Length`, починаючи з  $n$ -го символу.

```

>> var S1, S2: String;
 begin
 S1 := 'This is a test for Copy(OK) function.';
 S2 := Copy(S1, 11, 4); // S2 = 'test'
 end;

```

- 4) Процедура `Delete(S: String; i, n: Integer)` вилучає  $n$  символів із рядка `S`, починаючи з  $i$ -го символу.

```

>> var S: String;
 begin
 S := 'Hello, world!';
 Delete(S, 6, 7); // S = 'Hello!'
 end;

```

- 5) Процедура `Insert(SubS: String; S: String; Pos: Integer)` вставляє в рядок `S` підрядок `SubS` у позицію `Pos`.

```

>> var S: String;
 begin
 S := 'Hello, world!';
 Insert(' my', S, 8); // S = 'Hello, my world!'
 end;

```

- 6) Функція `StringReplace(S, OldS, NewS, [rfReplaceAll, rflgnoreCase])` замінює в рядку `S` підрядок `OldS` на рядок `NewS`. У квадратних дужках задаються параметри заміни:  
**rfReplaceAll**: замінити всі входження підрядка `OldS` у рядку `S`;  
**rflgnoreCase**: нехтувати регістром під час пошуку підрядка `OldS` у рядку `S`.

```

>> var before, after: String;
 begin // Заміна всіх входжень A на THE
 before := 'This is a way to live A big life';
 after := StringReplace(before, 'a', 'THE', [rfReplaceAll, rflgnoreCase]);
 { after= 'This is THE wTHEy to live THE big life }
 end;

```

- 7) Функція `ReverseString(S: String)` розташовує символи рядка `S` у зворотному порядку. Для застосування цієї функції необхідно підключити до програми модуль `StrUtils`.

```

>> uses StrUtils;
 var S1, S2: String;
 begin
 S1 := '1234abcd';
 S2 := ReverseString(S1); // S2 = 'dcba4321'
 end;

```

## Порівняння рядків

Рядки можна порівнювати один з одним, використовуючи лексикографічний порядок (подібний до порядку слів у словнику). Рядки порівнюються посимвольно згідно з їхніми кодами в таблиці кодів.

```

var S1, S2, S3: String; B1, B2: Boolean;
begin
 S1 := '123'; S2 := '456';
 B1 := (S1 = S2); // B1 = False
 B2 := (S1 < S2); // B2 = True
 S1 := 'abcd'; S2 := 'bcd';
 B1 := (S1 < S2); // B1 = True
end;

```

## Питання для самоперевірки



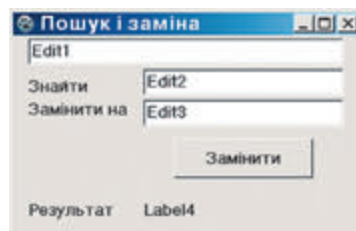
1. Що таке довжина рядка? Як її визначити?
2. Як можна виконати злиття рядків?

3. Яка функція повертає позицію першого входження підрядка у рядок?
4. Яке значення повертає функція Pos(SubS, S), якщо підрядок SubS у рядку S відсутній?
5. Як можна отримати підрядок заданого рядка?
6. Як можна вилучити підрядок із заданого рядка?

### Вправа 33



- Скласти програму, що виконує задану заміну символів у рядку.
- 1) Створіть новий проект. Змініть заголовок форми на «Пошук і заміна». Розмістіть на формі кнопку Button1 і змініть заголовок створеної кнопки на «Замінити».
  - 2) Розмістіть на формі чотири компоненти Label згідно з рисунком. Для елемента Label4 задайте такі значення властивостей: AutoSize: False; WordWrap: True.
  - 3) Розмістіть на формі три компоненти Edit згідно з рисунком.
  - 4) Створіть процедуру обробки події OnClick для кнопки Замінити. У програмному коді створеної процедури запишіть оператори:



```
var S, S1, S2, S3: String;
begin
 S := Edit1.Text;
 S1 := Edit2.Text; S2 := Edit3.Text;
 S3 := StringReplace(S, S1, S2, [rfReplaceAll, rflgnoreCase]);
 Label4.Caption := S3;
end;
```

- 5) Запустіть проект на виконання. Перевірте роботу програми для S = 'Why do You cry, Willy, why do You cry?'; S1 = 'cry'; S2 = 'smile'. Збережіть проект у папці Вправа 33.
- 6) Змініть програмний код так, щоб у реченні, вказаному в п. 5, літери «W» і «w» були замінені символом «\*».



### Комп'ютерне тестування



Виконайте тестове завдання 33 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## § 34. Перетворення величин одного типу на інший

Вам уже доводилося перетворювати рядкові дані на числові, числові — на рядкові тощо. Розглянемо функції мови Object Pascal для перетворення величин різних типів.

### Функції перетворення чисел на рядки і рядків на числа

Ці функції — `StrToInt`, `StrToFloat`, `IntToStr` і `FloatToStr` — ви вже використовували. Розглянемо ще декілька прикладів.

» Перетворити ціле число на рядок цифр:

```
var a: Integer; s: String;
begin
 a := 12345;
 s := IntToStr(a); // s = '12345'
end;
```

» Перетворити рядок цифр на ціле число:

```
var a: Integer; s: String;
begin
 s := '12345';
 a := StrToInt(s); // a = 12345
end;
```

» Перетворити рядок символів на дійсне число (рис. 34.1):

```
var a: Real;
begin
 a := StrToFloat('12,45');
 a := 2 * a;
 Edit1.Text := FloatToStr(a);
end;
```

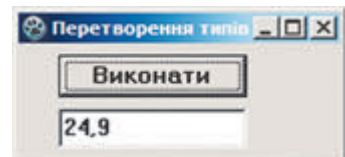


Рис. 34.1

! При введенні десяткових дробів слід пам'ятати, що вибір роздільника залежить від налаштування системи (див. с. 156).



У разі неправильного вибору роздільника програма припиняє роботу і виводиться вікно з повідомленням про помилку (рис. 34.2).



Рис. 34.2

## Виведення дробових чисел за форматом

Для виведення дробових чисел із заданою кількістю знаків у дробовій частині використовується функція

`FormatFloat(const Format: String; Value: Extended): String;`

Ця функція перетворює значення дійсного числа (Value) на рядок, поданий за шаблоном (параметр Format). Шаблон може містити такі символи форматування:

|   |                                                                                                            |
|---|------------------------------------------------------------------------------------------------------------|
| . | Місце розташування десяткового роздільника                                                                 |
| 0 | Виведення цифри десяткового дробу у відповідну позицію. Якщо цифри в позиції немає — виводиться "0"        |
| # | Виведення цифри десяткового дробу у відповідну позицію. Якщо цифри в позиції немає — не записується нічого |

» Вивести дійсне число з двома знаками після коми (рис. 34.3):

```
var S: String;
```

```
begin
```

```
 S := FormatFloat('#.##', Pi); // S := '3.14'
```

```
 MessageDlg('Pi :=' + S, mtInformation, [mbOk], 0);
```

```
end;
```

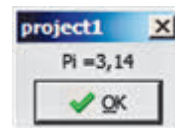


Рис. 34.3

## Неявне перетворення типів даних

**Неявними** називають ті перетворення типів змінних, які відбуваються без безпосередньої вказівки користувача. Зокрема, при присвоєнні змінній дійсного типу цілого значення, воно перетворюється на дійсне.

» `var d: Double; a: Integer;`

```
begin
```

```
 a := 100;
```

```
 d := a; // правильно
```

```
 a := d; // помилка
```

```
end;
```

При наданні змінній певного типу значення, яке не може бути автоматично перетворено відповідно до типу змінної, виникає помилка компіляції:

Incompatible types:... (*Несумісні типи...*).

» Знайти помилку та пояснити приклад.

```
var a: Integer;
begin
 a := Sqrt(100); // Несумісні типи
end;
```

Функція `Sqrt(x)` повертає результат дійсного типу.

Під час виконання програми буде отримано повідомлення про помилку `Incompatible types: 'Integer' and 'Extended'`.

## Функції перетворення дат і часу

При створенні проектів вам можуть стати в пригоді такі функції:

| Функція                                 | Тип результату         | Призначення                                                     |
|-----------------------------------------|------------------------|-----------------------------------------------------------------|
| <code>Date</code>                       | <code>TDateTime</code> | Повертає поточну дату                                           |
| <code>Time</code>                       | <code>TDateTime</code> | Повертає поточний час                                           |
| <code>DayOfWeek(Date: TDateTime)</code> | <code>Integer</code>   | Повертає поточний номер дня тижня: від 1 (неділя) до 7 (субота) |

Функції перетворення дати і часу з формату `TDateTime` на рядковий формат (у всіх функцій тип результату — `String`):

| Функція                                         | Призначення                                          |
|-------------------------------------------------|------------------------------------------------------|
| <code>DateTimeToStr(DateTime: TDateTime)</code> | Перетворює дату і час <code>DateTime</code> на рядок |
| <code>DateToStr(Date: TDateTime)</code>         | Перетворює дату <code>Date</code> на рядок           |
| <code>TimeToStr(Time: TDateTime)</code>         | Перетворює час <code>Time</code> на рядок            |

Для задавання, введення і виведення значень типу `DateTime` застосовують такий формат: `hh:mm:ss` — для часу; `dd.mm.yyyy` — для дати.

» Присвоїти значення типу `DateTime`:

```
var s: String; t, d: TDateTime;
begin
 s := '8:30:15'; t := StrToDate(s);
 s := '19.02.2016'; d := StrToDate(s);
end;
```

» Вивести значення поточної дати і часу (рис. 34.4):

```
var t, d: TDateTime;
begin
 t := Time; d := Date;
 ShowMessage ('поточна дата =' + DateToStr(d));
 ShowMessage ('поточний час =' + TimeToStr(t));
end;
```



Рис. 34.4

### Питання для самоперевірки



1. Як перетворити значення дійсного типу на рядок символів?
2. Як перетворити значення цілого типу на рядок символів?
3. Як перетворити рядок символів на значення числового типу?
4. Як вивести десятковий дріб із заданою кількістю знаків у дробовій частині?
5. В яких випадках відбувається неявне перетворення типів даних?
6. Які функції призначені для роботи з датою та часом?

### Вправа 34



» Скласти програму «Годинник».

- 1) Створіть новий проект. Змініть заголовок форми на «Годинник». Розмістіть на формі два компоненти Edit для виведення поточних значень часу і номера дня тижня. Поряд із полем Edit2 створіть напис (Label1) «День тижня:».
- 2) Додайте на форму компонент Timer (див. рисунок).
- 3) Створіть процедуру обробки події OnTimer для об'єкта Timer. У програмному коді запишіть оператори виведення у поля Edit1 і Edit2 поточних часу і дня тижня відповідно.

```
procedure TForm1.Timer1Timer(Sender: TObject);
```

```
var t: TDateTime; day: String;
```

```
begin
```

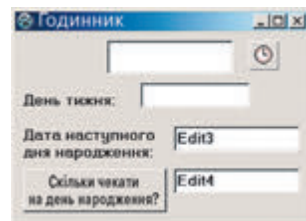
```
 t := Time;
```

```
 Edit1.Text := TimeToStr(t);
```

```
 day := IntToStr(DayOfWeek(Date));
```

```
 Edit2.Text := day;
```

```
end;
```



- 4) Додайте на форму кнопку Скільки чекати на день народження? та текстові поля Edit3, Edit4 згідно з рисунком. Поряд із полем Edit3 створіть напис (Label2) «Дата наступного дня народження:».

- 5) Створіть процедуру обробки події OnClick для кнопки. У програмному коді створеної процедури запишіть оператори для визначення кількості днів між датою наступного дня народження і поточною датою:
- ```
var day: Double; S: String;
begin
    S := Edit3.Text; // введення дати наступного дня народження
    day := Date — StrToDate(S); // різниця між двома датами
    Edit4.Text := FloatToStr(day); { перетворення дійсного числа day на
    рядок символів }
end;
```
- 6) Перевірте роботу програми. Чому в полі виводиться від'ємне число? Виправте помилку. Збережіть проект у папці Вправа 34.

Комп'ютерне тестування

Виконайте тестове завдання 34 з автоматичною перевіркою на сайті interactive.ranok.com.ua.



Практична робота 14

Налагодження готової програми

Завдання: створити проект для розв'язання задачі.

Відомі назва планети, радіус (км) її орбіти, швидкість руху (км/год) по орбіті. Обчислити тривалість року на планеті (період обертання планети).

Налагодити програмний код. Перевірити роботу програми для заданих тестових даних:

Назва планети	Радіус орбіти (км)	Швидкість руху (км/год)	Період обертання (доби)
Земля	149 600 000	107 250	365,18
Венера	108 200 000	126 110	
Нептун	4 496 600 000	19 550	
Марс	227 900 000	86 870	

Обладнання: ПК із встановленим середовищем програмування Lazarus.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.

I. Розміщення елементів керування на формі

- ▶ 1. Створіть новий проект. Розробіть інтерфейс програми (рис. 1). Змініть для об'єктів Label значення властивості WordWrap на True.

II. Налаштування програмного коду

- ▶ 2. Створіть процедуру обробки події OnClick для кнопки Обчислити. У програмному коді створеної процедури запишіть оператори:

```
var Nazva: String; R, Period: Real;
Const Pi = 3.1415926;
begin
    Nazva := Edit1.Text; R = Edit2.Text;
    V = Edit3.Text;
    Period := 2 * Pi * R / V * 24;
    Edit4.Text := FloatToStr(Period);
    ShowMessage('Пік на планеті ' + 'Nazva'
    + ' триває ' + FormatFloat('#0.00',Period) + ' діб');
end;
```

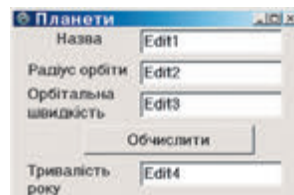


Рис. 1

У програмному коді допущено помилки, можливо, ви помітите їх під час набору тексту програми і виправите, якщо ні — вам допоможе виправити помилки компілятор.

III. Тестування програми

- ▶ 3. Після виправлення синтаксичних помилок перевірте роботу програми для даних планети Земля (рис. 2). Чи збігається відповідь із вказаною в таблиці? Якщо ні — проаналізуйте вираз, який обчислює значення Period:
$$\text{період обертання} = \frac{\text{довжина орбіти}}{\text{швидкість руху} \cdot 24}$$
, та виправте помилку.
- ▶ 4. Виконайте обчислення для Венери, Нептуна, Марса. Доповніть таблицю.
- ▶ 5. Які зміни потрібно внести, щоб результат обчислень виводився в земних роках?
- ▶ 6. Збережіть проект у папці Практична робота 14. Завершіть роботу за комп'ютером.

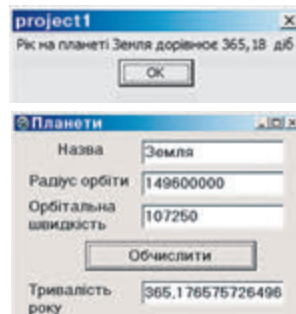


Рис. 2

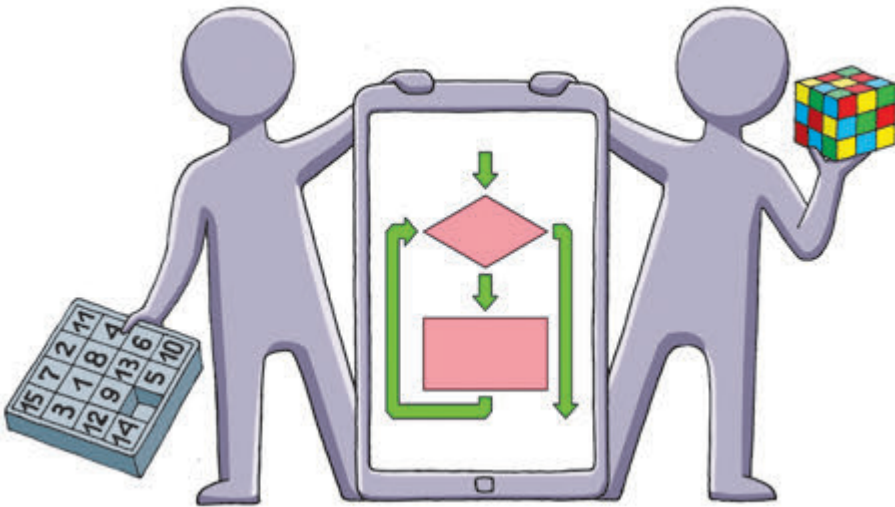
Зробіть висновок: як виправляти синтаксичні та логічні помилки у програмному коді й аналізувати результат роботи програми.

РОЗДІЛ 8

АЛГОРИТМИ

З ПОВТОРЕННЯМИ

ТА РОЗГАЛУЖЕННЯМИ



- § 35. Алгоритми з розгалуженнями
 - § 36. Складання програм із розгалуженнями
 - § 37. Поліваріантне розгалуження
 - § 38. Компоненти вибору
 - § 39. Алгоритми з повтореннями. Цикл із передумовою
 - § 40. Алгоритми з повтореннями. Цикл із післяумовою
 - § 41. Алгоритми з повтореннями. Цикл із параметром
 - § 42. Складання циклічних алгоритмів опрацювання величин
- Практична робота 15. Складання та виконання алгоритмів із повтореннями та розгалуженнями для опрацювання величин

ПОВТОРЮЄМО

Алгоритм — це чітко визначена послідовність вказівок для виконання дій, спрямована на розв'язання певної задачі.

З курсу математики вам добре відомі алгоритми виконання арифметичних операцій над багатоцифровими числами; знаходження коренів лінійного і квадратного рівнянь; поділу відрізка на рівні частини; побудови трикутника за заданими сторонами тощо. У курсі інформатики ви склали алгоритми не тільки обчислювальних, а й ігрових задач.

Ви вже знаєте, що будь-який алгоритм можна побудувати, використовуючи три базові алгоритмічні структури: слідування, розгалуження, повторення. Для графічного подання алгоритмів використовують блок-схеми.

У *лінійних алгоритмах* використовується тільки структура слідування. Алгоритми, в основі яких лежить структура розгалуження, називають *алгоритмами з розгалуженнями*. Алгоритми, в основі яких лежить структура повторення, називають *циклічними*.



1. Що таке алгоритм?
2. Які базові алгоритмічні структури використовують для конструювання алгоритмів?
3. З якою метою складають блок-схеми алгоритмів?
4. Яку алгоритмічну структуру використовують у лінійних алгоритмах?



У цьому розділі ви ознайомитесь із правилами реалізації алгоритмів із розгалуженнями і повтореннями у мові програмування *Object Pascal*.

§ 35. Алгоритми з розгалуженнями



Розгалуження — це алгоритмічна структура, в якій залежно від виконання чи невиконання деякої умови здійснюється або та, або інша послідовність дій.

Для організації розгалужень у мові Object Pascal призначені умовні оператори If...Then та If...Then...Else.

Умовний оператор If...Then

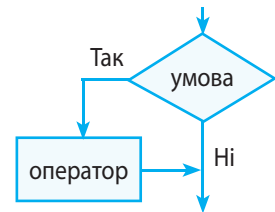
Оператор If...Then призначено для виконання деякої послідовності дій у тому випадку, якщо істинною є зазначена умова. Цей умовний оператор відповідає алгоритмічній конструкції «неповне розгалуження».

Синтаксис оператора:

If <умова> Then <оператор>;

Оператор If перевіряє істинність зазначеної умови. Як умова використовується логічний вираз (або логічна змінна як окремий випадок логічного виразу). Умова істинна, якщо логічний вираз набуває значення True, і хибна в протилежному випадку.

Якщо умова істинна (True), програма виконає дію, зазначену в частині <оператор>. Якщо ж умова хибна (False), то керування передається оператору, що міститься після оператора If.



» Збільшити значення змінної A на одиницю, якщо її поточне значення менше від 5.

Оператор розгалуження має вигляд: If A < 5 Then A := A + 1;

Оператор A := A + 1 виконується тільки в тому випадку, коли істинною є умова A < 5:

Початкове значення A	Значення умови	Оператор A := A + 1	Значення A після виконання оператора If
1	True	Виконується	2
5	False	Не виконується	5
10	False	Не виконується	10

Якщо у випадку істинності умови потрібно виконати деяку послідовність дій, оператори, що реалізують ці дії, беруться в операторні дужки **begin...end**. Таку конструкцію називають **складеним оператором**.

» Поміняти місцями значення змінних a і b , щоб була вірною нерівність $a \leq b$.

```
var a, b, x: Integer;
begin
  a := StrToInt(TextBox('Введіть a', 'a=', '0'));
  b := StrToInt(TextBox('Введіть b', 'b=', '0'));
  If a > b Then begin
    x := a; // для обміну значеннями між змінними a і b
    a := b; // використовується додаткова змінна x
    b := x
  end;
  Edit1.Text := IntToStr(a);
  Edit2.Text := IntToStr(b);
end;
```

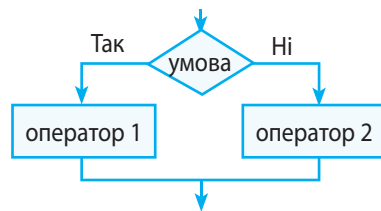
Якщо умова хибна, керування передається оператору `Edit1.Text := IntToStr(a)`.

Умовний оператор If...Then...Else

Алгоритмічній конструкції «повне розгалуження» відповідає умовний оператор `If...Then...Else`.

Синтаксис оператора:

```
If <умова> Then <оператор 1>
  Else <оператор 2>;
```



! Перед службовим словом `Else` крапка з комою не ставиться.

