



О. Ю. Корчук, В. І. Косяк

ОСНОВИ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

unit Unit1;

```
var a,b:real;code:integer;  
function tangens(x:real):real;  
begin
```

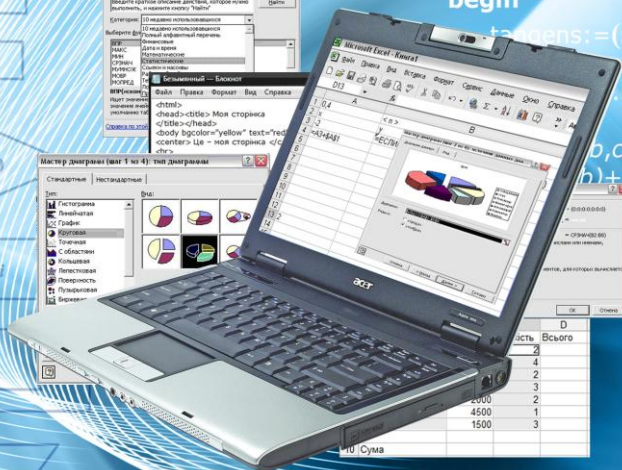
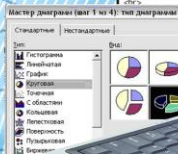
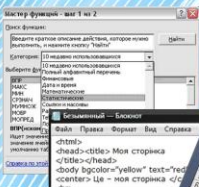
```
tangens:=(sin(x)/cos(x));
```

But

```
b,code
```

```
(b)+1/t
```

```
text
```



```
procedure TForm1.Button1.Click(Sender: TObject);  
var a, b, P, S:real;s1,s2:string;code:integer;
```

```
begin
```

```
val(edit1.text,a,code); val(edit2.text,b,code);
```

```
P:= 2*(a + b);S:= a*b; str(P:8:2,s1); str(S:8:2,s2);
```

```
edit3.text:=s1; edit4.text:=s2;
```

```
end;
```

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний авіаційний університет

О.Ю.Корчук, В.І.Косяк

ОСНОВИ ІНФОРМАТИКИ
ТА ОБЧИСЛЮВАЛЬНОЇ
ТЕХНІКИ

Навчальний посібник

Київ 2018

УДК 004(075.8)
ББК 3 810я7
К 703

Рецензенти: Рильцев Є.В. - доктор фіз.-мат. наук, професор (Міжрегіональна академія управління персоналом).

Єфіменко В.В. - кандидат пед. наук, доцент (Національний педагогічний університет ім. М.Драгоманова).

Каленик О.О. - кандидат фіз.-мат. наук, старший викладач (Київський національний університет ім. Т.Шевченка).

*Затверджено методично-редакційною радою
Національного авіаційного університету (протокол
№13 від 20.12.2017 р.).*

Корчук О.Ю., Косяк В.І.

К 703 Основи інформатики та обчислювальної техніки:
навч. посібник / О.Ю.Корчук, В.І.Косяк. – К.: НАУ, 2018. –
160 с.

Містить необхідні відомості про основні поняття інформатики.

Для іноземних студентів підготовчого відділення, які починають вивчення основ інформатики та обчислювальної техніки українською мовою.

**УДК 004(075.8)
ББК 3 810я7**

ISBN 978-966-932-083-4

© Корчук О.Ю., Косяк В.І., 2018
© НАУ, 2018

ЗМІСТ

ПЕРЕДМОВА	5
ВСТУП	6
Модуль 1. ОСНОВНІ ПОНЯТТЯ ІНФОРМАТИКИ. ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ. КОМП'ЮТЕРНІ МЕРЕЖІ	7
1.1. Кодування, збереження та оброблення інформації в комп'ютері	7
1.1.1. Поняття інформації	7
1.1.2. Ознайомлення з персональним комп'ютером	8
1.1.3. Кодування інформації	16
1.2. Операційна система	18
1.3. Основи роботи з дисками	20
1.4. Текстовий редактор	22
1.5. Графічний редактор	33
1.6. Електронні таблиці	40
ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 1	55
1.7. Системи керування базами даних	57
1.8. Комп'ютерна мережа. Глобальна мережа Інтернет. Створення web-сторінок	80
ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 2	85
Модуль 2. ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ	95
2.1. Типові структури алгоритмів	95
ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 3	101
2.2. Основні поняття програмування	104
2.3. Конструкції та оператори мови програмування	105
2.3.1. Поняття про візуальне програмування	105
2.3.2. Поняття проекту в об'єктно-орієнтованому програмуванні	107
2.3.3. Основні елементи мови програмування Object Pascal	121
2.3.4. Складання лінійних програм	127
2.4. Програмування розгалужень	131
ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 4	136
2.5. Програмування циклів	136
2.6. Робота з масивами	140
2.7. Процедури і функції	145

2.8. Робота з рядками.....	148
2.9. Засоби роботи з файлами та графікою.....	150
2.9.1. Робота з файлами	150
2.9.2. Поняття про графіку у візуальному програмуванні.....	154
ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 5	156
ПІСЛЯМОВА	158
<i>СПИСОК ЛІТЕРАТУРИ</i>	159

ПЕРЕДМОВА

Мета викладання дисципліни «Основи інформатики та обчислювальної техніки» на підготовчому відділенні центру міжнародної освіти – підготовка студентів до успішного навчання у вищих навчальних закладах України. Ця мета досягається шляхом:

- поглиблення та систематизації знань, здобутих на батьківщині;
- усунення прогалин шкільної освіти, зумовлених відмінностями у національних та українській програмах з інформатики;
- формування теоретичної бази знань, необхідної студентам для вивчення дисципліни у вищому навчальному закладі;
- засвоєння практичних навичок використання засобів комп'ютерної техніки і сучасних інформаційних технологій у науково-пізнавальній діяльності студентів;
- вивчення термінології дисципліни українською мовою, засвоєння лексичного матеріалу і конструкцій, властивих науковому стилю мовлення.

Необхідність створення посібника зумовлено недостатньою кількістю навчальної літератури з основ інформатики та обчислювальної техніки для іноземних студентів українською мовою.

Зміст розділів відповідає програмі курсу і забезпечує потрібну повноту знань з дисципліни. Матеріал чітко структурований, викладений послідовно, у відповідності до програми вивчення української мови. Стиль викладення відрізняється від аналогічних видань, призначених для україномовних студентів. Використовуються мовні конструкції, зрозумілі для тих, хто тільки починає вивчення української мови як іноземної.

За законами логіко-сислової побудови, способами подання навчальної інформації текстовий матеріал посібника близький до того, з яким студенти працюватимуть на першому курсі. Досягнуто компромісу між намаганням уніфікації та спрощення лексичних форм викладення матеріалу і бажанням забезпечення їх повноти та логічної завершеності.

ВСТУП

Умовно весь матеріал курсу основ інформатики та обчислювальної техніки можна розподілити так:

- комп'ютер як об'єкт вивчення;
- комп'ютер як інструмент для оброблення, пошуку та збереження різних видів інформації;
- комп'ютер як інструмент для створення програмних продуктів.

У посібнику розглядаються основні поняття інформатики (інформація, інформаційні процеси, комп'ютер), операційна система, основи роботи з дисками, текстовим (MS Word) та графічним (Paint) редакторами, електронними таблицями (MS Excel) і системою керування базами даних (MS Access), поняття про комп'ютерні мережі, Інтернет, створення web-сторінок за допомогою мови HTML, а також основи алгоритмізації та програмування в середовищі Delphi (складання лінійних програм, програмування розгалужень, циклів, робота з масивами, рядками, засоби роботи з графікою та файлами).

Навчальний матеріал структурований за модульним принципом і складається з двох навчальних модулів. Наприкінці кожного наведено запитання або завдання, які можна використовувати під час як самостійної роботи студентів, так і роботи з викладачем. Посібник містить завдання для п'яти контрольних робіт у 20 варіантах, а також багато ілюстрацій, які дозволяють детально прослідкувати хід виконання навчальних завдань-прикладів та практичних робіт.

Навчальний посібник може бути корисним для студентів та викладачів вищих навчальних закладів, учнів та вчителів загальноосвітніх шкіл.

Модуль 1

ОСНОВНІ ПОНЯТТЯ ІНФОРМАТИКИ. ПРИКЛАДНЕ ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ. КОМП'ЮТЕРНІ МЕРЕЖІ

1.1. Кодування, збереження та оброблення інформації в комп'ютері

1.1.1. Поняття інформації

Інформатика – це наука про інформацію.

Інформація – це відомості про навколишній світ. Є три види інформації: вхідна, вихідна, внутрішня (рис.1.1).



Рис. 1.1. Види інформації

Інформаційні процеси – це дії, які виконують з інформацією (рис.1.2):



Рис. 1.2. Інформаційні процеси

Форми подання інформації

1. **Числова:** цифра, число, вираз та ін.

2. **Текстова:** художня (роман), наукова (стаття), технічна (інструкція) та ін.
3. **Графічна:** картина, креслення, графік та ін.
4. **Звукова:** мовлення, музика.
5. **Комбінована:** фільм, відеокліп, вистава та ін.

Носій інформації – це середовище, яке містить інформацію (наприклад, книга, оптичний диск, USB flash накопичувач).

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Що таке інформатика? Що таке інформація?
2. Які ви знаєте види інформації?
3. Які є інформаційні процеси?
4. Які є форми подання інформації?
5. Що таке носій інформації? Наведіть приклад.

1.1.2. Ознайомлення з персональним комп'ютером

Комп'ютер – це пристрій для роботи з інформацією (рис.1.3).



Рис. 1.3. Основні пристрої комп'ютера

Розглянемо компоненти системного блока (рис.1.4):

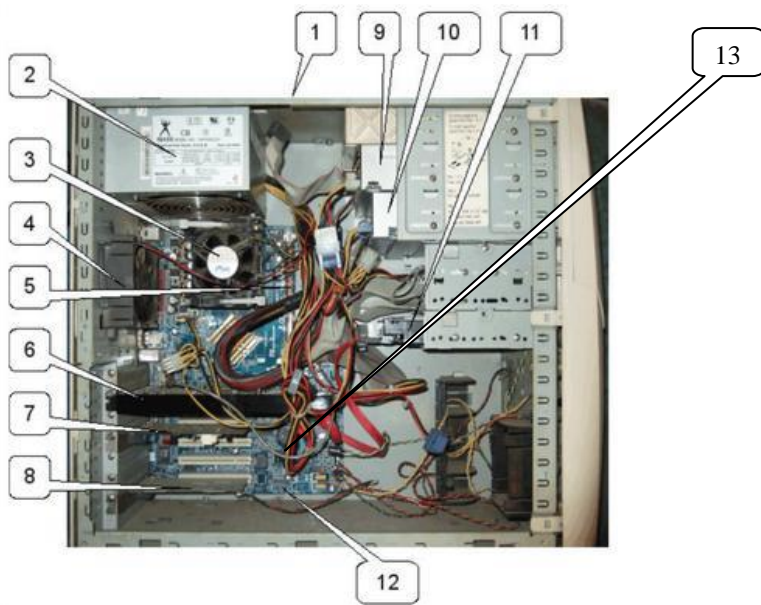


Рис. 1.4. Системний блок у розрізі

1 – Корпус.

2 – Блок живлення – пристрій для забезпечення електричним живленням усіх пристроїв системного блока (рис.1.5).

3 – Процесор – це мікросхема для обробки інформації і управління роботою всього комп'ютера (рис.1.6).



Рис. 1.5. Блок живлення



Рис. 1.6. Процесор

4 – Вентилятор – це пристрій для зниження температури всередині системного блока (рис.1.7).



Рис. 1.7. Вентилятор



Рис. 1.8. Оперативна пам'ять

5 – Оперативна пам'ять (RAM – Random Access Memory) – це пристрій для зберігання інформації, яка обробляється в поточний момент часу (рис.1.8).

Оперативна пам'ять – це пристрій **внутрішньої пам'яті** комп'ютера.

Пристроями **внутрішньої пам'яті** також є:

постійна пам'ять (ROM – Read Only Memory) – це пам'ять для зберігання інформації, яка потрібна постійно. Зберігає інформацію і тоді, коли комп'ютер вимкнено. Інформація записується у постійну пам'ять на заводі (рис.1.9);

кеш-пам'ять (cache) – це пам'ять для зберігання інформації, яка використовується дуже часто (рис.1.10).



Рис. 1.9. Постійна пам'ять



Рис. 1.10. Кеш-пам'ять

Контролери – це мікросхеми, які керують роботою зовнішніх пристроїв комп'ютера (відеокарта, мережева карта, звукова карта та ін.).

6 – Відеокарта – це пристрій для оброблення та виведення на екран графічної інформації (рис.1.11).

7 – Модем – це пристрій зв'язку (використовується для підключення до мережі Інтернет) (рис.1.12 – 1.14).



Рис. 1.11. Відеокарта

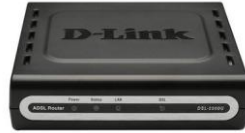


Рис. 1.12. Зовнішній модем



Рис. 1.13. Внутрішній модем



Рис. 1.14. Безпроводний модем

8 – Мережева карта – це пристрій для підключення до локальної чи глобальної мережі (рис.1.15).

9, 10 – CD/DVD привід – це пристрій для тривалого зберігання інформації (рис. 1.16, 1.17).



Рис. 1.15. Мережева карта



Рис. 1.16. CD/DVD привід

11 – Жорсткий диск (вінчестер) – HDD (Hard Disk Drive) – це пристрій для зберігання інформації (рис. 1.18).



Рис. 1.17. Оптичний диск



Рис. 1.18. Жорсткий диск

Для зберігання інформації також використовується **USB флеш-накопичувач** (рис.1.19).

12 – Материнська плата (*MotherBoard*) – це пристрій, який з'єднує інші пристрої комп'ютера (рис.1.20).



Рис. 1.19. USB флеш-накопичувач



Рис. 1.20. Материнська плата

Жорсткий диск (вінчестер), CD/DVD привід, USB флеш накопичувач – це пристрої **зовнішньої пам'яті** комп'ютера.

13 – Шина – це набір паралельних проводів, кожний з яких передає електричний сигнал. Шина використовується для обміну інформацією між пристроями комп'ютера.

Для **введення інформації** використовуються: **клавіатура** (рис.1.21), **сканер** (рис.1.22 – 1.24), **миша** (рис.1.25), **трекбол** (рис.1.26), **джойстик** (рис.1.27), **кермо** (рис.1.28), **сенсорний екран** (рис.1.29), **електронний планшет** (рис.1.30).



Рис.1.21. Клавіатура



Рис.1.22. Настільний (планшетний) сканер



Рис.1.23. Ручний сканер



Рис.1.24. Настільний сканер



Рис.1.25. Миша



Рис.1.26. Трекбол



Рис.1.27. Джойстик



Рис.1.28. Кермо



Рис.1.29. Сенсорний екран



Рис.1.30. Планшет

Для виведення інформації на екран використовується *монітор* (рис.1.31).

Для виведення інформації на папір використовується *принтер* (рис.1.32-1.34).

Для виведення звукової інформації використовуються *звукові колонки* та *наушники* (рис.1.35, 1.36).



Рис. 1.31. Монітор



Рис. 1.32. Лазерний принтер



Рис. 1.33. Матричний принтер



Рис. 1.34. Струменевий принтер



Рис. 1.35. Звукові колонки



Рис. 1.36. Наушники

Будову комп'ютера можна подати у вигляді схеми (рис.1.37):

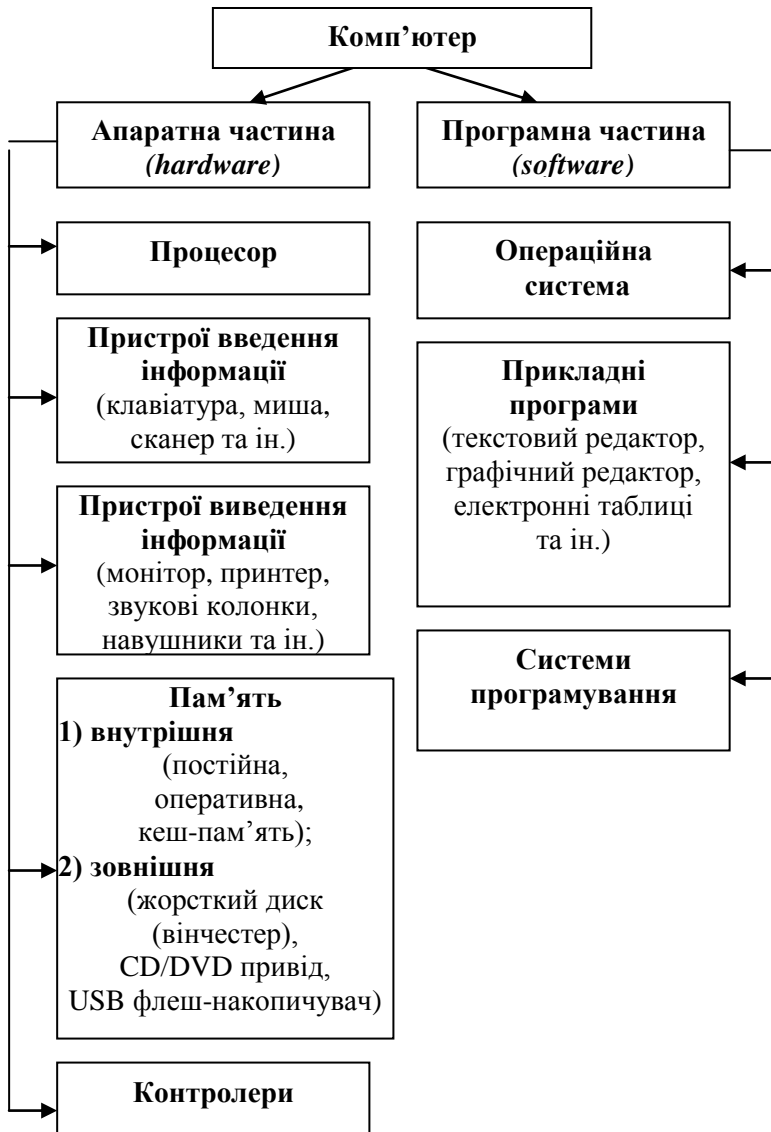


Рис. 1.37. Будова комп'ютера

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Що таке комп'ютер? Які є основні пристрої комп'ютера?
2. Що таке процесор? Для чого потрібен вентилятор?
3. Які ви знаєте пристрої введення інформації?
4. Які є пристрої виведення інформації? Наведіть приклад.
5. Які ви знаєте пристрої внутрішньої пам'яті?
6. Які є пристрої зовнішньої пам'яті? Наведіть приклад.
7. Що таке шина? Для чого вона потрібна?

1.1.3. Кодування інформації

Комп'ютер зберігає та обробляє інформацію у вигляді електричних сигналів двох типів. Один з них позначають цифрою **1**, а другий – цифрою **0** (*двійковий код*). Найменша кількість інформації – це одне з двох значень – 0 або 1. Вона називається *бітом* (*binary digit – двійкова цифра*).

Одиниці вимірювання інформації

1 байт (byte) = 8 біт (*байт* – це послідовність з восьми нулів та одиниць).

1 кілобайт (кб) (kilobyte) = 2^{10} байт = 1024 байт;

1 Мегабайт (Мб) (Megabyte) = 2^{10} кбайт = 1024 кбайт;

1 Гігабайт (Гб) (Gigabyte) = 2^{10} Мбайт = 1024 Мбайт;

1 Терабайт (Тб) (Terabyte) = 2^{10} Гбайт = 1024 Гбайт;

1 Петабайт (Пб) (Petabyte) = 2^{10} Тбайт = 1024 Тбайт;

1 Ексабайт (Еб) (Exabyte) = 2^{10} Пбайт = 1024 Пбайт;

1 Дзетабайт (Дб) (Zettabyte) = 2^{10} Ебайт = 1024 Ебайт;

1 Йотабайт (Йб) (Yottabyte) = 2^{10} Дбайт = 1024 Дбайт.

Для кодування *алфавітно-цифрової* інформації використовується **таблиця комп'ютерних кодів ASCII** (*American Standard Code for Information Interchange – американський стандартний код для обміну інформацією*).

Для кодування одного символу використовується *1 байт*. Усього існує 256 різних байтів. Їх нумерують цифрами від 0 до 255:

0	00000000
1	00000001
2	00000010
...	
50	00011010
...	
255	11111111

Байти з номерами *від 0 до 31* використовуються для кодування службових символів, які не використовуються у текстових документах.

Байти з номерами *від 32 до 127* використовуються для кодування цифр, літер англійського алфавіту і службових символів, які використовуються у текстових документах.

Байти з номерами *від 128 до 255* використовуються для національних алфавітів та символів псевдографіки.

Наприклад:

<i>Номер коду</i>	<i>Символ</i>	<i>Номер коду</i>	<i>Символ</i>
32	пропуск	58	:
33	!	64	@
43	+	65	A
47	/	66	B
48	0	67	C
49	1	90	Z
57	9	97	a

Кодування графічної інформації

У разі кодування *чорно-білого зображення* кожній точці відповідає *один байт*. Він означає *відтінок сірого кольору*. У разі кодування *кольорового зображення* кожній точці відповідають *три байти*. Вони означають яскравість трьох основних кольорів: *червоного (Red)*, *зеленого (Green)* і *синього (Blue)* – від 0 до 255. У комп'ютерній графіці таку палітру називають **RGB-палітрою**, де *червоний* колір має код (255, 0, 0), *зелений* – (0, 255, 0), *синій* – (0, 0, 255). Всього різних кольорів існує 256·256·256, тобто більше ніж 16 мільйонів.

Кодування звуку

Звук має дві основні характеристики: *силу* і *тривалість*. Силу звуку градуюють на 256 значень. У кожний момент часу звуку відповідає значення деякого байта.

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Як кодується інформація в комп'ютері? Що таке один біт?
2. Які ви знаєте одиниці вимірювання інформації?
3. Як кодується інформація з допомогою таблиці ASCII-кодів?
4. Скільки бітів потрібно для кодування одного символу алфавіту? Як кодується графічна та звукова інформація?

1.2. Операційна система

Операційна система – це сукупність програм, які *керують* роботою комп'ютера (рис. 1.38).



Рис.1.38. Складові операційної системи

Ядро – це *командний інтерпретатор*, який перекладає команди з мови програм у машинні коди. **Драйвери** – це спеціальні програми для керування пристроями комп'ютера. **Інтерфейс** – це зручна *оболонка*, з якою спілкується користувач. **Файлова система** – це сукупність файлів та правила їх організації. Об'єкти файлової системи – це *диски, файли та папки*.

Комп'ютер звичайно має один **жорсткий диск (HDD – Hard Disk Drive)**. Для ефективного використання дискової пам'яті на ньому створюють *розділи* (їх теж називають *дисками*) і позначають латинськими літерами **C:**, **D:** і т.д.

Файл (file) – це порція інформації на диску, яка має ім'я (<ім'я>.<розширення>).

Наприклад: $\underbrace{\text{file1}}_{\text{ім'я}} \underbrace{\text{.txt}}_{\text{розширення}}$

Позначення для складання імен файлів:

* - будь-яка кількість символів;

? - тільки один символ.

Наприклад: *.doc, ?ab.txt.

Розширення визначає тип файла. **Наприклад:**

<i>Розширення</i>	<i>Тип файла</i>
.doc	Файл документа
.txt	Текстовий файл
.exe	Виконувана програма
.pas	Вихідний текст програми на Паскалі
.mdb	Дані СКБД MS ACCESS
.xls	Файл електронної таблиці MS Excel
.sys	Драйвер керування пристроєм
.bmp	Точкове зображення
.com	Командний файл

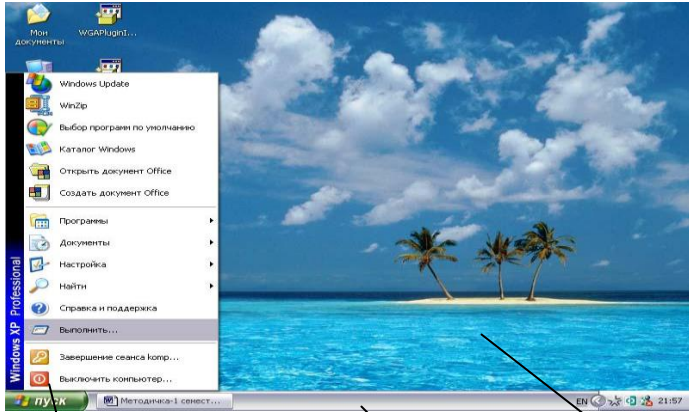
Папка – це ділянка диска, де реєструються відомості про файли: імена, розширення, розмір, дата редагування або створення і т.ін., а також назви вкладених папок.

Повна назва (абсолютне ім'я) файла має вигляд:

дисковод : \ шлях до файла \ назва файла

Наприклад: $E: \underbrace{\text{folder1}}_{\text{диск}} \underbrace{\text{folder2}}_{\text{папка1}} \underbrace{\text{file1.txt}}_{\text{папка2 файл}}$

На рис.1.39 зображено вигляд екрана операційної системи *Microsoft Windows XP Professional* після натиснення кнопки «Пуск».



Головне меню
(Main menu)

Панель завдань
(Status Bar)

Робочий стіл
(Desktop)

Рис. 1.39. Microsoft Windows XP Professional

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Що таке операційна система? Які обов'язкові частини має кожна операційна система?
2. Що таке ядро, інтерфейс, драйвери операційної системи?
3. Що таке файлова система? Що таке файл, папка?
4. Які ви знаєте стандартні розширення файлів?

1.3. Основи роботи з дисками

Форматування – це розбиття поверхонь жорсткого магнітного диска на *доріжки (треки)*. Кожна з них радіальними лініями розбивається на *сектори* (рис.1.40). Уздовж кожного сектора розміщуються *намагнічені* та *не намагнічені* ділянки. *Намагнічена* ділянка відповідає *двійковому значенню 1*, *не намагнічена* – значенню *0*. *Сектор містить 512 байт інформації*.

Кластер – це група логічно послідовних секторів на диску. Файл записується на диск у вигляді фрагментів, кожний фрагмент записується в один кластер. Кожний кластер має свій номер.

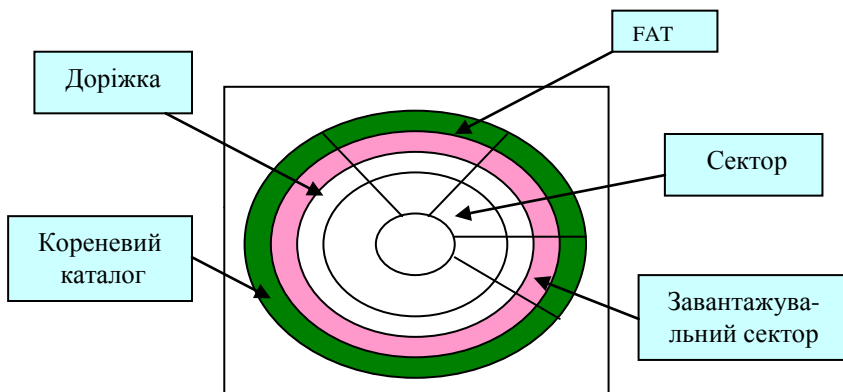


Рис. 1.40. Структура диска

Таблиця розміщення файлів (FAT-таблиця – File Allocation Table) – це список кластерів диска, за яким операційна система знаходить всі частини необхідного файлу. FAT-таблиця міститься на початку диска. За таблицею розміщена її копія, а далі – опис кореневої папки. Решту місця на диску призначено для збереження файлів. **Під час форматування диска вся інформація, яка на ньому записана, знищується.**

Сканування диска – це перевірка диска на наявність дефектів. **Дефрагментація диска** – це впорядкування кластерів на диску, видалення порожніх кластерів.

Архівація – це процес перекодування файлів з метою зменшення їх розміру. Архівацію виконують спеціальні програми – архіватори. Найбільш поширені програми-архіватори: *WinRar, WinZip*.

Комп'ютерний вірус – це спеціально створена програма, яка може записувати свої копії в інші комп'ютерні програми, системну ділянку диска і т.д. Вірус знаходить та заражає інші програми, а також сам може виконати якісь шкідливі дії. **Антивірусні програми** – це програми, які знаходять віруси та лікують заражені файли і диски.

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Що таке форматування?
2. Що таке кластер? Що таке FAT-таблиця?
3. Що таке сканування та дефрагментація диска?
4. Що таке архівація файлів? Які є програми-архіватори?
5. Що таке вірус? Які ви знаєте антивірусні програми?

1.4. Текстовий редактор

Текстовий редактор – це програма, призначена для оброблення текстових документів. **Текстовий процесор** – це текстовий редактор, який має розширені можливості. **Microsoft Word** – це текстовий процесор. Запустити програму можна за допомогою команди:

Пуск → Програми → Microsoft Word.

Після запуску на екрані побачимо вікно програми (рис. 1.41).

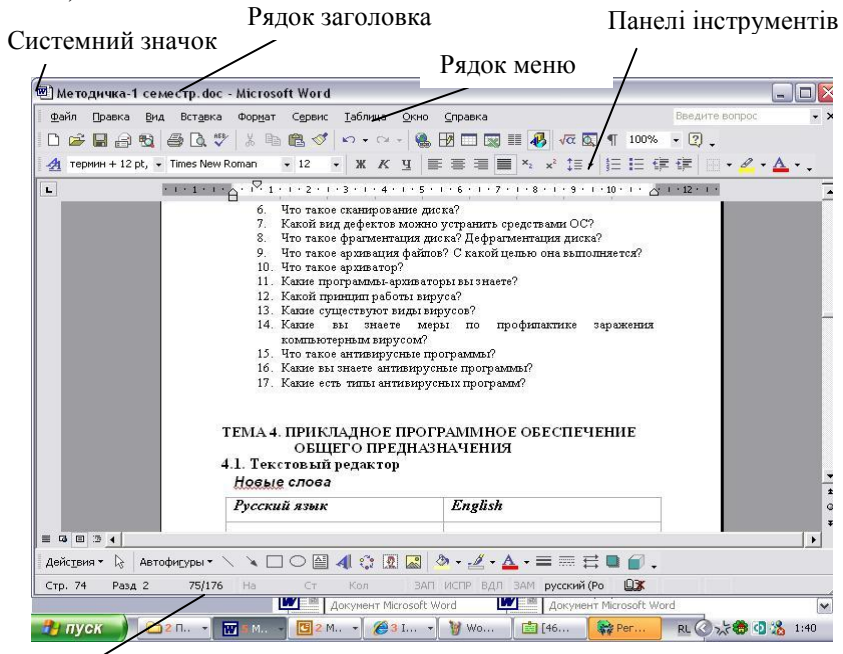


Рис. 1.41. Вікно програми Microsoft Word

Рядок заголовка – це верхній рядок екрана. Він містить ім'я документа, ім'я програми і чотири кнопки. Ліва кнопка (рис.1.42) викликає меню, яке керує вікном програми:

Кнопка виклику
меню керування
вікном програми

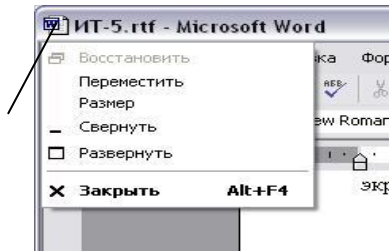


Рис. 1.42. Меню керування вікном програми

Призначення кнопок, які містяться справа (рис.1.43):

- 1 – **згорнути** вікно до піктограми;
- 2 – **розгорнути** вікно на весь екран;
- 3 – **закрити** вікно.



Рис. 1.43. Кнопки керування вікном програми

Головне меню містить пункти:

- Файл** – робота з файлами документів;
- Правка** – редагування документів;
- Вид** – перегляд документів;
- Вставка** – вставка об'єктів;
- Формат** – форматування документів;
- Сервис** – сервісні функції (налаштування Word);
- Таблиця** – робота з таблицями;
- Окно** – робота з вікнами документів;
- Справка** – отримання довідкової інформації про Word.

Символи клавіатури вводяться в позицію текстового курсору (мерехтливий вертикальний штрих) в режимі **вставки** або **заміни**, які перемикає клавіша **Insert**.

Рядок стану містить інформацію про активне вікно:

Стр *i* – курсор міститься на *i*-й сторінці;

Разд *b* – курсор міститься в *b*-му розділі;

k/p – від початку документа до курсору *k* сторінок, весь документ містить *p* сторінок;

на 5 см – відстань від курсору до верхньої межі тексту;

Ст *t* – курсор міститься в *t*-му рядку сторінки;

Кол *W* – курсор міститься в *W*-му стовпці.

Панель інструментів – це ряд кнопок, дії яких аналогічні командам головного меню. Для вибору панелі треба виконати команду *Вид* → *Панели инструментов* (або *Сервис* → *Настройка* → *Панели инструментов*).

Форматування тексту – це задання його параметрів. Перед форматуванням текст потрібно *виділити*. Колір виділеного фрагмента *інвертується*. Щоб виділити весь текст документа, треба виконати: *Правка* → *Выделить все*.

Зміна параметрів шрифту: *Формат* → *Шрифт* (рис.1.44):

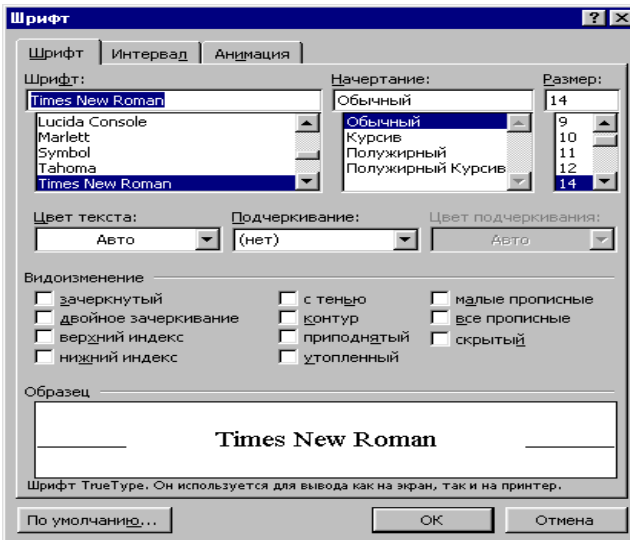


Рис.1.44. Параметры шрифту

Здати параметри шрифту можна також за допомогою панелі інструментів «*Форматирование*» (рис. 1.45):

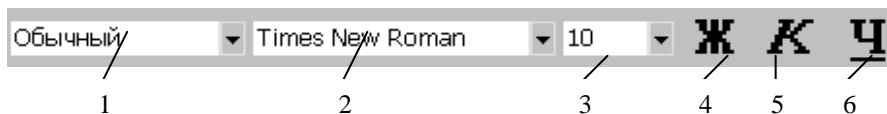


Рис.1.45. Панель інструментів «*Форматирование*»

- | | |
|-------------------------|---------------------------|
| 1 – стиль форматування; | 4 – жирний шрифт; |
| 2 – тип шрифту; | 5 – <i>курсив</i> ; |
| 3 – розмір шрифту; | 6 – <u>підкреслений</u> . |

Зміна регістра символів: *Формат* → *Регистр*.

Задання міжсимвольного інтервалу виконується командою *Формат* → *Шрифт* → *Інтервал* (рис. 1.46):

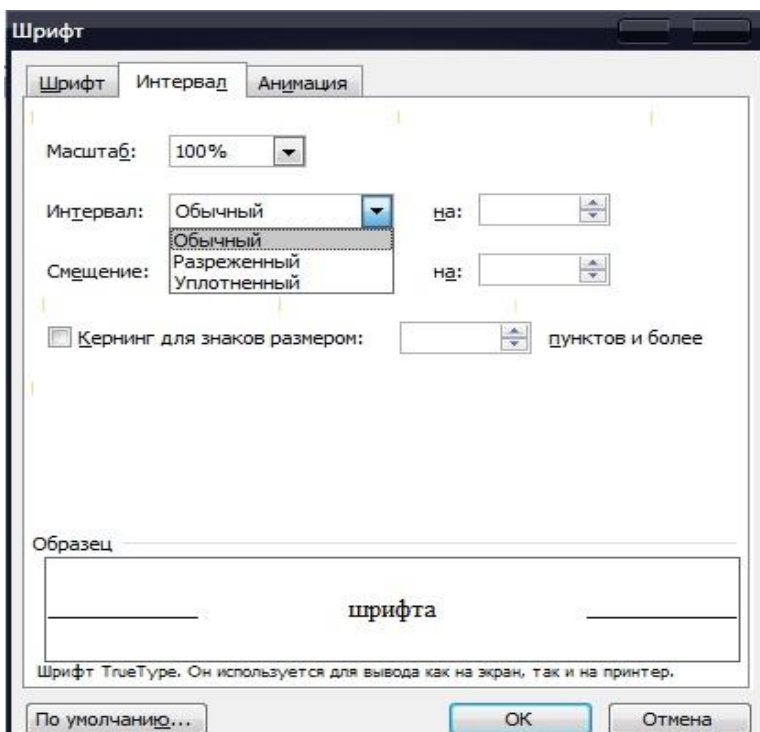


Рис.1.46. Міжсимвольний інтервал

На *горизонтальній координатній лінійці* (рис.1.47) містяться: маркер першого рядка (1), маркер лівої (2) і правої меж абзацу:



Рис.1.47. Горизонтальна лінійка

Розглянемо призначення деяких кнопок (рис. 1.48):

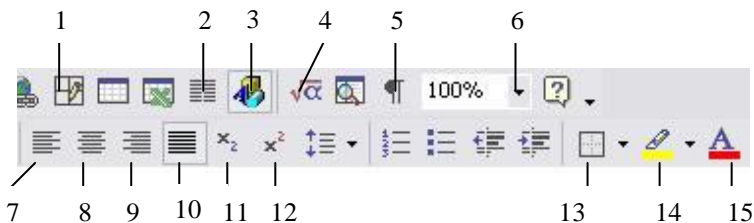


Рис.1.48. Кнопки панелі інструментів «Стандартная»

- | | |
|--|--|
| 1 – створення таблиць; | 9 – вирівнювання по правому краю (рис.1.49); |
| 2 – стовпці; | 10 – вирівнювання по ширині; |
| 3 – виведення панелі інструментів «Рисование»; | 11 – нижній індекс; |
| 4 – редактор формул; | 12 – верхній індекс; |
| 5 – недруковані символи; | 13 – границі таблиць; |
| 6 – масштаб; | 14 – виділення кольором; |
| 7 – вирівнювання по лівому краю; | 15 – колір шрифту. |
| 8 – вирівнювання по центру; | |

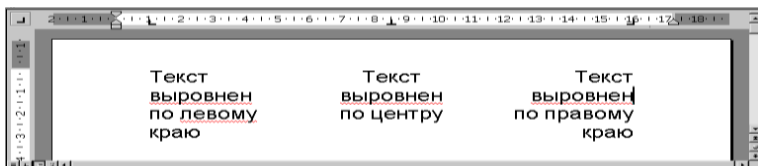


Рис.1.49. Вирівнювання тексту

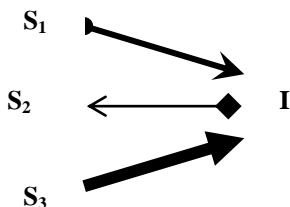
Завдання. Уведіть і відформатуйте текст:

Інформатика – це наука про *властивості, методи* та *засоби* оброблення інформації.



- ♪ **Числова:** цифра, число, вираз, таблиця та ін.
- ♪ **Текстова:** художня (роман), наукова (стаття) та ін.
- ♪ **Графічна:** картина, креслення, графік та ін.
- ♪ **Звукова:** мовлення, музика та ін.
- ♪ **Комбінована:** фільм, відеокліп, вистава.

Види інформації



*Одну і ту саму
інформацію
можна отримати з
різних повідомлень*

Способи подання інформації				
<i>число</i>	$\left\{ \begin{array}{l} \frac{1}{x^{2+3x}} - \sqrt[4]{25x-9} \\ 2x\sqrt{x+8} - 5x^2 \end{array} \right.$	<i>графік</i>		<i>мовлення</i>
<i>таблиця</i>		<i>фільм</i>		<i>вистава</i>
<i>креслення</i>	<i>музика</i>	<i>цифра</i>	<i>концерт</i>	<i>картина</i>

Вказівки до виконання завдання

1. Створення буквиці. Виділіть абзац і виконайте команду *Формат* → *Буквиця* (рис. 1.50):

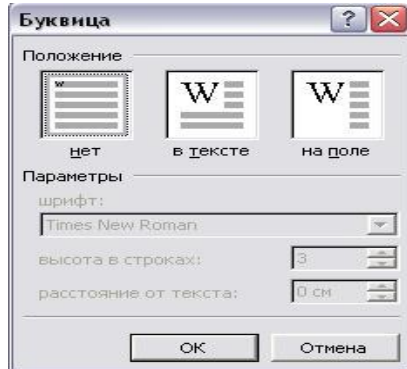


Рис.1.50. Створення буквиці

2. Задання меж сторінки, абзацу або тексту (*Формат* → *Границы и заливка*) (рис. 1.51):

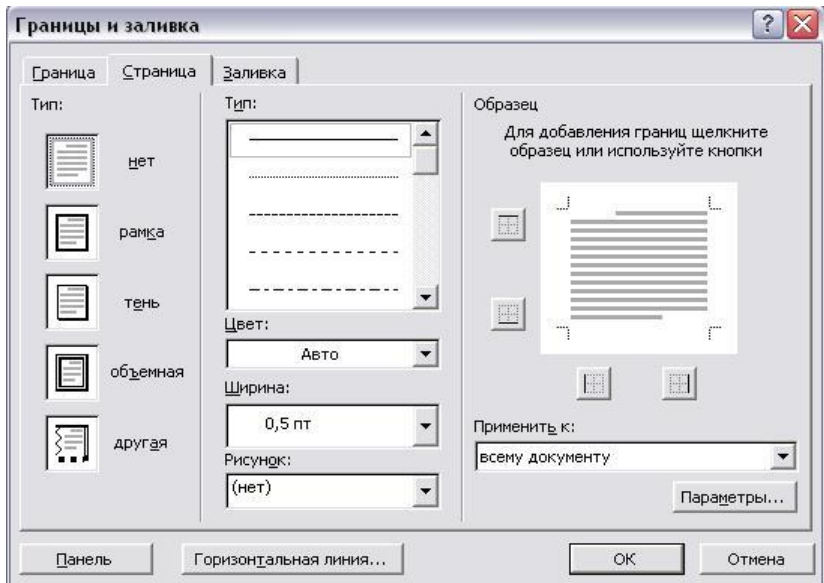


Рис.1.51. Задання меж і заповнення

3. Створення автофігур (рис. 1.52):

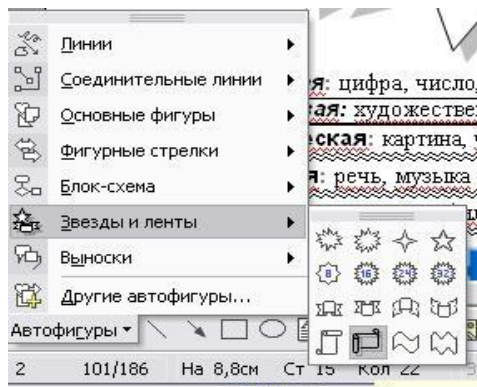


Рис.1.52. Створення автофігур

4. **Створення списку.** Виділіть текст, який потрібно перетворити у список, і виконайте команду *Формат* → *Список* (рис. 1.53):

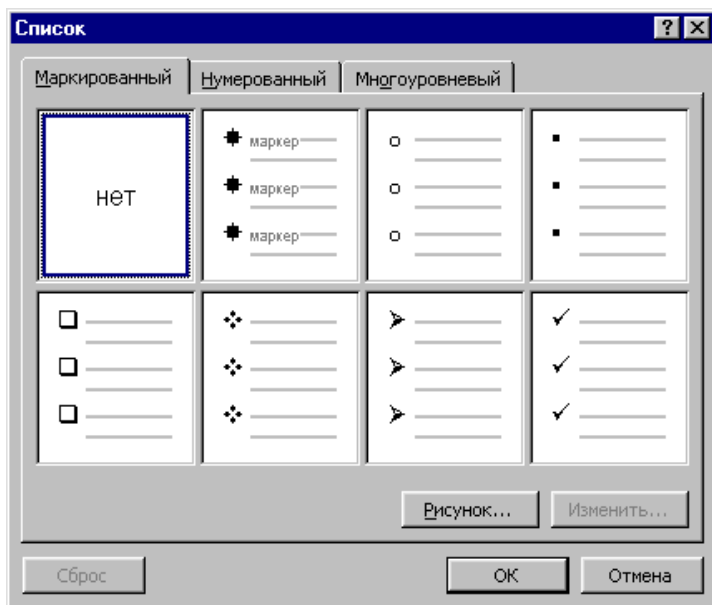


Рис.1.53. Створення списку

5. **Зміна списку** (*Формат* → *Список* → *Изменить*) (рис.1.54):

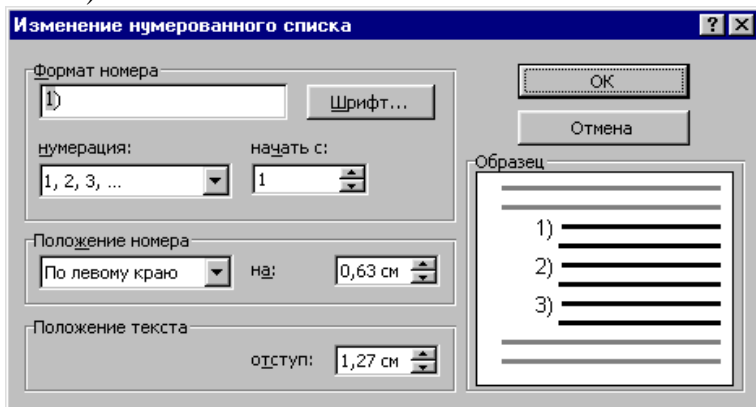


Рис.1.54. Зміна списку

6. **Задання параметрів абзацу** (*Формат* → *Абзац*) (рис. 1.55):

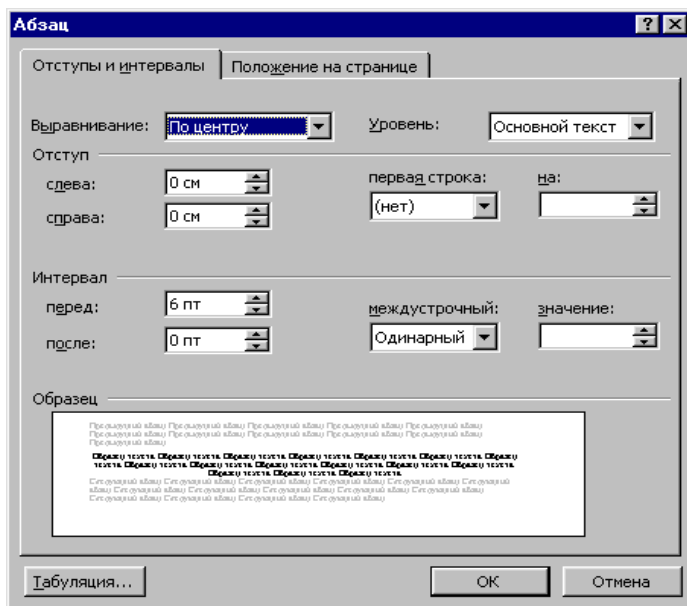


Рис.1.55. Параметры абзацу

Для задання *міжсимвольного* інтервалу треба виконати команду *Формат* → *Шрифт* → *Інтервал*.

Для задання *міжрядкового* інтервалу треба виконати команду *Формат* → *Абзац* → *Отступы и интервалы*.

7. Панель інструментів «Рисование» (рис. 1.56):

- | | |
|-----------------------|----------------------|
| 1 – лінія; | 8 – колір ліній; |
| 2 – прямокутник; | 9 – колір шрифту; |
| 3 – напис; | 10 – тип лінії; |
| 4 – колір заповнення; | 11 – вигляд стрілки; |
| 5 – стрілка; | 12 – стиль тіні; |
| 6 – еліпс; | 13 – об'єм. |
| 7 – об'єкт WordArt; | |

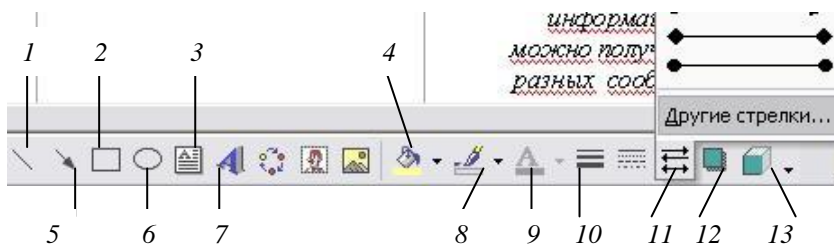


Рис.1.56. Панель інструментів «Рисование»

8. **Вставка таблиці.** Виконайте команду *Таблица* → *Вставить* → *Таблица*, задати параметри (рис.1.57).

9. **Об'єднання клітинок таблиці.** Виділіть клітинки і виконайте команду *Таблица* → *Объединить* *ячейки*.

10. **Видалення рядків або стовпців таблиці.** Виділіть рядки або стовпці і виконайте команду *Таблица* → *Удалить* → *Строки* або *Таблица* → *Удалить* → *Столбцы*.

11. **Вставка рядків або стовпців таблиці.** Установіть курсор у потрібну клітинку і виконайте команду *Таблица* → *Вставить* → *Строки* («*выше*» або «*ниже*») або *Таблица* → *Вставить* → *Столбцы* («*слева*» або «*справа*»).

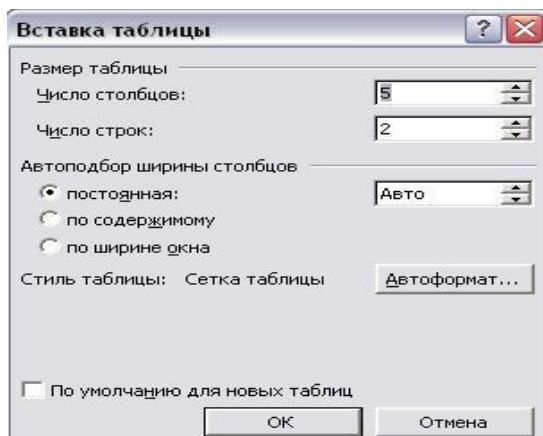


Рис.1.57. Параметры таблицы

12. Вставка формул. Установіть курсор у потрібну клітинку і виконайте команду *Вставка* → *Формула* або натисніть кнопку «*Редактор формул*» на панелі інструментів (рис.1.48). З'явиться поле редагування формул і панель інструментів «Формула» (рис.1.58), за допомогою якої вводяться формули.

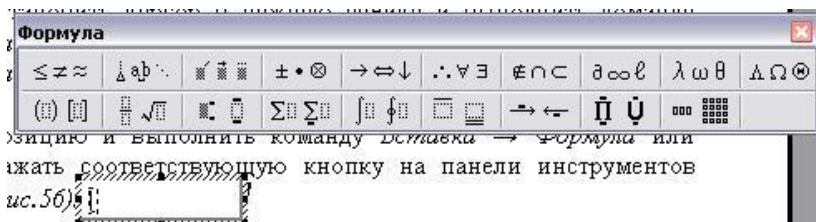


Рис.1.58. Панель інструментів «Формула»

13. Параметри програми Word задаються за допомогою команди *Сервис* → *Параметры*.

14. Додавання кнопок до панелей інструментів. До панелей інструментів можна додавати окремі кнопки за допомогою команди *Сервис* → *Настройка* → *Команды* (рис.1.59). Виберіть розділ, необхідну кнопку і витягніть її на панель інструментів за допомогою миші.

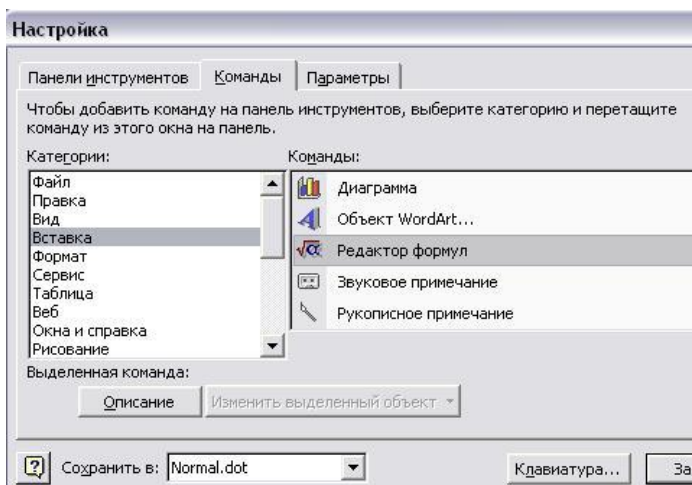


Рис.1.59. Додавання кнопок до панелей інструментів

1.5. Графічний редактор

Комп'ютерна графіка – це створення та оброблення зображень за допомогою комп'ютера. Є два види комп'ютерної графіки – *растрова* та *векторна*.

Растрова графіка. Зображення складається з різнокольорових точок (*пiкселiв*). *Перевага*: висока якість зображення. *Недолiк*: файл, що містить растровий малюнок, має великий розмір.

Векторна графіка. Зображення складається з простих об'єктів (*примітивiв*), таких, наприклад, як *лiнiя*, *коло*, *прямокутник*. Вони задаються за допомогою математичного опису. *Перевага*: файли, що зберігають векторне зображення, мають у 10 – 1000 разів менший розмір, ніж аналогічні растрові файли. *Недолiк*: неприродність зображення.

Операційна система *Windows* містить растровий графічний редактор **Paint**. Програма запускається командою *Пуск* → *Програми* → *Стандартные* → *Paint*. Після запуску з'явиться вікно програми (рис.1.60).

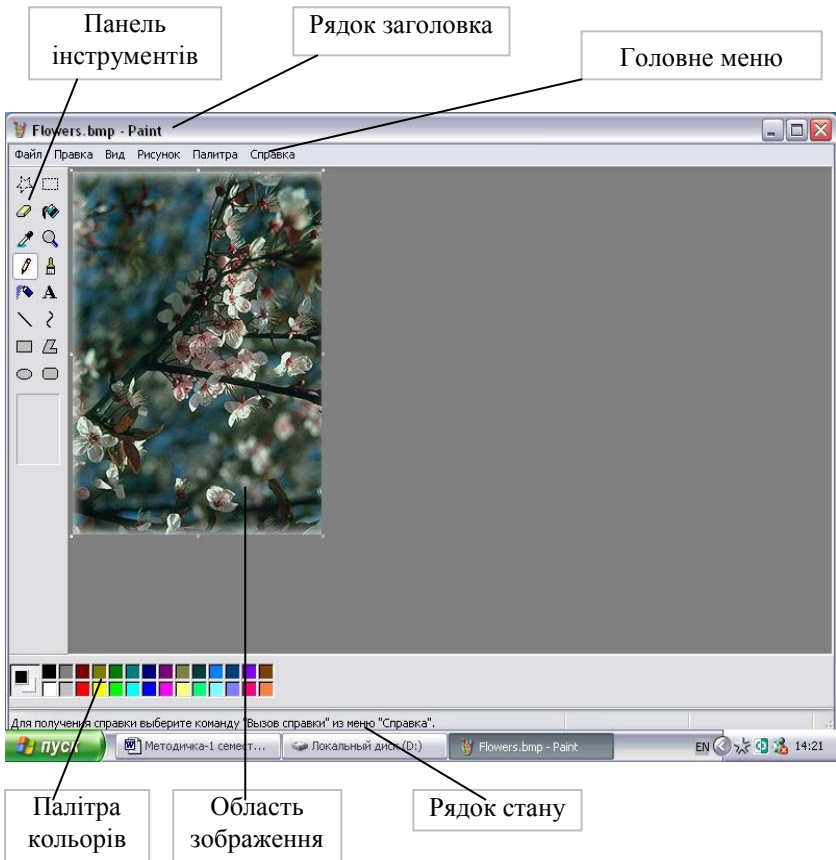


Рис.1.60. Вікно графічного редактора

Рядок заголовка містить ім'я програми Paint та ім'я відкритого графічного файлу.

Рядок меню містить такі пункти:

Файл – робота з файлами (створення, відкриття, збереження, друк).

Правка – редагування зображення.

Вид – керування зображенням деяких елементів вікна.

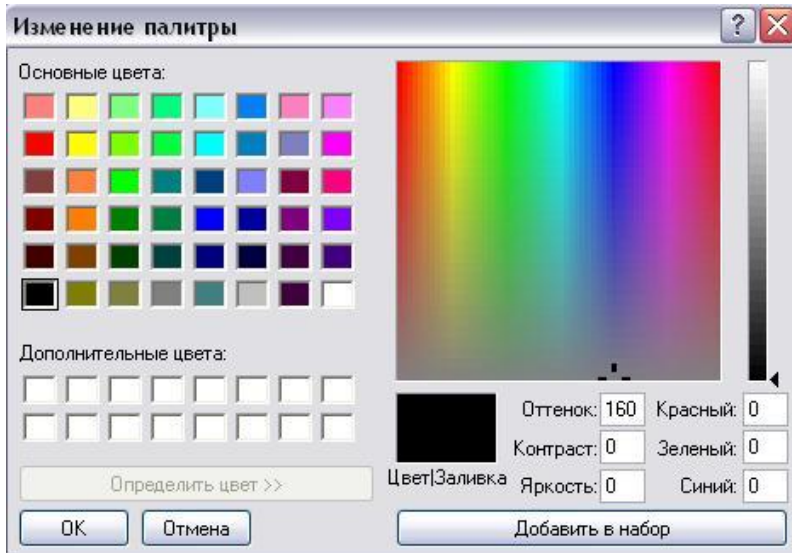


Рис.1.62. Додавання кольору

Рядок стану містить довідкову інформацію про призначення пунктів меню, кнопок інструментів, хід виконання деяких операцій, координати курсору в пікселях.

Область зображення призначена для створення зображення. Її розміри задаються командою *Рисунок* → *Атрибути*.

Завдання 1. Зобразіть шахову дошку розміром 8×8 клітинок (рис.1.63).

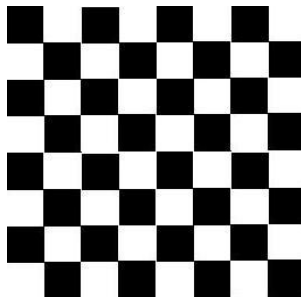


Рис.1.63. Шахова дошка

План виконання

1. Виберіть інструмент **«Прямокутник»** і побудуйте квадрат, тримаючи натиснутою клавішу **«Shift»**.
2. Виберіть інструмент **«Заливка»** і зафарбуйте квадрат.
3. Виберіть інструмент **«Виділення»** і виділіть квадрат.
4. Утримуючи клавішу **«Ctrl»**, перетягніть копію квадрата в потрібне місце, щоб утворилась заготовка шахової дошки розміром 2×2.
5. Виберіть інструмент **«Виділення»** і виділіть заготовку.
6. Утримуючи клавішу **«Ctrl»**, перетягніть копію заготовки в потрібне місце, щоб утворилась заготовка шахової дошки розміром 4×4.
7. Аналогічно отримайте зображення шахової дошки розміром 8×8.

Завдання 2. Створіть зображення обкладинки блокнота (рис.1.64).

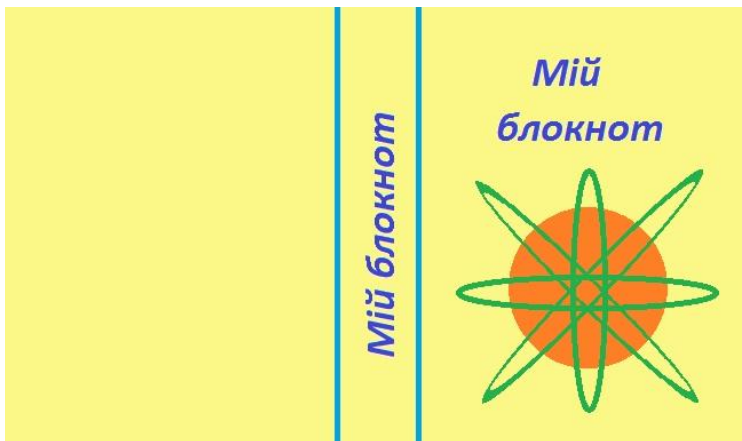


Рис. 1.64. Обкладинка блокнота

План виконання

1. Виконайте команду **Рисунок** → **Атрибути** і в діалоговому вікні (рис.1.65) задайте ширину зображення 25 см і висоту 15 см.



Рис. 1.65. Атрибути зображення

2. Виберіть інструмент «*Линія*» і побудуйте по центру дві вертикальні лінії, утримуючи клавішу «*Shift*».

3. Виконайте команду *Рисунок* → *Отразить/Повернуть* і в діалоговому вікні (рис.1.66) задайте кут повороту 90° .



Рис. 1.66. Відбиття та поворот

4. Виберіть інструмент «*Напис*» і між лініями створіть напис з текстом «*Мій блокнот*».

5. Виконайте команду *Рисунок* → *Отразить/Повернуть* і в діалоговому вікні задайте кут повороту 270° .

6. Виберіть інструмент «*Надпис*» і в правій частині зображення створіть напис з текстом «*Мій блокнот*».

7. Виберіть інструмент «*Эллипс*» і побудуйте коло, утримуючи клавішу «*Shift*».

8. Виберіть інструмент «*Заливка*» і зафарбуйте еліпс.

9. Виберіть інструмент «*Эллипс*» і побудуйте вертикально і горизонтально розташовані еліпси.

10. Побудуйте еліпс у лівій частині малюнка.

11. Виберіть інструмент «*Выделение*» і виділіть еліпс.

12. Виконайте команду *Рисунок* → *Растянуть/Наклонить* і в діалоговому вікні задайте кут нахилу *по горизонталі 45°* і *по вертикалі 45°* (рис.1.67).

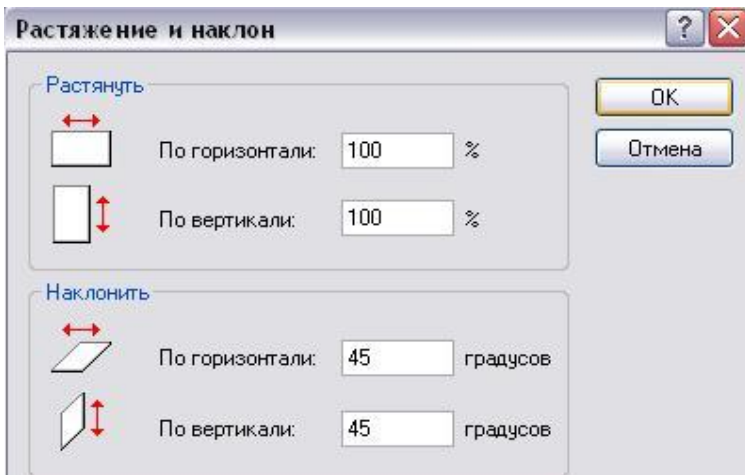


Рис. 1.67. Розтягування і нахил

13. Натисніть кнопку «*Кнопка 2*» (див. рис.1.61).