Якщо результатом перевірки умови є значення `True`, то виконується блок дій `<оператор 1>`, який міститься після службового слова `Then`. Якщо перевірка умови дала результат `False`, виконується блок дій `<оператор 2>`, який міститься після службового слова `Else`.

» Збільшити змінну a на 1, якщо $a < 5$, і зменшити на 1, якщо $a \geq 5$.

```
If a < 5 Then a := a + 1 Else a := a - 1;
```

► Перевірити існування трикутника зі сторонами a , b , c .
Умова існування трикутника із заданими довжинами сторін: сума двох будь-яких довжин сторін повинна бути більшою за третю.

var a, b, c : Integer;

begin

$a := \text{StrToInt}(\text{InputBox}(\text{'Введіть } a, \text{'a='}, 0))$;

$b := \text{StrToInt}(\text{InputBox}(\text{'Введіть } b, \text{'b='}, 0))$;

$c := \text{StrToInt}(\text{InputBox}(\text{'Введіть } c, \text{'c='}, 0))$;

If $(a < b + c)$ and $(b < a + c)$ and $(c < a + b)$ Then

 Edit1.Text := 'трикутник існує'

Else Edit1.Text := 'трикутника не існує';

end;

Питання для самоперевірки



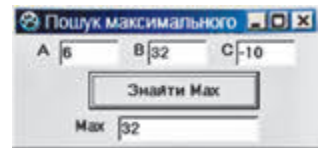
1. Дайте означення розгалуження як алгоритмічної конструкції.
2. Які вирази використовують як умову в умовному операторі?
3. Поясніть, як виконується умовний оператор в неповній формі.
4. Як записується і виконується умовний оператор в повній формі?
5. Початкові значення змінних $A = -3$; $B = 5$. Знайдіть значення змінних A і B після виконання таких операторів:
 - а) If $A > B$ Then $A := A - B$ Else $B := B - A$;
 - б) If $A <> B$ Then $A := B$;
 - в) If $A < B$ Then $A := 2 * A$ Else $B := B * A$;
6. Запишіть умовні оператори, за допомогою яких можна виконати такі дії:
 - а) перевірити, чи є число a парним;
 - б) від більшого із заданих чисел a і b відняти менше;
 - в) перевірити, чи є серед чисел a , b , c рівні.

Вправа 35



►► Визначити більше з трьох чисел a , b , c .

- 1) Створіть новий проект. Змініть заголовок форми на «Пошук максимального числа». Розмістіть на формі чотири компоненти Edit для введення значень a , b , c та виведення значення Max найбільшого з трьох чисел згідно з рисунком.
- 2) Додайте на форму кнопку Button1 і змініть заголовок доданої кнопки на «Знайти Max».



- 3) Створіть процедуру обробки події OnClick для кнопки Знайти Max. У програмному коді створеної процедури запишіть оператори:


```
var a, b, c, Max: Integer;
begin
  a := StrToInt(Edit1.Text); b := StrToInt(Edit2.Text); c := StrToInt(Edit3.Text);
  If a > b Then Max := a Else Max := b;
  If c > Max Then Max := c;
  Edit4.Text := IntToStr(Max);
end;
```
- 4) Перевірте роботу програми для різних наборів чисел. Збережіть проект у папці Вправа 35.
- 5) Змініть програмний код так, щоб програма знаходила найменше з трьох чисел.
- 6) Додайте на форму кнопку Button2 і змініть її заголовок на «Упорядкувати a, b, c». Створіть процедуру обробки події OnClick для цієї кнопки, щоб значення змінних a, b і c були впорядковані за неспаданням (поміняйте місцями a, b і c так, щоб стало $a \leq b \leq c$).



Комп'ютерне тестування



Виконайте тестове завдання 35 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 36. Складання програм із розгалуженнями

Вкладені розгалуження

Під час розв'язування багатьох задач виникає необхідність при виконанні (або невиконанні) деякої умови знову робити вибір. У такій ситуації застосовують так звані **вкладені розгалуження**: в умовному операторі після службового слова Then (або Else) використовують оператор If.

» Записати умовний оператор для обчислення значення функції:

$$y = \begin{cases} x, & \text{якщо } x < 0; \\ x^2, & \text{якщо } 0 \leq x \leq 5; \\ 2x, & \text{якщо } x > 5; \end{cases}$$

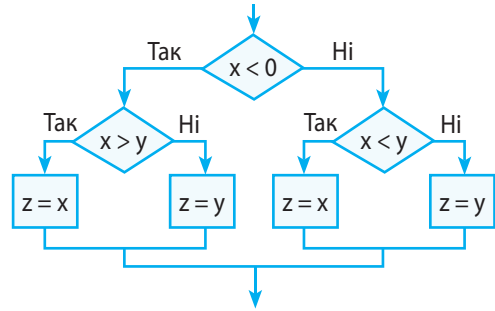
```
If x < 0 Then y := x
Else If x <= 5 Then y := x * x
Else y := 2 * x;
```

Під час запису програмного коду використовуйте структурований запис операторів, тобто пов'язані пари ключових слів або їх групи (наприклад, Then...Else, **begin...end**) пишуть на одній вертикалі. Це зробить текст програми більш наочним і зрозумілим.

» Обчислити:

$$Z = \begin{cases} \max(x, y) & \text{при } x < 0; \\ \min(x, y) & \text{при } x \geq 0. \end{cases}$$

```
If x < 0 Then
  If x > y Then z := x
  Else z := y
Else
  If x < y Then z := x
  Else z := y;
```



» Знайти корені квадратного рівняння вигляду $ax^2 + bx + c = 0$. Процедура обробки події для кнопки Знайти корені (рис. 36.1):

```
var a, b, c, x1, x2, D: Real;
{ введення коефіцієнтів a, b, c }
a := StrToFloat(Edit1.Text);
b := StrToFloat(Edit2.Text);
c := StrToFloat(Edit3.Text);
D := b*b - 4*a*c; // обчислення дискримінанта
Edit4.Text := FloatToStr(D); // виведення значення дискримінанта
If D > 0 Then begin
  x1 := (-b + Sqrt(D)) / (2*a); // обчислення і виведення значень коренів
  x2 := (-b - Sqrt(D)) / (2*a);
  Edit5.Text := FormatFloat('0.##', x1);
  Edit6.Text := FormatFloat('0.##', x2);
end
Else
  If D = 0 Then begin
    x1 := -b / (2*a);
    Edit5.Text := FormatFloat('0.##', x);
  end
  Else Edit5.Text := 'Коренів немає';
end;
```

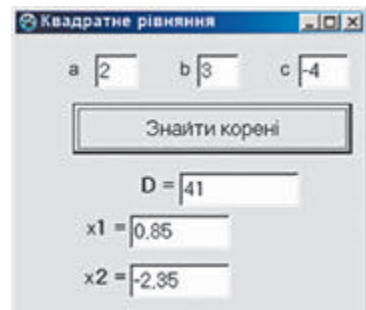


Рис. 36.1

Питання для самоперевірки



1. Що таке вкладені розгалуження?
2. У яких випадках у записі оператора If використовують операторні дужки **begin...end**?
3. Яких значень набудуть змінні A і B після виконання умовного оператора If для наведених наборів початкових значень?

Фрагмент програмного коду	Початкові значення A, B	A	B
If A < 0 Then	A = -3, B = 5		
If A > B Then A := B	A = 3, B = 5		
Else B := A	A = 8, B = 5		
Else If A > B Then B := A	A = -3, B = -5		
Else A := B;			

4. Запишіть умовний оператор, який виконує таку дію: вивести в текстове поле Edit1 кількість цифр у цілому числі A, що належить діапазону від 1 до 999.
5. Запишіть умовний оператор, який виконує таку дію: найбільше з трьох різних значень змінних дійсного типу X, Y і Z зменшити на 0,3.
6. Створіть програму, за допомогою якої визначається сторіччя, до якого належить зазначений рік у (у > 0). (Врахувати, що, наприклад, початком XX ст. був 1901, а не 1900 рік.)

Вправа 36



- Скласти програму «Вгадай число» за таким планом:

- 1) Комп'ютер загадує випадкове число від 1 до 100.
- 2) Користувач робить спробу вгадати число.
- 3) Комп'ютер порівнює введене число із загаданим і дає відповідь «Мало» або «Забагато». Якщо користувач відгадав число, виводиться напис «Ти вгадав число!».

- 1) Розмістіть на формі компоненти згідно з рис. 1*.

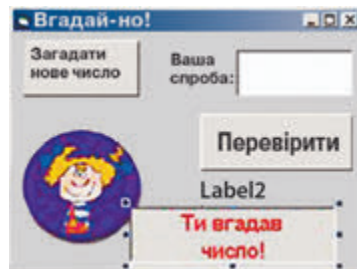


Рис. 1

* З тим, як додавати рисунки із зовнішніх файлів, ви ознайомитеся в § 47.

- 2) У вікні Інспектор об'єктів задайте значення властивостей компонентів згідно з таблицею:

Властивість	Значення	Властивість	Значення
Form1.Caption	Вгадай число	Button1.Caption	Загадати нове число
Label1.Caption	Ваша спроба:	Button1.WordWrap	True
Label3.Caption	Ти вгадав число!	Button2.Caption	Перевірити

- 3) Опишіть змінну b (загадане комп'ютером число) як глобальну:
var Form1: TForm1;
 b: Integer;

- 4) Створіть процедуру обробки події OnClick для кнопки Загадати нове число. У програмному коді процедури запишіть оператори:
begin

```

Randomize;
b := Random(100);
Label2.Visible := False;
Label3.Visible := False;
Edit1.Clear;
Edit1.SetFocus;

```

end;

- 5) Створіть процедуру обробки події OnClick для кнопки Перевірити (рис. 2, 3). У програмному коді процедури запишіть оператори:

```

var a: Integer;
begin
a := StrToInt(Edit1.Text); { a — число, яке вводиться користувачем }
{ Якщо число вгадано, елемент Label3 стає видимим, а елемент Label2 (для слів «Мало» і «Забагато») — невидимим }
If a = b Then begin
    Label3.Visible := True;
    Label2.Visible := False;
end
Else If a < b Then begin
    Label2.Visible := True;
    Label2.Caption := 'МАЛО';
end

```



Рис. 2

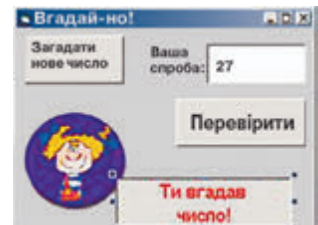


Рис. 3


```

Else begin
    Label2.Visible := True;
    Label2.Caption := 'ЗАБАГАТО';
end;
end;

```

- 6) Запустіть проект на виконання. Збережіть проект у папці Вправа 36.
- 7) Як потрібно діяти, щоб відгадати число якомога швидше?



Комп'ютерне тестування

Виконайте тестове завдання 36 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 37. Поліваріантне розгалуження

У багатьох випадках, коли доводиться використовувати декілька вказівок розгалуження, вкладених одна в одну, краще скористатися оператором вибору.

Оператор вибору Case

Оператор вибору Case дозволяє вибрати один із декількох операторів залежно від значення селектора.

Синтаксис оператора вибору Case:

```

Case <Селектор> of
    <список міток 1>: <оператор1>;
    ...
    <список міток N>: <операторN>;
end;

```

end;

Тут <Селектор> — це змінна або вираз, значення якої буде перевірятися; <список міток> — деякий набір можливих значень.



Селектор — змінна або вираз порядкового типу (цілого, символічного тощо).

Список міток можна задавати у вигляді окремого значення, переліку значень або у вигляді діапазону значень.

Виконання оператора Case починається з обчислення значення селектора. Далі виконується оператор, одна з міток якого збігається зі значенням селектора. Оператор може бути як простим, так і складеним. Після виконання цього оператора керування передається оператору, записаному після ключового слова **end**. Тобто може бути виконана тільки одна з гілок оператора вибору.

Гілки оператора Case будуть перебиратися послідовно, до першого збігу значення <Селектор> зі значенням однієї з міток. Якщо значення <Селектор> не дорівнює жодному зі значень міток, жодну з гілок виконано не буде.

» Визначити, яким є одноцифрове число x — простим чи складеним.

Case x of

1: ShowMessage ('Це — 1');

2, 3, 5, 7: ShowMessage ('Це — просте число');

4, 6, 8, 9: ShowMessage ('Це — складене число');

end;

Якщо є дії, які потрібно виконати у випадку, коли значення <Селектор> не збігається з жодним зі значень міток, то використовують оператор Case у повній формі:

Case <Селектор> of

<список міток 1>: <оператор 1>;

...

<список міток N-1>: <оператор N-1>;

Else <оператор N>;

end;

При використанні повної форми запису оператора у випадку, коли значення <Селектор> не збігається з жодним зі значень міток, буде виконано групу операторів, розташовану в гілці Else.

» Зазначити словесну оцінку залежно від кількості отриманих балів:

Case N of // N — кількість балів

4..6: ShowMessage ('Ваша оцінка — задовільно');

7..9: ShowMessage ('Ваша оцінка — добре');

10..12: ShowMessage ('Ваша оцінка — відмінно');

Else ShowMessage ('Виконайте тест ще раз');

end;

Питання для самоперевірки



1. Для розв'язування яких задач використовують оператор вибору?
2. Які вирази можна використовувати як селектор в операторі Case?
3. Як в операторі Case можна задати список міток?
4. Доповніть таблицю виконання оператора Case для різних початкових значень змінної A.

Фрагмент програмного коду	Початкове значення A	Значення A після виконання оператора Case
Case A mod 10 of 0..3: A := A * 2; 3..5: A := A + 10; 6..9: A := A * 10; end;	13	
	40	
	88	
	11	
	26	

5. Розгляньте фрагмент програми для визначення пори року за номером місяця. Подайте наведений фрагмент програми у вигляді команди вибору:

```

If (N = 12) or (N = 1) or (N = 2) Then S := 'зима'
  Else If (N >= 3) and (N <= 5) Then S := 'весна'
    Else If (N >= 6) and (N <= 8) Then S := 'літо'
      Else If (N >= 9) and (N <= 11) Then S := 'осінь'
        Else ShowMessage ('Такого номера місяця немає');
  
```

6. Напишіть програму, яка за номером дня тижня визначає, скільки цього дня у вашому класі уроків: K1 — кількість уроків з гуманітарних предметів, K2 — із природничо-математичних дисциплін.

Вправа 37



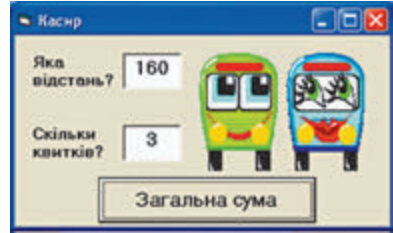
- Створити проект для розв'язання задачі.
Касир продає квитки на автобус, який курсує за маршрутом від міста A до міста B. Ціна квитка залежить від відстані, на яку потрібно їхати пасажиру:

$$X = \begin{cases} 20 \text{ грн, до } 50 \text{ км;} \\ 75 \text{ грн, від } 51 \text{ до } 100 \text{ км;} \\ 125 \text{ грн, від } 101 \text{ до } 150 \text{ км;} \\ 175 \text{ грн, від } 151 \text{ км.} \end{cases}$$

Скільки коштуватимуть N квитків до населеного пункту, відстань* до якого вводиться з клавіатури?

1) Створіть новий проект. Змініть заголовок форми на «Касир».

2) Можливий інтерфейс програми наведено на рисунку. Розмістіть на формі два компоненти Edit для введення значень відстані та кількості квитків і два компоненти Label. Змініть заголовки написів.



3) Додайте на форму кнопку Button1 і змініть заголовок доданої кнопки на «Загальна сума». Створіть процедуру обробки події OnClick для цієї кнопки:

procedure TForm1.Button1Click(Sender: TObject);

4) Опишіть необхідні змінні.

var X, V: Double; R, N: Integer; { R — відстань, X — ціна квитка, N — кількість квитків, V — загальна вартість квитків }

5) У програмному коді процедури Button1Click запишіть оператори:

```
begin
  R := StrToInt(Edit1.Text);
  N := StrToInt(Edit2.Text);
  Case R of
    1..50: X := 20;
    51..100: X := 75;
    101..150: X := 125;
    Else X := 175;
  end;
  V := X*N;
  ShowMessage ('Вартість квитків ' + FloatToStr(V) +' грн');
end;
```

6) Запустіть проект на виконання. Перевірте роботу програми для різних значень відстані й кількості квитків. Збережіть проект у папці Вправа 37.

Комп'ютерне тестування

Виконайте тестове завдання 37 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

* Значення відстані є цілим числом, яке більше або дорівнює 1.

§ 38. Компоненти вибору

Під час роботи з різними програмами користувачеві часто доводиться робити вибір певних елементів або налаштувань із запропонованих у діалоговому вікні за допомогою таких елементів керування, як перемикач і прапорець (рис. 38.1).



Рис. 38.1

Компонент перемикач (RadioButton)

Перемикачі використовують у ситуаціях, коли потрібно вибрати один із декількох наявних варіантів. Перемикачі завжди розміщуються на формі групами, в яких потрібно вибрати тільки один перемикач. При виборі іншого перемикача попередній вимикається.

Компонент RadioButton (вкладка Standard на Палітрі компонентів) складається з круглого віконця і текстового рядка. У віконці з'являється позначка вибору даного варіанта, а текстовий рядок пояснює його призначення. За зміст текстового рядка відповідає властивість Caption. Основна властивість елемента RadioButton — Checked (*Вибрано*) логічного типу (Boolean) може набувати значень True або False. У вибраного елемента властивість Checked дорівнює True.

Щоб перемикачі були зручно розташовані на формі, їх потрібно вирівняти.

Для **вирівнювання перемикачів** треба:

- 1) вибрати всі перемикачі групи, клацнувши їх по черзі, утримуючи натиснутою клавішу Shift;
- 2) клацнути по групі правою кнопкою миші для виклику контекстного меню;
- 3) вибрати команду Вирівняти;
- 4) у діалоговому вікні Вирівнювання (рис. 38.2), що відкриється, вибрати потрібні параметри вирівнювання, наприклад горизонтальне вирівнювання — Ліві сторони, вертикальне вирівнювання — Вирівняти по висоті. Об'єкти будуть розташовані відповідно до вибраних параметрів.

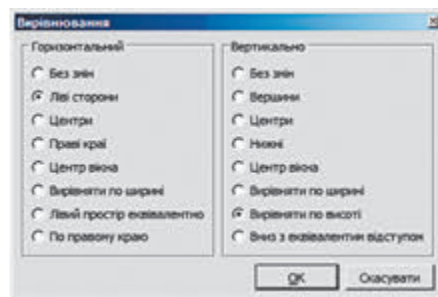


Рис. 38.2

» Запрограмувати ввімкнення перемикача
 RadioButton2 (рис. 38.3).

```
RadioButton1.Checked = False;
RadioButton2.Checked = True;
RadioButton3.Checked = False;
```

Для зняття вибору в програмі потрібно записати оператор:
 RadioButton2.Checked := False; // тепер жоден із компонентів не вибраний.

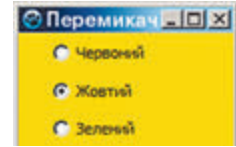


Рис. 38.3

Основною подією компонента RadioButton є OnClick. Можна **створити спільний обробник події OnClick** для всіх перемикачів на формі (в групі):

- 1) вибрати групу перемикачів;
- 2) у вікні Інспектор об'єктів на вкладці Події двічі клацнути в рядку OnClick.

Після цього процедура TForm1.RadioButton1Click буде виконуватися для всіх перемикачів, які були вибрані в момент її створення.

```
» procedure TForm1.RadioButton1Click(Sender: TObject);
var V: Integer;
begin
  with (Sender as TRadioButton) do
    V := StrToInt(copy(Name,12,1)); // Для RadioButton1 V = 1,
                                   // для RadioButton2 V = 2 і т. д.
  case V of
    1: Form1.Color := clRed;
    2: Form1.Color := clYellow;
    3: Form1.Color := clGreen;
  end;
end;
```

Оператор with (Sender as TRadioButton) означає, що властивість Name, яка в ньому використовується, є властивістю саме того з елементів керування RadioButton, із яким відбулася подія.

Компонент прапорець (CheckBox)

Елемент керування CheckBox використовується в програмі для візуалізації станів увімкнено/вимкнено, коли серед незалежних параметрів або характеристик користувачеві необхідно вибрати потрібні.

Компонент CheckBox (вкладка Standard на Палітрі компонентів) є поєднанням квадратного віконця, яке і показує стан компонента (вибраний/невибраний), і текстового рядка (властивість Caption) — заголовка, що пояснює призначення прапорця.

Основною властивістю компонента `CheckBox` є властивість `Checked` логічного типу (`Boolean`). Клацання компонента `CheckBox` змінює значення властивості `Checked` на протилежне. Зміна стану перемикача доступна і в програмі:

```
CheckBox1.Checked = False; // зняти позначку
CheckBox2.Checked = True; // встановити позначку
```

Під час кожної зміни стану прапорця, тобто при його встановленні або скиданні, для даного елемента керування відбувається подія `OnClick`. З'ясувати, встановлений прапорець чи скинутий, можна за значенням властивості `Checked`.