14. Перетягніть виділене зображення в потрібне місце.

15. Побудуйте ще один еліпс у лівій частині малюнка, нахиліть його на -45° і скопіюйте у потрібне місце.

16. Виберіть інструмент «*Заливка*» і зафарбуйте область зображення.

1.6. Електронні таблиці

Електронні таблиці – це програми, призначені для автоматичного оброблення числової інформації, поданої у формі таблиці. Табличний процесор *Microsoft Excel* входить у стандартний пакет *Microsoft Office*.

Для запуску програми треба виконати команду *Пуск* → *Програми* → *Microsoft Excel* (*Start* → *Programs* → *Microsoft Excel*). Після запуску на екрані з'явиться вікно програми (рис.1.68).

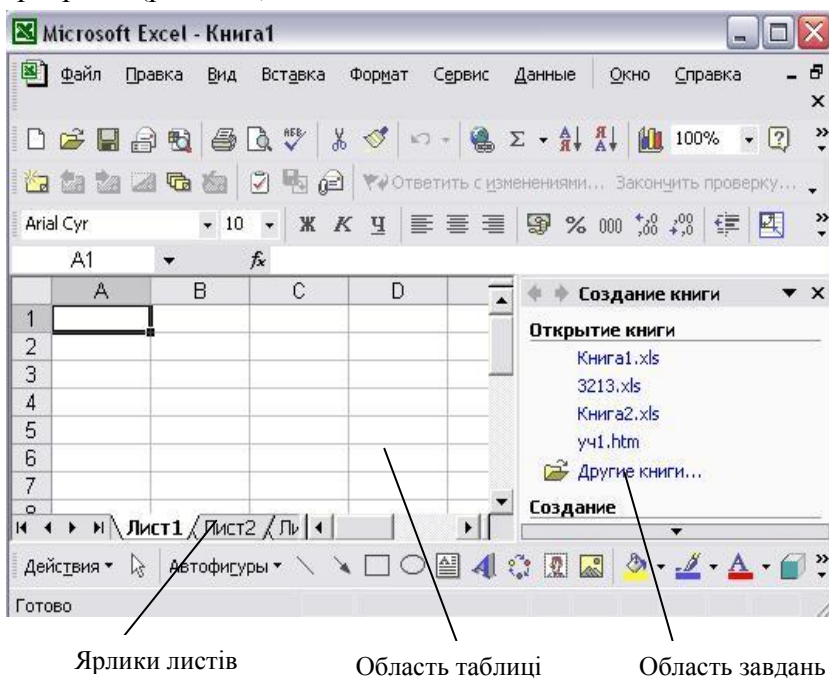


Рис. 1.68. Вікно програми Microsoft Excel

Документи, що створюються в середовищі Excel, називаються **книгами**. Кожна книга складається з **листів**. Робочі листи – це електронні таблиці, що складаються зі **стовпців** та **рядків**. Максимальна кількість стовпців таблиці 255, рядків – 65536.

Стовпці позначаються зліва направо літерами: перші 26 – літерами А..Z, наступні 26 – літерами АА..AZ і так далі до останнього 255-го стовпця, котрий позначається як IV.

Рядки позначаються згори донизу числами від 1 до 65536. На перетині стовпця та рядка містяться **клітинки**.

Адреса клітинки складається з позначення стовпця та номера рядка. Наприклад: A2, C5, D7.

Адреса діапазону складається з адрес лівої верхньої та правої нижньої клітинок. Наприклад: A3:C6, D5:F7 (рис.1.69).

Клітинку, у якій розміщена **курсорна рамка**, називають **активною**. Символи вводяться в активну клітинку. Клітинка може містити число, формулу або текст.

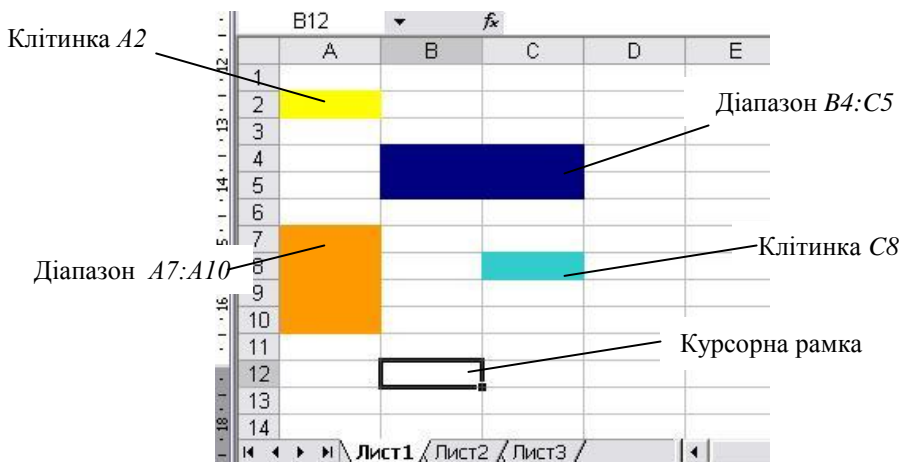


Рис. 1.69. Адреси клітинок та діапазонів

Завдання 1. Складіть електронний касовий чек, у якому зафіксовано купівлю 7 товарів у вигляді таблиці.

Побудуйте:

- кругову діаграму цін товарів;
- гістограму кількості товарів.

Назва	Ціна	Кількість	Усього
Холодильник	2200	2	
Праска	295	4	
Телевізор	2600	2	
Магнітофон	1200	3	
Фотоапарат	2000	2	
Комп'ютер	4500	1	
Принтер	1500	3	
<i>Сума</i>			
<i>Середнє значення</i>			
<i>Мінімум</i>			
<i>Макимум</i>			

План виконання

1. Уведіть назви товарів, ціни, кількість (рис.1.70).
2. У клітинку D2 введіть формулу $=B2*C2$. Введена формула відобразиться в рядку формул. **Формула повинна починатись тільки зі знака « = ».**

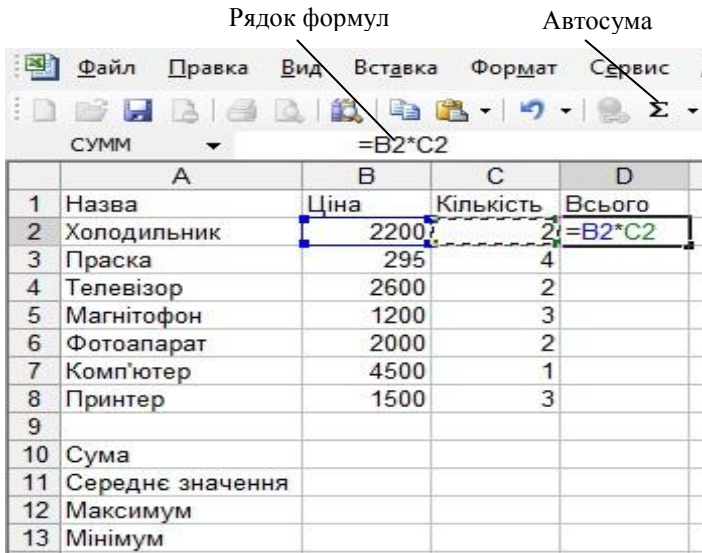


Рис. 1.70. Задача про касовий чек

Натисніть клавішу «*Enter*». Замість уведеної формули з'явиться число – результат обчислень за формулою.

Для *копіювання* формули на сусідні клітинки поставте курсор на маркер (чорний квадрат в правому нижньому куті курсорної рамки) (рис.1.71) і протягніть лівою кнопкою миші на потрібні клітинки.

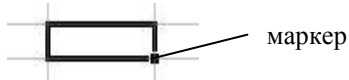


Рис. 1.71. Клітинка

3. Для обчислення *суми* виділіть діапазон *B2:B10* і натисніть кнопку «*Автосумма*» (див. рис.1.70).

4. Аналогічно обчисліть суму для *C2:C10* і *D2:D10*.

5. Для обчислення *середнього значення* встановіть курсор у клітинку *B11* і виконайте команду *Вставка* → *Функція*. У вікні *Мастера функций* виберіть категорію «*Статистические*» (рис.1.72) і функцію *СрЗнач* (рис.1.73).

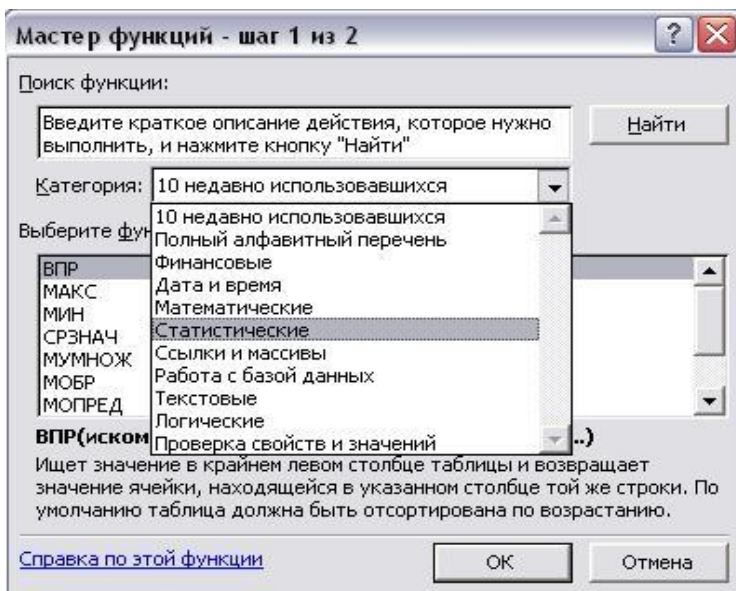


Рис. 1.72. Вікно «Мастера функций»

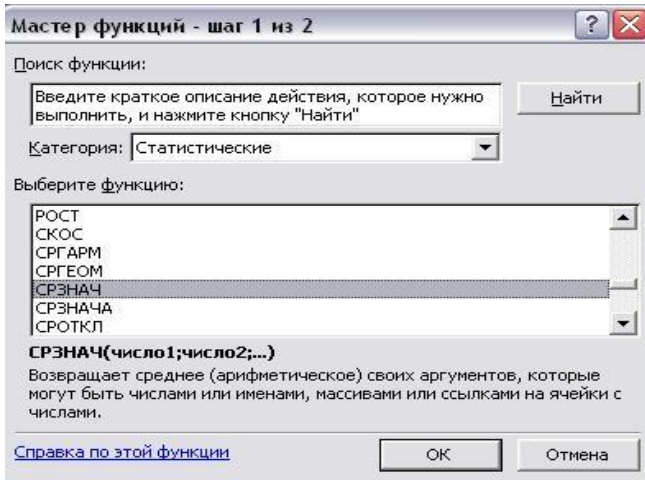


Рис. 1.73. Вибір функції *СрЗнач*

У вікні «*Аргументы функции*» в полі «*Число1*» задайте діапазон *B2:B8* (рис.1.74), натисніть кнопку «**ОК**».

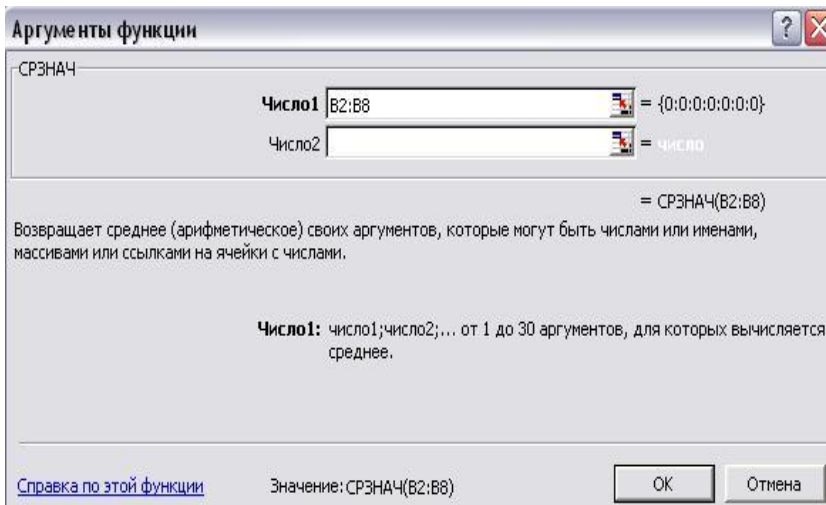


Рис. 1.74. Задання діапазону

6. Аналогічно обчисліть середнє значення для діапазонів *C2:C8* та *D2:D8*.

7. Аналогічно обчисліть мінімум та максимум для діапазонів $B2:B8$, $C2:C8$, $D2:D8$.

8. Щоб сховати **сітку**, виконайте команду *Сервіс* → *Параметри* → *Сетка*.

9. Задайте колір символів і заповнення клітинок, використовуючи кнопки панелі інструментів «*Рисование*» (рис.1.75).

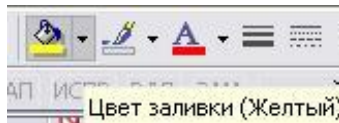


Рис. 1.75. Задання кольору заповнення та символів

Побудова кругової діаграми

1. Виділіть діапазон $A2:B8$ (рис.1.76).

	A	B	C	D
1	Назва	Ціна	Кількість	Всього
2	Холодильник	2200	2	
3	Праска	295	4	
4	Телевізор	2600	2	
5	Магнітофон	1200	3	
6	Фотоапарат	2000	2	
7	Комп'ютер	4500	1	
8	Принтер	1500	3	
9				
10	Сума			
11	Середнє значення			
12	Максимум			
13	Мінімум			

Рис. 1.76. Виділення діапазону

2. Виконайте команду *Вставка* → *Діаграма* (рис.1.77).

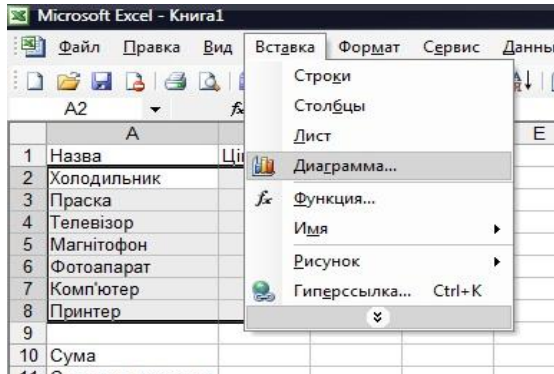


Рис. 1.77. Вставка діаграми

3. Виберіть тип діаграми: **Круговая** (рис.1.78).

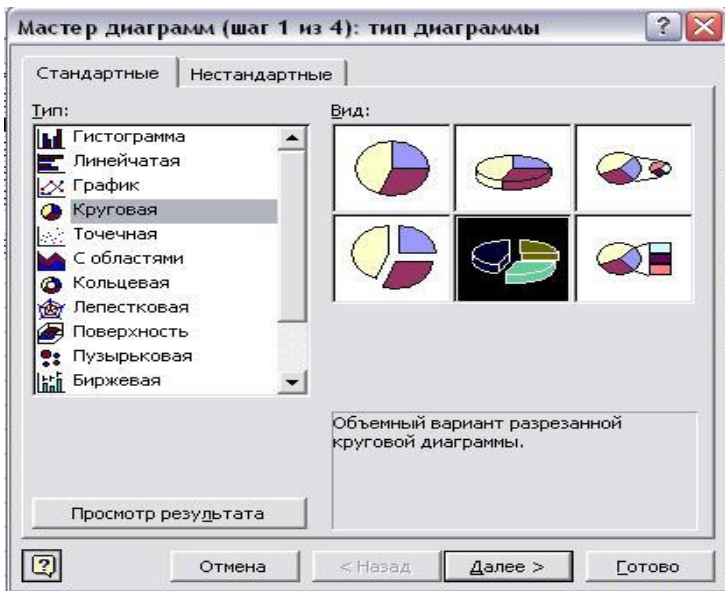


Рис. 1.78. Вибір типу діаграми

4. Натисніть кнопку «Далее» (рис.1.79).

5. Виберіть розділ «Подписи данных» (рис.1.80), у ньому – пункти: «имена категорий», «значения», «доли».

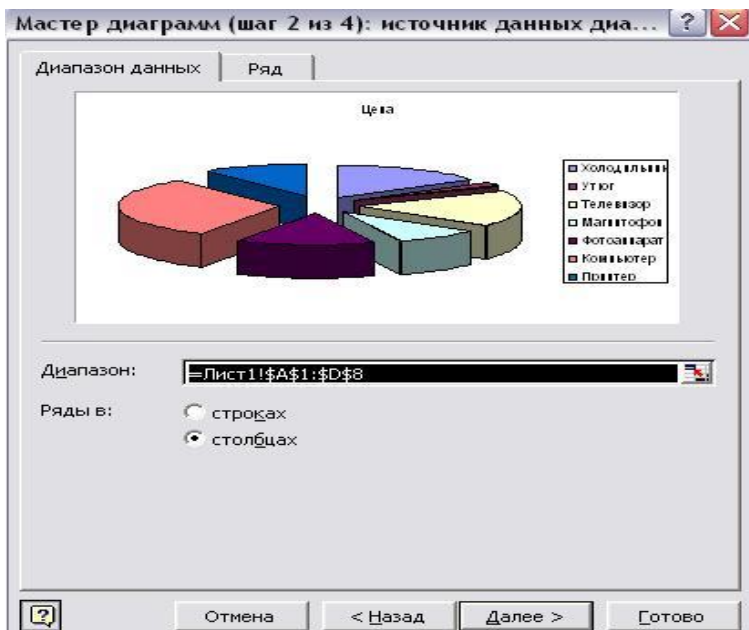


Рис. 1.79. Задання діапазону даних

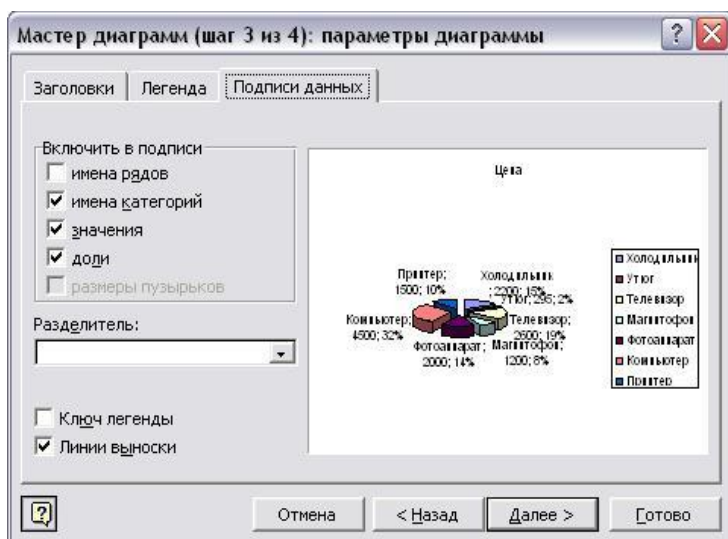


Рис. 1.80. Підписи даних

6. Виберіть лист розміщення діаграми (рис.1.81), натисніть кнопку **«Готово»**.

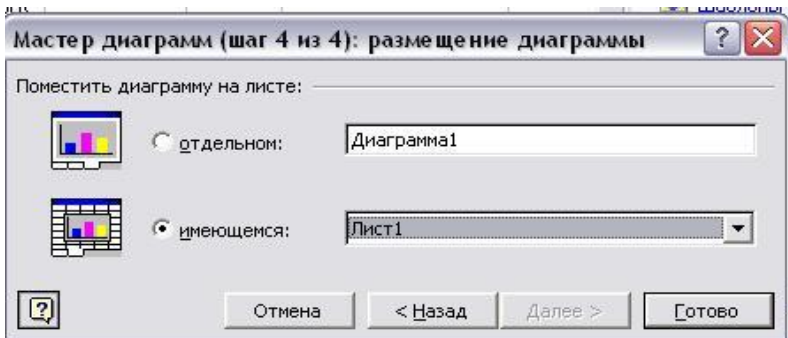


Рис. 1.81. Вибір листа розміщення діаграми

Після цього отримаємо **кругову діаграму** (рис.1.82):



Рис. 1.82. Кругова діаграма

Побудова гістограми

1. Для побудови гістограми кількості товарів виділіть діапазони **A2:A8** і **C2:C8** (рис.1.83). **Щоб виділити несуміжні діапазони, треба тримати натисненою клавішу «Ctrl».**

	A	B	C	D
1	Назва	Ціна	Кількість	Всього
2	Холодильник	2200	2	
3	Праска	295	4	
4	Телевізор	2600	2	
5	Магнітофон	1200	3	
6	Фотоапарат	2000	2	
7	Комп'ютер	4500	1	
8	Принтер	1500	3	
9				
10	Сума			

Рис. 1.83. Виділення несуміжних діапазонів

2. Виберіть тип «Гистограмма» (рис.1.84):

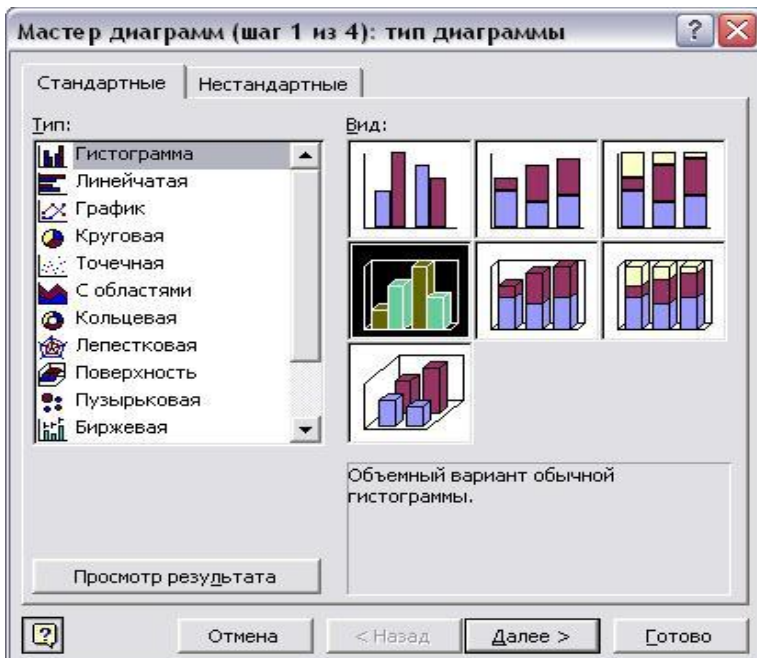


Рис. 1.84. Вибір типу «Гистограмма»

3. Натисніть кнопку «Далее» (рис.1.85):

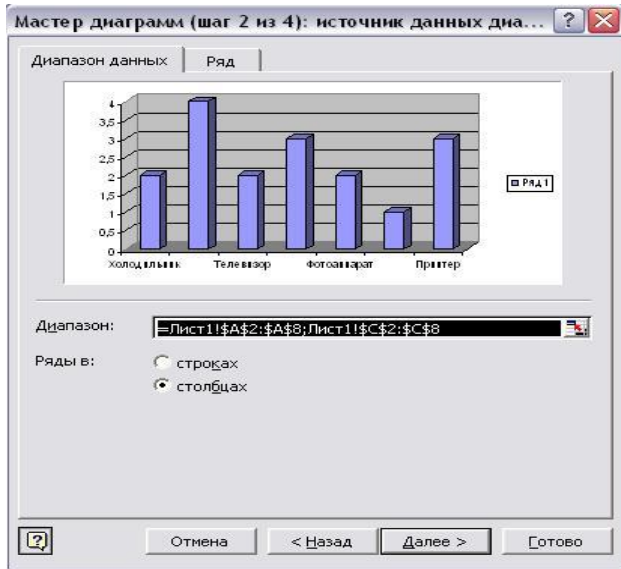


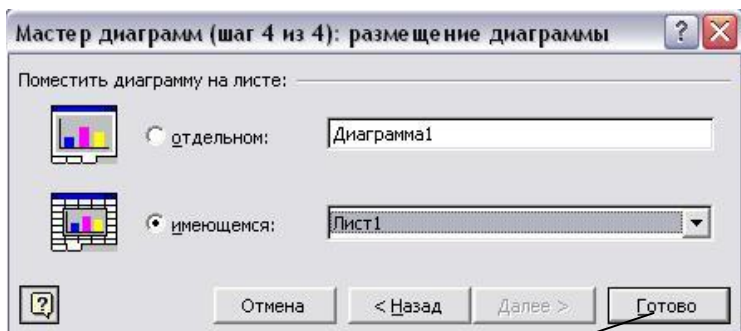
Рис. 1.85. Джерело даних діаграми

4. Натисніть кнопку «Далее» (рис.1.86).



Рис. 1.86. Параметри діаграми

5. Виберіть лист розміщення діаграми («Имеющийся») (рис.1.87), натисніть кнопку «Готово».



Натисніть «Готово»

Рис. 1.87. Розміщення діаграми

Після цього отримуємо гістограму, яка відображає співвідношення кількості різних товарів (рис.1.88).

Діаграма призначена для ілюстрування співвідношення величин.

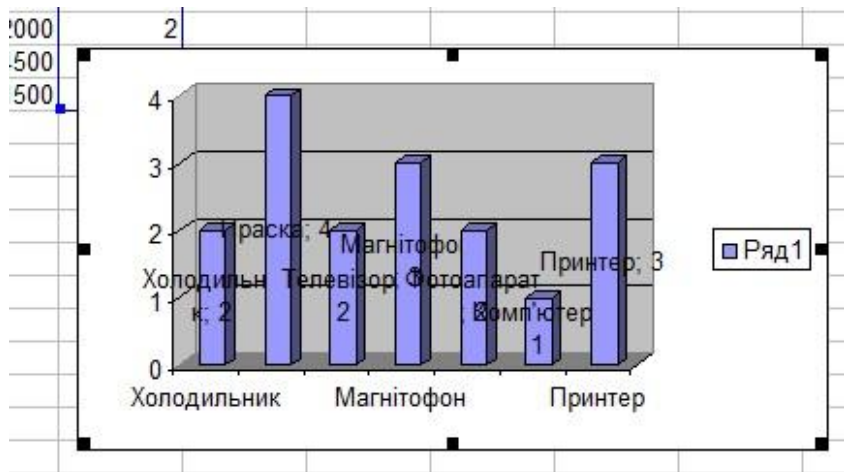


Рис. 1.88. Гістограма

Завдання 2. Складіть відомість про зарплату у вигляді таблиці (рис.1.89):

	A	B	C	D
1	Прізвище, ім'я	Нараховано	Податок (15%)	Отримати
2	Токіл Айлінж	4500	=B2*0,15	=C2-B2
3	Ерхан Аяр	5600		
4	Ци Бінъ	6300		
5	Ерен Тимур	3700		
6	Лю Сінъ	4120		
7				
8	Сума			
9	Середнє значення			
10	Мінімум			
11	Максимум			

Рис. 1.89. Відомість про зарплату

Побудуйте: а) кругову діаграму; б) гістограму.

Сховайте сітку. Задайте заповнення клітинок.

Абсолютні та відносні адреси клітинок. Логічна функція **ЕСЛИ**

A1, A2, B3, ... - це **відносні** адреси клітинок (у разі копіювання формул можуть змінюватись).

\$A\$1, \$B\$3, ... – це **абсолютні** адреси клітинок (не змінюються у разі копіювання формул).

\$A1, A\$1, ... - це **змішані** адреси клітинок (у разі копіювання змінюється тільки номер стовця або тільки номер рядка).

Логічна функція **ЕСЛИ** має такий загальний вигляд:

ЕСЛИ (умова; дія 1; дія 2)

Функція **ЕСЛИ** перевіряє виконання деякої умови; якщо умова виконується, то виконується дія 1, якщо умова не виконується, то виконується дія 2.

Наприклад, розглянемо функцію $y = |x|$:

$$y = \begin{cases} x, & \text{якщо } x \geq 0, \\ -x, & \text{якщо } x < 0. \end{cases}$$

Функція **ЕСЛИ** запишеться у вигляді: $\boxed{=ЕСЛИ(x \geq 0; x; -x)}$.

Завдання. Побудуйте на проміжку $[-2;2]$ з кроком $h=0,4$ таблицю значень функції $y = \begin{cases} x^2 + 0,1nx & \text{для } x \leq 0, \\ \sin(0,1nx) & \text{для } x > 0 \end{cases}$,

де n – день вашого народження.

Побудуйте:

- а) кругову діаграму для x і y ;
- б) гістограму для x і y .

План виконання

Побудуйте таблицю значень функції за рис.1.90.

1. У клітинку **A1** уведіть крок ($h = 0,4$).
2. У клітинку **B1** замість n уведіть **число** (день вашого народження).
3. У клітинку **A2** уведіть назву аргумента X .
4. У клітинку **B2** уведіть назву функції Y .
5. У клітинку **A3** уведіть лівий кінець відрізка (-2).
6. У клітинку **A4** уведіть формулу $=A3+\$A\1 і скопіюйте на діапазон **A5:A13** (поки не з'явиться значення 2).

У задачі $\boxed{x = A3}$, $\boxed{h = 0,4 = \$A\$1}$, $\boxed{n = \$B\$1}$.

7. У клітинку **B3** уведіть функцію **ЕСЛИ** у вигляді:

$\boxed{=ЕСЛИ(A3 <= 0; A3^2 + 0,1 * \$B\$1 * A3; SIN(0,1 * \$B\$1 * A3))}$.

	A	B
1	0,4	< n >
2	x	y
3	-2	=ЕСЛИ(А3<=0,А3^2+0,1*В\$1*А3; SIN(0,1*В\$1*А3))
4	=А3+А\$1	
5		
6		
7		
8		
9		
10		
11		
12		
13	2	
14		
15		

Рис. 1.90. Таблица значений функции

Выполните команду *Вставка* → *Функция*. Выберите категорию «Логические» и функцию **ЕСЛИ** (рис.1.91).

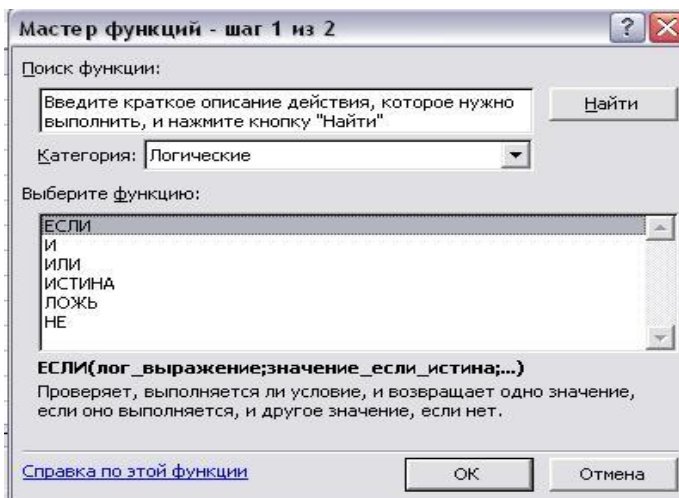


Рис. 1.91. Выбор функции «ЕСЛИ»

У наступному вікні (рис. 1.92) заповніть поля:

Лог_выражение: $A3 \leq 0$

Значение_если_истина: $A3^2 + 0,1 * \$B\$1 * A3$

Значение_если_ложь: $SIN(0,1 * \$B\$1 * A3)$.