Після подвійного клацання компонента `CheckBox` на формі в режимі розробки створюється процедура обробки події:

```
procedure TForm1.CheckBox1Click(Sender: TObject);
```

У коді процедури перевіряється стан прапорця й активізуються параметри, пов'язані з прапорцем:

```
If CheckBox1.Checked Then
    <активізація параметрів, які пов'язані з прапорцем>
Else
    <відключення параметрів, які пов'язані з прапорцем>;
```

Питання для самоперевірки



1. У яких випадках доцільно використовувати компонент `RadioButton`?
2. Яких значень може набувати властивість `Checked` компонента `RadioButton`?
3. Яка подія є основною для компонента `RadioButton`?
4. Як створити спільний обробник події `OnClick` для групи перемикачів?
5. З якою метою в програмі може бути використаний компонент `CheckBox`?
6. Як організувати перевірку значення властивості `Checked` при зміні стану прапорця?

Вправа 38



- Скласти програму, в якій реалізовано головоломку Лойда: із заданого набору чисел вибрати ті, сума яких дорівнює 50.
- 1) Створіть новий проект. Змініть заголовок форми на «Головоломка Лойда».

- 2) Додайте на форму десять компонентів CheckBox згідно з рисунком. Змініть заголовки компонентів CheckBox на числа з набору, запропонованого С. Лойдом: 25, 27, 3, 12, 6, 15, 9, 30, 21, 19.

Додайте на форму два компоненти Label. Змініть заголовок напису Label1 на «Сума чисел:». Напис Label2 призначений для виведення суми чисел, які відповідають вибраним прапорцям.

- 3) Опишіть змінну S як глобальну:

```
var Form1: TForm1;
```

```
    S: Integer;
```

- 4) Створіть процедуру обробки події OnCreate для форми.

Змінна S набуває значення 0 при завантаженні форми:

```
procedure TForm1.FormCreate(Sender: TObject);
```

```
begin
```

```
    S := 0;
```

```
end;
```

- 5) Створіть спільний для всіх прапорців обробник події OnClick. У програмному коді реалізується такий алгоритм: якщо прапорець установлений, відповідне число додається до суми S, якщо прапорець скинутий, відповідне число віднімається від суми S. Після цього значення суми виводиться в напис Label2.

```
procedure TForm1.CheckBox1Click(Sender: TObject);
```

```
begin
```

```
    with (Sender as TCheckBox) Do begin
```

```
        If Checked Then
```

```
            S := S + StrToInt(Caption)
```

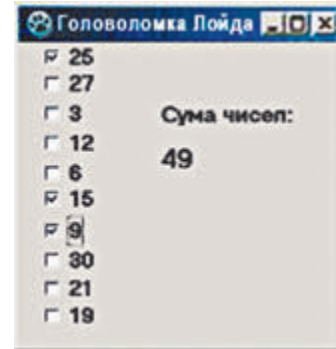
```
        Else S := S - StrToInt(Caption);
```

```
    end;
```

```
    Label2.Caption := IntToStr(S);
```

```
end;
```

- 6) Запустіть проект на виконання. Перевірте роботу програми. Розв'яжіть головоломку Лойда. Збережіть проект у папці Вправа 38.



Комп'ютерне тестування

Виконайте тестове завдання 38 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 39. Алгоритми з повтореннями. Цикл із передумовою



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

Для запису алгоритмів із повторенням (циклів) мовою Object Pascal використовують *три види операторів циклу*: з параметром, з передумовою і з післяумовою.



Серію команд, що повторюється під час виконання циклу, називають **тілом циклу**.

Кожне виконання тіла циклу називають **ітерацією**.

Оператор циклу з передумовою While

Оператор повторення While (*цикл «Поки»*) призначений для організації повторного виконання серії команд, поки залишається істинною умова виконання циклу.

Синтаксис оператора:

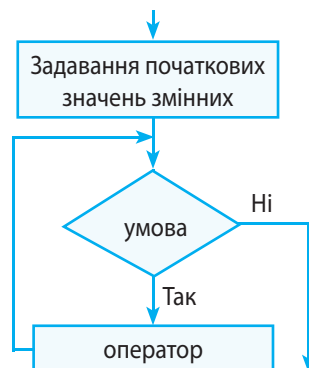
```
While <умова> do <оператор>;
```

Тут <умова> — логічний вираз, що є умовою виконання циклу;

<оператор> — простий або складений оператор, який виконується при кожній ітерації.

Виконання оператора циклу While починається з обчислення значення логічного виразу — умови повторення циклу. Якщо умова істинна, то виконуються оператори тіла циклу і керування повертається на перевірку умови. Якщо ж умова хибна, то виконується оператор, який є наступним після оператора While.

Якщо при першій перевірці умова виявиться хибною, тіло циклу While не виконається жодного разу.



» Для різних початкових значень змінної X визначити значення цієї змінної після виконання циклу та кількість повторень:

```
While X <= 10 do X := X + 1;
```

Початкове значення X	Значення X після виконання циклу	Кількість повторень
7	11	4
10	11	1
15	15	0

» Наведений фрагмент програми ілюструє нескінченний цикл.

```
Num := 0;
While Num < 20 do
    Label1.Caption := IntToStr(Num)
```

У тілі циклу змінна Num має постійне значення, тому умова Num < 20 (0 < 20) завжди правильна і поданий цикл є нескінченним.

! Для переривання нескінченного циклу треба натиснути сполучення клавіш Ctrl + Break.

Щоб побачити зміну значень у ході виконання циклу, в тіло циклу включають виклик методу Application.ProcessMessages.

» Розробити програму-таймер, що буде послідовно виводити числа від 1 до 100 з інтервалом 0,1 с.

Нехай змінна A містить поточне значення числа, що виводиться. Поки A менше за 100, збільшувати значення A на одиницю і виводити значення змінної у Label (рис. 39.1). Для призупинення роботи програми використати процедуру Sleep.

```
var A: Integer;
begin
    A := 1; // початкове значення A
    While A < 100 do
        begin
            A := A + 1;
            Label1.Caption := IntToStr(A);
            Application.ProcessMessages; // опрацювання повідомлень
            Sleep(100); // зупинка на 0,1 с
        end;
end;
```

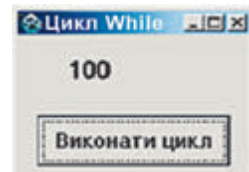


Рис. 39.1

Алгоритм обчислення суми n членів числової послідовності

Розглянемо алгоритм обчислення n -го члена деякої послідовності чисел і суми n її членів:

- 1) задати значення A_1 — першого члена послідовності, n — кількості членів, які потрібно обчислити;
- 2) задати початкові значення лічильника членів послідовності ($i := 0$) і суми ($S := 0$);
- 3) поки номер i -го члена послідовності, який обчислено, не досягне значення n — заданої кількості членів, повторювати дії:
 - номер поточного доданка збільшити на 1;
 - обчислити значення наступного доданка A ;
 - обчислене значення A додати до суми S .

» Знайти суму 20 елементів послідовності чисел 2,5; 3,0; 3,5; 4,0...

{ Задаються початкові значення }

$S := 0$; // суми S

$i := 0$; // лічильника доданків

$A := 2.5$; // першого доданка A

While $i < 20$ do **begin**

$S := S + A$; // A додається до суми S

$A := A + 0.5$; // обчислюється наступне значення доданка A

$i := i + 1$

end;

» Знайти суму всіх елементів послідовності

$1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{5}, \dots$, значення яких не менше за

0,01 (рис. 39.2).

var i : Integer; A, S : Double;

begin

{ Задаються початкові значення }

$S := 0$; // суми S

$A := 1$; // першого доданка A

$i := 0$; // лічильника доданків

While $A > 0.01$ do **begin**

$S := S + A$;

$i := i + 1$;

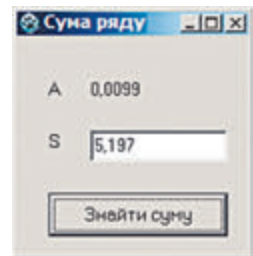


Рис. 39.2

```

A := 1 / i; // обчислення поточного доданка A
Label2.Caption := FormatFloat('0.####', A);
end;
Edit1.Text := FormatFloat('0.####', S);
end;

```

Питання для самоперевірки



1. Дайте означення циклу як алгоритмічної структури.
2. Поясніть синтаксис і правила виконання оператора циклу з передумовою.
3. В якому випадку тіло циклу While не виконається жодного разу?
4. Визначте значення змінної S після виконання операторів:
 - a) $S := 0; i := 0; \text{While } i < 5 \text{ do } i := i + 1; S := S + i;$
 - б) $S := 0; i := 5; \text{While } i > 1 \text{ do } \mathbf{begin} S := S + i; i := i - 1; \mathbf{end};$
5. Складіть програму для знаходження суми чисел, кратних 3, які розташовані в інтервалі (100; 300).
6. Складіть програму для знаходження суми десяти елементів послідовності 1,0; 1,2; 1,4; 1,6...

Вправа 39



- Розробити проект «Числа Фібоначчі», призначений для обчислення n -го числа Фібоначчі.

Числами Фібоначчі називають числа, які знаходять за таким правилом:

$$F_1 = F_2 = 1; F_n = F_{n-1} + F_{n-2}.$$

$F_1 = F_2 = 1$, тому обчислення починається з $n = 3$.

Для обчислення n -го числа Фібоначчі виконується така послідовність дій (рис. 1):

- 1) виділяються змінні A і B для зберігання двох поточних чисел Фібоначчі F_{n-1} і F_{n-2} ;
 - 2) сума чисел $A + B$ заноситься в змінну C ;
 - 3) на наступній ітерації циклу:
 - $B = F_{n-1}$ стає $(n-2)$ -м членом ряду, тому $A := B$;
 - $C = F_n$ стає $(n-1)$ -м членом, тому $B := C$.
- 1) Створіть новий проект.

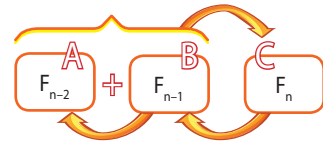


Рис. 1

2) Розробіть інтерфейс програми згідно з рис. 2.

3) Опишіть змінні, необхідні для реалізації алгоритмів виконання завдань:

- n — змінна цілого типу для збереження номера шуканого елемента послідовності чисел Фібоначчі;
- i — змінна цілого типу для збереження номера поточного елемента послідовності;
- A, B, C — змінні цілого типу для збереження $(i-2)$ -го, $(i-1)$ -го, i -го елементів послідовності при кожній ітерації циклу.

4) Створіть процедуру обробки події OnClick для кнопки Обчислити число з номером n . У програмному коді процедури Button1Click запишіть оператори для обчислення n перших чисел Фібоначчі:

$A := 1; B := 1; i := 2;$

While $i < n$ do begin

$i := i + 1;$

$C := A + B;$

$A := B; B := C;$

 Application.ProcessMessages; // опрацювання повідомлень

 Sleep(500); // зупинка на 0,5 с

 Edit2.Text := IntToStr (C);

end;

5) Запустіть проект на виконання. Перевірте роботу програми. Збережіть проект у папці Вправа 39.

6) Додайте на форму компонент Edit. Доповніть програмний код операторами для обчислення суми n перших чисел Фібоначчі і виведення результату до створеного текстового поля (Edit3). Перевірте дію кнопки.



Рис. 2



Комп'ютерне тестування

Виконайте тестове завдання 39 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 40. Алгоритми з повтореннями. Цикл із післяумовою

Оператор циклу з післяумовою REPEAT

Оператор циклу Repeat використовують у тих випадках, коли потрібно спочатку виконати деяку послідовність дій, а потім визначити, чи є потреба повторювати ці дії ще раз.

Синтаксис оператора Repeat:

```
Repeat
<оператор>
Until <умова>;
```

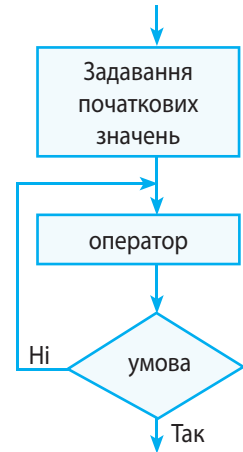
Тут <умова> — логічний вираз, що є умовою виходу з циклу;

<оператор> — оператор, який виконується при кожній ітерації (тіло циклу).

Порядок виконання циклу Repeat:

- 1) виконується <оператор> — вказівки, які входять у тіло циклу;
- 2) перевіряється <умова>. Якщо умова хибна, то оператори тіла циклу повторюються, якщо ж умова істинна — цикл Repeat припиняє свою роботу і виконується наступний оператор програми.

! Тіло циклу — блок операторів між Repeat і Until — обов'язково виконується хоча б один раз.



➤ Вивести парні числа від 2 до 12 (рис. 40.1).

```
var k: Integer;
begin
  k := 0;
  Repeat
    k := k + 2;
    Label1.Caption := IntToStr(k);
  Until k >= 12;
end;
```

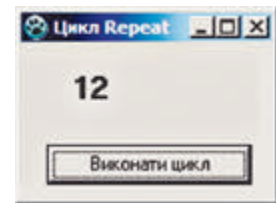


Рис. 40.1

» Визначити, скільки перших натуральних чисел потрібно додати, щоб одержати суму, більшу за 1000 (рис. 40.2).

```
var n, S: Integer;
begin
    n := 0; S := 0;
    Repeat
        n := n + 1; S := S + n;
    Until S > 1000;
    Edit1.Text := IntToStr(n);
end;
```

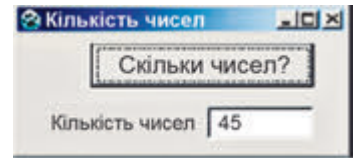


Рис. 40.2

» Знайти суму цифр натурального числа n (рис. 40.3).

```
var n, d, S: Integer;
begin
    n := StrToInt(Edit1.Text);
    S := 0;
    Repeat
        d := n mod 10; { знайдено останню цифру числа }
        S := S + d; // цифра d додається до суми
        n := n div 10; // відкидається остання цифра числа n
    Until n=0; { Якщо n=0, цикл припиняє роботу, оскільки всі цифри числа розглянуто }
    Edit2.Text := IntToStr(S);
end;
```

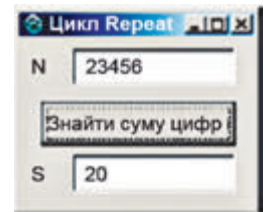


Рис. 40.3

» Полічити кількість слів у введеному користувачем тексті (рис. 40.4). Вважатимемо, що слова відокремлюються виключно одним пробілом, а перший і останній символи текста (рядка S) не є пробілами. Отже, кількість слів у рядку S на 1 більша за кількість пробілів. Для введення текстового рядка використовується компонент Мемо

(вкладка Standard на Палітрі компонентів): він дозволяє вводити багаторядковий текст із клавіатури як значення властивості Text.

```
var S: String;
    i, count: Integer;
```

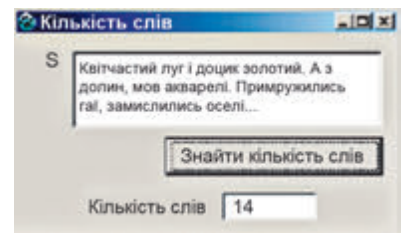


Рис. 40.4

begin

```
S := Memo1.Text;
i := 0;
If length(S) > 0 Then begin // Якщо довжина рядка S не нульова,
    count := 1; // є хоч одне слово
    Repeat
        i := i + 1; // Перевіряється кожен символ рядка S
        If S[i] = ' ' // Якщо знайдено пропуск,
            Then count := count + 1; // то збільшується лічильник слів на 1
    Until i >= length(S);
end
Else count := 0;
Edit2.Text := IntToStr(count);
end;
```

Питання для самоперевірки



1. Поясніть, як виконується оператор циклу з післяумовою.
2. Чи може виникнути ситуація, за якої тіло циклу Repeat не виконається жодного разу?
3. Дано фрагмент програмного коду. Доповніть таблицю.

Фрагмент програмного коду	Початкове значення X	Значення X після виконання циклу	Скільки повторень відбулося
Repeat X := X + 1 Until X > 10;	7		
	10		
	11		

4. Якого значення набуде змінна F після виконання циклу?
i := 1; F := 2;
Repeat
 i := i + 1; F := F * i
Until i < 6;
5. Поясніть призначення оператора циклу:
Repeat
 x := StrToFloat(InputBox('Введення X', 'Введіть додатне число', '0'));
Until x > 0;
6. Складіть програму для знаходження найменшої кількості перших парних чисел, потрібної для того, щоб одержати суму, більшу від 100.

Вправа 40



- Організувати введення послідовності натуральних чисел, яка завершується 0. Знайти кількість і середнє арифметичне введених чисел.

- 1) Створіть новий проект.
- 2) Розробіть інтерфейс програми згідно з рисунком.
- 3) Створіть процедуру обробки події OnClick для кнопки Ввести числа.
- 4) У програмному кодї процедури Button1Click запишіть оператори:

begin

 K := 0; S := 0;

 Repeat

 { введення чисел за допомогою функції InputBox }

 A := StrToInt(InputBox('Введення чисел', 'A=>', '0'));

 K := K + 1; // лічильник чисел, що вводяться

 S := S + A; // додавання введеного числа до суми

 Until A = 0; // цикл припиняє роботу, якщо A = 0 — вводиться 0
 Edit1.Text := IntToStr(K-1); { виведення кількості чисел без урахування останнього введеного 0 }

 Sr := S / (K-1); // обчислення середнього арифметичного

 Edit2.Text := FloatToStr(Sr);

end;

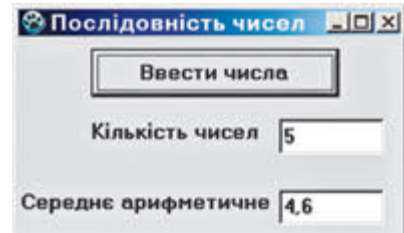
- 5) Запустіть проект на виконання. Перевірте роботу програми для різних наборів чисел. Збережіть проект у папці Вправа 40.



- 6) Змініть програмний код так, щоб програма визначала порядковий номер n найменшого з уведених чисел.

Алгоритм знаходження найменшого (мінімального) з уведених чисел та його номера:

- 1) змінній Min присвоїти значення першого з уведених чисел, а змінній n — значення 1;
- 2) кожне наступне число порівняти з Min. Якщо це число менше за Min ($A < \text{Min}$), то змінній Min присвоїти значення цього числа ($\text{Min} := A$), а змінній n — його порядковий номер.



Після введення $A = 0$ цикл припиняє роботу, Min містить мінімальне число з уведеної послідовності, а змінна n — порядковий номер цього числа в послідовності.

Комп'ютерне тестування



Виконайте тестове завдання 40 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 41. Алгоритми з повтореннями. Цикл із параметром

Оператор циклу з параметром For

Цикл із параметром (лічильником) використовують у випадках, коли відома кількість повторень циклу. Оператор For забезпечує повторення команд, кероване параметром циклу.

Синтаксис оператора For:

For <параметр> := NV to KV do <оператор>;

Тут <параметр> — змінна порядкового типу;

NV і KV — константи, змінні або вирази того самого типу, що й параметр;

<оператор> — простий або складений оператор (тіло циклу).

Порядок виконання циклу:

- 1) параметр отримує значення виразу NV;
- 2) поки параметр не перевищує значення KV, виконується <оператор> і значення параметра збільшується на одиницю відповідного типу.

» Повторити п'ять разів виведення вікна повідомлення:

```
For i := 1 to 5 do ShowMessage('i = '+IntToStr(i));
```

Якщо кінцеве значення параметра циклу менше від початкового ($KV < NV$), то цикл не виконається жодного разу.

» Проаналізувати, чому цикл не виконується жодного разу.

```
For A := 10 to 0 do S = S + A;
```

Якщо при використанні форми To кінцеве значення параметра циклу менше за початкове, то цикл не виконається жодного разу.

Для того щоб реалізувати цикл зі зміною параметра від більшого значення до меншого, використовують форму циклу For зі словом DownTo. Значення параметра з кожною ітерацією зменшується на одиницю:

```
For <параметр> := NV downto KV do <оператор>;
```

У такому випадку параметр щоразу отримує не наступне, а попереднє значення свого типу.

» Знайти значення виразу A^n .

У класичній мові Pascal відсутня стандартна функція піднесення до степеня. Щоб обчислити $St = A^n$, скористаємося означенням степеня: $St = A \cdot A \cdot \dots \cdot A$, де кількість множників дорівнює n .

```
St := 1;
```

```
For i := 1 to N do St := St * A;
```

» Реалізувати зворотний відлік від 10 до 0 у компоненті Label.

Щоб значення лічильника змінювалось від більшого значення до меншого, скористаємось формою циклу For зі словом downto:

```
For A := 10 downto 0 do begin
    Label1.Caption := IntToStr(A);
    Application.ProcessMessages;
    Sleep(1000)
end;
```

» Обчислити факторіал числа n (рис. 41.1) за формулою

$$n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot n.$$

```
var i, n, f: Integer;
begin
    n := StrToInt (Edit1.Text);
    f := 1;
    For i := 2 to n do f := f * i;
    Edit2.Text := IntToStr(f);
end;
```

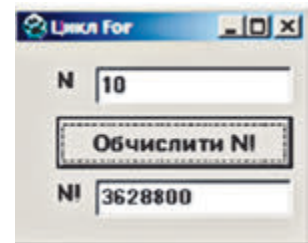



Рис. 41.1

» З клавіатури вводяться п'ять дійсних чисел, серед яких можуть бути як додатні, так і від'ємні числа. Знайти середнє арифметичне додатних чисел.

```
var i, k: Integer; a, Sr: Double;
begin
  Sr := 0;
  k := 0; // лічильник додатних чисел
  For i := 1 to 5 do begin
    a := StrToFloat(InputBox('Ввести 5 чисел', IntToStr(i), '0'));
    If a > 0 Then begin
      k := k + 1; Sr := Sr + a;
    end;
  end;
  Sr := Sr / k;
  Edit1.Text := FloatToStr(Sr);
end;
```

Компонент список (ListBox)

Для виведення значень змінних у кілька рядків зручно скористатися компонентом ListBox (*Список*).

Компонент ListBox  (вкладка Standard на Палітрі компонентів) призначений для зберігання й опрацювання текстових даних. Кожен рядок ListBox зберігає рядок символів (String).

Доступ до рядків надає властивість Items цього компонента.

Для додавання рядка до списку програмним шляхом призначено метод Add.

Синтаксис виклику методу Add:

```
ListBox1.Items.Add('Новий_рядок');
```

» Додати до списку ListBox значення виразу $2x$:

```
ListBox1.Items.Add(IntToStr (2 * x));
```

Щоб очистити вміст списку, для елемента керування ListBox ви-кликають метод Clear:

```
ListBox1.Clear;
```

Властивість Count зберігає кількість рядків, доданих до списку.

» Додати до списку ListBox літери із заданого діапазону (рис. 41.2):

```
var i, nv, kv: Char;
```

```
begin
```

```
  nv := Edit1.Text[1];
```

```
  kv := Edit2.Text[1];
```

```
  For i := nv to kv do ListBox1.Items.Add (i);
```

```
end;
```

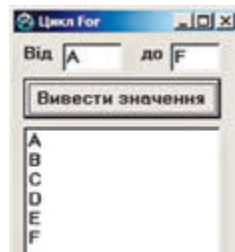


Рис. 41.2

Питання для самоперевірки

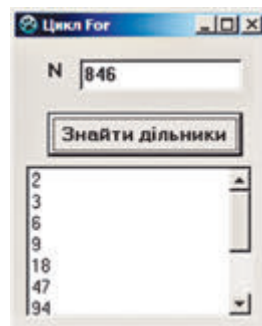


1. Поясніть структуру і правила виконання оператора циклу з параметром. Якого типу може бути параметр циклу?
2. Як змінюється параметр циклу? З яким кроком? У якому діапазоні? Скільки разів виконується тіло циклу?
3. В якому випадку застосовується форма циклу For зі словом downto?
4. Які переваги дає використання елемента керування ListBox при виведенні значень виразів (змінних)?
5. Чому дорівнює значення змінної S після виконання циклу:
 - а) $S := 0$; For $i := 5$ to 7 do $S := S + 1$;
 - б) $S := 0$; For $i := 10$ downto 6 do $S := S + 1$;
 - в) $S := 0$; For $i := 'A'$ to $'D'$ do $S := S + 1$;
6. Знайдіть помилки в наведених фрагментах програми:
 - а) $Y := 0$; For $X := 0.1$ to 0.9 do $Y := Y + \text{Sin}(X)$;
 - б) $K := 81$; $Y := 1$; For $i := 1$ to $\text{Sqrt}(K)$ do $Y := Y * 2$;
 - в) $K := 0$; For $i := 9$ to 1 do $K := K * 2$;

Вправа 41



- » Визначити всі дільники числа n (крім 1 і самого числа). Якщо дільники відсутні, повідомити, що n — просте число.
- 1) Створіть новий проект. Змініть заголовок форми на «Цикл For». Розмістіть на формі компонент Edit для введення числа n згідно з рисунком.
 - 2) Додайте на форму компонент ListBox згідно з рисунком.
 - 3) Додайте на форму кнопку Button1. Змініть заголовок доданої кнопки на «Знайти дільники».



- 4) Перевірте, чи є число n простим:
- переберіть всі числа від 2 до $(n \div 2)$. Якщо число i є дільником числа n , то i додається до списку `ListBox`;
 - якщо після завершення перебору можливих дільників числа елементів списку залишилось рівним 0, то повідомити, що n — просте число.
- 5) Створіть процедуру обробки події `OnClick` для кнопки Знайти дільники. У програмному коді цієї процедури запишіть оператори, що реалізують алгоритм, зазначений у п. 4:

```
var i, n: Integer;
begin
  ListBox1.Clear;           // очищення списку
  n := StrToInt(Edit1.Text); // введення числа n
  For i := 2 to n div 2 do
    If n mod i = 0 Then
      ListBox1.Items.Add(IntToStr (i));
  If ListBox1.Count = 0
    Then ListBox1.Items.Add(IntToStr (n) + ' — просте число');
end;
```

- 6) Запустіть проект на виконання. Перевірте роботу програми для різних значень n . Збережіть проект у папці Вправа 41.

Комп'ютерне тестування

Виконайте тестове завдання 41 з автоматичною перевіркою на сайті interactive.ranok.com.ua.

§ 42. Складання циклічних алгоритмів опрацювання величин



Цикл називають **вкладеним**, якщо він міститься в тілі іншого циклу.

Такий цикл також називають *внутрішнім*, а цикл, у якому він міститься, — *зовнішнім*.

Вкладені цикли організовані таким чином: внутрішній цикл повністю вміщується в тілі зовнішнього циклу (рис. 42.1).

- А — зовнішній цикл;
- В — внутрішній цикл.

Внутрішній і зовнішній цикли можуть бути циклами з параметром, з передумовою або з післяумовою.

Принцип роботи вкладених циклів такий: при першій ітерації зовнішній цикл викликає внутрішній, який виконується до свого завершення, після чого керування передається в тіло зовнішнього циклу. При другій ітерації зовнішній цикл знову викликає внутрішній. І так доти, поки не завершиться зовнішній цикл.



Рис. 42.1

» Надрукувати числа у вигляді таблиці:

```
5 5 5 5 5
5 5 5 5 5
5 5 5 5 5
```

Рядок із п'яти цифр можна сформувавши за допомогою циклу For:
`S := ' ';`

`For i := 1 to 5 do S := S + '5 ';` // S — рядок

Щоб повторити виведення даного рядка тричі, вставляємо цей цикл всередину іншого:

`For k := 1 to 3 do` // тричі виконується цикл із параметром k

begin

`S := ' ';`

`For i := 1 to 5 do S := S + '5 ';` // 5 разів виконується цикл із параметром i
`ListBox1.Items.Add(S);`

end;

! Параметрами вкладених циклів For повинні бути різні змінні.

» Надрукувати таблицю множення (рис. 42.2).

`var N, K: Integer;`

begin

`For N := 2 to 9 do begin`

`For K := 1 to 9 do begin`

`ListBox1.Items.Add(IntToStr(N) + ' x ' + IntToStr(K) + ' = ' + IntToStr(N * K));`

end;

`ListBox1.Items.Add('-----');`

end;

end;

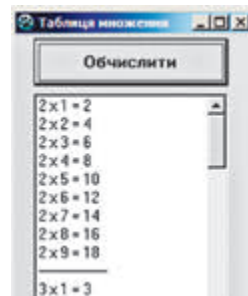


Рис. 42.2



Знайти всі трицифрові натуральні числа, сума цифр яких дорівнює їхньому добутку, та визначити кількість таких чисел (рис. 42.3).

Для розв'язання задачі можна перебрати всі можливі сполучення цифр, з яких утворюється десятковий запис трицифрового числа, і перевіряти для кожного сполучення умову задачі.

var A, B, C, K: Integer;

begin

 K := 0;

 For A := 1 to 9 do // перша цифра

 For B := 0 to 9 do // друга цифра

 For C := 0 to 9 do // третя цифра

 If A + B + C = A*B*C Then

begin

 K := K + 1;

 ListBox1.Items.Add(IntToStr(100 * A + 10 * B + C));

end;

 ListBox1.Items.Add('K=' + IntToStr (K));

end;

У задачі використано вкладені цикли. Зверніть увагу, що лише тіло внутрішнього циклу з параметром C взято в операторні дужки **begin..end**, оскільки складається з двох операторів. Для циклів вищих рівнів у цьому немає потреби, оскільки в кожному з них тіло циклу містить лише один оператор: цикл із параметром C є тілом циклу з параметром B, а той, у свою чергу, є тілом циклу з параметром A.

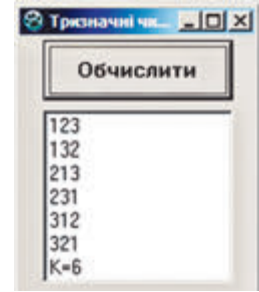


Рис. 42.3

Питання для самоперевірки



1. Які види циклів ви знаєте? У яких випадках застосовують кожний із цих циклів?
2. У чому полягає правило вкладення циклів?
3. Проаналізуйте циклічну конструкцію:

 For i := 1 to 2 do

 For j := 1 to 3 do

 For k := 1 to 3 do

 ListBox1.Items.Add(IntToStr(i) + IntToStr(j) + IntToStr(k));

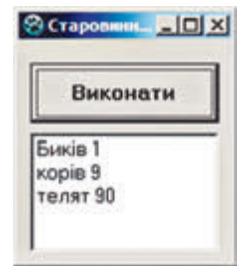
- а) Назвіть тіло кожного циклу.
 б) Скільки разів буде виконаний кожний цикл?
 в) Який результат буде виведений після виконання програми?
4. Який рядок S буде сформовано в результаті виконання фрагмента програми?
- ```
S := ' ';
For i := 0 to 1 do
 For j := 0 to 1 do
 S := S + IntToStr(i) + IntToStr(j);
```
5. Складіть програму для обчислення значення виразу  $2k + n$  при всіх значеннях змінних  $n = 1, 2, 3$  і  $k = 2, 4, 6, 8$ .
6. Складіть програму для обчислення  $F = x + x^2 + x^3 + \dots + x^n$  для  $x = 0,1; 0,2; \dots; 1,0$  і введеного з клавіатури значення  $n$ .

### Вправа 42



- Скласти програму для розв'язання старовинної задачі.  
*Плата за одного бика — 20 карбованців, за корову — 10 карбованців, за теля — 1 карбованець. Скільки можна купити биків, корів і телят, якщо на 200 карбованців треба купити 100 голів худоби?*

- 1) Створіть новий проект. Змініть заголовок форми на «Старовинна задача».
- 2) Розмістіть на формі компонент ListBox згідно з рисунком.
- 3) Додайте на форму кнопку Button1. Змініть заголовок доданої кнопки на «Виконати».
- 4) Розв'язання задачі.



Позначимо літерою  $b$  кількість биків;  $k$  — кількість корів;  $t$  — кількість телят. Загальна кількість голів дорівнює 100:  $b + k + t = 100$ .

За биків заплатили  $20b$  карбованців, за корів —  $10k$  карбованців, за телят —  $t$  карбованців, отже,  $20b + 10k + t = 200$ . На 200 карбованців можна купити: не більше 10 биків, тобто  $0 \leq b \leq 10$ ; не більше 20 корів, тобто  $0 \leq k \leq 20$ ; не більше 200 телят, тобто  $0 \leq t \leq 200$ . Таким чином, необхідно перебрати всі можливі значення змінних  $b, k, t$  і вивести в поле ListBox той набір значень, для яких виконується умова  $(20 \cdot b + 10 \cdot k + t = 200)$  and  $(b + k + t = 100)$ .

- 5) Створіть процедуру обробки події OnClick для кнопки Старовинна задача. У програмному коді створеної процедури запишіть такі оператори:

```
var b, k, t: Integer;
begin
 For b := 0 to 10 do
 For k := 0 to 20 do
 For t := 0 to 200 do
 If (20*b + 10*k + t = 200) and (b + k + t = 100)
 Then begin
 ListBox1.Items.Add('Биків ' + IntToStr(b));
 ListBox1.Items.Add('корів ' + IntToStr(k));
 ListBox1.Items.Add('телят ' + IntToStr(t));
 end;
 end;
 end;
 end;
```

- 6) Запустіть проект на виконання. Перевірте роботу програми. Збережіть проект у папці Вправа 42.

### Комп'ютерне тестування



Виконайте тестове завдання 42 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).

## Практична робота 15



### Складання та виконання алгоритмів із повтореннями та розгалуженнями для опрацювання величин

**Завдання:** скласти програму для обчислення найбільшого спільного дільника (НСД) чисел  $A$  і  $B$  за алгоритмом Евкліда.

**Обладнання:** ПК із встановленим середовищем програмування Lazarus.

#### Хід роботи

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

#### 1. Розміщення елементів керування на формі

- ▶ 1. Створіть новий проект.
- ▶ 2. Розробіть інтерфейс програми згідно з рис. 1.

## II. Написання процедури обробки подій

- ▶ 3. Створіть процедуру обробки події OnClick для кнопки Обчислити NSD, за допомогою якої обчислюється НСД чисел A і B: поки  $A \neq B$ , від більшого числа віднімати менше (рис. 2).



Рис. 1

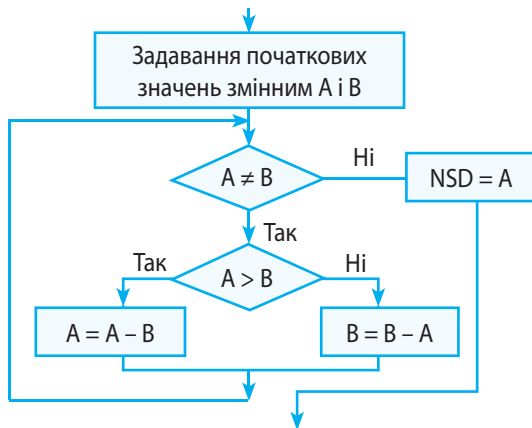


Рис. 2

- ▶ 4. Додайте на форму текстове поле Edit4. Змініть заголовок кнопки на «Обчислити NSD і NSK».
- ▶ 5. Додайте до програмного коду оператори для обчислення значення найменшого спільного кратного (НСК) чисел A і B та виведення цього значення в поле Edit4. Скористайтеся формулою:  $NSK(A, B) = A * B / NSD(A, B)$ .

## III. Тестування проекту

- ▶ 6. Запустіть проект на виконання. Перевірте роботу програми для різних наборів чисел. Збережіть проект у папці Практична робота 15. Завершіть роботу за комп'ютером.

**Зробіть висновок:** як складати та виконувати циклічні алгоритми з розгалуженням усередині циклу для розв'язування задач.

# РОЗДІЛ 9

## ГРАФІЧНЕ ВІДОБРАЖЕННЯ ДАНИХ



- § 43. Базові графічні примітиви
  - § 44. Налаштування властивостей графічних примітивів
  - § 45. Створення програм із графічним відображенням даних
  - § 46. Графічні компоненти
  - § 47. Виведення рисунків із зовнішніх файлів
  - § 48. Розробка програм із графічними компонентами
- Практична робота 16. Створення програми з графічними компонентами



## ПОВТОРЮЄМО



*Комп'ютерна графіка* — розділ інформатики, який вивчає технології опрацювання графічних зображень за допомогою комп'ютерної техніки. Залежно від способу побудови графічного зображення розрізняють растрові та векторні графічні зображення.

*Растрове графічне зображення* складається з окремих маленьких прямокутників — пікселів. Растровий малюнок зберігається в пам'яті комп'ютера у вигляді таблиці різнокольорових пікселів. Ви вже створювали графічні зображення в середовищі графічного редактора Paint.

*Векторне графічне зображення* будується з окремих базових об'єктів — графічних примітивів: відрізків, багатокутників, кривих, овалів тощо. У ході вивчення текстового редактора Word, програми для створення презентацій PowerPoint ви дізналися про те, як створювати векторні зображення із геометричних примітивів.

1. Що таке комп'ютерна графіка?
2. Що таке растрове графічне зображення?
3. Що таке векторне графічне зображення?
4. Наведіть приклади графічних примітивів.
5. Наведіть приклади графічних редакторів.
6. Опишіть принцип побудови векторного зображення.



У цьому розділі ви ознайомитеся із графічною системою координат, дізнаєтеся про принципи формування зображень на екрані, навчитесь застосовувати графічні можливості середовища програмування для створення зображень.

## § 43. Базові графічні примітиви

Середовище програмування Lazarus має різноманітні засоби для роботи з комп'ютерною графікою.

### Графічні примітиви


Ознайомлення з графічними можливостями середовища програмування Lazarus почнемо з графічних примітивів.



**Графічні примітиви** — це графічні елементи простої форми (лінії, кола, дуги тощо), з яких будують більш складні зображення.

Будь-який векторний малюнок можна розглядати як сукупність графічних примітивів. Щоб зображення з'явилося на екрані, в тексті програми потрібно передбачити креслення всіх графічних примітивів, з яких складається це зображення.

### Компонент зображення (Image)

Для роботи з графікою призначений компонент Image  (Зображення), який міститься на вкладці Additional Палітри компонентів. Після розміщення на формі компонент Image набуває вигляду прямокутної пунктирної рамки (рис. 43.1).

Змінити місце розташування на формі (властивості Left, Top), розміри (властивості Width, Height), колір фону (властивість Color) компонента Image можна у вікні Інспектор об'єктів або запрограмувавши зміну значень властивостей у ході виконання програми.

» Встановити в процедурі завантаження форми OnCreate такі значення властивостей зображення Image1, щоб відстань від нього до меж форми з усіх боків становила 20 пікселів (рис. 43.2).



Рис. 43.1

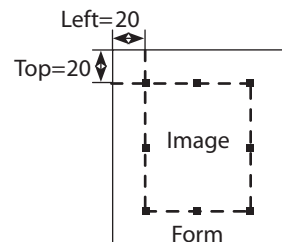


Рис. 43.2

```

procedure TForm1.FormCreate (Sender: TObject);
begin
 Image1.Left := 20;
 Image1.Top := 20;
 Image1.Width := Form1.Width - 40;
 Image1.Height := Form1.Height - 40;
end;

```

## Полотно для малювання

Компонент Image має властивість Canvas (*Полотно*). Малювання на полотні здійснюється, коли деяким його точкам надається колір. Властивостями полотна є Pen (*Олівець*), Brush (*Пензель*) і Font (*Шрифт*). Також полотно має властивість Pixels (*Точки*), яка надає можливість перекрасити кожен піксель полотна, тобто кожний піксель зображення. Координати конкретної точки вказують у квадратних дужках.

- » Вивести точку червоного кольору в задану точку екрана:

```
Image1.Canvas.Pixels [20,50] := clRed;
```

- » Полотно має систему координат *XOY* (рис. 43.3), верхня ліва точка якої має координати (0,0).



Рис. 43.3

- » Зафарбувати полотно заданим кольором.

```
Image1.Canvas.Brush.Color := clGreen; // встановлюємо зелений колір
Image1.Canvas.FillRect (Image1.ClientRect); // замальовуємо
```

Завдяки цим командам полотно стає видимим на екрані у вигляді зеленого прямокутника.

Колір можна вказати за допомогою колірних констант, які зручно використовувати для роботи з основними кольорами.

### Колірні константи

| Значення  | Опис    | Значення | Опис         | Значення | Опис       |
|-----------|---------|----------|--------------|----------|------------|
| clBlack   | Чорний  | clGray   | Темно-сірий  | clRed    | Червоний   |
| clWhite   | Білий   | clLtGray | Світло-сірий | clTeal   | Бірюзовий  |
| clBlue    | Синій   | clMaroon | Малиновий    | clPurple | Бузковий   |
| clGreen   | Зелений | clNavy   | Темно-синій  | clSilver | Сріблястий |
| clYellow  | Жовтий  | clOlive  | Оливковий    | clAqua   | Блакитний  |
| clFuchsia | Рожевий | clCream  | Кремовий     | clLime   | Салатовий  |

## Графічні методи

Полотно надає простір для створення графічних об'єктів.

Для створення графічних примітивів викликають графічні методи полотна Canvas, що забезпечують малювання певного графічного примітива.

**Відрізок.** Метод `LineTo` креслить відрізок від поточної позиції олівця до точки з координатами, вказаними під час виклику методу: `Image1.Canvas.LineTo (250,65);`

Спочатку олівець розташований над точкою  $(0, 0)$ . Встановити олівець в іншу позицію можна викликом методу `MoveTo`.

» Намалювати відрізок чорного кольору від точки  $(20, 20)$  до точки  $(100, 20)$ .

```
Image1.Canvas.Brush.Color := clBlack;
Image1.Canvas.MoveTo (20,20);
Image1.Canvas.LineTo (100,20);
```

**Прямокутник.** Метод `Rectangle` малює незаповнений прямокутник, протилежні кути якого збігаються з точками  $(x_1, y_1)$  і  $(x_2, y_2)$ .

Сторони прямокутника паралельні сторонам екрана.

```
Image1.Canvas.Rectangle (x1, y1, x2, y2);
```

» Намалювати квадрат із довжиною сторони 100 пікселів:

```
Image1.Canvas.Rectangle (20,30,120,130);
```

**Еліпс.** Метод `Ellipse` малює еліпс, вписаний в прямокутник, який задано координатами двох протилежних вершин  $x_1, y_1$  і  $x_2, y_2$  (рис. 43.4).

```
Image1.Canvas.Ellipse (x1, y1, x2, y2);
```

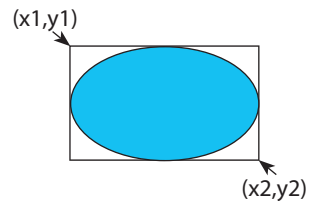


Рис. 43.4

» Намалювати коло діаметром 100 пікселів. `Image1.Canvas.Ellipse (20, 30, 120, 130);`

**Сектор.** Метод `Pie` малює сектор еліпса.

Еліпс, вписаний у прямокутник, задано координатами  $(x_1, y_1, x_2, y_2)$  (рис. 43.5). Сектор обмежений прямими, що проходять через

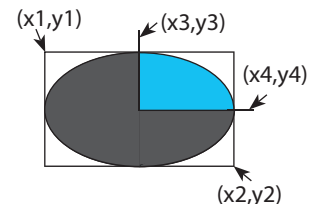


Рис. 43.5

центр еліпса і точки  $(x_3, y_3)$  і  $(x_4, y_4)$ . Дуга малюється від точки  $(x_3, y_3)$  до точки  $(x_4, y_4)$  проти ходу годинникової стрілки.