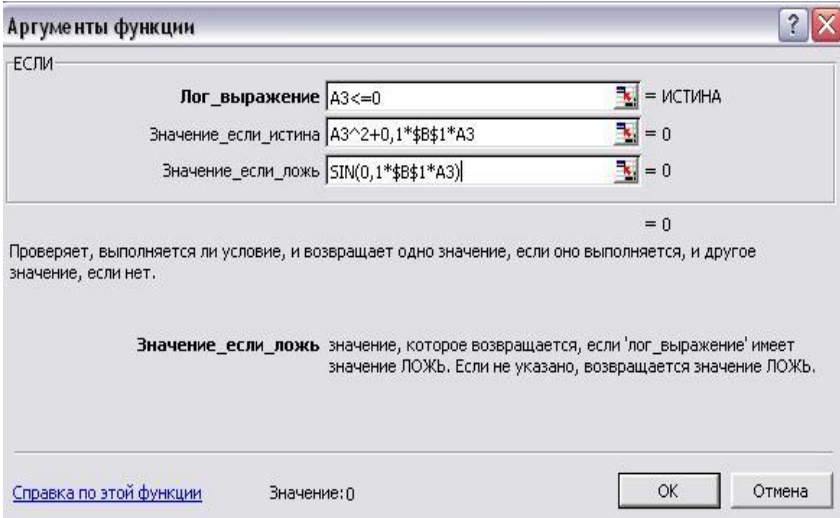


Рис. 1.92. Задання аргументів функції **ЕСЛИ**

1. Скопіюйте формулу в клітинки діапазону $B4:B13$.
2. Побудуйте *кругову діаграму* для значень x і y .
3. Побудуйте *графік залежності* значень x і y .

ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 1

Завдання 1. Побудуйте на проміжку $[a;b]$ з кроком h таблицю значень функції, де n – будь-яке натуральне число.

	a	b	h	Функція
Варіант 1	-3	3	0,2	$z = \begin{cases} 8y^2 - 4ny + 9, & \text{якщо } y \geq -1, \\ y^3 + 5y - n, & \text{якщо } y < -1. \end{cases}$
Варіант 2	-6	6	0,5	$y = \begin{cases} 5x^3 - nx + 2, & \text{якщо } x \geq 4, \\ x^2 + nx - n, & \text{якщо } x < 4. \end{cases}$

Варіант 3	-9	9	1	$y = \begin{cases} 4z^2 - 7nz + 5, & \text{якщо } z \geq 4, \\ z^2 - 3n, & \text{якщо } z < 4. \end{cases}$
Варіант 4	-1	1	0,1	$z = \begin{cases} 7y^2 - y + n, & \text{якщо } y \geq 0, \\ 4y^3 + 3ny - 2, & \text{якщо } y < 0. \end{cases}$
Варіант 5	-7	7	0,5	$y = \begin{cases} nx + 1, & \text{якщо } x \geq 4, \\ 5x^2 + nx - 6, & \text{якщо } x < 4. \end{cases}$
Варіант 6	-8	8	0,4	$z = \begin{cases} 4nx^3 - 2x + 2, & \text{якщо } x \geq 3, \\ 8x^2 + x - n, & \text{якщо } x < 3. \end{cases}$
Варіант 7	-10	10	2	$y = \begin{cases} 3x^2 - x + n, & \text{якщо } x \geq 5, \\ 7x^2 + nx - n, & \text{якщо } x < 5. \end{cases}$
Варіант 8	-15	15	1,5	$z = \begin{cases} 4nx^3 + 2x + 2, & \text{якщо } x \geq 1, \\ 8x^2 + nx - 1, & \text{якщо } x < 1. \end{cases}$
Варіант 9	-4	4	0,2	$y = \begin{cases} 4nx + 6, & \text{якщо } x \geq -0,3, \\ 5x^2 + 7nx - 2, & \text{якщо } x < -0,3. \end{cases}$
Варіант 10	-5	5	0,2	$y = \begin{cases} 4nx + 2, & \text{якщо } x \geq 0,1, \\ x^2 + nx - 8, & \text{якщо } x < 0,1. \end{cases}$
Варіант 11	-1	1	0,1	$z = \begin{cases} 7y^2 - ny + 9, & \text{якщо } y \geq 0, \\ 3ny - 2, & \text{якщо } y < 0. \end{cases}$
Варіант 12	-3	3	0,3	$z = \begin{cases} 2y^2 - 4ny + 7, & \text{якщо } y \geq 1, \\ 9y^3 + 2y - n, & \text{якщо } y < -1. \end{cases}$
Варіант 13	-7	7	0,7	$y = \begin{cases} 5x^3 + nx + 0,2, & \text{якщо } x \geq 4, \\ 7x + 1, & \text{якщо } x < 4. \end{cases}$
Варіант 14	-9	9	0,9	$z = \begin{cases} 3nx^2 - 4x + n, & \text{якщо } x \geq 6, \\ 2x^2 + nx - 6, & \text{якщо } x < 6. \end{cases}$
Варіант 15	-13	13	1,3	$y = \begin{cases} x^3 + nx - 4, & \text{якщо } x \geq 8, \\ 5x^3 + 6x - n, & \text{якщо } x < 8. \end{cases}$
Варіант 16	-4	4	0,8	$y = \begin{cases} 6x^3 - x + n, & \text{якщо } x \geq 2, \\ 4x^2 + nx + 9, & \text{якщо } x < 2. \end{cases}$

Варіант 17	-3	3	0,6	$z = \begin{cases} 3x^3 - 4x + 6, & \text{якщо } x \geq -1, \\ x^2 + 7nx, & \text{якщо } x < -1. \end{cases}$
Варіант 18	-2	2	0,2	$y = \begin{cases} x^3 - x + n, & \text{якщо } x \geq 0,3, \\ 7nx^2 + x - 1, & \text{якщо } x < 0,3. \end{cases}$
Варіант 19	-8	8	0,8	$z = \begin{cases} nx^3 + 4x + 3, & \text{якщо } x \geq 6, \\ 3x^2 + 6nx - 8, & \text{якщо } x < 6. \end{cases}$
Варіант 20	-10	10	1	$z = \begin{cases} 5y^3 - 3y + 2, & \text{якщо } y \geq -1, \\ 2y^2 + ny - 5, & \text{якщо } y < -1. \end{cases}$

Завдання 2. Побудуйте:

- кругову діаграму значень x і y ;
- графік залежності значень x і y .

1.7. Системи керування базами даних

База даних – це велика кількість даних однакової структури з певної галузі. *Наприклад*, телефонний довідник, каталог книг в бібліотеці, розклад руху потягів, дані про співробітників підприємства і т.ін.

Система керування базами даних (СКБД) – це програма, призначена для створення баз даних і пошуку в них інформації. Однією з таких СКБД є **Microsoft Access**. База даних у *Microsoft Access* може містити такі об'єкти, як показано на рис.1.93.

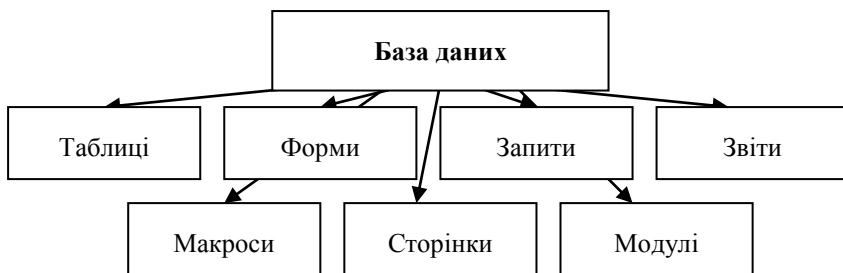


Рис.1.93. Об'єкти бази даних

Таблиці містять дані у вигляді двовимірного масиву, який має рядки (*записи*) і стовпці (*поля*) (рис.1.94).

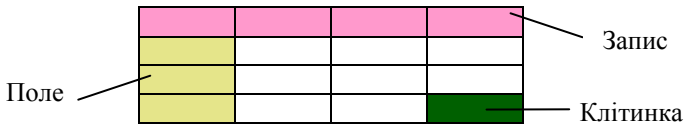


Рис.1.94. Елементи таблиці бази даних

Запис містить інформацію *тільки про один об'єкт* і може складатись з даних *різних типів*. **Поле** може містити дані про багато *об'єктів*, але ці дані мають бути *тільки одного типу*.

Запити створюються для пошуку і вибору певних даних з таблиць.

Форми дозволяють відбирати дані з однієї або кількох таблиць і виводити їх на екран, використовуючи макет.

Звіти готують до друку дані з таблиць або запитів.

Макроси автоматизують стандартні дії.

Модулі автоматизують складні операції.

Створення бази даних починається зі створення **структури запису** (*визначення кількості і типу полів, задання їхніх назв і властивостей*).

Тип даних визначає, які дані можна вводити в поле. Допустимі такі **типи даних**:

Текстовий – це один рядок тексту (до 256 символів).

Поле MEMO – кілька рядків тексту (до 65535 символів).

Числової – це число будь-якого типу.

Дата/время – поле, яке містить дату або час.

Денежний – значенням поля є грошові одиниці.

Счетчик – значення поля вводиться автоматично.

Логический – поле містить одне із значень: *TRUE* (*істина*) або *FALSE* (*хибність*).

Поле об'єкта OLE – містить зображення, звукові файли, таблиці *Excel*, документи *Word* і т.ін.

Кожна таблиця повинна містити *ключове поле*. **Ключове поле** – це поле або декілька полів, які однозначно визначають запис (не містять даних, що повторюються).

Об'єкти бази даних (таблиці, форми, запити і т.ін.) можна відкривати в режимі *введення і редагування даних*, а також у режимі *конструктора* (для редагування структури).

Завдання 1. Створіть базу даних, яка містить таблиці:
Студенти

<i>N1</i>	Прізвище	Ім'я	Дата народження
1			
2			
...			
5			

Адреси, телефони

<i>N2</i>	Прізвище	Адреса	Телефон
1			
2			
...			
5			

План створення таблиці

1. Пуск → Програми → Microsoft Access (Start → Programs → Microsoft Access).
2. Виберіть команду «Новая база данных» (рис.1.95).

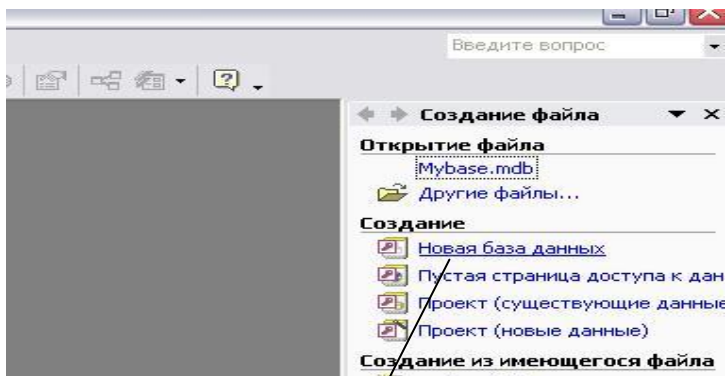


Рис.1.95. Створення нової бази даних

2. Задайте *ім'я* вашої бази даних, натисніть кнопку «*Создать*» (рис.1.96).

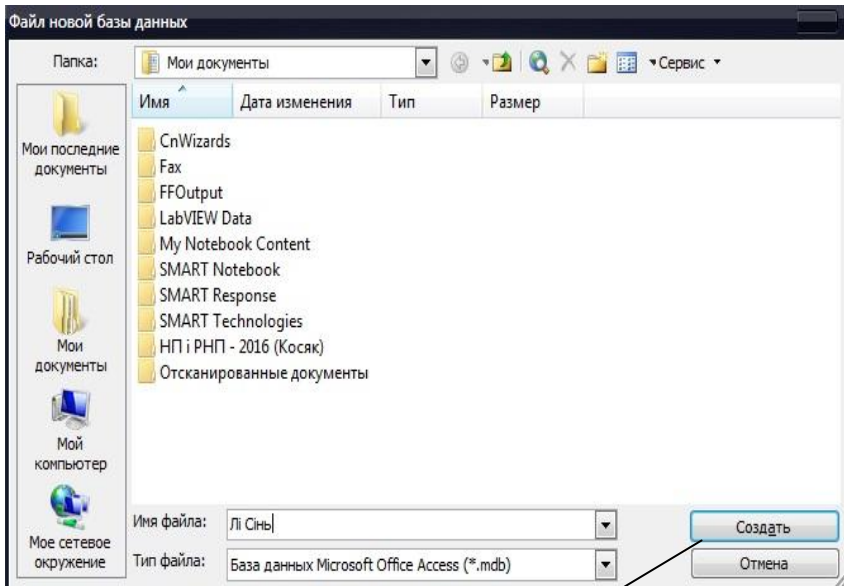


Рис.1.96. Задання імені бази даних

3. У розділі «*Таблиці*» виберіть команду «*Создание таблицы в режиме конструктора*» (рис.1.97).

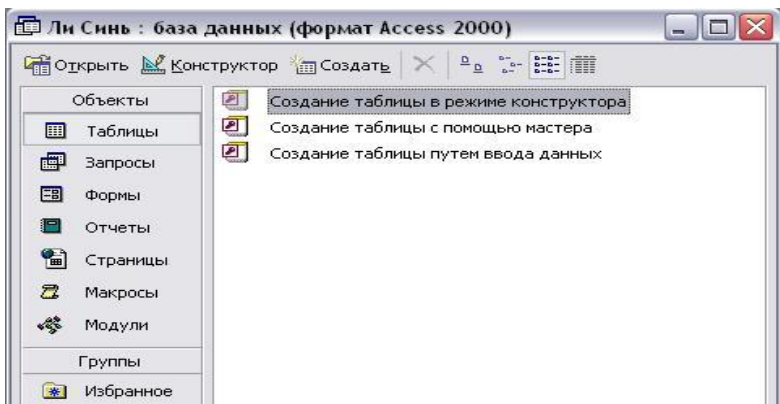


Рис.1.97. Створення таблиці в режимі конструктора

4. Створіть *структуру запису*, тобто введіть *назви полів* і вкажіть їх *тип* (рис.1.98).

Имя поля	Тип данных
N1	Счетчик
Прізвище	Текстовый
Ім'я	Текстовый
Дата народження	Дата/время

Рис.1.98. Створення структури запису

5. Задайте *ключове поле* (це поле «N1», оскільки тільки воно не може містити даних, що повторюються). Виділіть рядок «N1» (рис.1.99) і *натисніть кнопку «Ключевое поле»* на панелі інструментів (рис.1.100).

Имя поля	Тип данных
N1	Счетчик
Прізвище	Текстовый
Ім'я	Текстовый
Дата народження	Дата/время

Рис.1.99. Виділення рядка «N1»

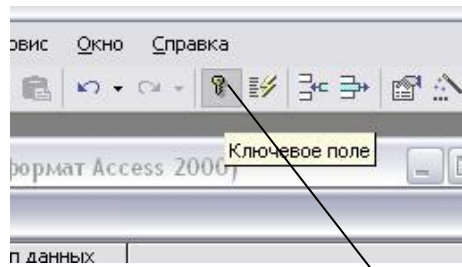


Рис.1.100. Кнопка «Ключевое поле»

6. Збережіть таблицю: *Файл* → *Сохранить* (рис.1.101).
7. Задайте ім'я таблиці «Студенти» (рис.1.102).
8. Відкрийте таблицю для введення даних (рис.1.103).

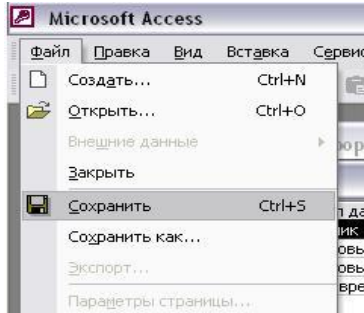


Рис.1.101. Збереження таблиці



Рис.1.102. Задання імені таблиці

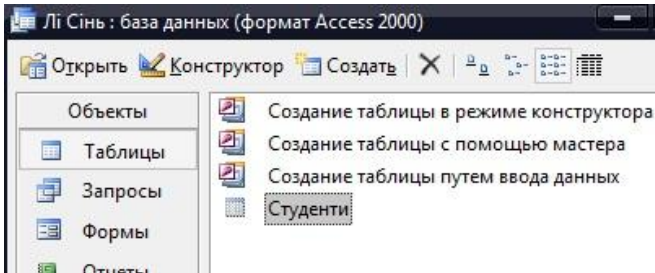


Рис.1.103. Відкриття таблиці для введення даних

9. Заповніть таблицю (рис.1.104).

	<i>N1</i>	Прізвище	Ім'я	Дата народжен
	1	Оу	Міньюе	15.09.2000
	2	Еркен	Мустафа	11.04.1999
	3	Цао	Фейфей	16.07.2000
	4	Токіл	Айпінж	09.05.1999
	5	Ерен	Тимур	19.03.2000
	*	(Счетчик)		

Рис.1.104. Заповнення таблиці

10. Аналогічно створіть Таблицю 2 (рис.1.105).

Адреси, телефони : таблиця				
	N2	Прізвище	Адреса	Телефон
	1	Оу	м.Київ, вул. Гарматна, 4	0671112233
	2	Еркен	м.Київ, пр-т Комарова, 7	0975556677
	3	Цао	м.Київ, вул. Перова, 67	0957778888
	4	Токіл	м.Київ, пр-т Комарова, 15	0639992277
	5	Ерен	м.Київ, пр-т Комарова, 10	0674455533
	* (Счетчик)			

Рис.1.105. Таблица 2

Завдання 2. Зв'яжіть таблиці за допомогою перенесення поля «Адреса» з таблиці «Адреси, телефони» у таблицю «Студенти» (рис.1.106).

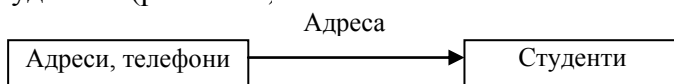


Рис.1.106. Перенесення поля «Адреса»

План зв'язування таблиць

1. Відкрийте *Таблицю1* в режимі конструктора (у розділі «Таблиці» виділіть таблицю «Студенти» і натисніть кнопку «**Конструктор**») (рис.1.107).

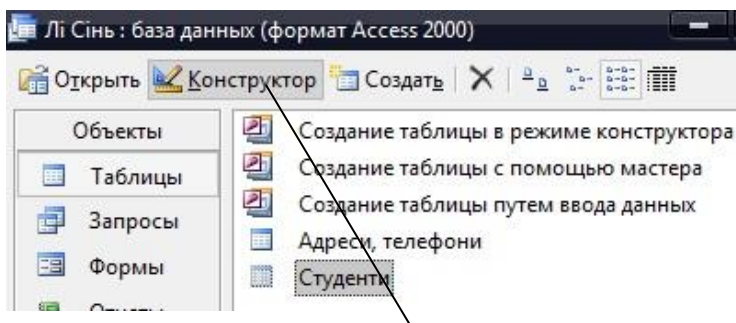


Рис.1.107. Відкриття таблиці в режимі конструктора

2. У структурі *Таблиці 1* додайте ім'я нового поля «Адреса», задайте тип «**Мастер подстановок**» (рис.1.108).

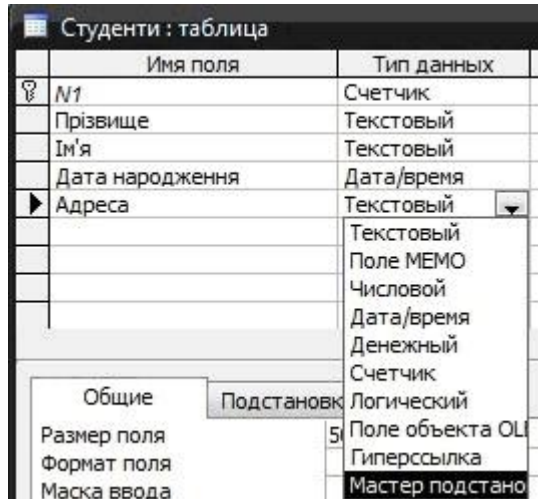


Рис.1.108. Зв'язування таблиць за допомогою Мастера подстановок

3. У вікні «Мастера подстановок» натисніть кнопку «Далее» (рис.1.109).

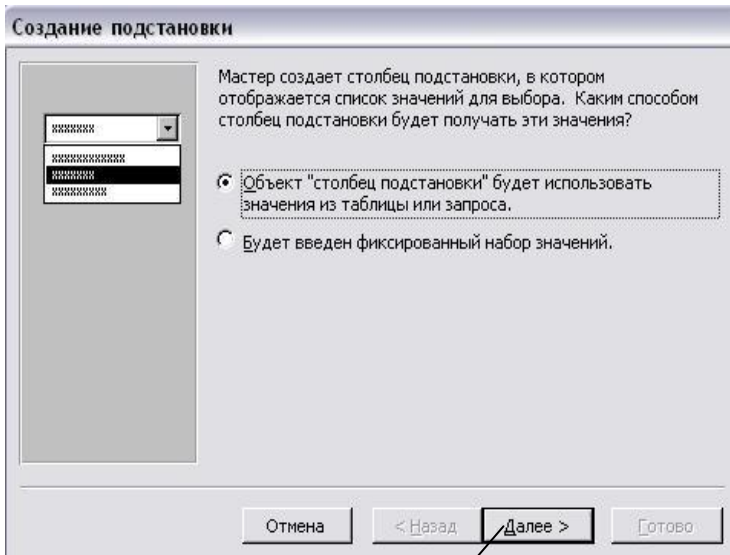


Рис.1.109. Створення підстановки

4. Натисніть кнопку «Далее» (рис.1.110).

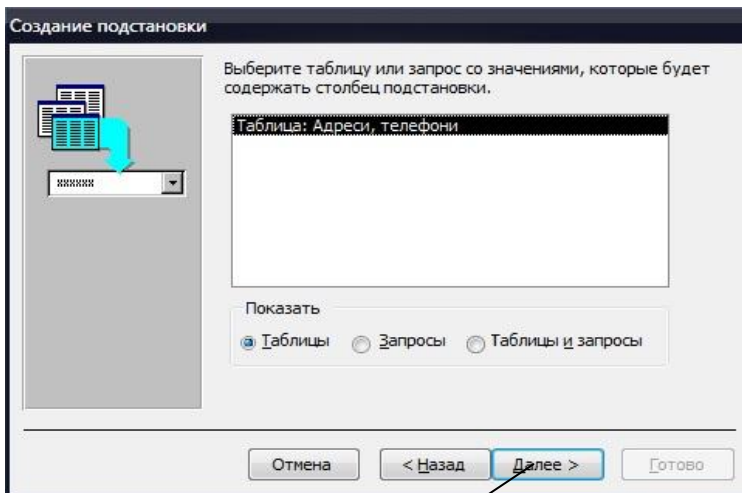
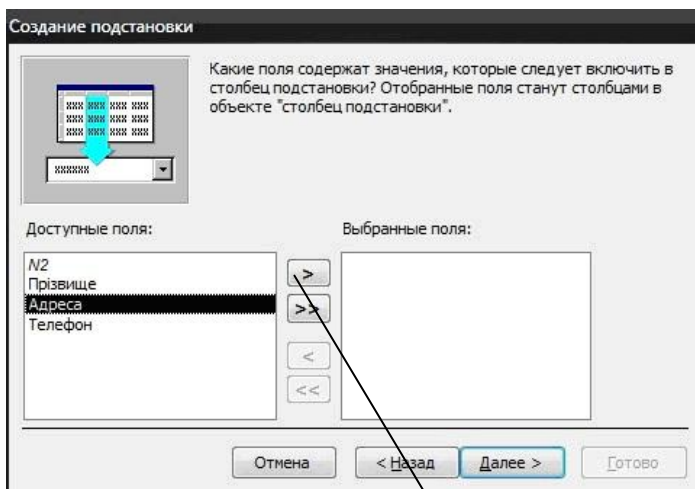


Рис.1.110. Вибір таблиці для підстановки

5. З лівого стовпця виберіть поле «Адреса» і перенесіть у правий за допомогою кнопки « > » (рис.1.111, 1.112).



Натисніть кнопку « > »

Рис.1.111. Створення підстановки (продовження)

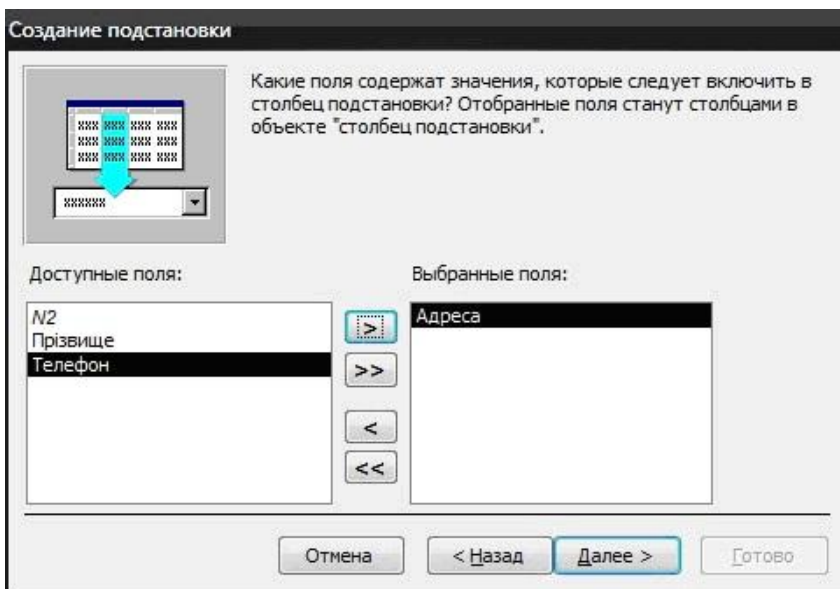


Рис.1.112. Вибір поля «Адреса»

6. Відредагуйте ширину стовпця і натисніть кнопку «Далее» (рис.1.113).

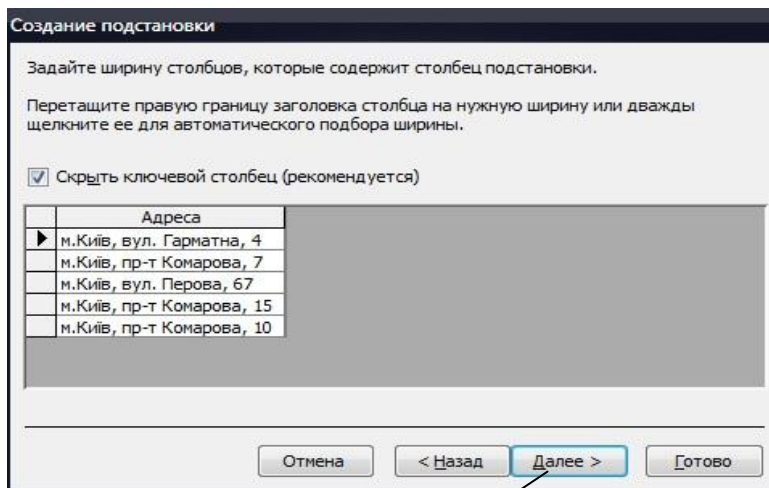


Рис.1.113. Створення підстановки (продовження)

7. Задайте назву нового стовпця «*Адреса*» і натисніть кнопку «*Готово*» (рис.1.114).

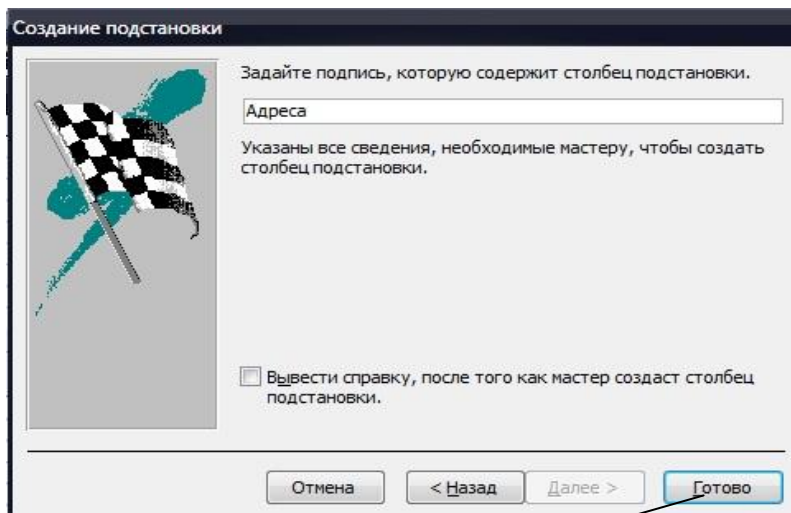


Рис.1.114. Задання підпису стовпця підстановки

8. Підтвердіть необхідність збереження таблиці (натисніть кнопку «*Да*») (рис.1.115).

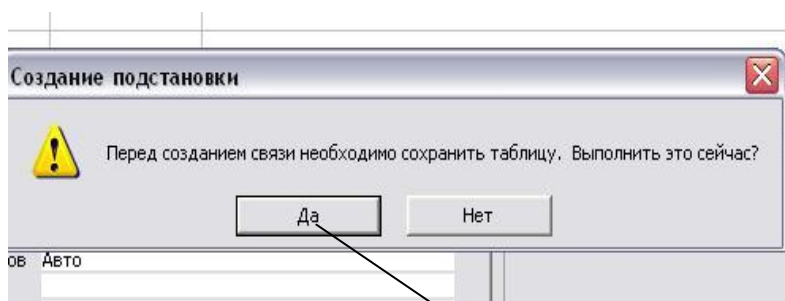


Рис.1.115. Збереження таблиці

9. Відкрийте таблицю «Студенти». Заповніть нове поле «*Адреса*», вибираючи значення з *випадного списку* (рис.1.116).

	N1	Прізвище	Ім'я	Дата народжен	Адреса
▶	1	Оу	Міньюе	15.09.2000	м.Київ, вул. ...
	2	Еркен	Мустафа	11.04.1999	м.Київ, вул. Гарматна,
	3	Цао	Фейфей	16.07.2000	м.Київ, пр-т Комарова,
	4	Токіл	Айлінж	09.05.1999	м.Київ, вул. Перова, 67
	5	Ерен	Тимур	19.03.2000	м.Київ, пр-т Комарова,
*	(Счетчик)				м.Київ, пр-т Комарова,

Рис.1.116. Заповнення поля «Адреса»

10. Закрийте таблицю «Студенти», підтвердивши необхідність збереження (кнопка «**Да**») (рис. 1.117).

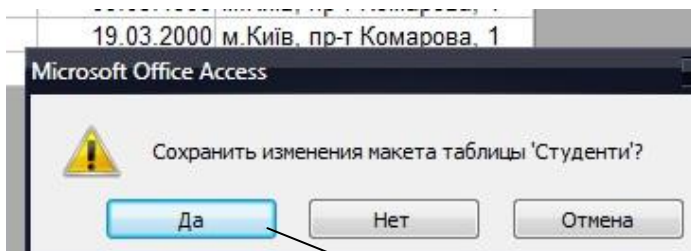


Рис.1.117. Збереження змін макета таблиці

11. Щоб переконатись у наявності зв'язку між таблицями, виконайте *Сервис* → *Схема данных* (рис.1.118).

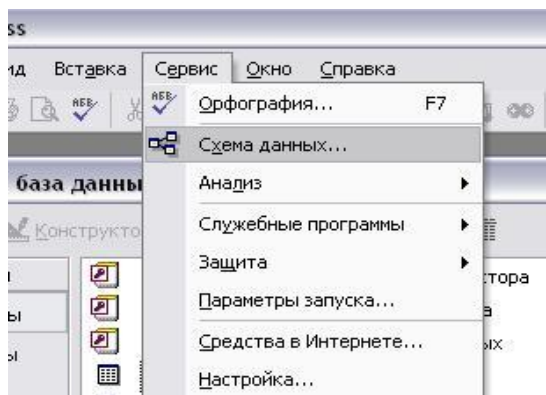


Рис.1.118. Схема даних

13. Якщо зв'язок між таблицями встановлено правильно, ключове поле таблиці «Адреси, телефони» буде з'єднано з полем «Адреса» таблиці «Студенти» (рис.1.119).