`Image1.Canvas.Pie (x1, y1, x2, y2, x3, y3, x4, y4)`

- » Намалювати фігуру за зразком (рис. 43.6).  
`Image1.Canvas.Rectangle (0,0,100,100); // квадрат`  
`Image1.Canvas.Pie (0,0,100,100,100,0,0,0); // сектор`

**Ламана лінія.** Метод `Polyline` малює ламану лінію за точками її вершин, що задані у вигляді масиву\* (таблиці) `Points`:

`Polyline (Points: array of Tpoint);`

- » Намалювати ламану лінію за п'ятьма точками, заданими координатами (рис. 43.7).  
`Image1.Canvas.Polyline ([Point (10,10), Point (30,100), Point (50,10), Point (70,100), Point (90,10)]);`

Ми розглянули лише частину методів для побудови графічних примітивів, але їх достатньо для того, щоб будувати складні малюнки. Пам'ятайте, що їх можна застосовувати для всіх компонентів, які мають властивість `Canvas`.

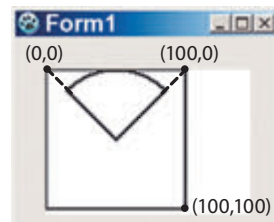


Рис. 43.6



Рис. 43.7

### Питання для самоперевірки



1. Що таке графічні примітиви?
2. Для чого призначене полотно `Canvas`?
3. Які властивості має полотно `Canvas`?
4. Які графічні методи застосовують до властивості `Canvas`?
5. Як накреслити відрізок на полотні компонента `Image`?
6. Як намалювати прямокутник; еліпс; ламану лінію на полотні `Canvas`?

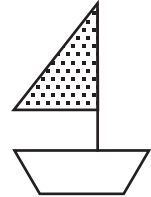
### Вправа 43



- » Скласти програму для малювання графічних примітивів.
  - 1) Створіть новий проект. Розмістіть на формі чотири командні кнопки, при клацанні яких буде виведено відповідний графічний примітив.

\* Детально з масивами ви ознайомитесь у 9 класі.

- 2) Побудуйте трикутник із вершинами в точках (100, 100), (200, 200), (100, 200).
- 3) Побудуйте прямокутник із вершинами (200, 10), (320, 10), (320, 120), (200, 120).
- 4) Побудуйте круг радіусом 50 пікселів, центр якого збігається з центром полотна Image1.
- 5) Запустіть проект на виконання. Перевірте роботу програми. Збережіть проект у папці Вправа 43.
- 6) Додайте нову кнопку для малювання човника (див. рисунок).



### Комп'ютерне тестування

Виконайте тестове завдання 43 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 44. Налаштування властивостей графічних примітивів

Для малювання рисунків і введення тексту застосовують інструменти Pen, Brush і Font. Олівець використовують для малювання ліній, пензель — для зафарбовування фігур, а шрифт — для вибору шрифту тексту.

### Властивість олівець (Pen)

**Олівець** (Canvas.Pen) використовують для малювання фігур, що складаються з ліній. Розглянемо деякі властивості олівця.

Pen.Color встановлює колір олівця, тобто колір, яким буде намальований контур фігури. Властивість Color може набувати значення колірних констант. Також колір малюнка можна задавати за допомогою функції RGBToColor, аргументи якої задають ступінь насиченості червоного, зеленого і синього кольорів:

```
Pen.Color := RGBToColor (r, g, b);
```

Змінні  $r$ ,  $g$ ,  $b$  можуть набувати значень від 0 до 255. Так можна отримати більше ніж 16 млн різних кольорів:

| Колір    | R(ed) | G(reen) | B(lue) |
|----------|-------|---------|--------|
| Чорний   | 0     | 0       | 0      |
| Білий    | 255   | 255     | 255    |
| Червоний | 255   | 0       | 0      |
| Зелений  | 0     | 255     | 0      |
| Синій    | 0     | 0       | 255    |

Pen.Style визначає стиль лінії і може мати, зокрема, такі значення:

| Значення | Вигляд | Значення | Вигляд    |
|----------|--------|----------|-----------|
| psSolid  | ————   | psDash   | - - - - - |

Pen.Width визначає товщину лінії. Значення цієї властивості може бути цілим числом. Лінія завтовшки більш ніж 1 може бути тільки суцільною.

» Намалювати на формі піраміду за зразком (рис. 44.1).

```
Form1.Canvas.Pen.Style := psSolid;
Form1.Canvas.Pen.Width := 3;
Form1.Canvas.Polygon ([Point (150,50),
 Point (50,250), Point (250,250),
 Point (150,50), Point (270,170),
 Point (250,250)]);
Form1.Canvas.Pen.Width := 1;
Form1.Canvas.Pen.Style := psDash;
Form1.Canvas.MoveTo (150,50);
Form1.Canvas.LineTo (130,170);
Form1.Canvas.LineTo (270,170);
Form1.Canvas.MoveTo (130,170);
Form1.Canvas.LineTo (50,250);
```

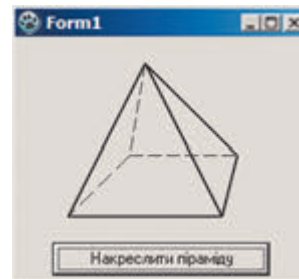


Рис. 44.1

## Властивість пензель (Brush)

Пензель (Canvas.Brush) використовують для зафарбовування (залитки) замкнених геометричних фігур. Властивості пензля:



- `Brush.Color` встановлює колір, яким можна зафарбовувати фігури;
- `Brush.Style` визначає стиль заповнення фігури, який, зокрема, може набувати таких значень:
  - `bsSolid` — суцільне заповнення кольором;
  - `bsDiagonal` — штрихування справа наліво;
  - `bsHorizontal` — горизонтальні лінії та інші.

## Виведення тексту на полотно

**Шрифт** (`Canvas.Font`) визначає шрифт, яким буде виведено текст. Можна встановити такі властивості шрифту: розмір, накреслення тощо.

- » Змінити значення властивостей шрифту.
 

```
Canvas.Font.Size := 20; // розмір у пунктах (20)
Canvas.Font.Color := clRed; // колір (червоний)
Canvas.Font.Name := 'Comic Sans MS'; // гарнітура (Comic Sans MS)
Canvas.Font.Style := [fsBold, fsItalic]; // накреслення (напівжирне, курсив)
або
Canvas.Font.Style := []; // накреслення (звичайне)
Canvas.Font.Style := [fsBold]; // накреслення (напівжирне)
```

**Виведення тексту на полотно** може здійснюватися за допомогою методу `TextOut` (`x, y: Integer; const Text: String`);

Цей метод виводить рядок тексту `Text` на полотно, починаючи з позиції з координатами (`x, y`).

```
Canvas.TextOut (10, 10, 'Виведення тексту');
```

- » Заповнити прямокутники зеленим кольором із використанням різних стилів пензля за зразком (рис. 44.2).

Для того щоб біля кожного виклику методу не писати назву об'єкта `Canvas`, використовуємо оператор приєднання `With`. При цьому в тілі оператора до кожного рядка, початок якого збігається з назвою властивості або методу об'єкта `Canvas`, під час компіляції буде додано `Canvas`.

```
With Canvas do begin
```

```
 Font.Size := 16;
 TextOut (10, 10, 'bsSolid'); // виведення назви стилю пензля
 Brush.Color := clGreen; // колір заливки
```

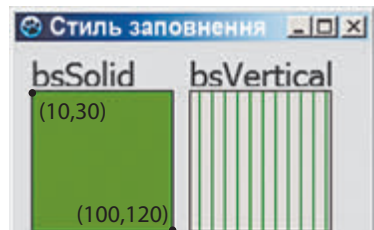


Рис. 44.2

```

Brush.Style := bsSolid; // стиль заливки
Rectangle (10, 30, 100, 120);
Brush.Style := bsClear;
TextOut (110, 10, 'bsVertical');
Brush.Color := clGreen;
Brush.Style := bsVertical; // стиль заливки
Rectangle (110, 30, 200, 120);
end;

```

## Заповнення багатокутників і складних фігур

Під час малювання простих геометричних фігур можна автоматично їх зафарбовувати відповідно до налаштувань пензля. Для будь-якої замкненої області використовують спеціальний метод FloodFill:

```
Canvas.FloodFill (x, y, Color, FillStyle);
```

Тут  $x, y$  — це координати будь-якої точки всередині фігури.

Якщо  $\text{FillStyle} = \text{fsBorder}$ , то фарба «розтікається» від точки з координатами  $(x, y)$  доти, поки не зустрине на шляху колір, заданий в параметрі  $\text{Color}$ . Якщо  $\text{FillStyle} = \text{fsSurface}$ , то колір  $\text{Color}$  замінюється кольором, вказаним у властивості  $\text{Brush.Color}$ .

» Намалювати зелений трикутник зі сторонами червоного кольору (рис 44.3).

```

Form1.Canvas.Pen.Color := clRed;
Form1.Canvas.Polygon ([Point (150,50), Point (50,250),
 Point (250,250), Point (150,50)]);
Form1.Canvas.Brush.Color := clGreen;
Form1.Canvas.FloodFill (200, 200, clRed, fsBorder);

```

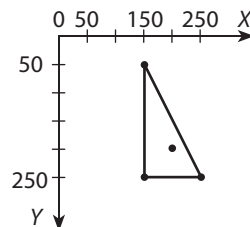


Рис. 44.3

Описані можливості Lazarus значно розширюють ваш арсенал художника-програміста. Тому обов'язково випробуйте всі наведені приклади, а потім спробуйте запрограмувати побудову власного малюнка.

## Питання для самоперевірки



1. Які властивості має інструмент Pen?
2. Як задати колір для побудови контуру графічних примітивів?
3. Які властивості має інструмент Brush?
4. Як зафарбувати фігуру?
5. Як побудувати і зафарбувати зеленим кольором коло радіусом 100, центр якого збігається з центром форми?

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

### Вправа 44



- ▶▶ Скласти програму для малювання ялинки за зразком (рис. 1).

- 1) Створіть новий проект. Додайте на форму кнопку Button1, при клацанні якої буде виконуватися малювання на полотні форми.
- 2) Створіть процедуру обробки події OnClick для форми.

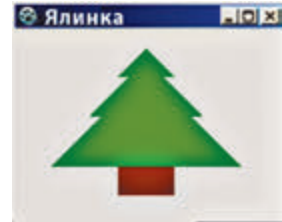


Рис. 1

- 3) Запишіть програмний код для малювання ялинки:

```
With Form1.Canvas do // працюємо з об'єктом Form1.Canvas
begin
 Pen.Width := 1; // встановлюємо товщину олівця
 Pen.Color := clGreen; // встановлюємо колір олівця
 Brush.Color := clGreen; // встановлюємо колір заливки
 { Малювання ялинки }
 Polygon ([Point (40,200), Point (160,100), Point (180,100), Point (140,160),
 Point (160,160), Point (100,250), Point (300,250), Point (240,160), Point
 (260,160), Point (220,100), Point (240,100)]);
 Pen.Color := RGB (128,64,0); // встановлюємо колір олівця
 Brush.Color := RGB (128,64,0); // встановлюємо колір заливки
 { Малювання стовбура }
 Polygon ([Point (250,251), Point (250,301), Point (150,301), Point (150,251)]);
```

- 4) Запустіть проект на виконання. Додайте оператори малювання жовтого сонечка (рис. 2).
- 5) Перевірте роботу програми. Збережіть проект у папці Вправа 44.



Рис. 2

- 6) Наша ялинка «висить» у повітрі. Додайте оператори для малювання лісової галявини, на якій росте ялинка, і хмаринок над ялинкою.



### Комп'ютерне тестування

Виконайте тестове завдання 44 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 45. Створення програм із графічним відображенням даних

У попередньому параграфі ви малювали графічні примітиви на формі в результаті виконання процедури обробника події `OnClick` для кнопки. При перекритті вікна форми іншими вікнами або зміні розмірів форми частина зображення може бути втрачена. Існують способи для відновлення зображення під час активізації форми та налаштування розмірів зображення під поточні розміри форми.

### Перемальовування форми

Подія `OnPaint` відбувається за необхідності перемальовати форму, наприклад, у разі її активізації, якщо до цього частина форми була закрита іншими вікнами.

Для створення процедури обробки події `OnPaint` потрібно:

- 1) клацнути мишею на формі;
- 2) у вікні Інспектор об'єктів перейти на вкладку Події;
- 3) двічі клацнути в полі події `OnPaint` (рис. 45.1).

Створена процедура `TForm1.FormPaint` виконуватиметься під час завантаження форми і при кожному її перемальовуванні під час зміни розмірів або відображення форми на екрані.

```
procedure TForm1.FormPaint (Sender: TObject);
begin
```

```
 Form1.Canvas. Rectangle (200,0,300,100);
 { далі слідують команди для побудови малюнка }
```

```
end;
```

Для візуалізації процесу зміни розміру форми потрібно очистити полотно перед перемальовуванням форми нового розміру. Для цього в процедурі обробки події `OnResize` слід викликати метод `Refresh ()`. Він очищує полотно і генерує подію `OnPaint`:

```
procedure TForm1.FormResize (Sender: TObject);
begin
```

```
 Refresh ();
```

```
end;
```

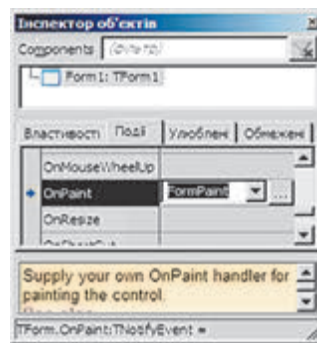


Рис. 45.1

» Намалювати різнокольорові лінії (відрізки) випадковим чином (рис. 45.2).

```
procedure TForm1.FormPaint (Sender: TObject);
const Num_Lines = 100; // кількість ліній
var
i: Integer;
begin
 { ініціалізація генератора випадкових чисел }
 Randomize;
 For i := 0 to Num_Lines do
 begin
 Canvas.Pen.Color := RGBToColor(Random (256), Random (256), Random
(256)); // випадковий вибір кольору
 Canvas.LineTo (Random (ClientWidth), Random (ClientHeight)); { малю-
вання відрізка з випадковими координатами кінців }
 end;
 end;
procedure TForm1.FormResize (Sender: TObject);
begin
 Refresh ();
end;
```

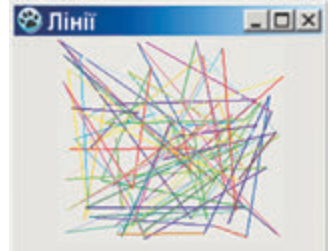


Рис. 45.2

## Масштабування зображення

Під час зміни розмірів форми в режимі виконання бажано, щоб розміри нашого малюнка підлаштувалися під ці зміни. Властивості форми ClientWidth, ClientHeight визначають ширину і висоту області, на якій можна малювати.

Щоб малюнок масштабувався разом із формою, необхідно при ви-клику графічних методів указувати не абсолютні значення координат, а залежності їх координат від властивостей ClientWidth, ClientHeight.

» Масштабувати малюнок за зразком (рис. 45.3, 45.4):

```
procedure TForm1.FormPaint (Sender: TObject);
const x = 10; y = 10;
var dx, dy: Integer;
begin
 dx := Form1.ClientWidth div x;
 dy := Form1.ClientHeight div y;
 with Form1.Canvas do begin
 Font.Size := 20;
```

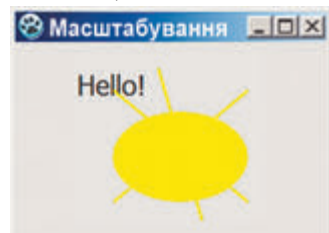


Рис. 45.3

```

TextOut (0, dy, ' Hello!');
Pen.Color := clYellow;
Pen.Width := 3;
Brush.Color := clYellow;
Ellipse (dx, 2*dy, 3*dx, 5*dy);
MoveTo (dx, 2*dy);
LineTo (3*dx, 5*dy);
MoveTo (3*dx, dy);
LineTo (dx, 5*dy);
MoveTo (0, 4*dy);
LineTo (4*dx, 3*dy);
Brush.Style := bsClear;
end;
end;
procedure TForm1.FormResize (Sender: TObject);
begin
 Refresh ();
end;

```

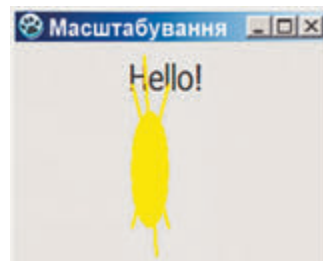


Рис. 45.4

### Питання для самоперевірки



1. Як створити процедуру обробки події OnPaint для форми?
2. Яку дію виконує метод Refresh ()?
3. Якими властивостями визначаються розміри форми?
4. Як запрограмувати масштабування зображення?
5. Як запрограмувати задавання кольору малювання випадковим чином?
6. Напишіть фрагмент програми для заповнення форми різнокольоровими точками з випадковими координатами.

### Вправа 45



- Скласти програму для малювання в центрі форми чотирьох концентричних еліпсів, що зафарбовано різними кольорами (рис. 1).
- 1) Створіть новий проект. Змініть заголовок форми на «Еліпси».
  - 2) Створіть процедуру обробки події OnPaint для форми.
  - 3) У програмному кодї процедури запишіть оператори для малювання чотирьох еліпсів (рис. 2).



Рис. 1

```

procedure TForm1.FormPaint (Sender: TObject);
const x = 10; y = 10; // умовне розбиття форми на комірки
var dx, dy, i: Integer;
begin
 dx := Form1.ClientWidth div x;
 dy := Form1.ClientHeight div y;
 For i := 1 to 4 do begin
 Canvas.Brush.Color := RGBToColor
 (255, 255 div i, 255 div i);
 Canvas.Ellipse (i*dx, i*dy, (10-i) *dx,
 (10-i) *dy);
 end;
end;
procedure TForm1.FormResize (Sender: TObject);
begin
 Refresh ();
end;

```

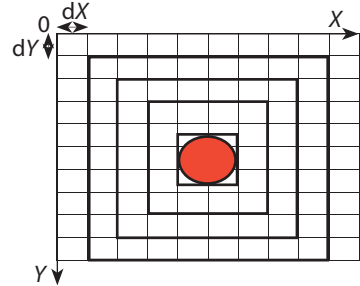


Рис. 2

- 4) Змініть кольори заливки еліпсів на відтінки зеленого.
- 5) Запустіть проект на виконання. Перевірте роботу програми. Збережіть проект у папці Вправа 45.
- 6) Змініть програмний код таким чином, щоб виконувалось малювання п'яти вкладених прямокутників.



### Комп'ютерне тестування


Виконайте тестове завдання 45 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).



## § 46. Графічні компоненти

Під час конструювання форми для створення візуальних ефектів і зображень можна використовувати такі графічні компоненти, як зображення Image і автофігура Shape.

### Автофігури (Shape)

Компонент Shape  (Автофігури) міститься на вкладці Additional Палітри компонентів і призначений для виведення геометричних фігур.



Вигляд фігури, що виводиться цим компонентом, визначається властивістю Shape. Форму фігури можна вибрати в списку властивості Shape (рис. 46.1) у вікні Інспектор об'єктів (рис. 46.2).

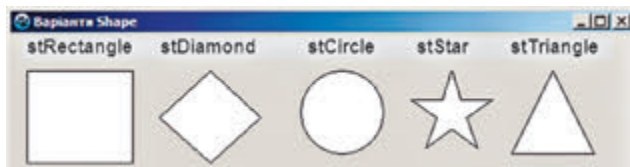


Рис. 46.1

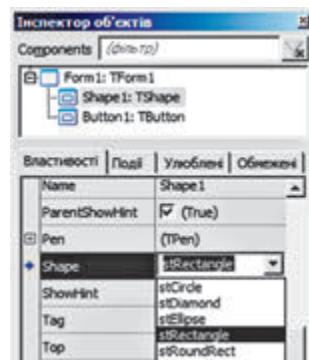


Рис. 46.2

Щоб задати розміри фігури, у вікні Інспектор об'єктів встановлюють значення властивостей Height і Width. Для точного розміщення фігури на формі призначені властивості Left і Top.

Ви вже знайомі з властивостями Pen і Brush полотна Canvas компонента Image. Аналогічні властивості має і компонент Shape.

Властивість Pen визначає колір і стиль контуру фігури. Властивість Brush визначає стиль і колір зафарбованої фігури. Значення цих властивостей можна задати як у вікні Інспектор об'єктів, так і програмним способом.

» Змінити значення властивостей компонента Shape програмним шляхом за зразком (рис. 46.3).

```
With Shape1 do begin
 Shape := stEllipse;
 Pen.Color := clBlue;
 Pen.Width := 3;
 Brush.Color := clRed;
 Brush.Style := bsDiagCross;
end;
```

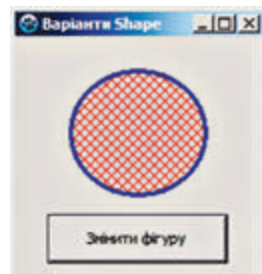


Рис. 46.3

## Переміщення об'єктів мишею

Розглянемо алгоритм переміщення об'єкта за допомогою миші. Для **переміщення об'єкта** треба:

- 1) натиснути ліву кнопку миші на об'єкті (подіяMouseDown);
- 2) перемістити вказівник при натиснутій кнопці миші (подіяMouseMove);
- 3) відпустити кнопку миші (подіяMouseUp).

Щоб запрограмувати переміщення по формі об'єкта `Shape1`, треба створити для нього процедури обробки подій: `OnMouseDown` (натиснення кнопки миші, коли вказівник миші знаходиться в області об'єкта), `OnMouseMove` (переміщення миші), `OnMouseUp` (відпускання кнопки миші). Текст програми буде таким:

**var**

```
Form1: TForm1;
x1, y1: Integer; // опис глобальних змінних
flag: Boolean = False; // кнопка миші не натиснута
```

**implementation**

```
{$R * .lfm}
```

**procedure TForm1.Shape1MouseDown** (Sender: TObject; Button: TMouseButton; Shift: TShiftState; x, y: Integer);

**begin**

```
x1 := x; y1 := y; // запам'ятовування поточних координат вказівника миші
flag := True; // кнопка миші натиснута
```

**end;**

**procedure TForm1.Shape1MouseMove** (Sender: TObject; Shift: TShiftState; x, y: Integer);

**begin**

```
If flag Then // якщо кнопка миші натиснута
```

**begin**

```
Shape1.Left := Shape1.Left + x - x1;
```

```
Shape1.Top := Shape1.Top + y - y1;
```

**end;**

**end;**

**procedure TForm1.Shape1MouseUp** (Sender: TObject; Button: TMouseButton; Shift: TShiftState; x, y: Integer);

**begin**

```
flag := False; // кнопка миші не натиснута
```

**end;**

Розглянемо роботу процедур даного проекту. Якщо кнопку миші натиснуто в момент, коли вказівник знаходиться в області об'єкта `Shape1`, то в процедурі `Shape1MouseDown` змінним `x1`, `y1` будуть надані значення координат вказівника, а змінній `flag` — значення `True`. Якщо не відпускати кнопку миші й перетягувати об'єкт, то в процедурі `Shape1MouseMove` відбувається перерахунок координат об'єкта `Shape1` відносно системи координат форми. Якщо кнопку миші відпустити, то виконається процедура `Shape1MouseUp`, в якій змінна `flag` набуде значення `False`, і перетягування завершиться.

- » Розробити гру «Конструктор», створивши процедури обробки подій OnMouseDown, OnMouseMove, OnMouseUp для компонентів Shape (рис. 46.4).



Рис. 46.4

### Питання для самоперевірки



1. Для чого призначений компонент Shape?
2. Яка властивість компонента Shape визначає форму автофігури?
3. Як змінити розміри; колір заливки; контур автофігури?
4. Які події відбуваються під час натиснення; переміщення; відпускання миші?
5. Як запрограмувати переміщення об'єкта за допомогою миші?

### Вправа 46



- » Скласти програму, у якій реалізовано відображення на формі п'яти різних геометричних фігур (рис. 1).

- 1) Створіть новий проект. Змініть заголовки форми на «Стилі заповнення Shape». Розмістіть на формі п'ять компонентів Shape.
- 2) Надайте фігурам вигляду: прямокутника (stRectangle); кола (stCircle); еліпса (stEllipse); квадрата (stSquare); прямокутника із заокругленими кутами (stRoundRect).

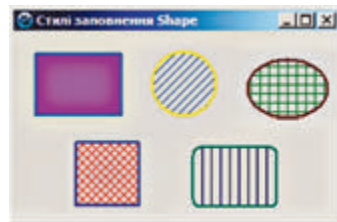


Рис. 1

- 3) Змініть контур фігур, використовуючи властивості Pen.Color, Pen.Width, Pen.Style.
- 4) Залийте фігури потрібними кольором та стилем, використовуючи властивості Brush.Color, Brush.Style.
- 5) Перевірте роботу програми. Збережіть проект у папці Вправа 46.
- 6) Створіть процедури обробки подій OnMouseDown, OnMouseMove, OnMouseUp для будь-якої автофігури.



- 7) За допомогою компонентів Shape створіть на формі прапори України, Франції, Польщі (рис. 2).



Рис. 2

### Комп'ютерне тестування

Виконайте тестове завдання 46 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).


## § 47. Виведення рисунків із зовнішніх файлів

Для створення проекту можна використовувати зображення, що містяться у зовнішніх графічних файлах. Зображення, яке міститься у файлі, додають на форму за допомогою компонента Image.

### Властивості компонента Image

| Властивість   | Що визначає                                                                                                                                                                  |
|---------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Picture       | Зображення, яке виводиться із зовнішнього файла в область компонента                                                                                                         |
| Width, Height | Ширина і довжина компонента. Якщо розміри компонента менші за розміри зображення і значення властивостей AutoSize і Stretch дорівнює False, то виводиться частина зображення |
| AutoSize      | Ознака автоматичної зміни розміру компонента відповідно до розміру зображення (має тип Boolean)                                                                              |
| Stretch       | Ознака автоматичного масштабування зображення відповідно до розміру компонента за умови, що AutoSize = False (має тип Boolean)                                               |
| Visible       | Ознака відображення компонента, і, відповідно, зображення, на формі (має тип Boolean)                                                                                        |

Для **виведення зображення в область компонента Image** потрібно:

- у вікні Інспектор об'єктів вибрати властивість Picture і клацнути кнопку ;
- клацнути кнопку Завантажити вікна Діалог завантаження зображення;
- вибрати файл у діалоговому вікні вибору файла.

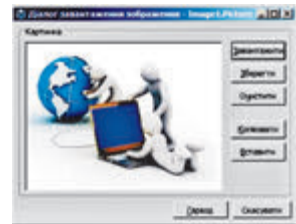


Рис. 47.1

Якщо розміри зображення більші від розмірів компонента, то властивості `Stretch` потрібно надати значення `True`. Розміри малюнка зміняться відповідно до реальних розмірів компонента `Image` (рис. 47.2).



Рис. 47.2

## Завантаження зображення під час виконання проекту

Щоб вивести зображення в область компонента `Image` під час роботи проекту, потрібно до властивості `Picture` застосувати метод `LoadFromFile`, вказавши як параметр ім'я файлу з потрібним зображенням.

» `Image1.Picture.LoadFromFile ('c:\Photo\portret.bmp')` — завантажує зображення із графічного файлу `portret.bmp`, що міститься в каталозі `Photo` кореневого каталогу диска `C:`, і виводить його в область об'єкта `Image1`.


» Якщо графічний файл міститься в поточному каталозі (у папці проекту), достатньо вказати лише його назву.  
`Image1.Picture.LoadFromFile ('Parrot.jpg');`

Метод `LoadFromFile` дозволяє виводити зображення різних графічних форматів, але для завантаження `jpg`-файлів необхідно включити до блоку `uses` стандартний модуль `JPEG`:

```
uses JPEG;
```

Очистити вміст об'єкта `Image1` можна таким способом:  
`Image1.Picture := nil;`

## Піктографічна кнопка (`BitBtn`)

Компонент `BitBtn`  (*Піктографічна кнопка*) міститься на вкладці `Additional` Палітри компонентів.

Компонент `BitBtn` надає можливість створювати кнопку, яка на своїй поверхні може містити не тільки напис, але й зображення, що задається властивістю `Glyph`. При клацанні кнопки в рядку `Glyph` у вікні Інспектор об'єктів відкривається вікно `Picture Editor`, як і при задаванні властивості `Picture` для компонента `Image`.

Можливості масштабування малюнка для компонента BitBtn немає, тому потрібно його розміри узгоджувати з розмірами створеної кнопки.

Властивість Kind компонента BitBtn надає можливість вибрати вигляд кнопки зі стандартного набору варіантів.

Клацання будь-якої піктографічної кнопки повертає в програму результат (назву цієї кнопки, наприклад Yes, No, Cancel), який визначає подальше виконання програми. Відмінність є у кнопки Close: її клацання приводить до завершення роботи програми.

## Вибір кольору (ColorDialog)

Компонент ColorDialog



(вкладка Dialogs на Палітрі компонентів) дозволяє під час роботи програми викликати діалогове вікно вибору кольору (рис. 47.3).

У вікні можна вибирати колір з базової палітри або, клацнувши кнопку Визначити колір, розкрити додаткову панель (див. рис. 47.3). Це дозволяє вибрати колір, відмінний від базових.

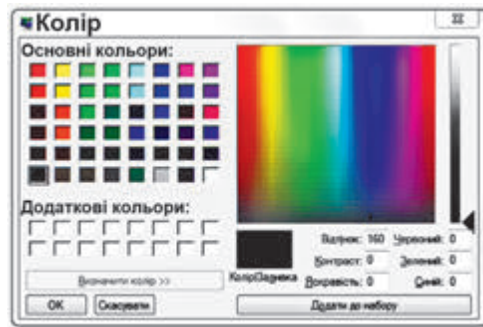


Рис. 47.3

Ви вже знаєте, що колір об'єкта визначається властивістю Color. Значення властивості Color можна задати на етапі проектування у вікні Інспектор об'єктів або під час виконання проекту. Вибір у програмі кольору з розширеної панелі здійснюється за допомогою методу Execute. Метод спрацьовує в момент клацання кнопки ОК у вікні вибору кольору.

» Задати колір фону автофігури Shape1 за допомогою оператора:  
`If ColorDialog1.Execute Then Shape1.Color := ColorDialog1.Color;`

## Питання для самоперевірки



1. Як розмістити на формі зображення, яке зберігається в зовнішньому файлі?
2. Опишіть основні властивості компонента Image.
3. Чому для властивості Stretch компонента Image корисно встановлювати значення True?



4. Запишіть оператор, який реалізує додавання на форму зображення з файла P1.bmp, який міститься на диску C: в папці Photo.
5. Чим відрізняються компоненти Button і BitBtn?
6. Запишіть оператор, який надає кольору олівця значення, вибраного у діалоговому вікні кольорів.

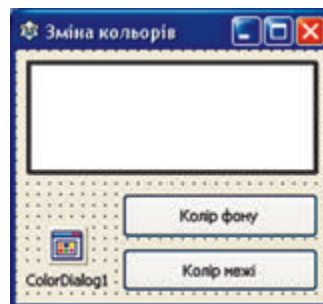
### Вправа 47



► Скласти програму, в якій автофігурі надається колір за допомогою компонента ColorDialog.

- 1) Створіть новий проект. Змініть заголовок форми на «Зміна кольорів».
- 2) Розмістіть на формі компоненти Shape і ColorDialog згідно з рисунком. У вікні Інспектор об'єктів змініть значення товщини межі прямокутника (об'єкта Shape1). Для цього у списку параметрів властивості Pen встановіть для параметра Width значення 3 пікселі.
- 3) Розмістіть на формі два компоненти Button і змініть заголовки створених кнопок на «Колір фону» і «Колір межі».
- 4) Створіть процедуру обробки події OnClick для кнопки Колір межі. У програмному коді створеної процедури запишіть оператори для відображення контуру автофігури Shape1 кольором, вибраним у діалоговому вікні кольорів:
 

```
procedure TForm1.Button2Click (Sender: TObject);
begin
 If ColorDialog1.Execute Then Shape1.Pen.Color := ColorDialog1.Color;
end;
```
- 5) Створіть процедуру обробки події OnClick для кнопки Колір фону. Додайте оператори для зміни кольору заливки автофігури Shape1 на колір, вибраний у діалоговому вікні кольорів. Запустіть проект на виконання. Збережіть проект у папці Вправа 47.
- 6) Додайте до процедури Button1Click оператор для зміни стилю заливки автофігури. Додайте до процедури Button2Click оператор для збільшення ширини межі прямокутника.



### Комп'ютерне тестування

Виконайте тестове завдання 47 з автоматичною перевіркою на сайті [interactive.ranok.com.ua](http://interactive.ranok.com.ua).







## § 48. Розробка програм із графічними компонентами

Із попереднього параграфу ви дізналися, як у проєкті завантажити зображення, що міститься в графічному файлі. Також у проєкті можна відкривати і зберігати файли.

### Робота з файлами зображень

Компоненти `OpenPictureDialog`  і `SavePictureDialog`  (вкладка `Dialogs` на Палітрі компонентів) дозволяють під час роботи програми вибирати в діалоговому вікні графічний файл для його відкриття або вказувати назву файла для збереження зображення.

Діалогове вікно компонента `OpenPictureDialog` має область попереднього перегляду файлів. Значення властивості `Filter` встановлено для показу графічних файлів з розширеннями `.bmp`, `.ico`, `.emf`, `.wmf`. Для того щоб у фільтрі була можливість вибору файлів із розширенням `.jpg`, потрібно додати модуль `JPEG` до блоку `uses`.

Під час виконання проєкту реалізація діалогу з користувачем для відкриття або збереження файла здійснюється за допомогою методу `Execute`. Метод спрацьовує в момент клацання кнопки `Відкрити` або `Зберегти` у вікні вибору файла. Якщо користувач клацне кнопку `Відкрити` (`Зберегти`), діалог завершиться, метод `Execute` поверне значення `True` і назва вибраного файла збережеться у властивості `FileName` компонента `OpenPictureDialog` (`SavePictureDialog`). Якщо ж користувач відмовиться від діалогу (клацне кнопку `Скасувати` або натисне клавішу `Esc`), то метод `Execute` поверне значення `False`.

У програмі потрібно передбачити відповідний компонент, за допомогою якого розпочнеться діалог користувача з програмою щодо відкриття (збереження) файла.


Для відкриття файла в процедурі обробки події для відповідного компонента (наприклад, кнопки) записують оператор:

```
If OpenPictureDialog1.Execute Then
 Image1.Picture.LoadFromFile (OpenPictureDialog1.FileName);
```

Для збереження зображення у файлі в процедурі обробки події для відповідного компонента слід записати оператор:

```
If SavePictureDialog1.Execute Then
 Image1.Picture.SaveToFile (SavePictureDialog1.FileName);
```

## Головне меню (MainMenu)

Компонент MainMenu  (вкладка Standard на Палітрі компонентів) призначений для додавання до програми головного меню. Редактор меню викликається подвійним клацанням компонента MainMenu.

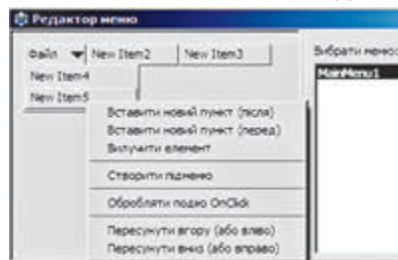


Рис. 48.1

У вікні Редактор меню автоматично створюється пункт меню верхнього рівня New Item1. Для додавання пунктів меню і підменю зручно користуватися командами контекстного меню (рис. 48.1). Меню складається з окремих пунктів, значення властивості Caption кожного з яких потрібно задати у вікні Інспектор об'єктів.

Кожна команда меню — це об'єкт зі своїми властивостями і методами. Для того щоб меню виконувало свої функції, потрібно створити процедуру обробки події OnClick для кожного пункту меню.

### Питання для самоперевірки



1. Який компонент дозволяє відкривати графічний файл за допомогою діалогового вікна під час виконання проекту?
2. Який компонент дозволяє зберігати графічний об'єкт за допомогою діалогового вікна під час виконання проекту?
3. Поясніть призначення методу Execute.
4. Поясніть призначення компонента MainMenu.
5. Як додавати пункти до головного меню?
6. Як створити процедуру обробки події для пунктів меню?

### Вправа 48



- ▶ Скласти програму, в якій реалізовані основні функції графічного редактора.
  - 1) Створіть новий проект. Змініть заголовок форми на «Графічний редактор». Розмістіть на формі компоненти згідно з рис. 1: Image; OpenFileDialog; SaveFileDialog; MainMenu; ColorDialog; три компоненти BitBtn.

Пункти меню назвіть так, як показано на рис. 1–3.

  - 2) Створіть процедуру обробки події OnCreate для форми. У програмному коді процедури запишіть оператори:

```

procedure TForm1.FormCreate (Sender: TObject);
begin
 Image1.Canvas.Brush.Color := clWhite;
 Image1.Canvas.FillRect (ClientRect);
end;

```

- 3) У редакторі меню створіть пункти меню Файл і Малюнок та відповідні пункти підменю за зразком (рис. 2, 3).

- 4) Створіть процедури обробки подій для пунктів Відкрити і Зберегти меню Файл.
- ```

procedure TForm1.MenuItem2Click (Sender: TObject);
begin

```

```

    If OpenFileDialog1.Execute Then
    Image1.Picture.LoadFromFile
        (OpenFileDialog1.FileName);

```

```

end;

```

```

procedure TForm1.MenuItem3Click (Sender: TObject);
begin

```

```

    If SavePictureDialog1.Execute Then
    Image1.Picture.SaveToFile (SavePictureDialog1.
        FileName);

```

```

end;

```

- 5) Самостійно створіть процедури обробки подій для пункту Вихід меню Файл і пункту Очистити меню Малюнок.
- 6) Додайте на піктографічні кнопки BitBtn1, BitBtn2, BitBtn3 відповідні зображення — олівець, заливка, колірна палітра. Кнопки BitBtn1, BitBtn2 відповідають за вибір інструмента для малювання, кнопка BitBtn3 — за вибір кольору.
- 7) До блоку опису глобальних змінних додайте опис змінних, що відповідають за вибір інструментів для малювання:

```

var Form1: TForm1;
    pen1, brush1: Boolean; cl: TColor;
    flag: Boolean; // допоміжна змінна

```

- 8) Створіть процедури обробки подій для кнопок BitBtn1, BitBtn2.

```

procedure TForm1.BitBtn1Click (Sender: TObject);
begin

```

```

    pen1 := True; brush1 := False;
end;

```



Рис. 1

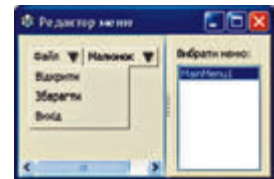


Рис. 2

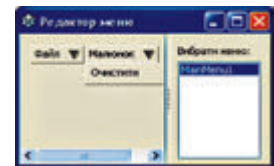


Рис. 3

```

procedure TForm1.BitBtn2Click (Sender: TObject);
begin
    pen1 := False; brush1 := True;
end;

```

- 9) Створіть процедуру обробки подій для кнопки BitBtn3, що реалізує вибір кольорів за допомогою компонента ColorDialog.
- 10) Запрограмуйте можливість малювання за допомогою миші (рис. 4). При клацанні кнопки миші в області компонента Image1 у процедурі Image1MouseDown перевіряється, який інструмент обрано. Якщо обрано інструмент Олівець (pen1 = True), то змінній flag надається значення True, малювання лінії починається з поточної координати (x, y) вказівника. Якщо обрано інструмент Заливка (brush1 = True), то відбувається заливка від точки (x, y) вибраним кольором (FillStyle = fsSurface).

Якщо не відпускати кнопку миші і пересувати вказівник по поверхні Image1, то у процедурі TForm1.Image1MouseMove відбувається малювання лінії за поточними координатами вказівника миші. Якщо кнопку миші відпустити, то виконається процедура TForm1.Image1MouseUp, в якій flag набуде значення False, і малювання лінії завершиться.

```

procedure TForm1.Image1MouseDown (Sender: TObject; Button:
TMouseButton; Shift: TShiftState; x, y: Integer);

```

```

begin
    If pen1 Then
        begin
            Image1.Canvas.Pen.Color := cl;
            flag := True;
            Image1.Canvas.MoveTo (x, y);
        end;

```

```

    end;
    If brush1 Then
        begin

```

```

            Image1.Canvas.Brush.Color := cl;
            Image1.Canvas.FloodFill (x, y, Image1.Canvas.Pixels [x, y], fsSurface);
        end;

```

```

    end;

```

```

procedure TForm1.Image1MouseMove (Sender: TObject; Shift: TShiftState;
x, y: Integer);

```

```

begin
    If flag Then Image1.Canvas.LineTo(x, y);
end;

```

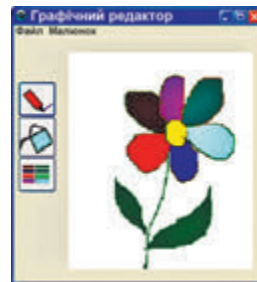


Рис. 4

```
procedure TForm1.Image1MouseDown (Sender:
  TObject; Button: TMouseButton; Shift:
  TShiftState; x, y: Integer);
```

```
begin
  flag := False;
end;
```

- 11) Перевірте роботу програми. Збережіть проект у папці Вправа 48. Створений графічний редактор дозволяє відкривати графічні файли, створювати і редагувати зображення, зберігати їх.



Комп'ютерне тестування

Виконайте тестове завдання 48 з автоматичною перевіркою на сайті interactive.ranok.com.ua.



Практична робота 16

Створення програми з графічними компонентами

Завдання: скласти програму для перегляду малюнків.

Обладнання: ПК із встановленим середовищем програмування Lazarus.

Хід роботи

Під час роботи з комп'ютером дотримуйтеся правил безпеки.


- ▶ 1. Знайдіть чотири малюнки з петриківським розписом і в графічному редакторі встановіть їхні розміри 80×80 пікселів. Створіть папку Практична робота 16 і збережіть у ній опрацьовані графічні файли.

I. Розміщення елементів керування на формі

- ▶ 2. Створіть новий проект і збережіть його у папці Практична робота 16.
- ▶ 3. Змініть заголовок форми на «Елементи петриківського розпису». Додайте на форму чотири компоненти `BitBtn` (рис. 1), розміри яких дорівнюють розмірам підготовлених зображень. Для цього задайте для властивостей `Width` і `Height` значення 80 пікселів. Очистіть вміст поля властивості `Caption` у вікні Інспектор об'єктів. На кожну з кнопок установіть зображення різних малюнків (властивість `Glyph`).