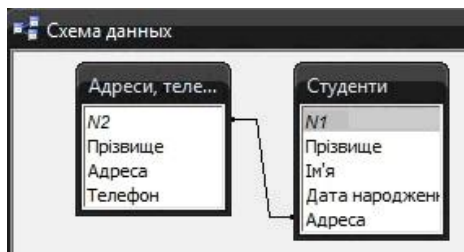


Рис.1.119. Схема даних

Завдання 3. Створіть форму, яка містить поля: «Прізвище», «Ім'я», «Дата народження», «Телефон». Додайте кнопки: «Следующая запись», «Предыдущая запись», «Первая запись», «Последняя запись», «Закреть форму».

План створення форми

1. Перейдіть у розділ «Форми», виберіть команду «Создание формы с помощью мастера» (рис.1.120).

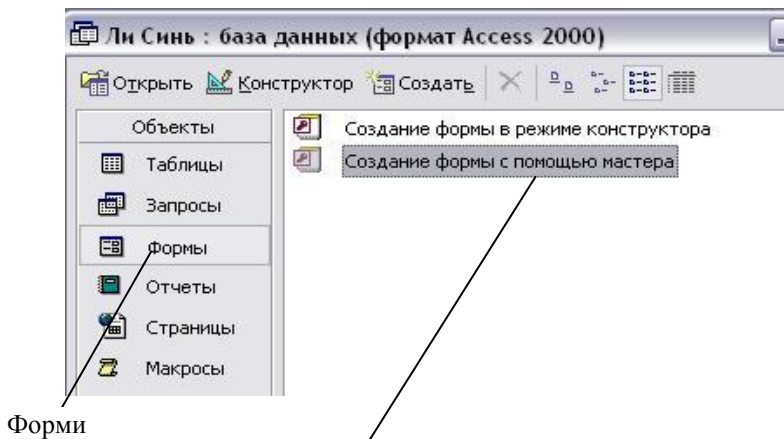


Рис.1.120. Створення форми за допомогою майстра

2. У списку таблиць та запитів виберіть таблицю «Студенти», поля «Прізвище», «Ім'я», «Дата народження» (рис.1.121).

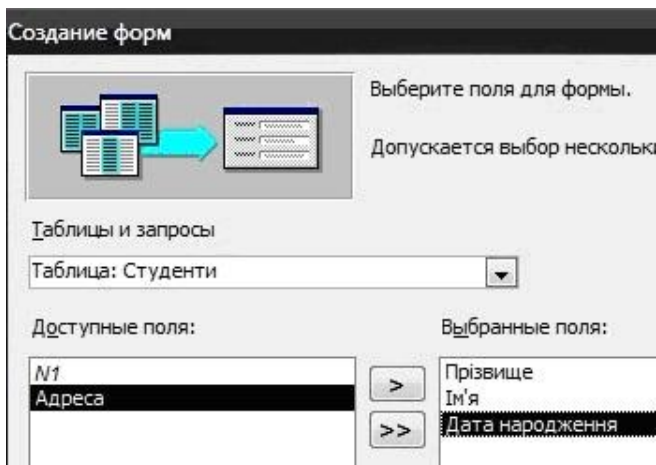


Рис.1.121. Вибір полів таблиці «Студенти»

3. Виберіть таблицю «Адреси, телефони», поле «Телефон» (рис.1.122).

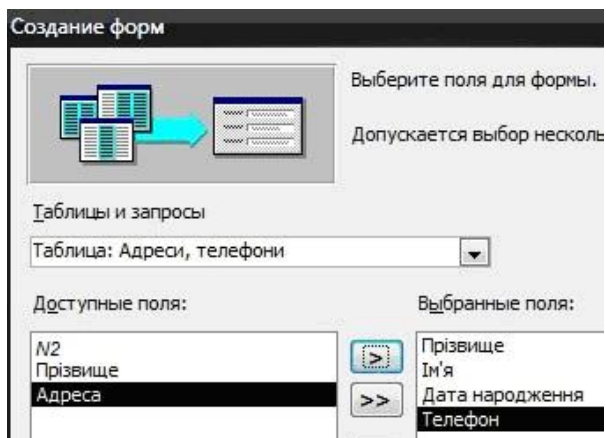


Рис.1.122. Вибір поля «Телефон» таблиці «Адреси, телефони»

4. Натисніть кнопку «Далее» (рис.1.123).

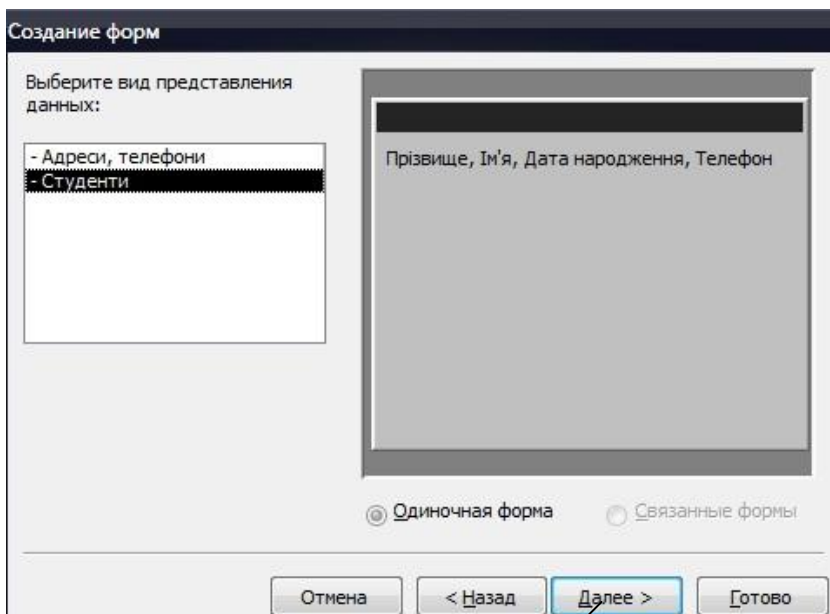


Рис.1.123. Вибір виду подання даних

5. Виберіть вигляд подання форми, наприклад, «Ленточный» (рис.1.124).

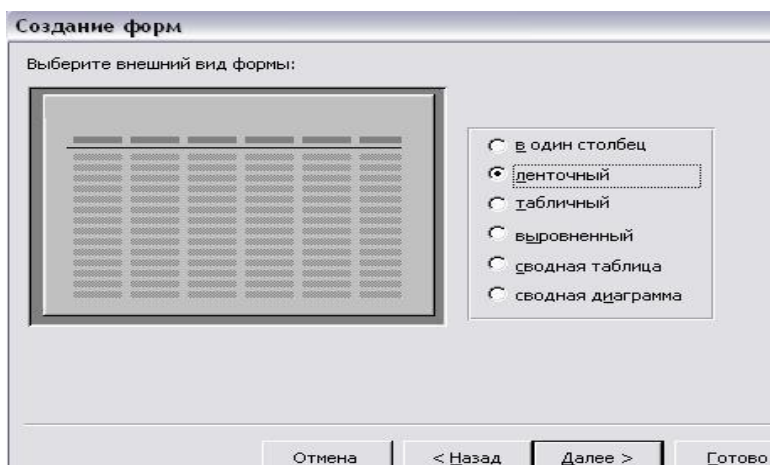


Рис.1.124. Вибір вигляду подання форми

6. Виберіть стиль форми, наприклад, «*Международный*» (рис.1.125).

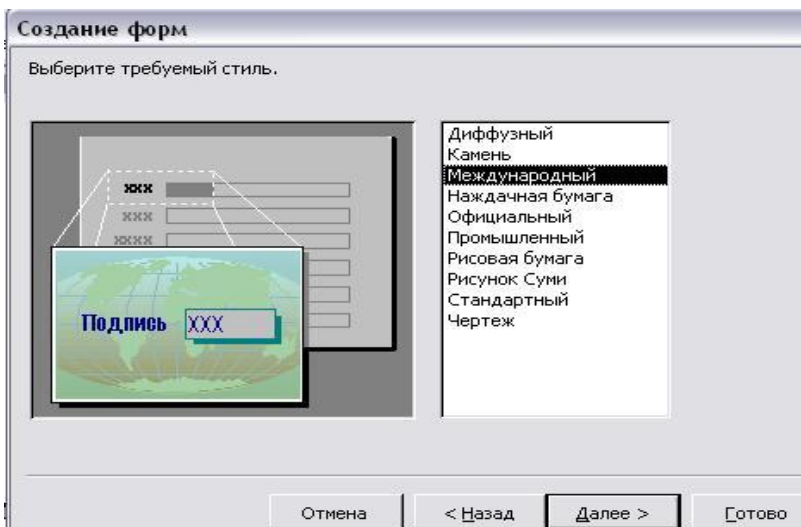


Рис.1.125. Вибір стилю форми

7. Задайте ім'я форми «Форма 1» і натисніть кнопку «*Готово*» (рис.1.126).

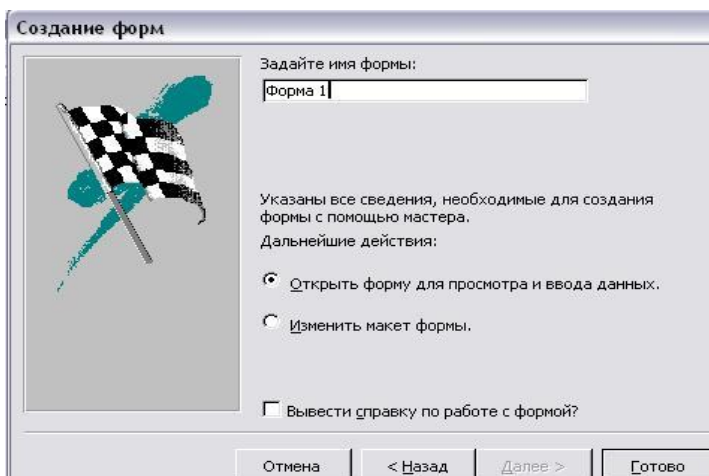


Рис.1.126. Задання імені форми «Форма 1»

8. Відкриється вікно форми (рис.1.127).



Рис.1.127. Форма

9. Закрийте це вікно і відкрийте форму в режимі «Конструктора» (виділіть назву форми «Форма 1» і натисніть кнопку «Конструктор») (рис. 1.128).

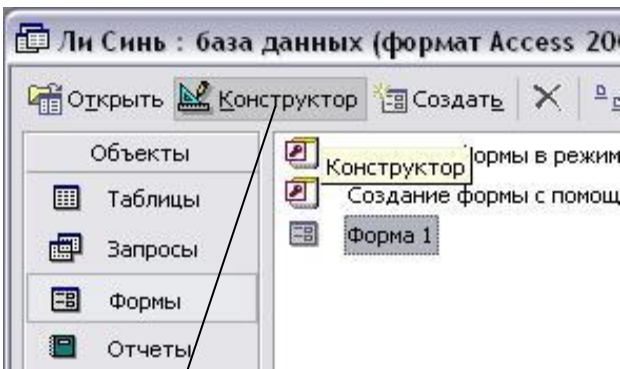


Рис.1.128. Відкриття форми у режимі «Конструктора»

10. На панелі елементів виберіть елемент «Кнопка» і розмістіть його на формі (рис.1.129).

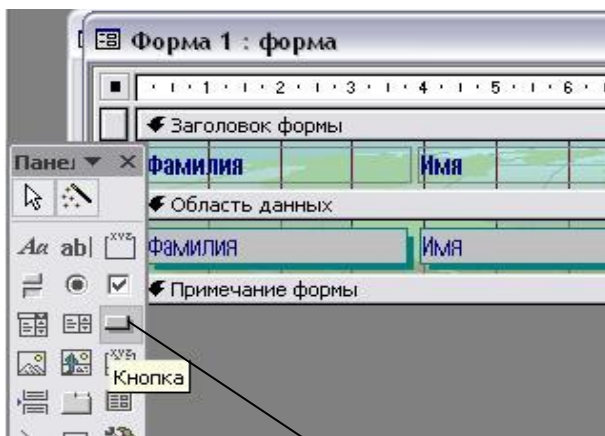


Рис.1.129. Элемент «Кнопка»

11. Виберіть категорію «*Переходы по записям*» і в ній дію «*Следующая запись*» (рис.1.130).

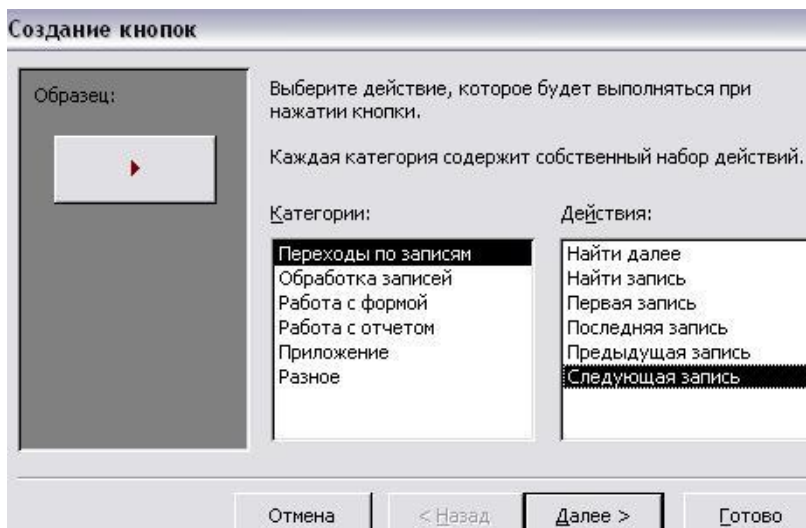


Рис.1.130. Вибір дії кнопки «Следующая запись»

12. У наступному вікні виберіть *зображення* або *підпис* для кнопки і натисніть кнопку «*Далее*» (рис.1.131).

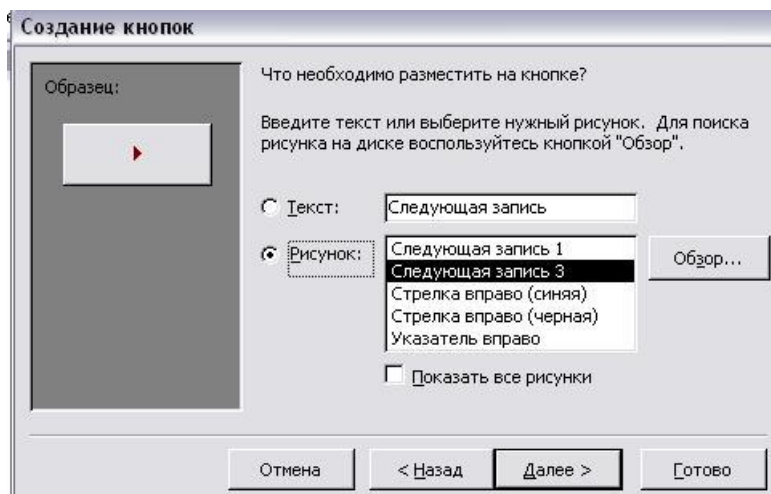


Рис.1.131. Вибір зображення або підпису для кнопки

13. У наступному вікні можна задати ім'я кнопки (краще залишити ім'я *«Кнопка 1»*), натисніть *«Готово»* (рис.1.132).

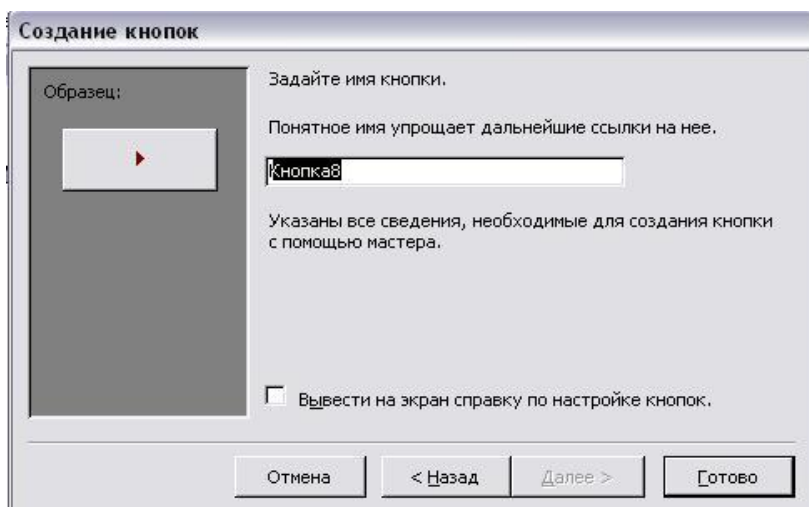


Рис.1.132. Задання імені кнопки

14. Аналогічно створить кнопки *«Предыдущая запись»*, *«Первая запись»*, *«Последняя запись»*.

15. Створіть кнопку «Закри́ть форму», вибравши її дію з категорії «Работа с формой» (рис.1.133).

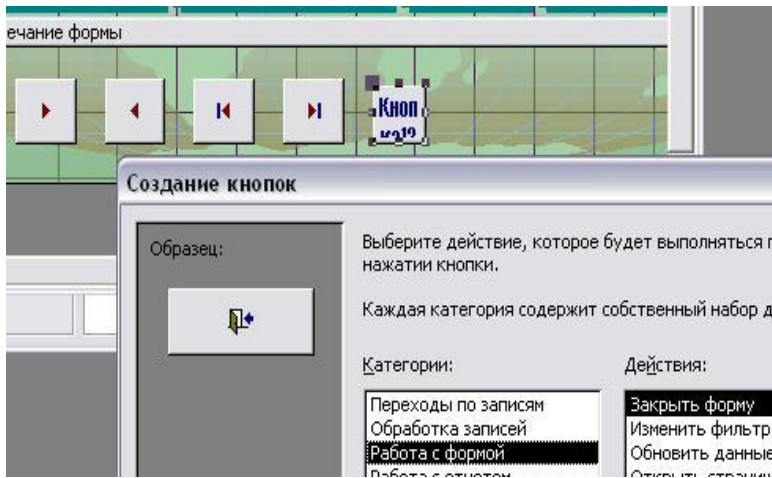


Рис.1.133. Задання дії «Работа с формой → Закри́ть форму»

16. Закри́йте структуру форми, підтвердивши необхідність її збереження (рис.1.134).

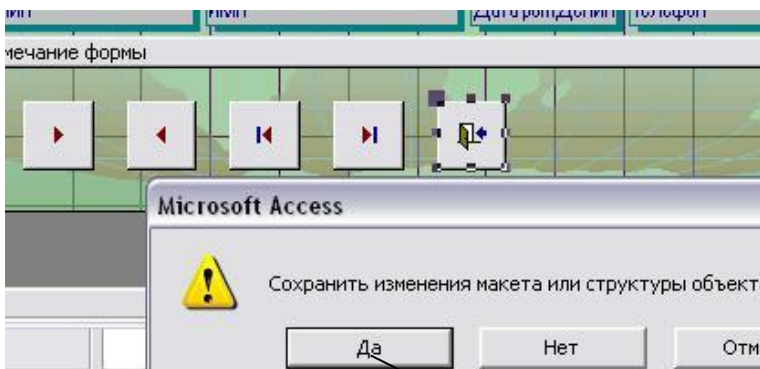


Рис.1.134. Збереження змін у формі

17. Відкри́йте форму, переконайтесь у дієвості кнопок (рис.1.135).



Рис.1.135. Форма з кнопками

Завдання 4. Створіть запит, який містить інформацію про *прізвища, імена та телефони* людей, які народились *пізніше 1.01.2000*.

План створення запиту

1. Перейдіть у розділ *«Запроси»*, виберіть команду *«Создание формы с помощью мастера»* (рис.1.136).

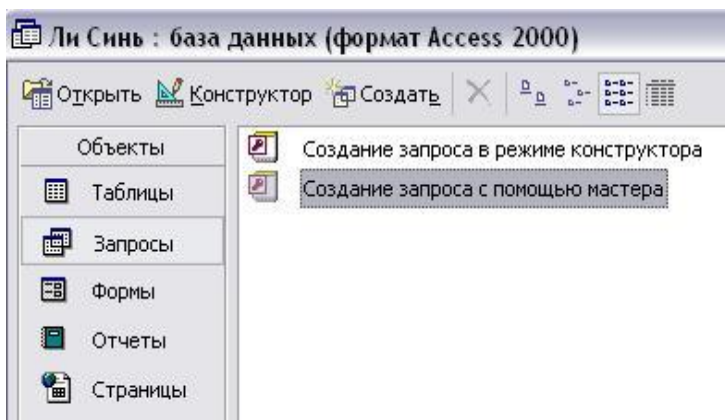


Рис.1.136. Створення запиту за допомогою майстра

2. У списку таблиць та запитів виберіть таблицю «Студенти», в ній виберіть поля «*Прізвище*», «*Ім'я*», «*Дата народження*» (рис.1.137).

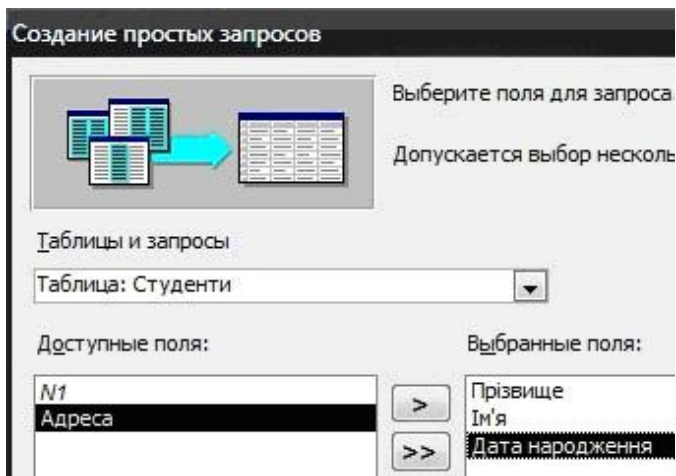


Рис.1.137. Вибір полів таблиці «Студенти» для запиту

3. Виберіть таблицю «Адреси, телефони», в ній виберіть поле «*Телефон*» (рис.1.138).

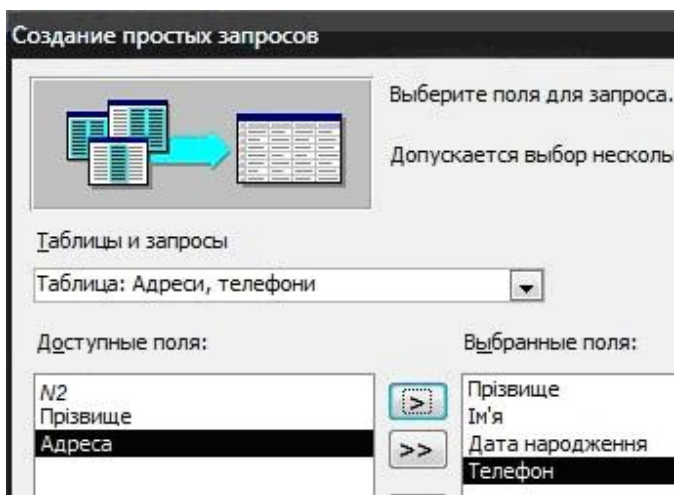


Рис.1.138. Вибір поля «Телефон» таблиці «Адреси, телефони»

4. Задайте ім'я запиту «Запит 1» і натисніть кнопку «Готово» (рис.1.139).

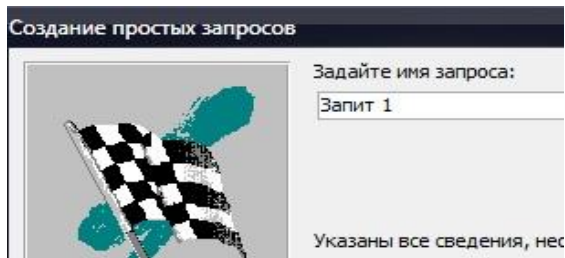


Рис.1.139. Задання імені запиту

5. Відкриється вікно запиту (рис.1.140):

	Прізвище	Ім'я	Дата народження	Телефон
	Оу	Міньюе	15.09.2000	0671112233
	Еркен	Мустафа	11.04.1999	0975556677
	Цао	Фейфей	16.07.2000	0957778888
	Токіп	Айлінж	09.05.1999	0639992277
	Ерен	Тимур	19.03.2000	0674455533

Рис.1.140. Вікно запиту

6. Закрийте це вікно і відкрийте запит у режимі «Конструктора» (рис. 1.141).

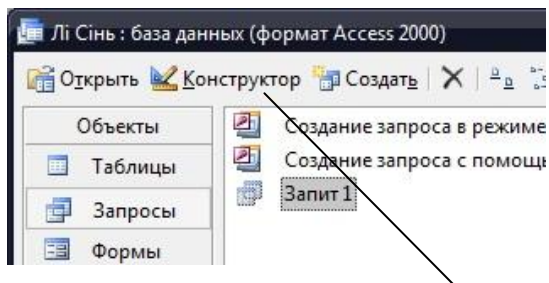


Рис.1.141. Відкриття запиту в режимі Конструктора

7. У рядку «Условие отбора» у полі «Дата народження» введіть «>1.01.2000» (рис. 1.142) і натисніть кнопку «Запуск» («!») на панелі інструментів (рис.1.143).

Адреса	Ім'я	Дата народж		
Телефон				
Поле:	Прізвище	Ім'я	Дата народження	Телефон
Ім'я таблиці:	Студенти	Студенти	Студенти	Адреси, телефони
Сортировка:				
Вивод на екран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Умовне відбору:			>1.01.2000	

Рис.1.142. Умова відбору

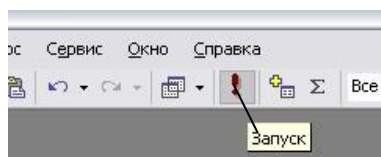


Рис.1.143. Кнопка «Запуск»

8. Відкриється вікно запити, що містить інформацію тільки про людей, які народились після 1.01.2000 (рис.1.144).

Запит 1 : запит на вибірку				
	Прізвище	Ім'я	Дата народжен	Телефон
▶	Оу	Міньюе	15.09.2000	0671112233
	Цао	Фейфей	16.07.2000	0957778888
	Ерен	Тимур	19.03.2000	0674455533
*				

Рис.1.144. Запит на вибірку

1.8. Комп'ютерна мережа. Глобальна мережа Інтернет. Створення web-сторінок

Комп'ютерна мережа – це декілька (або багато) комп'ютерів, об'єднаних між собою за допомогою кабелів, телефонних ліній або радіоканалів, через які передається інформація. **Інтернет** (англ. *Internet* – взаємодія мереж) – це об'єднання нескінченної кількості мереж, яке забезпечує поширення інформації по всій земній кулі.

Перетворення інформації для передавання по мережі забезпечує *мережеве обладнання*, наприклад, *мережева карта*, яка з'єднує комп'ютер з локальною мережею.

Система WWW (World Wide Web – всесвітня павутина) – це глобальна система поширення гіпертекстової інформації. **Гіпертекст** – це текстовий документ, котрий містить *гіперпосилання (links)* на інші документи. Гіпертекстові документи у WWW подаються у вигляді *web-сторінок*.

Web – сторінка – це *електронний документ, написаний мовою розмітки гіпертексту HTML (Hyper Text Markup Language)*. Web – сторінки звичайно містять різну інформацію: текст, графіку, звуки, відео, а також *гіперпосилання*.

HTML – документи – це текстові файли, які містять спеціальні коди розмітки (*теги*). Теги відокремлюються від основного тексту маркерами (< >). <...> – тег *відкривання*, </...> – тег *закривання*. **Контейнери** – це парні теги.

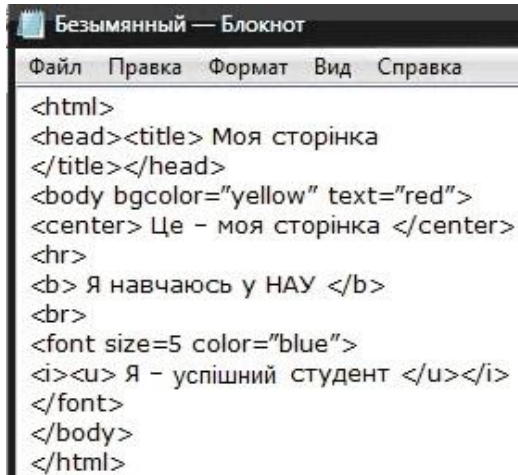
HTML – документи мають розширення *.html*.

Завдання 1. Створіть web-сторінку за зразком:

HTML-код сторінки	Призначення тегів
<html>	Обов'язковий тег документа
<head><title> <i>Моя сторінка</i>	Теги заголовка
</title></head>	
<body bgcolor="yellow"	Колір фону
text="red">	Колір тексту
<center> <i>Це – моя сторінка</i>	Центрування
</center>	
<hr>	Горизонтальна лінія
 <i>Я навчаюсь у НАУ</i> 	Жирний шрифт
 	Абзац
	Розмір та колір шрифту
<i><u> <i>Я – успішний студент</i>	Курсив, підкреслений
</u></i>	
 </body> </html>	Теги закривання

План створення web-сторінки

1. Пуск → Программы → Стандартные → Блокнот (Start → Programs → Accessories → Notepad).
2. Уведіть *html*-код сторінки (рис. 1.145).



```
Безымянный — Блокнот
Файл  Правка  Формат  Вид  Справка
<html>
<head><title> Моя сторінка
</title></head>
<body bgcolor="yellow" text="red">
<center> Це - моя сторінка </center>
<hr>
<b> Я навчаюсь у НАУ </b>
<br>
<font size=5 color="blue">
<i><u> Я - успішний студент </u></i>
</font>
</body>
</html>
```

Рис.1.145. Html-код сторінки

3. Збережіть документ (Файл → Сохранить) (рис.1.146).

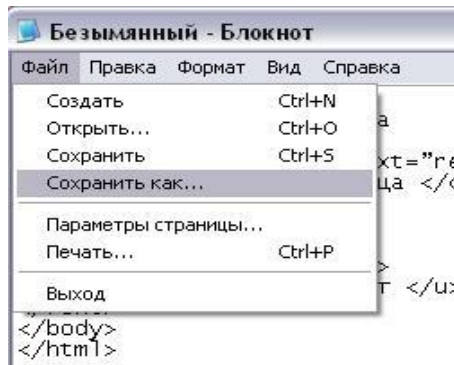


Рис.1.146. Збереження документа

3. Задайте *ім'я* документа, наприклад, *myweb1.txt* (рис.1.147).

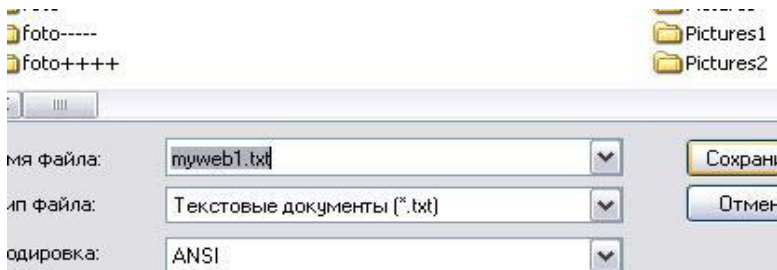


Рис.1.147. Задання імені документа

4. Переіменуйте документ, змінивши його розширення з *txt* на *html* (в контекстному меню файла виберіть команду «*Переіменувати*») (рис.1.148, 1.149).