Рис. 1

- ▶ 4. Додайте на форму компонент Panel  (Панель) із вкладки Standard Палітри компонентів. Панель використовують як контейнер для інших компонентів. Властивості BevelOuter (Зовнішня рамка) і BevelInner (Внутрішня рамка) відповідають за стиль оформлення зовнішньої і внутрішньої рамок панелі відповідно. Виберіть для цих властивостей значення bvLowered (Рамка утоплена). Властивості BorderWidth (Ширина рамки) надайте значення 2.
- ▶ 5. Розмістіть компонент Image на компоненті Panel. Властивості Stretch компонента Image надайте значення True. Властивості Align надайте значення alClient для того, щоб компонент Image займав увесь доступний простір компонента Panel.
- ▶ 6. Додайте на форму кнопки Збільшити малюнок і Зменшити малюнок.
- ▶ 7. Додайте на форму п'ятий компонент BitBtn. Властивість Kind кнопки змініть на «Close».

II. Створення програмного коду

- ▶ 8. Запрограмуйте для кожної кнопки з малюнком відкриття в області компонента Image відповідного файла (рис. 2). Замість назви Малюнок1.bmp напишіть назву того файла, зображення з якого ви помістили на цю кнопку: `Image1.Picture.LoadFromFile ('Малюнок1.bmp');`
- ▶ 9. Створіть процедуру обробки події OnClick для кнопки Збільшити малюнок. У програмному коді створеної процедури запишіть оператори: `Panel1.Height := Panel1.Height + 5;` `Panel1.Width := Panel1.Width + 5;` Аналогічно запрограмуйте зменшення розмірів об'єкта Panel1 на 5 пікселів.
- ▶ 10. Додайте на форму компонент Label. Додайте до процедур обробки подій для кнопок BitBtn1–BitBtn4 оператори виведення назви файла.



Рис. 2

III. Тестування програми

- ▶ 11. Перевірте роботу програми.
- ▶ 12. Збережіть проект. Завершіть роботу за комп'ютером.

Зробіть висновок: як додавати зображення на форму; як використовувати піктографічні кнопки для розробки наочного інтерфейсу.

РОЗДІЛ 10

РОЗВ'ЯЗУВАННЯ КОМПЕТЕНТІСНИХ ЗАДАЧ. ВИКОНАННЯ ІНДИВІДУАЛЬНИХ І ГРУПОВИХ НАВЧАЛЬНИХ ПРОЕКТІВ



§ 49. Розв'язування компетентісних задач

§ 50. Виконання навчальних проектів

ПОВТОРЮЄМО



У сучасному інформаційному суспільстві дедалі більшого значення набувають такі компетенції, як уміння збирати необхідну інформацію, критично її опрацьовувати, робити висновки й умовиводи, використовуючи для роботи новітні інформаційні технології.

Компетентність у перекладі з латинської *competentia* означає коло питань, у яких людина добре обізнана, має знання та досвід. З курсу інформатики 5–7 класів ви знаєте, що для розв'язування компетентнісних задач потрібно виконати низку завдань за запропонованим планом як під керівництвом вчителя, так і самостійно. Ваша робота буде більш ефективною за умови активного використання інтернет-ресурсів для пошуку інформації та оцінювання результатів виконання завдань.

1. Яке значення має інформація в сучасному суспільстві?
2. Які проекти вам доводилося виконувати?
3. Які інформаційні технології ви використовували під час роботи над проектами?
4. Як здійснювати пошук матеріалів у Інтернеті?
5. Яким чином можна подати результати виконаної роботи?



У цьому розділі ви дізнаєтесь, як спланувати діяльність при виконанні навчального проекту, розробити стратегію розв'язування задач з різних галузей людської діяльності, які інформаційні технології зручно застосувати, навчитися оцінювати результати вашої діяльності.

§ 49. Розв'язування компетентнісних задач

Людині в інформаційному суспільстві необхідно не лише орієнтуватися у потоці інформації, а й організувати життя таким чином, щоб при найменших витратах часу і засобів отримувати від діяльності найбільшу віддачу. Будемо цього вчитися, розв'язуючи компетентнісні задачі.

Компетентнісні задачі з інформатики — це проблемні завдання з різних галузей людської діяльності, які розв'язуються засобами ІКТ.

Для розв'язування задачі вам треба буде провести пошук і відбір потрібної інформації як у друкованих джерелах (підручники, довідники), так і в інтернет-ресурсах (електронні енциклопедії, інтернет-видання).

Етапи розв'язування задачі

1. Постановка задачі	Змістовий аналіз формулювання задачі: які дані є в умові; які дані, в якому вигляді повинні бути отримані; які дані треба додатково знайти
2. Пошук інформації	Пошук інформації, спрямований на конкретизацію змісту описаної ситуації. Формування списку друкованих джерел та інтернет-ресурсів
3. Побудова інформаційної моделі	Інформаційна модель — це опис істотних для поставленої задачі властивостей і закономірностей поведінки об'єктів. Інформаційні моделі подаються у вигляді текстів, малюнків, таблиць, схем, формул тощо
4. Визначення засобів опрацювання даних	Добір засобів опрацювання даних (текстовий процесор, графічний редактор, редактор презентацій, табличний процесор, середовище програмування), необхідних для поетапного розв'язання задачі
5. Опрацювання даних	Поетапне виконання задачі засобами ІКТ
6. Подання результатів	Вибір форми подання результатів розв'язання задачі. Оформлення і презентація отриманих результатів
7. Аналіз результатів	Оцінка повноти і вірогідності результатів розв'язання задачі. Формулювання відповіді на проблемне питання задачі

Розглянемо задачу.

Родина з трьох осіб планує подорож за маршрутом Київ — Львів — Одеса — Київ. Потрібно визначити, подорож яким транспортом — потягом «Укрзалізниці» чи автомобілем, наприклад Toyota Prius 1.8 (рис. 49.1), — буде дешевшою, й обґрунтувати вибір.

1. Постановка задачі. Визначити дані, які є в умові задачі. (*Кількість осіб; маршрут; транспортні засоби.*)



Рис. 49.1

Які дані повинні бути отримані? (*Вартість подорожі потягом та автомобілем для подальшого їх порівняння.*)

Які дані потрібно додатково знайти з інших джерел? (*Довжина маршруту; ціна на пальне; ціна одного квитка на потяг.*)

- Пошук інформації. Визначити ключові слова для пошуку потрібної інформації в Інтернеті. Вказати URL-адреси використаних сайтів.
- Побудова інформаційної моделі. Після визначення всіх даних можна побудувати інформаційну модель задачі.

Для задач, які потребують обчислень, доречно скласти математичну модель задачі. *Математична модель* — це спосіб подання інформаційної моделі, що відображає зв'язок різних параметрів об'єкта через математичні формули й поняття.

Математична модель задачі:

Дано: VK — вартість квитків на потяг на 1 людину, грн;
 VB — вартість 1 л бензину, грн;
 K — кількість пального на 100 км шляху, л;
 S_1, S_2, S_3 — відстані між містами, км.

Знайти: VP — вартість подорожі потягом, грн;
 VA — вартість подорожі автомобілем, грн;

Порівняти значення VP і VA , визначити менше з них;

Зв'язок між даними і результатом:

$$VP = VR * 3; VA = (S_1 + S_2 + S_3) / 100 * K;$$

Якщо $VP < VA$, то відповідь: «варто подорожувати потягом», якщо $VP > VA$, то відповідь: «варто подорожувати автомобілем», якщо $VP = VA$, то відповідь: «можна вибрати будь-який варіант».

- Визначення засобів опрацювання даних. Обчислення зручно виконувати в табличному процесорі Excel. Узагальнити результати розв'язання задачі можна у текстовому процесорі Word або в редакторі презентацій PowerPoint.
- Опрацювання даних. Створити електронну таблицю, в якій: вказати відстань між містами, норму витрат та ціну пального, кількість квитків та ціни на них; розрахувати суму, яку витратить сім'я для подорожі: потягом; автомобілем. Побудувати діаграму для порівняння отриманих результатів.
- Подання результатів. Для подальшого обговорення майбутньої подорожі створити звіт про результати розв'язання задачі, до якого можна включити:

- результати обчислень;
- діаграму, побудовану за даними обчислень, для порівняння результатів;
- зображення історичних місць, які варто відвідати (рис. 49.2);
- URL-адреси сайтів, на яких знайдено необхідні дані.

! Формат опису інтернет-ресурсу: Автор (якщо вказано). Заголовок статті. Назва сайту. URL-адреса сайту. Наприклад: Видатні українці. Спадщина. <http://www.spadshina.com/programs/vidatni-ukrayintsi/>

7. Аналіз результатів. На підставі звіту зробити висновки щодо найдешевшої подорожі.

Таким чином, для розв'язання компетентнісної задачі потрібно проаналізувати запропоновану ситуацію, поставити мету, спланувати результат, розробити алгоритм розв'язування задачі, обрати оптимальну для розв'язування конкретної задачі інформаційну технологію, проаналізувати результат і зробити висновки.



Рис. 49.2

Питання для самоперевірки



1. Що ви розумієте під поняттям «компетентнісна задача»?
2. З яких етапів складається процес розв'язування задачі?
3. Що таке інформаційна модель?
4. Чому при розв'язуванні різних задач можуть бути складені різні інформаційні моделі?
5. Як можна подати інформаційну модель задачі?
6. Що впливає на вибір інформаційної технології для опрацювання даних?

Вправа 49



- Розв'язати задачу про вибір транспорту для вашої подорожі по містах України.
- 1) Знайдіть дані, потрібні для розв'язання задачі: довжина маршруту Київ — Львів — Одеса — Київ; ціна на пальне; ціна проїзду однієї людини за даним маршрутом. У текстовому редакторі створіть перелік інтернет-ресурсів, на яких знайшли необхідну інформацію.

- 2) Створіть в табличному процесорі таблицю для розрахунку вартості подорожі відповідно до математичної моделі задачі:

	A	B	C	D	E	F	G
		вартість квитків на потяг на 1 людину, грн.	вартість подорожі потягом, грн.	вартість 1 л бензину, грн.	кількість пального на 100 км шляху, л.	відстань між містами, км	вартість подорожі автомобілем, грн.
1							
2	Київ-Львів		=B2*3				=F2/100*E2*D2
3	Львів-Одеса		=B3*3				=F3/100*E3*D3
4	Одеса-Київ		=B4*3				=F4/100*E4*D4
5			=SUM(C2:C4)				=SUM(G2:G4)
6	Відповідь:	F(C5<G5; "варті подорожувати потягом"); F(C5>G5; "варті подорожувати автомобілем"); можна вибрати будь-який варіант")					

- 3) Внесіть до таблиці дані, потрібні для розв'язання задачі.
- 4) Побудуйте діаграму, на якій відобразить вартості поїздки заданим маршрутом автомобілем і потягом.
- 5) Створіть звіт про результати розв'язання задачі.
- 6) Зробіть висновки щодо подорожі. Збережіть звіт у файлі Вправа 49.

Задачі для самостійного опрацювання



1. Тетянка звернулася до вас за допомогою. Дівчинка хоче запросити на свій день народження 9 друзів у дитяче кафе. Потрібно: 1) створити запрошення на свято; 2) обрати сценарій свята; 3) створити вітальний відеофайл; 4) скласти меню; 5) обчислити вартість проведення свята.
2. Учитель математики звернувся до вас із проханням розробити програму для знаходження коренів квадратного рівняння $ax^2 + bx + c = 0$. Потрібно: 1) створити програму (рис. 1); 2) створити електронну таблицю для перевірки правильності роботи програми за зразком (рис. 2); 3) у текстовому редакторі оформити звіт про виконання завдання за планом: постановка задачі; математична модель задачі; блок-схема алгоритму; програмний код; результати тестування програми.

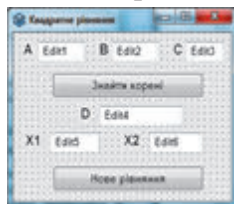


Рис. 1

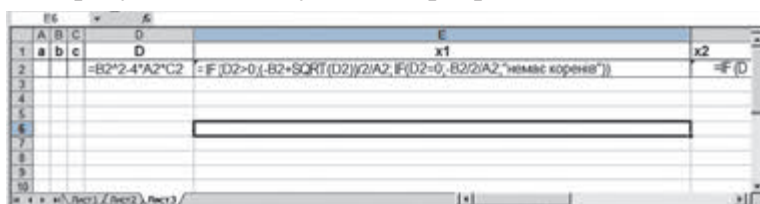


Рис. 2

3. Ваші однокласники після закінчення школи планують скласти ЗНО з історії України. Під час підготовки в учнів виникли проблеми з розпізнаванням історичних пам'яток за зображен-

ням. Потрібно створити довідник у вигляді таблиці, що містить назву об'єкта, його зображення та коротку історичну довідку.

Перелік історичних об'єктів:

- Софійський собор у Києві (I пол. XI ст.);
- Спасо-Преображенський собор у Чернігові (1036 р.);
- Вірменський собор у Львові (1363–1370);
- Замок Любарта, або Луцький замок (II пол. XIV ст.);
- Хотинська фортеця (II пол. XIII ст. — I пол. XIV ст.).

Комп'ютерне тестування

Виконайте тестове завдання 49 з автоматичною перевіркою на сайті interactive.ranok.com.ua.



§ 50. Виконання навчальних проєктів

Успіх у сучасному світі багато в чому визначається здатністю людини проєктувати своє життя: визначати далеку і найближчу перспективу, знаходити і залучати необхідні ресурси, намічати план дій та оцінювати досягнення поставлених цілей.

Проєкт являє собою самостійне доскональне вирішення певної проблеми з презентацією результатів роботи.

Навчальний проєкт — спільна навчально-пізнавальна, дослідницька, творча або ігрова діяльність учнів-партнерів, що має спільну мету, узгоджені методи, способи діяльності, спрямована на досягнення спільного результату з розв'язування певної проблеми.

Виконання проєкту на уроках інформатики дозволяє не тільки опанувати комп'ютерні технології, але й навчитися прийомам самостійної роботи від вибору теми до постановки цілей, від народження гіпотез до розробки алгоритму та створення готового електронного продукту.

Які бувають проєкти

Розглянемо деякі види навчальних проєктів.

- **Практико-орієнтований проєкт** націлений на соціальні інтереси учасників проєкту або зовнішнього замовника.
- **Дослідницький проєкт** за структурою нагадує наукове дослідження. Він включає обґрунтування актуальності обраної теми, визначення завдань дослідження, висунення гіпотези з подальшою її перевіркою, обговорення отриманих результатів. При цьому

використовуються методи сучасної науки: лабораторний експеримент, моделювання, соціологічне опитування й інші.

- **Інформаційний проект** спрямований на збір інформації про якийсь об'єкт або явище з метою її аналізу, узагальнення та подання для широкої аудиторії (публікація в ЗМІ, зокрема в Інтернеті).
- **Творчий проект** передбачає максимально вільний і нетрадиційний підхід до подання результатів. Це можуть бути літературні альманахи, театральні вистави, спортивні ігри, відеофільми тощо.
- **Рольово-ігровий проект.** Учасники беруть на себе соціальні ролі або ролі літературних чи історичних персонажів з метою відтворення через ігрові ситуації різних ситуацій соціальних або ділових відносин.

Робота над проектами проходить в кілька етапів:

1. Підготовка до роботи над проектом.
2. Вибір теми.
3. Постановка мети і завдання проекту.
4. Пошук інформації для проекту.
5. Виконання завдань проекту.
6. Підготовка до захисту проекту.
7. Презентація (захист) проектів.
8. Аналіз результатів проектної роботи.

Проект «Роль всесвітньої мережі Інтернет у нашому житті»

Розглянемо етапи роботи над проектом на прикладі проекту «Роль всесвітньої мережі Інтернет у нашому житті».

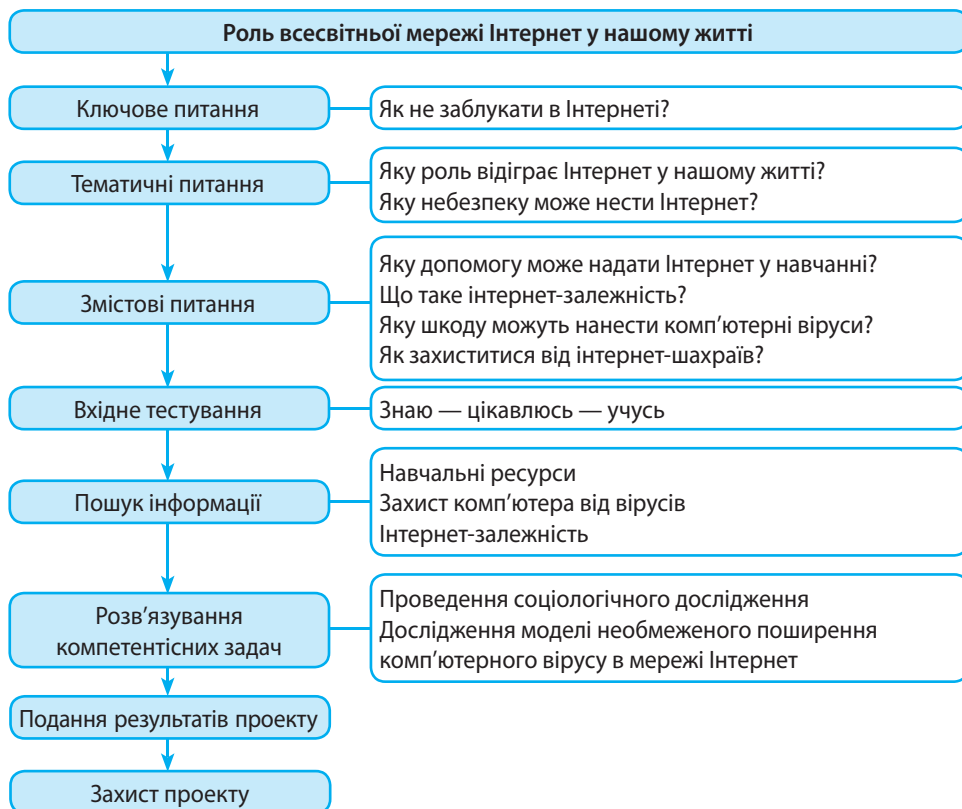
Мета проекту:

- дослідити, яке місце займає Інтернет у нашому житті;
- дізнатись, які небезпеки та можливості приховує в собі мережа;
- підготувати рекомендації для активних користувачів Інтернету.

Тривалість проекту: 4 уроки.

У ході виконання проекту ви проаналізуєте користь та можливу шкоду від Інтернету та знайдете відповідь на питання: як не заблукати у всесвітній мережі. Для пошуку шляхів вирішення проблеми потрібно визначити основні тематичні питання проекту. Вхідне тестування покаже, чи існують у вас проблеми в користуванні Інтернетом, а якщо існують, то які саме. Далі ви об'єднаєтесь в групи для роботи над питанням, яке вас зацікавило. Результати роботи варто оформити як презентацію, в якій підбиваються підсумки роботи групи учнів.

Для планування і контролю за виконанням завдань проекту доцільно створити інформаційну модель проекту у формі карти знань:



Реалізація проекту передбачає розв'язання низки компетентісних задач, які ви розглянете у вправі 50.

На етапі подання результатів проекту група учнів повинна створити звіт про результати виконання завдань проекту у вигляді текстового документа або презентації. Звіт повинний містити такі відомості:

- мета проекту;
- ключове і тематичні питання проекту;
- задачі, які розв'язувались у ході виконання проекту;
- URL-адреси сайтів, з яких використано необхідні дані;
- результати виконання завдань проекту;
- висновки, зроблені учасниками проекту на основі результатів виконання завдань;
- аналіз досягнутих результатів проекту.

Завершальним етапом роботи над проектом є його публічний захист, в ході якого відбувається презентація звіту про результати виконання завдань проекту. Під час захисту проекту потрібно оцінити рівень подання та оформлення результатів дослідження кожної групи за чітко прописаними критеріями оцінювання.

Робота над проектом дозволить вам спробувати свої сили, перевірити знання і оприлюднити досягнутий результат. Це діяльність, спрямована на розв'язування певної проблеми, що має важливе прикладне значення.

Питання для самоперевірки



1. Що таке навчальний проект?
2. Які бувають проекти?
3. З яких етапів складається робота над проектом?
4. Які відомості повинен містити звіт про результати виконання завдань проекту?
5. Яку допомогу ви отримуєте в Інтернеті під час роботи над проектом?
6. Які засоби захисту від інтернет-небезпеки ви використовуєте?

Вправа 50



- Розв'язати компетентнісні задачі проекту.
1. Дослідження моделі необмеженого розповсюдження комп'ютерного вірусу в мережі Інтернет.
 - 1) Учень отримав електронного листа, до якого був прикріплений файл, що містив комп'ютерний вірус ILOVEYOU. Коли учень відкрив файл, вірус видалив усі графічні та звукові файли на диску, а потім, використовуючи адресну книгу поштової програми, розіслав такого саме листа всім знайомим учня. Вірус ILOVEYOU заразив величезну кількість комп'ютерів по всьому світу за один день. Визначте, скільки комп'ютерів інфікував комп'ютер учня через 24 години, якщо середній коефіцієнт поширення вірусу дорівнює двом адресам за годину з кожного зараженого комп'ютера. Наявність засобів антивірусного захисту не враховувати.