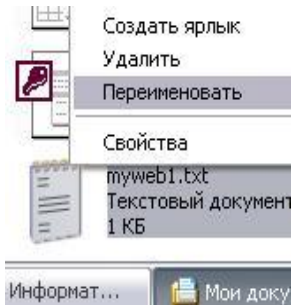


Рис.1.148. Переіменування файла



Рис. 1.149. Зміна розширення

5. Підтвердіть намір змінити розширення (натисніть кнопку «*Да*») (рис.1.150).



Рис.1.150. Підтвердження наміру змінити розширення

6. Після зміни розширення зміниться піктограма файла (рис. 1.151).



Рис.1.151. Файл myweb1.html

7. Сторінка відкриється у вікні браузера (рис.1.152).



Рис.1.152. Web-сторінка

Завдання 2. Створіть web-сторінку за зразком. Замість «...» треба вказати *шлях до файла*, який містить зображення.

HTML-код сторінки	Призначення тегів
<pre><html> <head><title> Web 2 </title></head> <body background = "... " bgproperties="fixed"> <i><u> Малюнок №1 </u></i> <hr> <marquee behavior="alternate" loop=2 scrollamount=9> </marquee> Гіперпосилання №1 </body></html></pre>	<ul style="list-style-type: none"> - Фоновий малюнок - Відміна прокручування - Анімація - Відштовхування від країв - Кількість проходжень - Швидкість - Малюнок - Вирівнювання - Ширина - Висота - Ширина рамки - Гіперпосилання

Завдання 3. Додайте два рухомих зображення з підписом та два гіперпосилання.

ЗАПИТАННЯ

1. Що таке комп'ютерна мережа? Що таке Інтернет?
2. Що таке HTML-документи?
3. Що таке гіпертекст? Що таке контейнери?

ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 2

Завдання 1. Створіть базу даних, яка містить таблиці:

Варіант 1 *Таблиця 1. Студенти*

<i>№1</i>	Прізвище	Ім'я	Країна
1	Чжу	Кай	Китай
2	Лі	Сінь	Китай
3	Ерхан	Аяр	Туреччина
4	Фан	Вей	В'єтнам
5	Лі	Лян	В'єтнам

Таблиця 2. Предмети

<i>№2</i>	Прізвище	Хімія	Фізика
1	Чжу	5	4
2	Лі	4	4
3	Ерхан	5	5
4	Фан	3	3
5	Лі	4	4

Варіант 2 *Таблиця 1. Професії*

<i>№1</i>	Прізвище	Ім'я	Професія
1	Ле Хоанг	Нам	Інженер
2	Фан	Лінфем	Викладач
3	Лі	Цзе	Лікар
4	Цзя	Пен	Інженер
5	Цзян	Лімін	Лікар

Таблиця 2. Місце роботи, зарплата

<i>№2</i>	Прізвище	Місце роботи	Зарплата, грн
1	Ле Хоанг	Завод	4000
2	Фан	Інститут	4400
3	Лі	Лікарня	5000
4	Цзя	Інститут	3900
5	Цзян	Лікарня	4780

Варіант 3*Таблиця 1. Студенти*

№1	Прізвище	Ім'я	Країна
1	Чжао	Шуань	Китай
2	Лі	Сінь	Китай
3	Ерен	Озгюр	Туреччина
4	Ле Хоанг	Нам	В'єтнам
5	Ле Хонг	Лонг	В'єтнам

Таблиця 2. Освіта

№2	Прізвище	Дата народження	Освіта
1	Чжао	5.09.1999	Вища
2	Лі	4.08.2000	Середня
3	Ерен	5.05.2000	Середня
4	Ле Хоанг	3.06.1998	Вища
5	Ле Хонг	4.12.1999	Середня

Варіант 4*Таблиця 1. Студенти*

№1	Прізвище	Ім'я	Телефон
1	Лі	Юань	2344565
2	Цзін	Яолян	5465476
3	Ван	Сюань	4565677
4	Дуань	Ченьхуа	4576588
5	Фан	Ле	3437767

Таблиця 2. Міста

№2	Прізвище	Місто	Дата народження
1	Лі	Київ	4.09.1999
2	Цзін	Пекін	4.12.1998
3	Ван	Київ	5.03.2000
4	Дуань	Харків	3.11.1999
5	Фан	Одеса	14.10.2000

Варіант 5*Таблиця 1. Студенти*

№1	Прізвище	Ім'я	Країна
1	Шен	Тюгберк	Турція
2	Лі	Сінь	Китай
3	Ерхан	Аяр	Туреччина
4	Уяник	Оркун	Туреччина
5	Лі	Лян	В'єтнам

Таблиця 2. Місяць та рік народження

N2	Прізвище	Місяць народження	Рік народження
1	Шен	Липень	1999
2	Ли	Липень	2000
3	Ерхан	Березень	1999
4	Уяник	Березень	1999
5	Лі	Квітень	2000

Варіант 6

Таблиця 1. Студенти

NI	Прізвище	Ім'я	Інформатика
1	Джаффарін	Диндар	4
2	Угуреллі	Озгюр	5
3	Цень	І	5
4	Фан	Вей	3
5	Лі	Лян	4

Таблиця 2. Оцінки

N2	Прізвище	Математика	Хімія
1	Джаффарін	5	5
2	Угуреллі	4	4
3	Цень	5	5
4	Фан	3	3
5	Лі	4	4

Варіант 7

Таблиця 1. Студенти

NI	Прізвище	Ім'я	Українська мова
1	Ічтен	Ефе	5
2	Танко	Халілу	3
3	Салех	Давид	4
4	Фан	Вей	3
5	Цянь	Дань	4

Таблиця 2. Оцінки

N2	Прізвище	Хімія	Математика
1	Ічтен	5	3
2	Танко	4	3
3	Салех	5	5
4	Фан	3	3
5	Цянь	4	4

Варіант 8*Таблиця 1. Студенти*

№1	Прізвище	Ім'я	Місяць народження
1	Топал	Ахмет	Травень
2	Шахін	Ілке	Вересень
3	Гун	Чуань	Березень
4	Чень	Бо	Січень
5	Лі	Лян	Травень

Таблиця 2. Місце народження

№2	Прізвище	Країна	Місто
1	Топал	Туреччина	Стамбул
2	Шахін	Туреччина	Анталія
3	Гун	Китай	Пекін
4	Чень	Китай	Пекін
5	Лі	Китай	Пекін

Варіант 9*Таблиця 1. Студенти*

№1	Прізвище	Ім'я	Рік народження
1	Се	Вей	2000
2	Абді	Абдулле	1999
3	Яшар	Чагатай	1999
4	Фан	Вей	2000
5	Аном	Терду	1998

Таблиця 2. Країна, освіта

№2	Прізвище	Країна	Освіта
1	Се	Китай	Вища
2	Абді	Ірак	Середня
3	Яшар	Туреччина	Вища
4	Фан	Китай	Середня
5	Аном	Кенія	Середня

Варіант 10*Таблиця 1. Вік студентів*

№1	Прізвище	Ім'я	Вік
1	Доган	Гунджай	24
2	Лі	Сінь	25
3	Захарен	Салімі	25
4	Атаа	Літ	23
5	Лі	Лян	24

Таблиця 2. Професія, зарплата

N2	Прізвище	Професія	Зарплата, грн
1	Доган	Лікар	4000
2	Лі	Інженер	4000
3	Захарен	Музикант	5000
4	Атаа	Викладач	3000
5	Лі	Економіст	4000

Варіант 11

Таблиця 1. Товари

NI	Назва	Ціна, грн	Кількість
1	Холодильник	2500	3
2	Праска	560	10
3	Телевізор	3500	4
4	Монітор	1500	10
5	Пилосос	2500	4

Таблиця 2. Фірма, рік випуску

N2	Назва	Фірма	Рік випуску
1	Холодильник	Indesit	2016
2	Праска	Tefal	2015
3	Телевізор	Samsung	2016
4	Монітор	Lg	2014
5	Пилосос	Lg	2016

Варіант 12

Таблиця 1. Назви товарів

NI	Назва	Фірма	Країна
1	Ноутбук	Samsung	Китай
2	Сканер	Canon	Китай
3	Флеш-диск	Lg	Туреччина
4	Монітор	Samsung	Таїланд
5	Фотоапарат	Canon	Таїланд

Таблиця 2. Ціни

N2	Назва	Рік випуску	Ціна, грн
1	Ноутбук	2016	4000
2	Сканер	2016	500
3	Флеш-диск	2015	150
4	Монітор	2016	1600
5	Фотоапарат	2014	1600

Варіант 13*Таблиця 1. Товари*

№1	Назва	Місяць	Кількість
1	Принтер	Травень	34
2	Сканер	Серпень	56
3	Монітор	Липень	37
4	Ноутбук	Липень	14
5	Клавіатура	Серпень	76

Таблиця 2. Цена, сорт

№2	Назва	Ціна	Сорт
1	Принтер	500	1
2	Сканер	400	1
3	Монітор	1500	2
4	Ноутбук	6000	1
5	Клавіатура	250	2

Варіант 14*Таблиця 1. Зарплата за січень*

№1	Прізвище	Ім'я	Січень
1	Радам	Алаа	2444
2	Ян	Таньян	7000
3	Лю	Хао	4500
4	Чжао	Цзіянь	7890
5	Меше	Джамі	4250

Таблиця 2. Зарплата за лютий, березень

№2	Прізвище	Лютий	Березень
1	Радам	5000	4450
2	Ян	4000	4765
3	Лю	5000	5980
4	Чжао	3000	3567
5	Меше	4000	4098

Варіант 15*Таблиця 1. Оцінки з математики*

№1	Прізвище	Ім'я	Математика
1	Дуань	Юнь	4
2	Цай	Хунбо	5
3	Ерхан	Аяр	3
4	Ка	Лі	5
5	Лі	Чень	5

Таблиця 2. Інформатика, фізика

N2	Прізвище	Інформатика	Фізика
1	Дуань	5	4
2	Цай	4	4
3	Ерхан	5	5
4	Ка	3	3
5	Лі	4	4

Варіант 16

Таблиця 1. Адреси студентів

№1	Прізвище	Ім'я	Адреса
1	Чжан	Да	вул.Польова, 8
2	Лі	Сінь	вул. Гарматна, 11
3	Юй	Тін	вул. Комарова, 15
4	Лю	Сен	вул. Комарова, 16
5	Чжоу	Тао	вул. Польова, 8

Таблиця 2. Зарплата

N2	Прізвище	Рік народження	Зарплата, грн
1	Чжан	1999	6700
2	Лі	1998	4000
3	Юй	2000	5000
4	Лю	1999	3340
5	Чжоу	1999	4980

Варіант 17

Таблиця 1. Студенти

№1	Прізвище	Ім'я	Телефон
1	Лю	Хао	4567654
2	Лі	Сінь	5667546
3	Чжоу	Кунь	1134565
4	Еркен	Мустафа	7345656
5	Аксу	Бурак	1135445

Таблиця 2. Професії

N2	Прізвище	Професія	Країна
1	Лю	Пілот	Китай
2	Лі	Інженер	Китай
3	Чжоу	Інженер	Китай
4	Еркен	Пілот	Туреччина
5	Аксу	Економіст	Туреччина

Варіант 18*Таблиця 1. Оцінки з хімії*

№1	Прізвище	Ім'я	Хімія
1	Бедуї	Кайєс	3
2	Садегі	Анахіта	4
3	Айдін	Хасан	3
4	Радам	Алаа	4
5	Лі	Лян	5

Таблиця 2. Математика, інформатика

№2	Прізвище	Математика	Інформатика
1	Бедуї	5	4
2	Садегі	4	4
3	Айдін	5	5
4	Радам	3	3
5	Лі	4	4

Варіант 19*Таблиця 1. Студенти*

№1	Прізвище	Ім'я	Рік народження
1	Лю	Мяо	1999
2	Діксіт	Акшей	2000
3	Ерхан	Аяр	1999
4	Фан	Вей	2000
5	Лі	Лян	1999

Таблиця 2. Телефони

№2	Прізвище	Країна	Телефон
1	Лю	Китай	4544456
2	Діксіт	Іран	4678999
3	Ерхан	Туреччина	5005456
4	Фан	Китай	3223443
5	Лі	Китай	4009876

Варіант 20*Таблиця 1. Товари*

№1	Назва	Фірма	Рік випуску
1	Праска	Vitek	2015
2	Лампа	Vitek	2016
3	Світильник	Vitek	2016
4	Дзеркало	Мойдодир	2015
5	Умивальник	Мойдодир	2014

Таблиця 2. Країна, місто виробництва

N2	Назва	Країна	Місто
1	Праска	Китай	Пекін
2	Лампа	Китай	Пекін
3	Світильник	Китай	Пекін
4	Дзеркало	Україна	Київ
5	Умивальник	Україна	Київ

Завдання 2. Зв'яжіть таблиці за допомогою перенесення з «Таблиці 2» в «Таблицю 1» поля:

<i>Варіант 1</i>	Хімія
<i>Варіант 2</i>	Зарплата
<i>Варіант 3</i>	Дата народження
<i>Варіант 4</i>	Місто
<i>Варіант 5</i>	Місяць народження
<i>Варіант 6</i>	Хімія
<i>Варіант 7</i>	Хімія
<i>Варіант 8</i>	Країна
<i>Варіант 9</i>	Освіта
<i>Варіант 10</i>	Професія
<i>Варіант 11</i>	Фірма
<i>Варіант 12</i>	Рік випуску
<i>Варіант 13</i>	Ціна
<i>Варіант 14</i>	Лютий
<i>Варіант 15</i>	Фізика
<i>Варіант 16</i>	Рік народження
<i>Варіант 17</i>	Професія
<i>Варіант 18</i>	Математика
<i>Варіант 19</i>	Телефон
<i>Варіант 20</i>	Місто

Завдання 3. Створіть форму, яка містить наведені нижче поля таблиць. Додайте кнопки: «*Наступний запис*», «*Попередній запис*», «*Перший запис*», «*Останній запис*», «*Закрити форму*».

Варіант 1	Прізвище, ім'я, країна, фізика
Варіант 2	Прізвище, ім'я, професія, місце роботи
Варіант 3	Прізвище, ім'я, країна, освіта
Варіант 4	Прізвище, ім'я, телефон, дата народження
Варіант 5	Прізвище, ім'я, країна, рік народження
Варіант 6	Прізвище, ім'я, інформатика, математика
Варіант 7	Прізвище, ім'я, українська мова, математика
Варіант 8	Прізвище, ім'я, місяць народження, місто
Варіант 9	Прізвище, ім'я, рік народження, країна
Варіант 10	Прізвище, ім'я, вік, зарплата
Варіант 11	Назва, ціна, кількість, рік випуску
Варіант 12	Назва, фірма, країна, ціна
Варіант 13	Назва, місяць, кількість, сорт
Варіант 14	Прізвище, ім'я, січень, березень
Варіант 15	Прізвище, ім'я, математика, інформатика
Варіант 16	Прізвище, ім'я, адреса, зарплата
Варіант 17	Прізвище, ім'я, телефон, країна
Варіант 18	Прізвище, ім'я, хімія, інформатика
Варіант 19	Прізвище, ім'я, рік народження, країна
Варіант 20	Назва, фірма, рік випуску, країна

Завдання 4. Створіть запит, який містить ті ж поля, що і форма, а також виводить таку інформацію:

Варіант 1	Студенти, які приїхали з Китаю
Варіант 2	Люди, у яких професія – інженер
Варіант 3	Люди, у яких вища освіта
Варіант 4	Студенти, які народились після 3.03.1999
Варіант 5	Студенти, які приїхали з Туреччини
Варіант 6	Студенти, у яких оцінка з інформатики «4»
Варіант 7	Студенти, у яких оцінка з математики «3»
Варіант 8	Студенти, які живуть в Пекіні
Варіант 9	Студенти, які народились у 1999 році
Варіант 10	Люди, яким 24 роки
Варіант 11	Товари, рік випуску яких – 2016
Варіант 12	Товари, виготовлені фірмою «Samsung»
Варіант 13	Товари, виготовлені у липні

Варіант 14	Товари, ціна яких вища 1000 грн
Варіант 15	Студенти, у яких оцінка з математики «5»
Варіант 16	Студенти, які народились пізніше 2.02.1999
Варіант 17	Студенти, які приїхали з Китаю
Варіант 18	Студенти, у яких оцінка з хімії «4»
Варіант 19	Студенти, які приїхали з Ірану
Варіант 20	Товари, виготовлені фірмою «Vitek»

Модуль 2

ОСНОВИ АЛГОРИТМІЗАЦІЇ ТА ПРОГРАМУВАННЯ

2.1. Типові структури алгоритмів

Мета вивчення програмування – навчитись розв’язувати різні задачі (математичні, економічні, фізичні, інженерні та інші) за допомогою комп’ютера. Існує єдиний план для розв’язання будь-якої задачі. Він складається з 6 етапів.

- 1 етап.** Формулювання умови задачі.
- 2 етап.** Створення моделі задачі. *Математична модель* – це опис задачі за допомогою формул.
- 3 етап.** Вибір методу розв’язання.
- 4 етап.** Складання алгоритму і переклад його мовою програмування.
- 5 етап.** Виконання програми на комп’ютері.
- 6 етап.** Аналіз результатів.

Алгоритм – це скінченна послідовність команд, які треба виконати над вхідними даними для отримання результату.

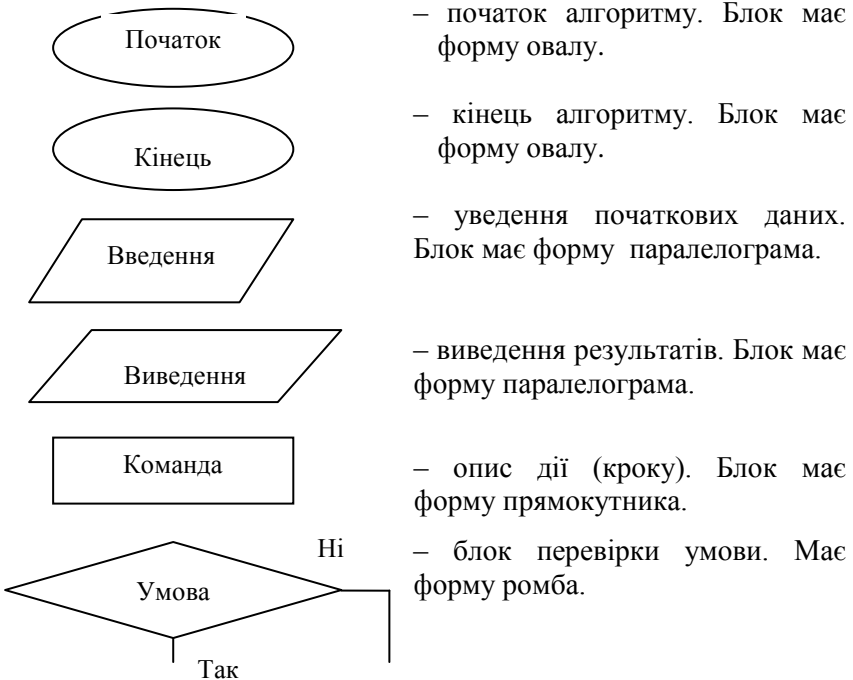
Приклад. Складіть алгоритм розв’язання квадратного рівняння $ax^2 + bx + c = 0$, $a \neq 0$.

Алгоритм

1. Задати a, b, c .
2. Обчислити $D := b^2 - 4ac$.

3. Якщо $D \geq 0$, то перейти до кроку 4, інакше перейти до кроку 7.
4. Обчислити $d = \sqrt{D}$.
5. Обчислити $x_1 = (-b - d)/2a$, $x_2 = (-b + d)/2a$.
6. Записати відповідь: x_1, x_2 . Перейти до кроку 8.
7. Записати відповідь: «Дійсних розв'язків немає». Перейти до кроку 8.
8. Кінець.

Алгоритм можна подати *графічно*, тобто у вигляді *схем*, які складаються з *блоків*, з'єднаних лініями або стрілками. Блоки позначають так:



Блоки з'єднуються *лініями потоків* передачі інформації. Вони напрямлені *згори-вниз* (\downarrow) і *зліва-направо* (\rightarrow). Якщо напрям інший, то лінія повинна мати *стрілку*. Блок «Умова»

має два виходи – вниз, якщо умова виконується, і вправо (або вліво), якщо не виконується.

Базові структури алгоритмів – це способи керування процесами обробки даних. Є три основні базові алгоритмічні структури: *лінійна, розгалуження та циклічна*.

1. Лінійний (простий) алгоритм – це алгоритм, який складається з *послідовності простих команд* (рис.2.1).

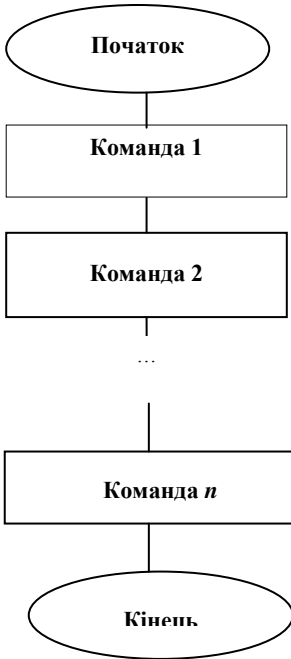


Рис. 2.1. Лінійний алгоритм

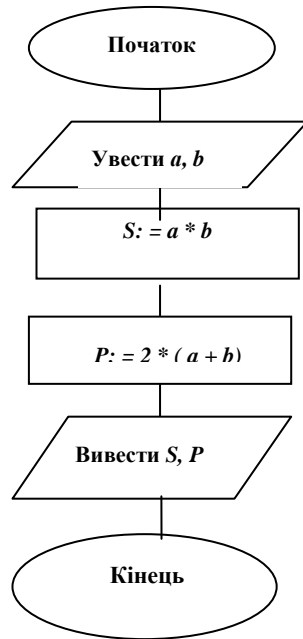
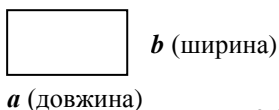


Рис. 2.2. Приклад лінійного алгоритму

Приклад. Складіть алгоритм у вигляді блок-схеми для обчислення *площі* і *периметра* прямокутника зі сторонами a і b (рис. 2.2, 2.3).



$$S = ab - \text{площа}$$

$$P = 2 \cdot (a + b) - \text{периметр}$$

Рис. 2.3. Прямокутник

2. Алгоритм розгалуження – це алгоритм, який містить *умову* (логічний вираз) (рис.2.4).

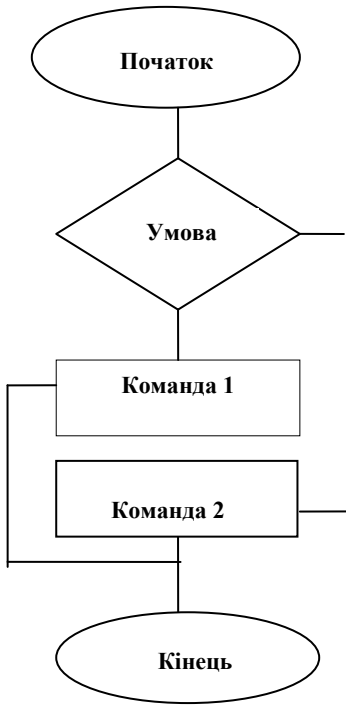


Рис. 2.4. Алгоритм розгалуження

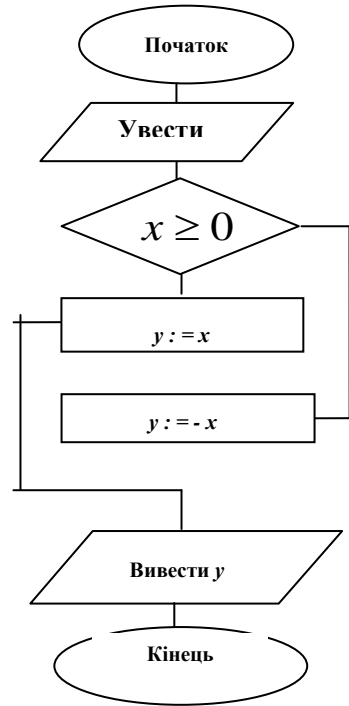


Рис. 2.5. Приклад

Приклад. Складіть алгоритм у вигляді блок-схеми для обчислення значень функції $y = |x|$ $\left(y = \begin{cases} x, & \text{якщо } x \geq 0, \\ -x, & \text{якщо } x < 0. \end{cases} \right)$ (рис. 2.5).

3. Циклічні алгоритми (повторення) – це алгоритми, які забезпечують *повторне виконання* команд скінченну кількість разів.

Повторення з передумовою (цикл - «поки») (рис. 2.6).

Приклад. Складіть алгоритм у вигляді блок-схеми для обчислення суми всіх натуральних чисел з відрізка [45; 100] ($i := 45, 46, 47, \dots, 100$; $S := 45 + 46 + 47 + \dots + 100$) (рис. 2.7).

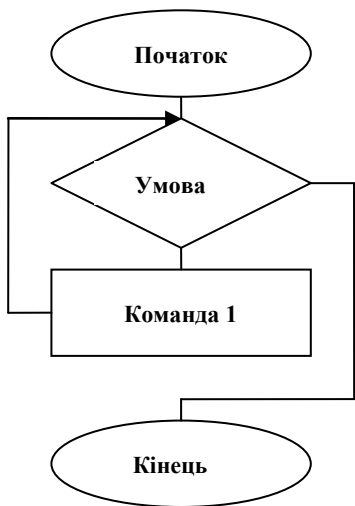


Рис. 2.6. Алгоритм повторення з передумовою (цикл-«поки»)

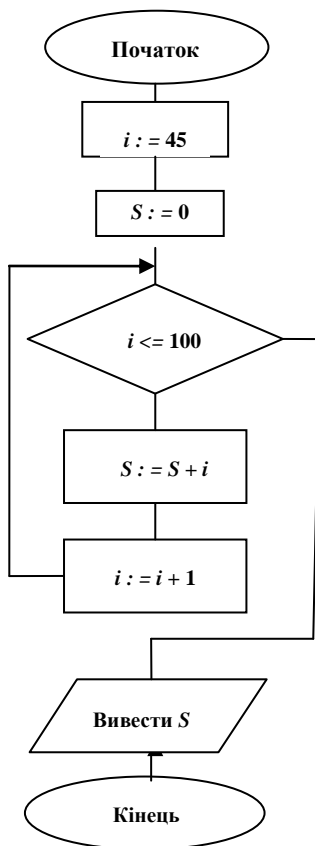


Рис. 2.7. Приклад передумови

Повторення з післяумовою (цикл - «до») (рис. 2.8).

Приклад. Складіть алгоритм у вигляді блок-схеми для обчислення добутку всіх натуральних чисел з відрізка $[8; 15]$ ($i := 8, 9, 10, \dots, 15$, $d := 1 \cdot 8 \cdot 9 \cdot \dots \cdot 15$) (рис. 2.9).

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Які є основні етапи розв'язання будь-якої задачі за допомогою комп'ютера? Що таке алгоритм?
2. Які ви знаєте три основні види алгоритмів?
3. Чим відрізняються циклічні алгоритми з передумовою і з післяумовою?

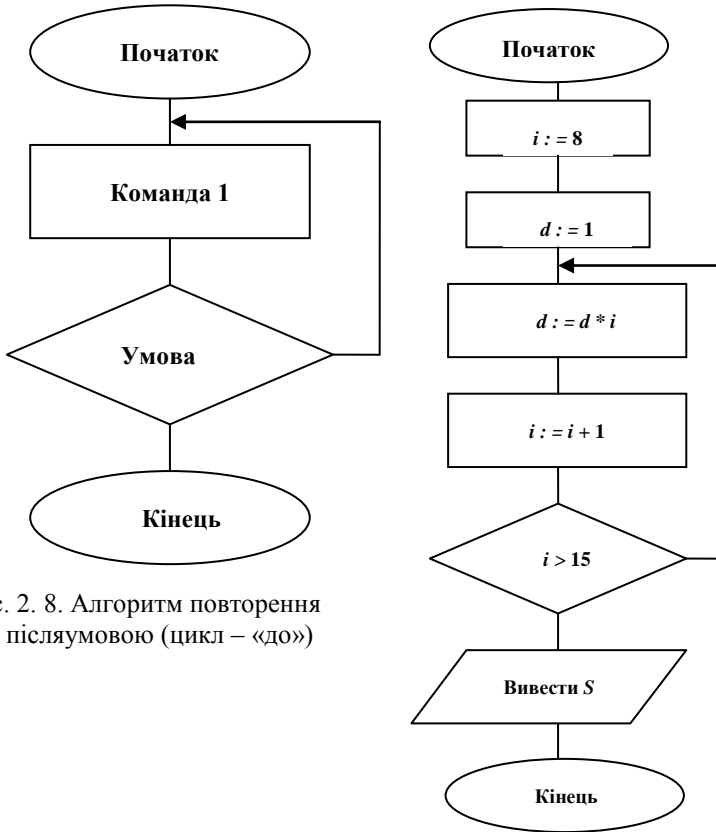


Рис. 2. 8. Алгоритм повторення з післяумовою (цикл – «до»)

Рис. 2.9. Приклад алгоритму з післяумовою

ЗАВДАННЯ

1. Складіть алгоритм для розв'язання квадратного рівняння $ax^2 + bx + c = 0$, $a \neq 0$.
2. Складіть алгоритм для обчислення значень функції:
 - а) $y = \frac{4x-2}{x+5} + \frac{1}{x}$;
 - б) $z = \begin{cases} 5x^3 - x + 6, & \text{якщо } x \geq -1; \\ 3x^2 + 8x - 9, & \text{якщо } x < -1. \end{cases}$
3. Складіть алгоритм для обчислення суми та добутку всіх натуральних чисел з відрізка $[12; 23]$.
4. Складіть алгоритм для обчислення кількості всіх натуральних чисел з відрізка $[17; 100]$, що діляться на 3.

ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 3

Завдання 1. Складіть алгоритм, за допомогою якого можна обчислити:

Варіант 1	площу S трикутника, якщо відомі його сторона a і відповідна їй висота h . Формула: $S = \frac{1}{2}ah$.
Варіант 2	об'єм V призми, якщо відомі площа її основи S і висота H . Формула: $V = SH$.
Варіант 3	об'єм V піраміди, якщо відомі площа її основи S і висота H . Формула: $V = \frac{1}{3}SH$.
Варіант 4	площу S трапеції, якщо відомі її основи a , b і висота h . Формула: $S = \frac{a+b}{2}h$.
Варіант 5	площу S трикутника, якщо відомі його сторони a і b і кут α між ними. Формула: $S = \frac{1}{2}ab\sin\alpha$.
Варіант 6	об'єм V прямокутного паралелепіпеда, якщо відомі його сторони a , b і c . Формула: $V = abc$.
Варіант 7	висоту H піраміди, якщо відомі площа її основи S і об'єм V . Формула: $H = \frac{3V}{S}$.
Варіант 8	площу S чотирикутника, якщо відомі його діагоналі d_1 і d_2 і кут між ними α . Формула: $S = d_1d_2 \sin\alpha$.
Варіант 9	площу S прямокутного трикутника, якщо відомі його сторони a і b . Формула: $S = \frac{1}{2}ab$.
Варіант 10	об'єм V конуса, якщо відомі радіус його основи R , число π і висота H . Формула: $V = \frac{1}{3}\pi R^2 H$.
Варіант 11	об'єм V куба, якщо відомі його сторона a . Формула: $V = a^3$.
Варіант 12	висоту h трапеції, якщо відомі її основи a і b і площа S . Формула: $h = \frac{2S}{a+b}$.