Динаміку поширення комп'ютерного вірусу можна описати за допомогою математичної моделі.

Дано: K_0 — кількість заражених комп'ютерів у початковий момент часу ($K_0 = 1$);

R — коефіцієнт поширення вірусу ($R = 2$);

T — загальний час поширення вірусу ($T = 24$);
 dt — період поширення вірусу ($dt = 1$).

Знайти: K — кількість заражених комп'ютерів за час T .

Зв'язок між даними і результатом: $N = T / dt$; $K_{i+1} = K_i + K_i * R$, $i=1..N$,
де K_i , K_{i+1} — число заражених комп'ютерів в i -й та $(i+1)$ -й години після початку розповсюдження вірусу;

2) Розробіть інтерфейс проекту згідно з рисунком.

3) У програмному коді процедури Button1Click для кнопки запишіть оператори:

```
var K, R, T, dt, N, i: Integer;
begin
```

```
  K := StrToInt(Edit1.Text);
```

```
  R := StrToInt(Edit2.Text);
```

```
  T := StrToInt(Edit3.Text);
```

```
  dt := StrToInt(Edit4.Text);
```

```
  N := Round(T / dt); // N — кількість періодів поширення вірусу за час T
```

```
  For i := 1 to N do K := K + K * R;
```

```
  Edit5.Text := IntToStr(K);
```

```
end;
```

4) Перевірте роботу програми для початкових значень задачі. Зробіть висновки. Збережіть проєкт у папці Вправа 50.

II. Проведення соціологічного дослідження

5) Підготуйте опитувальник в тестовому процесорі або з використанням технологій Web-2.0:

- З ким ви спілкуєтесь в Інтернеті?
- Скільки часу на день ви проводите в мережі Інтернет?
- На яку небезпеку можна натрапити в Інтернеті?
- Які сайти ви відвідуєте?
- Чи повідомляєте ви інформацію про себе в Інтернеті?
- Що корисного є в Інтернеті для навчання?

6) Узагальніть результати опитування, занесіть дані до таблиці Excel та побудуйте діаграми, на яких відобразіть результати опитування. Збережіть файл у папці Вправа 50.



Комп'ютерне тестування

Виконайте тестове завдання 50 з автоматичною перевіркою на сайті interactive.ranok.com.ua.



Навчальні проекти

1. Тема. Цифрова інформація в нашому житті.

Предметно-змістовна область: інформатика.

Тип проекту: дослідницький.

Мета: дослідити основні ідеї, теоретичні й практичні принципи, що лежать в основі сучасних цифрових технологій.

Ключове питання: як інформатика допомагає сучасній людині?

Тематичні питання

1. Як закодувати звук?

Змістові питання:

- У чому різниця між аналоговою й цифровою інформацією?
- Як з аналогової інформації зробити цифрову і навпаки?
- Які існують пристрої та програмні засоби для опрацювання цифрового звуку?
- Які існують формати звукових файлів і в чому їх відмінність?

2. Чому існують різні палітри кольорів?

Змістові питання:

- Що таке RGB-палітра і CMYK-палітра, де вони використовуються?
- Чи існують інші палітри кольорів?
- Які є пристрої та програмні засоби для опрацювання зображень?
- Які існують формати графічних файлів і в чому їх відмінність?

3. Як навчити комп'ютер грамоті?

Змістові питання:

- Що таке таблиця кодів?
- Які існують таблиці кодів символної інформації?
- Які існують проблеми кодування національних алфавітів?
- У чому відмінність різних таблиць кодів і як розрізняються текстові файли, створені в різних кодуваннях?

4. Як оцінити інформацію?

Змістові питання:

- Як оцінити якість інформації?
- Як оцінити кількість інформації?
- Які існують одиниці вимірювання інформації?
- Як теорія інформації допомагає вирішувати побутові задачі?

Форма подання результатів проекту: презентація, буклет, відеоматеріали, вікі-стаття.

2. Тема. Видатні імена та відкриття в історії інформатики.

Предметно-змістовна область: інформатика.

Тип проекту: інформаційний.

Мета: дослідити історію розвитку обчислювальної техніки, вплив видатних особистостей в галузі інформаційних технологій на розвиток інформатики.

Ключове питання: до чого приводять відкриття?

Тематичні питання:

1. В яких трьох основних напрямках розвивається інформатика?

Змістове питання:

- Виділіть три нерозривно й істотно пов'язані частини інформатики: hardware, software, brainware та опишіть їх.

2. Як змусили комп'ютер думати?

Змістові питання:

- Які відкриття були зроблені в області обчислювальної техніки?
- Які відкриття були зроблені в ХХ ст. в галузі інформатики?

3. З якими іменами пов'язане становлення інформатики як науки?

Змістові питання:

- Кого називають засновником науки інформатики?
- З іменами яких людей пов'язаний розвиток інформаційно-комунікаційних технологій?

Форма подання результатів проекту: презентація, буклет, відеоматеріали, вікі-стаття.

3. Тема. Вибір комп'ютера.

Предметно-змістовна область: інформатика.

Тип проекту: рольово-ігровий.

Мета: зробити усвідомлений вибір моделі комп'ютера, згідно з наявними початковими умовами; змоделювати ситуацію практичного застосування матеріалу з теми «технічні характеристики складових комп'ютера»; навчитися культурі поведінки в ситуації продавець — покупець.

Ключове питання: Як вибрати комп'ютер?

Тематичне питання: Як визначити, виходячи з потреб користувача, найбільш підходящу конфігурацію ПК?

Змістові питання:

- Які технічні характеристики складових ПК цікавлять користувача?
- Як зробити розрахунок вартості вибраної комплектації?

Форма подання результатів проекту: рекламна презентація, буклет.

КОМП'ЮТЕРНИЙ СЛОВНИК

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

Алгоритм — послідовність команд для виконавця, яка чітко визначає, які дії та в якому порядку потрібно виконати для розв'язування певної задачі.

Аналоговий сигнал — неперервний сигнал, що може набувати будь-яких значень із певного діапазону.

Апаратне забезпечення — сукупність пристроїв і приладів, які забезпечують функціонування інформаційної системи.

Архіватор — програма, що здійснює упакування файлів в архів шляхом стиснення даних.

Архітектура комп'ютера — опис будови комп'ютера й принципів роботи його складових.

Аудіоредактор — програма для опрацювання звукових (аудіо) файлів.

Байт — набір із 8 послідовних бітів, який сприймається комп'ютером як єдине ціле.

Біт — найменша одиниця інформації, що позначається двійковою цифрою 1 або 0.

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

Глобальна змінна — змінна, доступ до якої можливий із будь-якої точки програми.

Двійкове кодування — подання даних за допомогою двійкових цифр — 0 і 1.

Декодування — перетворення закодованого повідомлення у форму, прийнятну для приймача.

Дискретний сигнал — сигнал, рівень якого може набувати лише окремих значень із певного набору.

Електронна таблиця — структура даних, поданих у вигляді таблиці, що забезпечує автоматичні обчислення за введеними в комірці формулами.

Звукова карта — пристрій комп'ютера, призначений для перетворення аналогового електричного сигналу на дискретний і навпаки.

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

Імпорт файла — відкриття у середовищі прикладної програми файла, створеного в іншому програмному середовищі.

Інсталяція — процес встановлення програмного забезпечення на комп'ютер користувача (зворотній процес — деінсталяція).

Інтерфейс користувача — засоби взаємодії користувача з програмою.

Код — набір символів або сигналів і правил їх використання для кодування повідомлень.

Кодування — перетворення повідомлення в зручний для передавання, зберігання й опрацювання вигляд.

Колонтитул — структурний елемент документа, що міститься у верхній або нижній частині сторінки.

Компілятор — спеціальна програма, яка перетворює текст програми, написаної мовою програмування високого рівня, на програму, подану машинною мовою, придатну для виконання на комп'ютері.

Конвертер — програма, призначена для перекодування медіафайлів з одного формату в інший.

Локальна змінна — змінна, доступ до якої можливий тільки всередині процедури, у якій вона оголошена.

Мова програмування — штучна мова, яка являє собою систему позначень і правил для запису алгоритмів у формі, придатній для подальшого їх виконання на комп'ютері.

Мультимедіа (від латин. *multum* — багато та *medium* — середовище) — це здебільшого інтерактивні (діалогові) системи, які використовують дані різних типів.

Операційна система — система програм та даних, за допомогою якої організується робота комп'ютера.

Пам'ять внутрішня — швидкодіюча пам'ять, розміщена на системній платі комп'ютера; поділяється на оперативну й постійну.

Пам'ять зовнішня (зовнішній запам'ятовуючий пристрій) — пам'ять на зовнішніх носіях даних (диски, флеш-пам'ять).

Пам'ять оперативна — внутрішня пам'ять комп'ютера, призначена для зберігання програм, даних, які опрацьовуються в даний момент.

Пам'ять постійна (постійний запам'ятовуючий пристрій) — пам'ять, призначена для зберігання програм і даних, що забезпечують роботу комп'ютера після ввімкнення.

Периферійне обладнання — зовнішні пристрої комп'ютера.

Посилання на комірку — адреса комірки, до якої додається ім'я аркуша, якщо посилання здійснюється на комірку з іншого аркуша.

Програмне забезпечення — сукупність програм, установлених на комп'ютері і призначених для забезпечення його роботи та розв'язування задач користувача.

Програмування — процес створення комп'ютерних програм.

Редагування документа — виправлення помилок, внесення змін до документа.

Система програмування — сукупність програм та інших засобів, які забезпечують створення комп'ютерних програм певною мовою програмування.

Стиль — спосіб форматування структурного елемента текстового документа.

Умовне форматування — автоматична зміна формату комірки на заданий, якщо значення даних в ній відповідають заданій умові.

Форматування документа — зміна зовнішнього вигляду документа.

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

Алфавітний покажчик

А

Адаптер 24
 Архів 30
 Архівування даних 30
 Архітектура комп'ютера 21
 Аудіоадаптер 22, 24, 63

В

Вираз 82
 Відеоадаптер 22, 24
 Відеоефект 70

Г

Гіперпосилання 51
 Графічні примітиви 211

Д

Двійкове кодування 10
 Деінсталяція програми 32
 Декодування 7
 Діаграма 103
 Діапазон комірок 77
 Довжина рядка 163, 164

Е

Елемент керування 114, 125, 126

З

Захоплення
 — аудіо 61
 — відео 63
 Зміст 51
 Зовнішній запам'ятовуючий пристрій 24

І

Імпорт 66
 Інсталяція програми 32
 Інтерфейс середовища програмування 110

К

Код 8
 — символу 10

— ASCII 10
 — KOI8-U 11
 — Юнікод 11

Кодек 59
 Кодування 7, 8
 Колонки 44
 Колонтитули 50
 Коментар 131
 Компілятор 114
 Компонент 109
 Конвертер 64
 Константа 139
 Контролер 22, 24
 Конфігурація комп'ютера 26

Л

Ліцензія 31, 32

М

Масштабування 221
 Метод 109, 126
 Мова програмування 108
 Мультимедійний пристрій 25

Н

Недрукований знак 45
 Носій даних 30

О

Об'єкт 109, 111
 Об'єктно-орієнтоване програмування 109
 Обробник події 119, 120
 Оператор
 — вибору 182
 — повного розгалуження 176
 — присвоєння 140
 — розгалуження 175
 — складений 176
 — циклу з параметром 199
 — циклу з передумовою 190

— циклу з післяумовою 195
Операції
— логічні 157
— відношення 157
П
Пам'ять
— внутрішня 24
— комп'ютера 23
— оперативна 24
— постійна 24
Перемикач 186
Перетворення типів 143, 158, 167
Подійно-орієнтоване програмування 117
Подія 117, 126
Полотно 211
Посилання на комірку
— абсолютне 78
— відносне 78
— мішане 78
Прапорець 187
Приймач 7
Принтер 25
Програмне забезпечення 28
— прикладне 30
— системне 29
— службове 30
Проект 113
Проектор 26
Проміжні підсумки 101
Процедура обробки події 117, 119
Процесор 22
Р
Редактор коду 117, 118
Режим структури 50
Розархівування 31
Розділ 49
Розкадрування 65

Розрив розділу 49
Розширений фільтр 100
С
Селектор 182
Середовище програмування 109
Системна плата 22
Список 37
— багаторівневий 38
— маркований 38
— нумерований 38
Стиль 48
Структура документа 49
Т
Таблиця 41
Тип величини 139
У
Умова
— проста 157
— складена 157
Упорядкування даних 94
Ф
Форматування
— носія даних 30
— даних 96
— списку 39
— умовне 96
Форма 110
Формат
— аудіофайла 60
— відеофайла 59
— файла текстового документа 37
Формула 45
Функція 82, 83
— логічна 82
— математична 86
— статистична 87
Ш
Шаблон документа 48

Зміст

Розділ 1. Кодування даних	
§ 1. Кодування та декодування повідомлень	7
§ 2. Двійкове кодування	10
Практична робота 1	13
Розділ 2. Апаратно-програмне забезпечення комп'ютера	
§ 3. Історія опрацювання інформаційних об'єктів	17
§ 4. Архітектура комп'ютера. Технічні характеристики складових комп'ютера	21
Практична робота 2	27
§ 5. Програмне забезпечення комп'ютера	28
Практична робота 3	34
Розділ 3. Опрацювання текстових даних	
§ 6. Списки в текстовому документі	37
§ 7. Таблиці в текстовому документі	41
§ 8. Опрацювання текстового документа, що містить різні об'єкти	44
§ 9. Оформлення документів. Структура складного текстового документа	48
Практична робота 4	54
Практична робота 5	55
Розділ 4. Опрацювання об'єктів мультимедіа	
§ 10. Знайомство з форматами відео- та аудіофайлів	59
§ 11. Програмне забезпечення для опрацювання об'єктів мультимедіа	61
§ 12. Створення аудіо- та відеофрагментів	65
§ 13. Опрацювання, збереження та публікація відеокліпу	69
Практична робота 6	73
Практична робота 7	74
Розділ 5. Технології опрацювання числових даних у середовищі табличного процесора	
§ 14. Типи посилань на комірки в Excel	77
Практична робота 8	81
§ 15. Логічні функції	82
§ 16. Математичні та статистичні функції	86
§ 17. Параметри сторінки. Друкування таблиці	89
§ 18. Упорядкування даних. Умовне форматування	94
Практична робота 9	98
§ 19. Розширені фільтри. Проміжні підсумки	100
§ 20. Діаграми і графіки	103
Практична робота 10	106
Розділ 6. Основи подійно- та об'єктно-орієнтованого програмування	
§ 21. Знайомство із середовищем програмування	109
§ 22. Створення найпростішого проекту	113

§ 23. Запис програми	116
§ 24. Основні компоненти програми	121
§ 25. Властивості і методи елементів керування	125
§ 26. Налаштування програмного коду	128
Практична робота 11	133
Практична робота 12	135

Розділ 7. Алгоритми роботи з об'єктами та величинами

§ 27. Величини та їхні властивості	139
§ 28. Елементи для введення даних	142
Практична робота 13	147
§ 29. Величини цілих типів	149
§ 30. Величини дійсних типів	153
§ 31. Величини логічного типу	157
§ 32. Величини символьного типу	160
§ 33. Величини рядкового типу	163
§ 34. Перетворення величин одного типу на інший	167
Практична робота 14	171

Розділ 8. Алгоритми з повтореннями та розгалуженнями

§ 35. Алгоритми з розгалуженнями	175
§ 36. Складання програм із розгалуженнями	178
§ 37. Поліваріантне розгалуження	182
§ 38. Компоненти вибору	186
§ 39. Алгоритми з повтореннями. Цикл із передумовою	190
§ 40. Алгоритми з повтореннями. Цикл із післяумовою	195
§ 41. Алгоритми з повтореннями. Цикл із параметром	199
§ 42. Складання циклічних алгоритмів опрацювання величин	203
Практична робота 15	207

Розділ 9. Графічне відображення даних

§ 43. Базові графічні примітиви	211
§ 44. Налаштування властивостей графічних примітивів	215
§ 45. Створення програм із графічним відображенням даних	220
§ 46. Графічні компоненти	223
§ 47. Виведення рисунків із зовнішніх файлів	227
§ 48. Розробка програм із графічними компонентами	231
Практична робота 16	235

Розділ 10. Розв'язування компетентнісних задач.

Виконання індивідуальних і групових навчальних проєктів

§ 49. Розв'язування компетентнісних задач	239
§ 50. Виконання навчальних проєктів	243
Навчальні проєкти	248
Комп'ютерний словник	251
Алфавітний покажчик	253

Відомості про користування підручником

№ з/п	Прізвище та ім'я учня / учениці	Навчальний рік	Стан підручника	
			на початку року	в кінці року
1				
2				
3				
4				
5				

Навчальне видання

*БОНДАРЕНКО Олена Олександрівна
 ЛАСТОВЕЦЬКИЙ Василь Васильович
 ПИЛИПЧУК Олександр Павлович
 ШЕСТОПАЛОВ Євген Анатолійович*

«ІНФОРМАТИКА»

підручник для 8 класу загальноосвітніх навчальних закладів

Рекомендовано Міністерством освіти і науки України

Видано за рахунок державних коштів. Продаж заборонено

Провідний редактор *І. Л. Морева*. Редактори *Л. А. Каюда, Ю. М. Миронова*.
 Художник *В. Д. Хорошенко*. Художнє оформлення *О. С. Юхтман*.
 Технічний редактор *В. І. Труфен*. Комп'ютерна верстка *О. В. Сміян, С. В. Яшиш*.
 Коректор *Н. В. Красна*.

Підписано до друку 22.06.2016. Формат 70×90/16.

Папір офсетний. Гарнітура Шкільна. Друк офсетний.

Ум. друк. арк. 18,67. Обл.-вид. арк. 24,27. Наклад 19181 прим. Зам. № 4506-16.

ТОВ Видавництво «Ранок».

Свідоцтво ДК № 3322 від 26.11.2008. 61071 Харків, вул. Кібальчича, 27, к. 135.

Адреса редакції: 61145 Харків, вул. Космічна, 21-а, 7 поверх.

E-mail: office@ranok.com.ua. Тел. (057) 701-11-22, 719-48-65, тел./факс (057) 719-58-67.

www.ranok.com.ua

Під час підготовки видання були використані матеріали із сайтів:
 freepik.com, freeimages.com, morguefile.com, pixabay.com, uk.wikipedia.org

Надруковано у друкарні ТОВ «Тріада Принт»
 м. Харків, вул. Киргизька, 19. Тел. (057) 703-12-21. E-mail: sale@triada.kharkov.ua



Н. Вірт:

Ключ до таємниць комп'ютерів — у гармонії математики, інженерії та програмування.

Мова програмування Pascal була розроблена у 1971 р. Ніклаусом Віртом, професором комп'ютерних наук Швейцарської вищої технічної школи (Цюріх), як мова для навчання майбутніх програмістів. Мову названо на честь видатного французького математика, фізика, літератора і філософа Блеза Паскаля (1623–1662 рр.) — винахідника механічного калькулятора паскаліни.

Delphi — об'єктно-орієнтоване середовище для візуальної розробки програм, засноване на мові Object Pascal. Назву Delphi було вибрано з огляду на прислів'я, відоме в Стародавній Греції: «Хочеш поговорити з оракулом — йди в Дельфи». Вільним аналогом середовища програмування Delphi є **Lazarus**, заснований на компіляторі Free Pascal.

У світі існує близько 8500 мов програмування, але затребуваних не так вже й багато. Якщо ви збираєтесь у майбутньому працювати у сфері ІТ-технологій, вам не завадить ознайомитися з переліком найпопулярніших сучасних мов програмування.



Блез Паскаль



Паскаліна, створена Б. Паскалем у 1642 р.



Символом IDE Lazarus є гепард



Java — об'єктно-орієнтована мова програмування, розроблена компанією Sun Microsystems. Свою назву мова отримала на честь марки кави Java. На мові Java працює близько 3 млрд мобільних телефонів. У світі налічується близько 9 млн Java-програмістів.



C/C++ — це універсальний вибір для розробки настільного програмного забезпечення, комп'ютерних ігор, а також застосунків, що вимагають великого обсягу пам'яті для роботи.



C# (Сі Шарп) — популярна мова для розробки застосунків для Windows. Для розробки ігор Unity 3D використовує C# як одну з основних мов.



Objective C стане вам у пригоді, якщо ви збираєтесь зайнятися розробкою застосунків для Apple Mac OS X, а також для айфонів, айпадів.



PHP — інструмент для створення сучасних веб-застосунків. На PHP розроблено більшість сайтів, орієнтованих на великий обсяг даних.



Python. Веб-застосунки, статистика, аналіз даних, призначені для користувача інтерфейси — для кожної задачі в Python знайдеться відповідний фреймворк (програмне забезпечення, що полегшує розробку й об'єднання різних компонентів великого програмного проекту).



JavaScript використовує будь-який сучасний сайт. Це ключова мова для створення інтерактивних сайтів або побудови призначених для користувача інтерфейсів.



Visual Basic.NET — це об'єктно-орієнтована мова, що дозволяє створювати широкий спектр програмного забезпечення: графічні програми, ігрові програми, складні високонавантажені веб-сервіси.



Ruby on Rails набуває популярності серед компаній-початківців, оскільки є простішою порівняно з Java або Visual Basic.NET для швидкої розробки веб-застосунків.

Марк Цукерберг,
керівник компанії Facebook Inc.:

*Я вважаю, що в майбутньому всі,
а не тільки програмісти, будуть пов'язані
з елементами програмування.*