Варіант 13	сторону a трикутника, якщо відомі його сторона b , площа S і кут α між цими сторонами. Формула: $a = \frac{2S}{b \sin \alpha}$.
Варіант 14	сторону c прямокутного паралелепіпеда, якщо відомі його сторони a , b і об'єм V . Формула: $c = \frac{V}{ab}$.
Варіант 15	площу S піраміди, якщо відомі його об'єм V і висота H . Формула: $S = \frac{3V}{H}$.
Варіант 16	діагональ d_1 чотирикутника, якщо відомі його діагональ d_2 , площа S і кут між діагоналями α . Формула: $d_1 = \frac{S}{d_2 \sin \alpha}$.
Варіант 17	сторону a трикутника, якщо відомі відповідна їй висота h і площа трикутника S . Формула: $a = \frac{2S}{h}$.
Варіант 18	площу основи S призми, якщо відомі її об'єм V і висота H . Формула: $S = \frac{V}{H}$.
Варіант 19	висоту H конуса, якщо відомі радіус його основи R , число π і об'єм V . Формула: $H = \frac{3V}{\pi R^2}$.
Варіант 20	основу a трапеції, якщо відомі її основа b , висота h і площа S . Формула: $a = \frac{2S}{h} - b$.

Завдання 2. Складіть алгоритм для обчислення значень функції:

Варіант 1	$y = \begin{cases} 3x^2 - 4x + 5, & \text{якщо } x \geq 6; \\ 2x^2 + 3x - 6, & \text{якщо } x < 6. \end{cases}$
Варіант 2	$z = \begin{cases} 4x^3 + 2x + 2, & \text{якщо } x \geq 8; \\ 8x^2 + x - 1, & \text{якщо } x < 8. \end{cases}$
Варіант 3	$z = \begin{cases} 8y^2 - 4y + 9, & \text{якщо } y \geq -7; \\ y^2 + 5y - 1, & \text{якщо } y < -7. \end{cases}$

Варіант 4	$y = \begin{cases} 4z^2 - 7z + 1, & \text{якщо } z \geq 1; \\ z^2 + 3z - 3 & \text{якщо } z < 1. \end{cases}$
Варіант 5	$z = \begin{cases} 3x^3 - 4x + 6, & \text{якщо } x \geq -3; \\ x^2 + 7x - 2, & \text{якщо } x < -3. \end{cases}$
Варіант 6	$z = \begin{cases} 8x^3 + x - 4, & \text{якщо } x \geq -1; \\ 5x^2 + 6x - 9, & \text{якщо } x < -1. \end{cases}$
Варіант 7	$y = \begin{cases} 7x^2 - y + 9, & \text{якщо } x \geq -5; \\ 4x^2 + 3x - 2, & \text{якщо } x < -5. \end{cases}$
Варіант 8	$y = \begin{cases} 2x^2 - 6x + 4, & \text{якщо } x \geq -7; \\ x^2 + 9x - 6, & \text{якщо } x < -7. \end{cases}$
Варіант 9	$z = \begin{cases} 8x^3 + x - 1, & \text{якщо } x \geq 8; \\ x^2 + x - 1, & \text{якщо } x < 8. \end{cases}$
Варіант 10	$z = \begin{cases} 8x^3 + 3x + 1, & \text{якщо } x \geq -9; \\ 6x^2 + x - 3, & \text{якщо } x < -9. \end{cases}$
Варіант 11	$z = \begin{cases} 5y^2 - 4y + 9, & \text{якщо } y \geq -7; \\ y^2 + 5y - 1, & \text{якщо } y < -7. \end{cases}$
Варіант 12	$y = \begin{cases} 4z^2 - 7z + 1, & \text{якщо } z \geq 1; \\ z^2 + 3z - 3, & \text{якщо } z < 1. \end{cases}$
Варіант 13	$z = \begin{cases} 3x^3 - 4x + 6, & \text{якщо } x \geq -3; \\ x^2 + 7x - 2, & \text{якщо } x < -3. \end{cases}$
Варіант 14	$z = \begin{cases} 8x^3 + x - 4, & \text{якщо } x \geq -4; \\ 5x^2 + 6x - 9, & \text{якщо } x < -4. \end{cases}$
Варіант 15	$z = \begin{cases} 7y^2 - y + 9, & \text{якщо } y \geq -5; \\ 4y^2 + 3y - 2, & \text{якщо } y < -5. \end{cases}$
Варіант 16	$y = \begin{cases} 3x^2 - 4x + 5, & \text{якщо } x \geq 6; \\ 2x^2 + 3x - 6, & \text{якщо } x < 6. \end{cases}$
Варіант 17	$y = \begin{cases} 6x^3 - x + 2, & \text{якщо } x \geq 3; \\ 7x^2 + 9x - 1, & \text{якщо } x < 3. \end{cases}$
Варіант 18	$z = \begin{cases} 4x^3 + 2x + 2, & \text{якщо } x \geq 8; \\ 8x^2 + x - 1, & \text{якщо } x < 8. \end{cases}$

Варіант 19	$z = \begin{cases} 3y^2 - 4y + 9, & \text{якщо } y \geq -1; \\ 4y^2 + y - 1, & \text{якщо } y < -1. \end{cases}$
Варіант 20	$y = \begin{cases} 7x^3 + 2x - 6, & \text{якщо } x \geq -5; \\ 2x^2 + x - 4, & \text{якщо } x < -5. \end{cases}$

Завдання 3. Складіть алгоритм для обчислення:

Варіант 1	суми всіх натуральних чисел з [3; 178].
Варіант 2	добутку всіх натуральних чисел з [5; 34].
Варіант 3	суми всіх натуральних чисел з [7; 211].
Варіант 4	добутку всіх натуральних чисел з [25; 74].
Варіант 5	суми всіх натуральних чисел з [7; 233].
Варіант 6	добутку всіх натуральних чисел з [6; 27].
Варіант 7	суми всіх натуральних чисел з [19; 238].
Варіант 8	добутку всіх натуральних чисел з [28; 72].
Варіант 9	суми всіх натуральних чисел з [26; 115].
Варіант 10	суми всіх натуральних чисел з [7; 211].
Варіант 11	добутку всіх натуральних чисел з [7; 22].
Варіант 12	добутку всіх натуральних чисел з [7; 22].
Варіант 13	суми всіх натуральних чисел з [7; 233].
Варіант 14	добутку всіх натуральних чисел з [6; 27].
Варіант 15	суми всіх натуральних чисел з [19; 238].
Варіант 16	добутку всіх натуральних чисел з [28; 72].
Варіант 17	суми всіх натуральних чисел з [5; 109].
Варіант 18	добутку всіх натуральних чисел з [7; 33].
Варіант 19	добутку всіх натуральних чисел з [4; 13].
Варіант 20	суми всіх натуральних чисел з [2; 169].

2.2. Основні поняття програмування

Мова програмування – це алфавіт та правила складання програми. **Програма** – це алгоритм, перекладений мовою програмування. Для складання програми потрібно мати систему програмування, яка містить:

- редактор текстів програм;
- *транслятор* (програму, яка перекладає програми користувача у машинні коди);

- бібліотеки з корисними процедурами та функціями;
- засоби отримання підказок від комп'ютера.

Приклад систем програмування: Turbo Basic, Turbo Pascal, Delphi, Visual Basic і т.ін.

ЗАПИТАННЯ ДЛЯ САМОПЕРЕВІРКИ

1. Що таке мова програмування?
2. Що таке програма? Що таке транслятор?
3. Які складові має система програмування?
4. Наведіть приклад систем програмування.

2.3. Конструкції та оператори мови програмування

2.3.1. Поняття про візуальне програмування

Візуальне програмування використовує *графічні оболонки* для *об'єктно-орієнтованих* мов програмування.

Середовище візуального програмування **Delphi** – це графічна оболонка для мови програмування **Object Pascal**.

Для складання програми користувач отримує графічні (візуальні) засоби (*компоненти*), які мають деякі властивості (*атрибути*). Користувач розв'язує задачі за допомогою вибору компонентів, задання потрібних значень їх властивостям та складання підпрограм-процедур.

Форма – це компонент, який має властивості вікна *Windows* і призначений для розміщення інших компонентів.

Після запуску програми **Borland Delphi** на екрані отримуємо чотири вікна (рис. 2.10):

- 1) **вікно форми** (Form);
- 2) **вікно інспектора об'єктів** (Object Inspector) - призначене для задання початкових значень властивостям об'єктів та їх реакцію на стандартні події;
- 3) **вікно редагування коду програми** (Code Editor) ;
- 4) **головне вікно**, яке містить головне меню (Main Menu), панель інструментів та палітру компонентів.

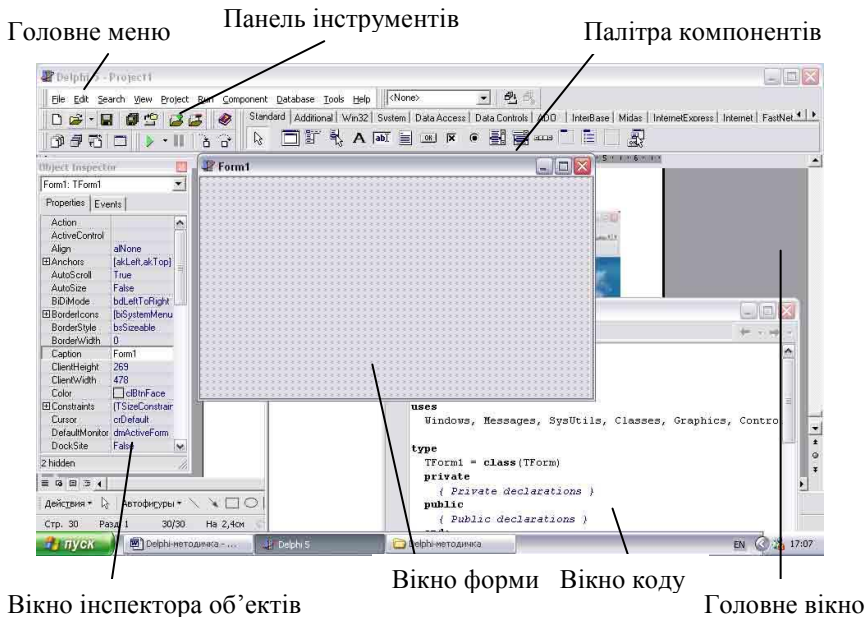


Рис. 2.10. Вікно середовища програмування Delphi

Головне меню

1. **File (файл)** – містить команди для роботи з файлами проекту:

New Application – створення нового проекту;

New Form – створення нової форми;

New Unit – створення нового модуля;

Open – відкрити файл;

Close – закрити файл;

Close All – закрити всі відкриті файли;

Save – зберегти;

Save as – зберегти як;

Save All – зберегти всі файли проекту.

2. **Edit (редагування)** – містить команди:

Align to Grid – вирівнювання компонентів відносно сітки;

Align – вирівнювання компонентів між собою;

Bring to Front – перемістити компонент на передній план;

Send to Back – перемістити компонент на задній план;

Size - зміна розміру вибраного компонента;

Scale - масштабування візуальних компонентів.

3. **Search (пошук)** – пошук та заміна фрагмента тексту.

4. **View** – візуалізація елементів середовища.

5. **Project** – компіляція та перевірка синтаксису програми .

6. **Run** – налаштування та запуск програми.

7. **Component** – створення нових компонентів.

8. **DataBase** – виклик інструментів бази даних.

9. **Tools** – команди для задання параметрів середовища.

10. **Help** – допомога.

Палітра компонентів має вигляд багатосторінкового блокнота. Вікно **інспектора об'єктів** містить список компонентів поточної форми, а також дві закладки: властивостей (*Properties*) та подій (*Events*). **Закладка властивостей** складається з двох стовпців: лівий містить назви властивостей компонентів, а правий – їх значення. **Закладка подій** також містить два стовпці. Лівий містить імена *стандартних подій*, а правий – імена *методів (процедур)*, які будуть реагувати на цю подію.

2.3.2. Поняття проекту в об'єктно-орієнтованому програмуванні

Проект – це сукупність файлів, з яких *Delphi* створює готову для виконання програму. Кожний проект обов'язково має файли:

- 1) *файл проекту* з розширенням *.dpr*. Це невеликий файл, який поєднує всі файли проекту;
- 2) *файли опису форм*: файл форми з розширенням *.dfm*, файл модуля, що їй відповідає, з розширенням *.pas*;
- 3) *файл ресурсів програми* з розширенням *.res*. Містить опис ресурсів, які не входять у форму, наприклад, піктограма програми;
- 4) *файл параметрів проекту* з розширенням *.dof*;
- 5) *файли параметрів середовища* з розширеннями *.drf*, *.dsk*, *.dsm*, *.ddp*, *.cfg*, *.dci*.

Для *збереження* Delphi-проекту треба задати імена модулів (автоматично пропонуються імена *Unit1.pas*, *Unit2.pas*, ...) та ім'я проекту (*Project1.dpr*). Для переміщення Delphi-проекту на інший комп'ютер необхідно мати тільки файли типів *.dpr*, *.dfm*, *.pas*, *.res*. Усі інші файли створюються автоматично.

Проект складається з *форм*. Кожній формі відповідає власний модуль. Файл проекту з розширенням *.dpr* їх об'єднує.

Під час створення нового проекту у модуль *Unit1.pas*, що відповідає формі *Form1*, автоматично вноситься **програмний код** опису цієї форми. Із додаванням нових компонентів до вікна форми у програму автоматично вноситься код опису параметрів цих компонентів (висота, ширина, розміщення, стиль і т.ін.).

Додавання об'єкта або застосування до нього методу призводить до появи **заготовки базового коду** відповідної процедури у вікні редактора. *Заготовка (шаблон)* складається із *заголовка процедури* та ключових слів **begin** та **end**. Заготовку заповнює користувач. Закінчується модуль власною командою **end**.

Модуль має такий загальний вигляд:

```

Unit Unit1; {ім'я модуля}
Interface {розділ оголошення процедур та функцій}
    Uses {список використаних модулів, наприклад}
    Windows, Messages, SysUtils, Classes, Graphics, Controls,
    Forms, Dialogs;
    Type {опис класів та типів об'єктів, наприклад}
        TForm = class(TForm)
            Button1: TButton;
            Procedure Button1.Click(Sender: TObject);
            ...
        private {приватні оголошення}
        public {загальнодоступні оголошення}
        end;
    Var {опис глобальних змінних}
        Form1: TForm1;
    {форвард-оголошення процедур та функцій користувача,
    наприклад}
    Procedure Information;
    Procedure SetPicture;
    ...
    function {ім'я функції};
    ...
implementation {розділ опису процедур і функцій}
    {$R *.DFM} {додається файл з описом форми}
    procedure TForm1.Button1Click(Sender: TObject);
        begin
            список операторів;
        end;
end. {кінець модуля}

```

ПРАКТИЧНА РОБОТА 1

Тема. Створення проекту «Анкета студента».

Мета. Отримання елементарних навичок роботи у середовищі Delphi.

Об'єкти: форма (*Form*), текстове поле (*Label*), зображення (*Image*), кнопка (*Button*).

Хід роботи

1. Завантажте програму Borland Delphi:

Start (Пуск) → Programs (Программы) → Borland Delphi x.0 → Delphi x.0, де *x* – версія програми (рис. 2.11).



Рис. 2.11. Запуск програми Delphi

2. Збережіть проект в окремій папці: *File → Save All («Зберегти все»)* (рис. 2.12).

З'явиться вікно «*Save Unit As*». У рядку «*Save in*» («Зберегти в») у випадковому списку виберіть ім'я робочого диска, створіть власну папку, задайте ім'я (рис. 2.13, 2.14).

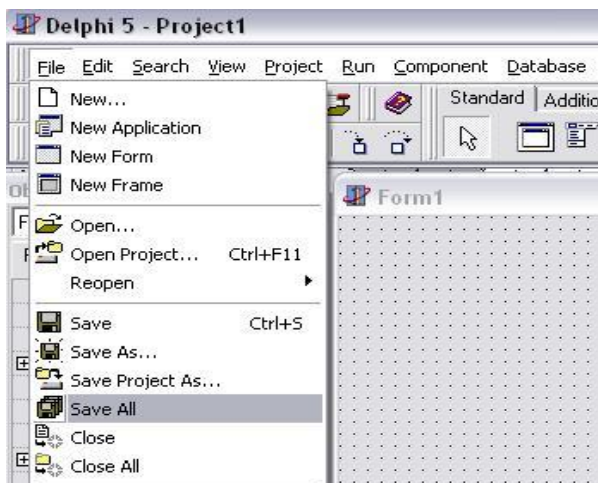


Рис. 2.12. Збереження проекту

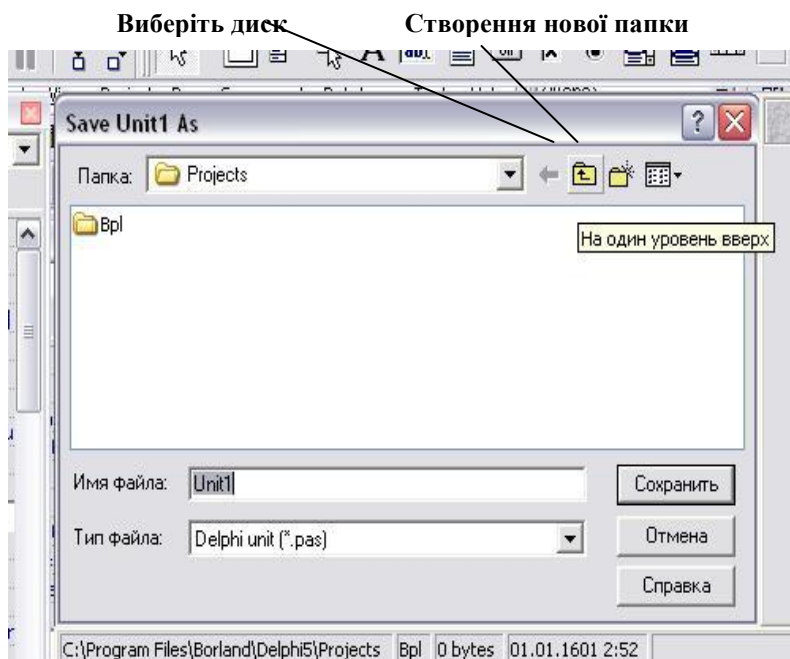


Рис. 2.13. Вибір диска і створення нової папки

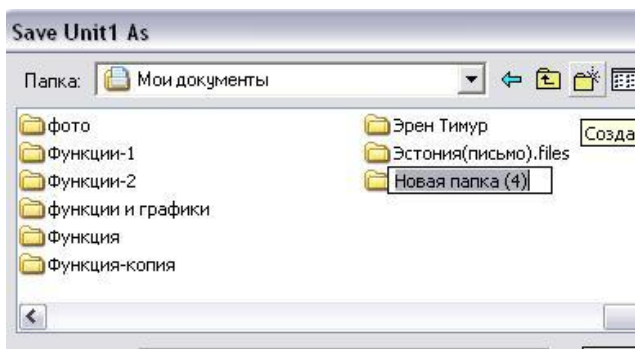


Рис. 2.14. Задання імені папки

Відкрийте папку. Збережіть: *Unit1.pas* → *Save* (рис. 2.15).

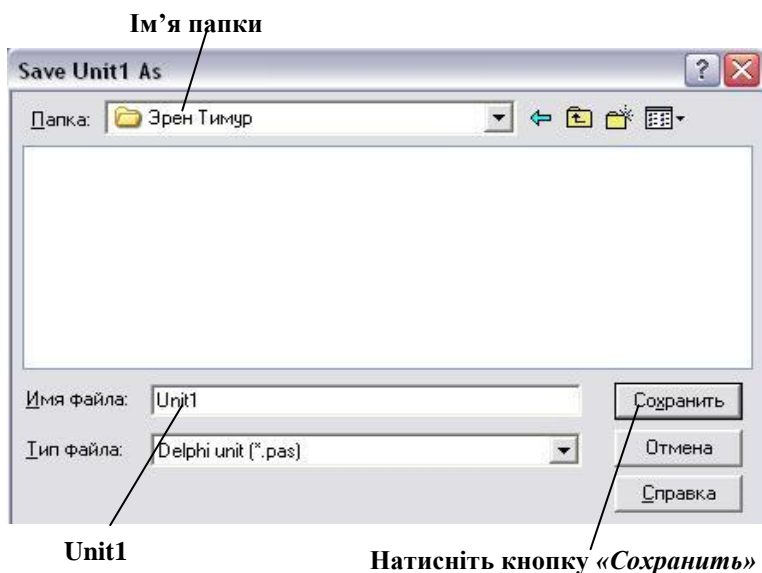
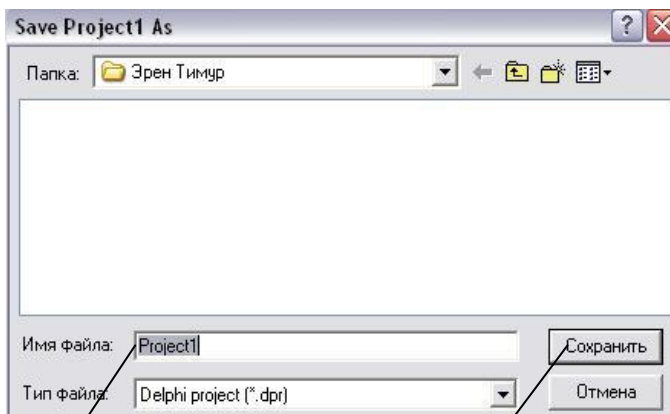


Рис. 2.15. Збереження модуля Unit1

У вікні “*Save Project As*” натисніть кнопку «*Сохранить*» («*Save*»). Ім'я файлу проекту *Project1.dpr* не змінюйте (рис. 2.16).



Project1

Натисніть кнопку «Сохранить»

Рис. 2.16. Збереження проекту Project1

3. Об'єкт **Form** використовується для створення програмою нового вікна. Задайте властивості форми:

Властивість	Опис властивості	Значення
<i>Caption</i>	Заголовок форми	Анкета
<i>Color</i>	Колір фону форми	Виберіть зі списку
<i>Icon</i>	Піктограма заголовка форми	Завантажте з будь-якого файла *.ico

Задання назви форми «Анкета» (рис. 2.17):

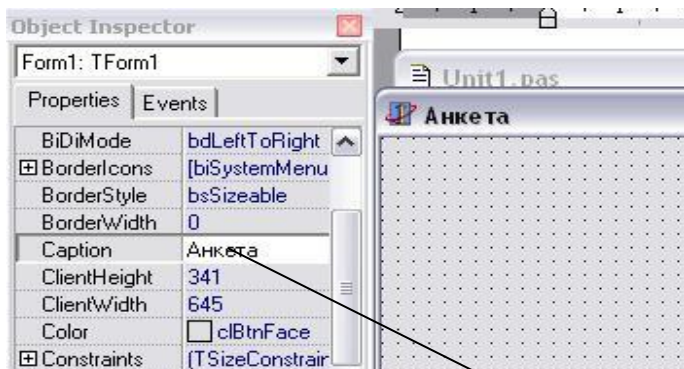


Рис. 2.17. Задання назви форми (**Caption**) «Анкета»

Вибір кольору фону форми (рис. 2.18):

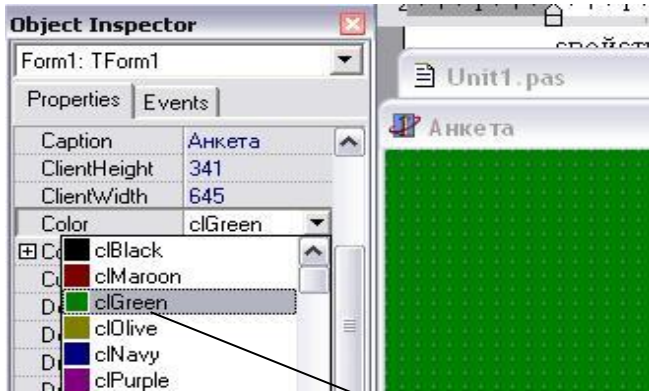
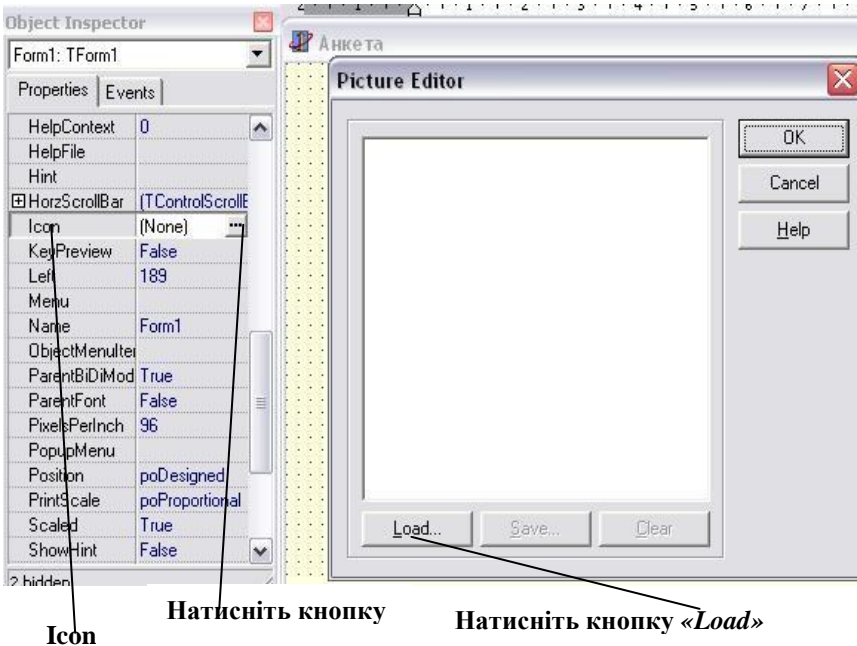


Рис. 2.18. Вибір кольору форми (*Color*)

Задання піктограми форми (рис. 2.19):



Натисніть кнопку

Натисніть кнопку «Load»

Icon

Рис. 2.19. Вибір піктограми заголовка форми

Натисніть кнопку «**Load**» (завантажити) і у вікні «*Load Picture*» виберіть будь-який файл зображення з розширенням *.ico. Натисніть кнопку «**Открыть**» («*Open*») (рис. 2.20). Підтвердіть свій вибір натисненням «**Ok**» (рис. 2.21).

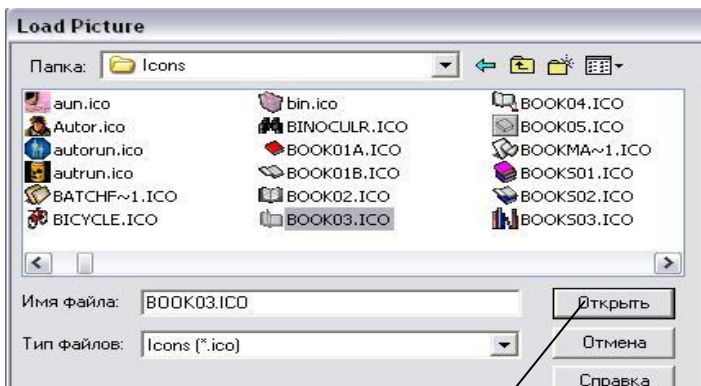


Рис. 2.20. Вибір файлу для піктограми

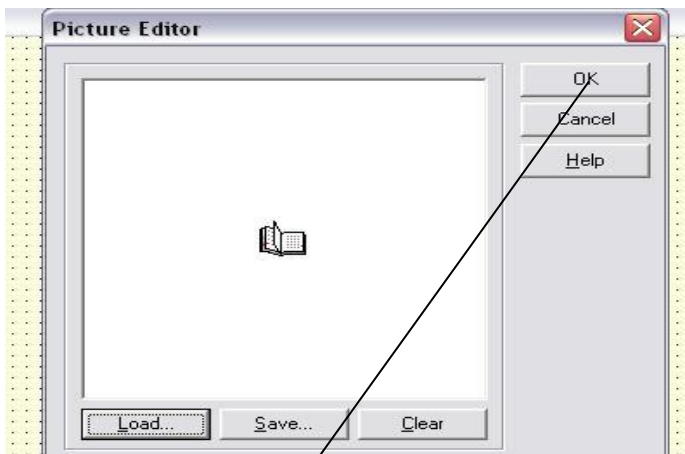



Рис. 2.21. Підтвердження вибору піктограми

Збережіть форму ще раз (*File* → *Save*).

4. Об'єкт **Label** використовують для створення текстових полів (підписів) у вікні програми. **Вставте у форму текстове поле (Label) з текстом «Анкета студента».**

Клацніть по піктограмі **Label**  у розділі «*Standard*» палітри компонентів (рис. 2.22).

Розділ «*Standard*»

Об'єкт *Label* (Підпис)

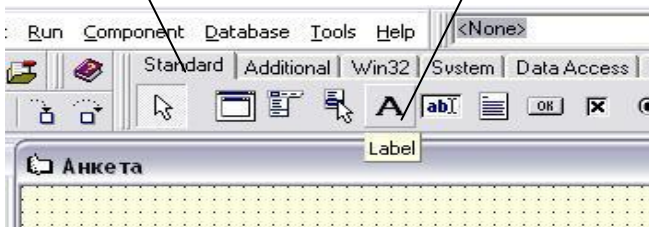


Рис. 2.22. Вибір об'єкта *Label*

Розмістіть вибраний об'єкт на формі (рис. 2.23).

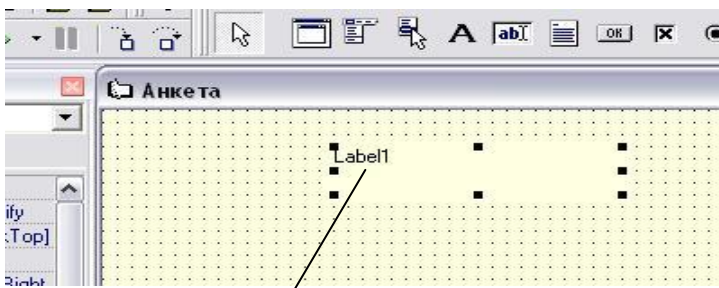


Рис. 2.23. Розміщення об'єкта *Label* на формі

Задайте властивість *Caption* об'єкта *Label* (рис. 2.24):

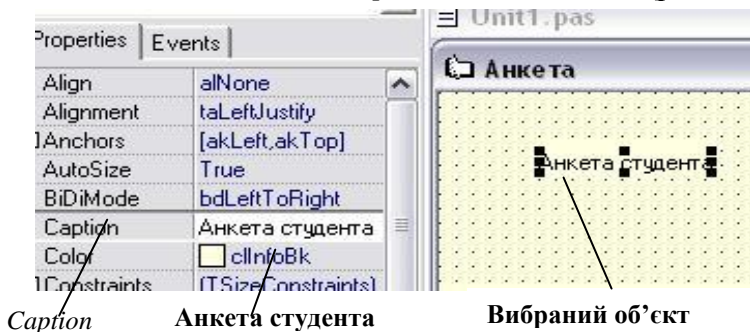



Рис. 2.24. Задання властивості *Caption* для об'єкта *Label*

У вікні «Object Inspector» задайте комплексну властивість *Font* для об'єкта *Label* так:

<i>Font</i> : Times New Roman Cyr;	<i>Size</i> : 16;
<i>Font style</i> : Bold;	<i>Color</i> : Green (зелений)

У вікні «Object Inspector» у рядку *Font* натисніть кнопку комплексної властивості  і задайте її значення (рис. 2.25).

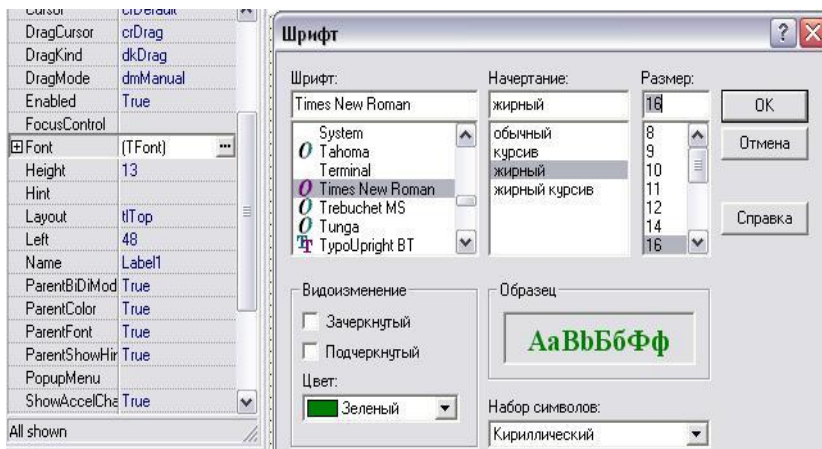


Рис. 2.25. Задання властивості *Font* для об'єкта *Label*

6. Аналогічно вставте у форму ще три об'єкти *Label*.

Label1: Ваше прізвище та ім'я.

Label2: Студент НАУ.

Label3: Номер групи. **Наприклад** (рис. 2.26):

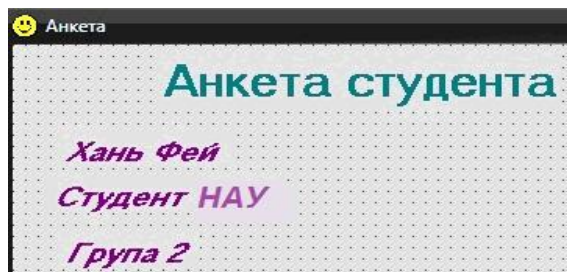
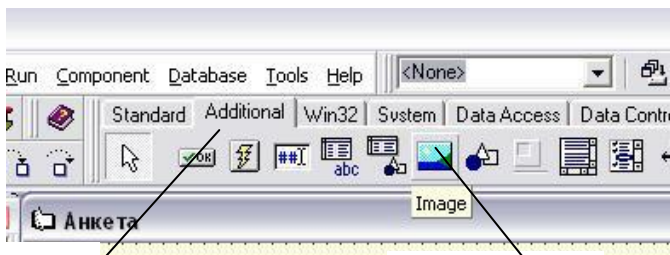


Рис. 2.26. Приклад анкетних даних

7. Об'єкт **Image** (зображення) використовують для додавання графічних об'єктів із файлів типу *.bmp, *.emf, *.ico, *.wmf. Вставте у форму об'єкт **Image** (рис. 2.27):



Закладка
«Additional»

Об'єкт **Image**

Рис. 2.27. Додавання об'єкта **Image**

Здайте властивості об'єкта **Image**:

Властивість	Опис властивості	Значення
<i>Picture</i>	Ім'я графічного файлу	Задасться в діалоговому вікні
<i>Stretch</i>	Зведення розміру зображення до заданих розмірів об'єкта	True
<i>Visible</i>	Видимість	False

8. Аналогічно додайте ще один об'єкт **Image2** поверх існуючого, вставте фото 2 за допомогою властивості **Picture**, задайте властивості **Stretch** (*True*) і **Visible** (*False*).

9. Об'єкт **Button** використовують для створення кнопок. Вставте у форму кнопки **Button1** і **Button2** (рис. 2.28):

Закладка «Standard»



Button (Кнопка)

Рис. 2.28. Об'єкт **Button** (Кнопка)

Здайте для них властивості *Caption* так (рис. 2.29):
Button1: Фото 1; **Button2**: Фото 2.



Рис. 2.29. Додавання кнопок

10. Запрограмуйте кнопку «*Фото 1*» так, щоб після її натиснення з'являлось перше фото. Для цього двічі клацніть по ній лівою кнопкою миші. З'явиться вікно тексту програми із заготовкою процедури *Button1Click*, котра обробляє подію клацання по кнопці «*Button1*»:

```
procedure Tform1.Button1Click(Sender: TObject);  
begin  
  
end;
```

Вставте текст програми:

```
procedure Tform1.Button1Click(Sender: TObject);  
begin  
Image1.Visible:= True;Image2.Visible:= False;  
end;
```

11. Аналогічно запрограмуйте кнопку «*Фото 2*»:

```
procedure Tform1.Button1Click(Sender: TObject);  
begin  
Image2.Visible:= True;Image1.Visible:= False;  
end;
```


12. Виконайте проект *Project1: Run* → *Run* (рис. 2.30):

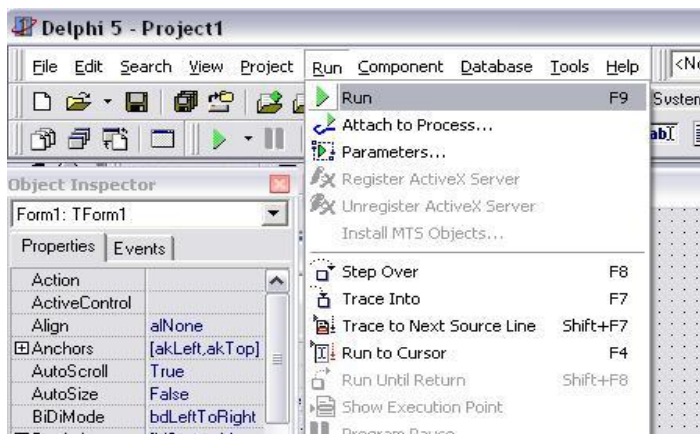


Рис. 2.30. Виконання проекту

13. Закрийте Delphi, відкрийте свою папку. Відкрийте



файл Project1.exe .

Додаткові завдання

1. Вставте у форму *третє* фото і відповідну кнопку «*Фото 3*».
2. Створіть проект «*Пори року*» (рис. 2.31):

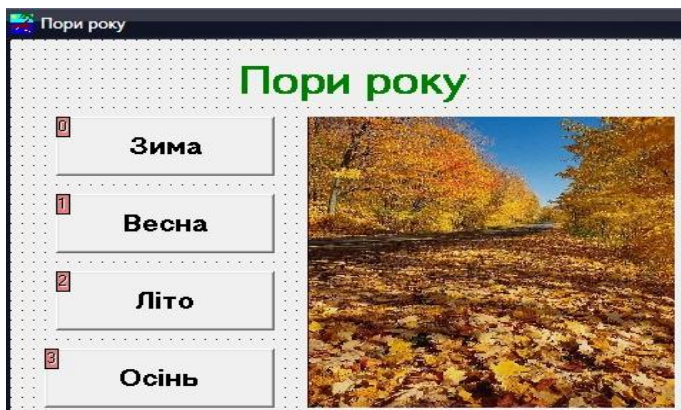


Рис. 2.31. Форма «Пори року»

2.3.3. Основні елементи мови програмування Object Pascal

Алфавіт мови *Object Pascal* містить літери, цифри, спеціальні символи, пропуски і зарезервовані слова. *Літери* – це літери латинського алфавіту від «a» до «z» і від «A» до «Z», а також знак підкреслення (_). *Цифри* – арабські цифри від 0 до 9.

Спеціальні символи Object Pascal:

+	– плюс	<	– менше
-	– мінус	>	– більше
*	– зірочка	[]	– квадратні дужки
/	– прямий слеш	{ }	– фігурні дужки
=	– дорівнює	()	– круглі дужки
.	– крапка	^	– тильда
'	– апостроф	\$	– долар
,	– кома	@	– комерційне а
:	– двокрапка	#	– номер
;	– крапка з комою		пропуск (не має позначення)

До спеціальних також належать такі пари символів:

<>	– не дорівнює	:=	– присвоїти
<=	– менше або дорівнює	//	– два прямих слеші
>=	– більше або дорівнює	..	– діапазон значень

(* *) - можна використовувати замість { }

(. .) - можна використовувати замість []

Ці пари символів не можна розділяти пропусками.

Ідентифікатори – це імена констант, змінних, міток, типів, об'єктів, класів, властивостей, процедур, функцій, модулів, програм і полів у записах. Ідентифікатор може складатись з літер і цифр, завжди починається з літери. Не може містити пропусків і спеціальних символів, має довільну довжину. В Object Pascal є *зарезервовані слова*, які (а також слова *at* і *on*) не можуть бути ідентифікаторами:

And	Exports	Library	Set
Array	File	Mod	Shl
As	Finalization	Nil	Shr
Asm	Finally	Not	String
Begin	For	Object	Then
Case	Function	Of	Threadvar
Class	Goto	Or	To
Const	If	Out	Try
Constructor	Implementation	Packed	Type
Destructor	In	Procedure	Unit
Dispinterface	Inherited	Program	Until
Div	Initialization	Property	Uses
Do	Inline	Raise	Var
Downto	Interface	Record	While
Else	Is	Repeat	With
End	Label	Resourcestring	xor
Except			

Константи – це величини, значення яких встановлені після службового слова **const** і не змінюються в процесі виконання програми. **Формат:**

const < ідентифікатор > = < значення константи >;

Наприклад: Const Max=1000; Min=1;

Змінні - це величини, значення яких змінюється в процесі виконання програми. Для опису змінних використовується службове слово **var**.

Формат: Var < список ідентифікаторів > : < тип >;

Цілі типи даних

Назва	Довжина, байт	Діапазон значень
<i>Byte</i>	1	0...255
<i>ShortInt</i>	1	-128...+127
<i>SmallInt</i>	2	-32768...+32767
<i>Word</i>	2	0...65535
<i>Integer</i>	4	-2 147 483 648...+2 147 483 647
<i>LongInt</i>	4	-2 147 483 648...+2 147 483 647
<i>LongWord</i>	4	0...4 294 967 295
<i>Int64</i>	8	-2 ⁶³ ...+2 ⁶³ -1
<i>Cardinal</i>	4	0...4 294 967 295

Наприклад: Var x, y:integer; z:word;

На множині **цілих** чисел визначені такі **операції**:

- + - додавання, * - множення,
- - віднімання, / - ділення,
- div** - ділення націло,
- mod** - остача від ділення націло.

Математичні функції:

Abs (x)	$- x $	Arctan (x)	$- arctg x$
Cos (x)	$- cos x$	Sin (x)	$- sin x$
Exp (x)	$- e^x$	Ln (x)	$- ln x$
Sqr (x)	$- x^2$	Sqrt (x)	$- \sqrt{x}$

Random (x) – випадкове число від 0 до x-1.

Odd (x) – надає значення *True*, якщо x – непарне число.

Результат виконання операцій +, -, *, *div*, *mod*, *abs(x)*, *sqr(x)* над цілими числами є цілим числом.

Результат виконання операції /, а також решти математичних функцій на множині цілих чисел є дійсним числом. Операції *div* і *mod* можна застосовувати тільки до цілих чисел. **Приклад:** $17 \text{ div } 3 = 5$; $17 \text{ mod } 3 = 2$.

Дійсні типи

Назва	Довжина, байт	Діапазон значень
<i>Real</i>	8	$5,0 \cdot 10^{-324} \dots 1,7 \cdot 10^{308}$
<i>Single</i>	4	$1,5 \cdot 10^{-45} \dots 3,4 \cdot 10^{38}$
<i>Double</i>	8	$5,0 \cdot 10^{-324} \dots 1,7 \cdot 10^{308}$
<i>Extended</i>	10	$3,4 \cdot 10^{-4951} \dots 1,1 \cdot 10^{4932}$
<i>Comp</i>	8	$-2^{63} \dots +2^{63-1}$
<i>Currency</i>	8	$\pm 922\ 337\ 203\ 685\ 477,5807$

Наприклад: Var a, b:real; c,d,f:single;

На множині **дійсних** чисел визначено такі **операції**:

- + - додавання; * - множення;
- - віднімання; / - ділення.

Математичні функції

Abs (x) – $ x $	Arctan (x) – $\arctg x$
Cos (x) – $\cos x$	Sin (x) – $\sin x$
Exp (x) – e^x	Ln (x) – $\ln x$
Sqr (x) – x^2	Sqrt (x) – \sqrt{x}

Int (x) – задає цілу частину числа x .

Trunc (x) – відкидає дробову частину числа x .

Frac (x) – задає дробову частину числа x .

Round (x) – округлення числа x .

Результатом виконання функцій *trunc (x)* і *round (x)* на множині дійсних чисел є ціле число.

Результатом виконання всіх інших операцій та функцій на множині дійсних чисел є дійсне число.

Приклад:

$$\text{Int}(3,6) = 3.0;$$

$$\text{Int}(-5,1) = -5.0;$$

$$\text{Int}(0,99) = 0.0;$$

$$\text{Trunc}(7,8) = 7;$$

$$\text{Frac}(7,8) = 0.8;$$

$$\text{Trunc}(-7,8) = -7;$$

$$\text{Frac}(-7,8) = -0.8;$$

$$\text{Round}(7,8) = 8;$$

$$\text{Round}(-7,8) = -8;$$

$$\text{Frac}(1) = 0.$$

Логічний (булевий) тип:

Назва	Діапазон
<i>boolean</i>	True, False

Наприклад: Var let:boolean;

Результат виконання операцій порівняння ($=$, $<$, $>$, \leq , \geq , $<>$) та операції *odd(x)* належить до логічного типу.

Символьний тип:

Назва	Діапазон
<i>char</i>	Таблиця кодів комп'ютера

Наприклад: Var m:char;

Функції для роботи із символьним типом

Ord (S) – визначає порядковий номер (код) символу S у кодовій таблиці. Результат є цілим числом.

Chr (I) – знаходить символ, порядковий номер (код) якого дорівнює I. Результат належить до символного типу.

Succ (S) – визначає символ, який міститься після символу S у кодовій таблиці.

Pred (S) – визначає символ, який зміститься перед символом S.

Upcase (s) – перетворює малі літери англійського алфавіту у великі.

Наприклад: Pred ('B') = "A"; Succ ('B') = "C"; Upcase ('n') = 'N'.

Математика	Object Pascal
$ax^2 + b$	$a*\text{sqr}(x)+b$
$\frac{a+b}{ c-d }$	$(a+b)/\text{abs}(c-d)$
$\sqrt{x^2 + 1}$	$\text{sqrt}(\text{sqr}(x)+1)$
$e^x \ln(x+2)$	$\text{exp}(x)*\ln(x+2)$
a^b	$\text{exp}(b*\ln(a))$

Пріоритет операцій

1. @, not.
2. *, /, div, mod, and, shl, shr.
3. +, -, or, xor.
4. =, <>, <, >, <=, >=, in.

Прості нестандартні типи даних

Перелічувальний тип задається переліком можливих значень. **Приклад:** type colors = (red, white, blue);
Var a,b:colors;

Тип-діапазон задається граничними значеннями всередині базового типу (цілого, логічного або перелічувального). **Приклад:** type digit='0' .. '9';
dig2=48 .. 57 ;
var a,b:digit;x,y:dig2;

Рядковий тип (string)

Рядок – це послідовність символів кодової таблиці. Рядок оточується з обох боків апострофами. Кількість символів у рядку може змінюватись від 0 до 255.

Формат:

Var < ідентифікатор > : string [максимальна довжина рядка];

Приклад: Var R1:string[10];R2:string[4];

Якщо довжину рядка не вказано, то вона автоматично набуває значення 255 байт.

Процедури і функції для роботи з рядками

1. **Процедура Str. Формат: Str** (< число >, < змінна >).

Переводить числові дані в дані типу рядок.

Наприклад: Str (2000, Kiev);

Результатом буде значення змінної Kiev = '2000'.

2. **Процедура Val. Формат: Val** (R1, S1, S2).

Присвоює числовій змінній S1 числовий образ рядка R1. Якщо це можливо, то змінна S2 отримує значення 0, інакше – числове значення номера першого недопустимого символу даного рядка.

Наприклад: Val ('1256', abc, code);

Результатом буде значення змінної abc=1256, змінна Code набуде значення 0.

3. **Функція StrToFloat. Формат: StrToFloat** (S1).

Перетворює символи рядка S1 у дійсне число.

Наприклад: S1:='123 '; N:=StrToFloat (S1);

Результатом буде значення змінної N = 123.

4. **Функція StrToInt. Формат: StrToFloat** (S1).

Перетворює символи рядка S1 у ціле число.

5. **Функція FloatToStr. Формат: FloatToStr**(S1).

Перетворює дійсне значення S1 у рядок символів.

6. **Функція IntToStr. Формат: IntToStr**(S1).

Перетворює ціле значення S1 у рядок символів.

2.3.4. Складання лінійних програм

ПРАКТИЧНА РОБОТА 2

Тема. Створення проекту «Периметр та площа прямокутника».

Мета. Отримання навичок складання лінійних програм.

Об'єкти: форма (*Form*), текстове поле (*Label*), кнопка (*Button*), поле редагування (*Edit*), кнопка із зображенням (*BitBtn*).

Хід роботи

1. Завантажте середовище програмування Delphi.
2. Об'єкт *Edit* використовують для введення рядка символів. Він міститься у розділі «Standard» (рис. 2.32):

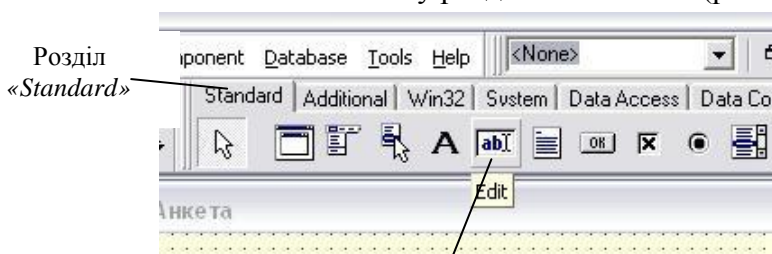


Рис. 2.32. Об'єкт *Edit* (поле редагування)

Об'єкт *BitBtn* – кнопка з певним типом дії. Міститься у розділі «Additional» палітри компонентів (рис. 2.33):

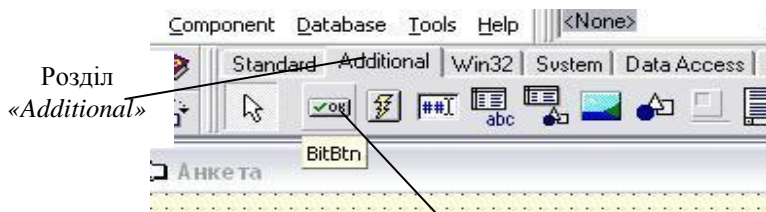


Рис. 2.33. Об'єкт *BitBtn* (кнопка з дією)

3. Розмістіть об'єкти на формі так, як на рис. 2.34:

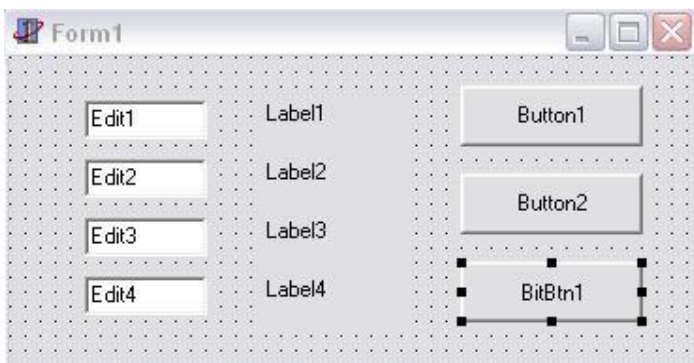


Рис. 2.34. Розміщення об'єктів на формі

4. Очистіть поля редагування *Edit1*, *Edit2*, *Edit3*, *Edit4* (у рядку властивості *Text* видаліть текст (рис. 2.35)):

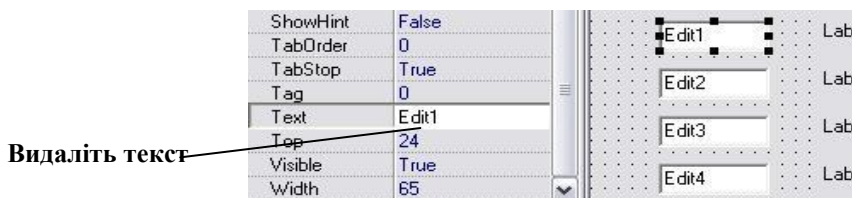


Рис. 2.35. Властивість «Text» об'єкта *Edit*

5. Задайте значення властивостям об'єктів:

Form

Caption: Периметр і площа прямокутника.

Color: будь-який колір.

Icon: будь-яке зображення із файлу з розширенням .ico.

Label1

<i>Caption</i> : a; <i>Font</i> : Times New Roman Cyr; <i>Font Style</i> : Bold;	<i>Size</i> :12; <i>Color</i> : Blue.
--	--

Label2

<i>Caption</i> : b; <i>Font</i> : Times New Roman Cyr; <i>Font Style</i> : Bold;	<i>Size</i> :12; <i>Color</i> : Blue.
--	--

Label3

<i>Caption: P;</i> <i>Font: Times New Roman Cyr;</i> <i>Font Style: Bold;</i>	<i>Size:12;</i> <i>Color: Red.</i>
---	---------------------------------------

Label4

<i>Caption: S;</i> <i>Font: Times New Roman Cyr;</i> <i>Font Style: Bold;</i>	<i>Size:12;</i> <i>Color: Red.</i>
---	---------------------------------------

Button1

Caption: Обчислити

Button2

Caption: Очистити

BitBtn

Kind: bkClose

Caption: Закрити

Після цього форма матиме вигляд, як на рис. 2.36:

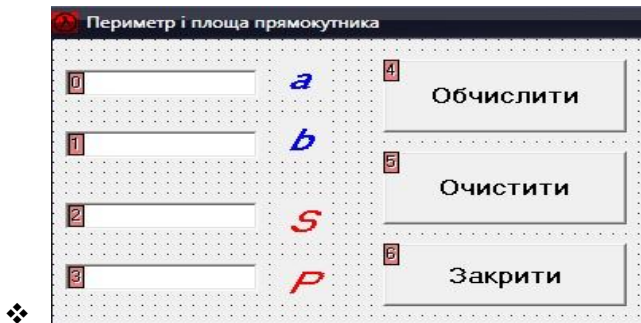


Рис. 2.36. Форма після задання властивостей об'єктів

6. Запрограмуйте кнопку «Очистити»:

```
procedure TForm1.Button2.Click(Sender: TObject);  
begin  
    edit1.clear;edit2.clear;  
    edit3.clear;edit4.clear;  
end;
```

7. Запрограмуйте кнопку «Обчислити»:

```

procedure TForm1.Button1.Click(Sender: TObject);
  var a, b, P, S: real; s1, s2: string; code: integer;
begin
  val(edit1.text, a, code); val(edit2.text, b, code);
  P := 2*(a + b); S := a*b; str(P:8:2, s1); str(S:8:2, s2);
  edit3.text := s1; edit4.text := s2;
end;

```

У запису **P:8:2** 8 – це максимальна кількість цифр перед комою у запису дійсного числа, 2 – максимальна кількість цифр після коми.

Наприклад: $\underbrace{\dots 656,53}_{\substack{8 \\ 2}}$

8. Збережіть проект у своїй папці (*File* → *Save All*).
9. Виконайте програму (*Run* → *Run*).

Додаткові завдання

1. Створіть проект для обчислення швидкості v , якщо відомо шлях s і час t (формула $v = \frac{s}{t}$).

2. Створіть проект для обчислення довжини кола l та площі круга S , якщо заданий його радіус R (рис. 2.37).

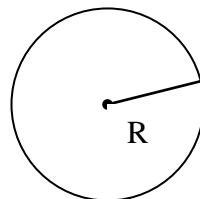


Рис. 2.37. Коло

Формули: $P = 2\pi R$,

$S = \pi R^2$, $\pi = 3,14$

3. Створіть проект для обчислення площі S трикутника, якщо задані координати вершин: (x_1, y_1) , (x_2, y_2) , (x_3, y_3) (рис. 2.38).

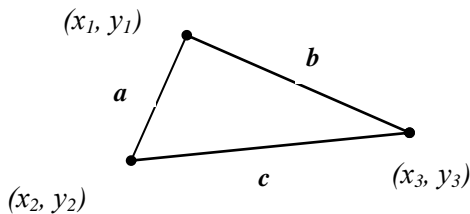


Рис. 2.38. Трикутник

Формули: $a = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$;
 $b = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2}$; $c = \sqrt{(x_3 - x_2)^2 + (y_3 - y_2)^2}$;
 $p = \frac{a+b+c}{2}$; $S = \sqrt{p(p-a)(p-b)(p-c)}$.

2.4. Програмування розгалужень

Команда розгалуження **if** має вигляд:

if < логічний вираз > **then** < команда 1 > **else** < команда 2 > ;

Приклад. Для функції $y = \begin{cases} \sin x, & \text{якщо } x < -1, \\ \cos x, & \text{якщо } -1 \leq x < 1, \\ \ln x, & \text{якщо } x \geq 1 \end{cases}$

команда розгалуження запишеться так:

if $x < -1$ **then** $y := \sin(x)$ **else if** $(x \geq -1)$ **and** $(x < 1)$ **then** $y := \cos(x)$ **else** $y := \ln(x)$;

Команда вибору (**case**) має вигляд:

case < вираз > **of**
 < список значень 1 > : < команда 1 > ;
 ...
 < список значень n > : < команда n > ;
else < команда n+1 >
end;

Приклад. Для функції $y = \begin{cases} \sin x, & \text{якщо } x = -1, \\ \cos x, & \text{якщо } x = 1, \\ \ln x, & \text{якщо } x < 1 \text{ або } x > -1 \text{ або } -1 < x < 1 \end{cases}$

команда вибору (**case**) запишеться так:

```
case x of
  -1:  y := sin(x);
   1:  y := cos(x);
  else y := ln(x)
end;
```

ПРАКТИЧНА РОБОТА 5

Тема. Створення проекту «Телефонний довідник».

Мета. Отримання навичок програмування розгалужень (зокрема команди вибору **case**).

Об'єкти: форма, текстове поле, поле редагування, головне меню (*MainMenu*), комбінований список (*Combobox*).

Хід роботи

1. Завантажите середовище програмування Delphi.
2. Об'єкти *MainMenu* та *Combobox* містяться у розділі *Standard* палітри компонентів (рис. 2.39):

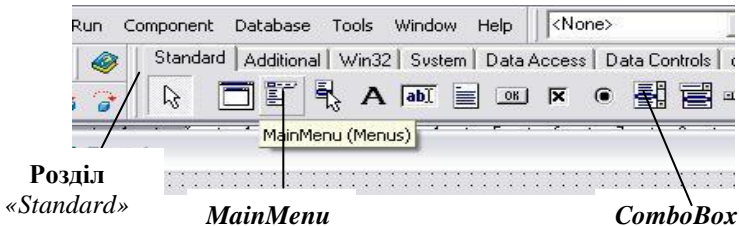


Рис. 2.39. Об'єкти *MainMenu* та *Combobox*

Об'єкт *Combobox* використовують для створення випадного списку. У програмному коді використовується його властивість *ItemIndex*. Значення 0 цієї властивості відповідає першому елементу списку, 1 – другому, 2 – третьому і т.д. Значення -1 відповідає порожньому рядку.

3. Розмістіть об'єкти на формі (рис. 2.40):

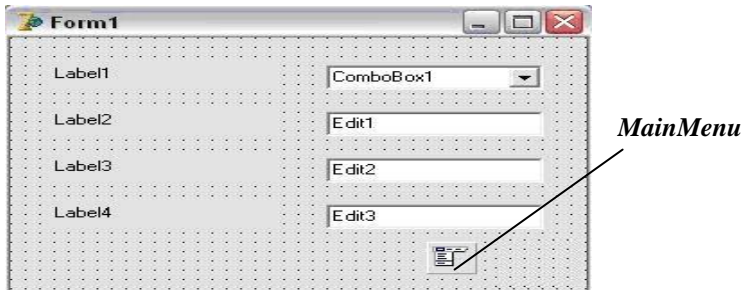
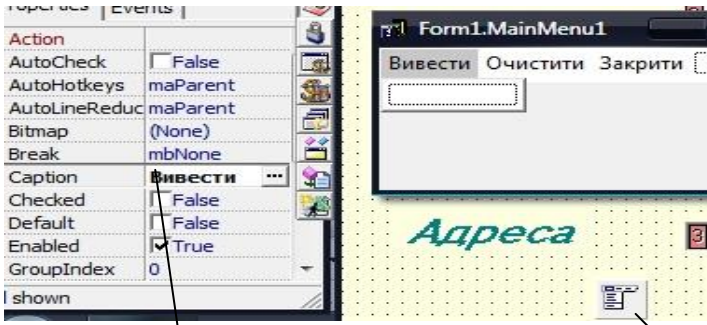


Рис. 2.40. Розміщення об'єктів на формі

4. Очистіть поля *Edit1, Edit2, Edit3, Combobox1*.
5. Задайте значення властивостей об'єктів:

Form	<i>Caption:</i> Телефонний довідник
Label1	<i>Caption:</i> Телефон
Label2	<i>Caption:</i> Прізвище
Label3	<i>Caption:</i> Ім'я
Label4	<i>Caption:</i> Адреса
Combobox1	<i>Items:</i> 2453565 5645637 2763567

6. Уведіть назви команд головного меню (рис. 2.41):



Пишіть: «Вивести»

Двічі клацніть по *MainMenu*

Рис. 2.41. Введення команд головного меню

Форма матиме вигляд (рис. 2.42):

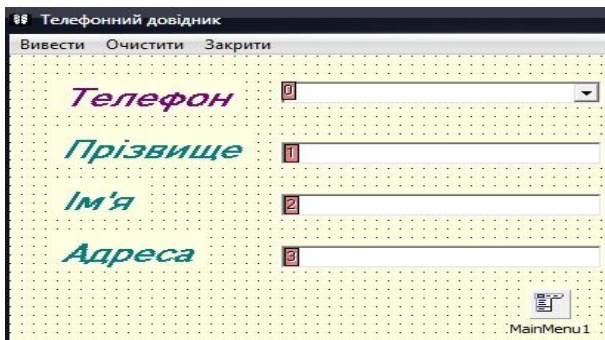


Рис. 2.42. Форма після задання властивостей об'єктів

7. Виконайте програму (*Run* → *Run*).
8. Збережіть проект у своїй папці (*File* → *Save All*).
9. Запрограмуйте команду «**Вивести**»:

```

procedure TForm1. N1Click(Sender: TObject);
begin
    case combobox1.itemindex of
        0: begin
            edit1.text:='Петров'; edit2.text:='Андрій';
            edit3.text:='вул. Жилианська, буд.34, кв.56';
        end;
        1: begin
            edit1.text:='Волков'; edit2.text:='Дмитро';
            edit3.text:='вул. Лугова, буд.125, кв. 238';
        end;
        2: begin
            edit1.text:='Ларіна'; edit2.text:='Ірина';
            edit3.text:='вул. Космонавтів, буд.7, кв.19';
        end;
    end;
end;

```

10. Запрограмуйте команду «**Очистити**»:

```

procedure TForm1. N3Click(Sender: TObject);
begin
    combobox1.itemindex:= -1;
    edit1.clear; edit2.clear; edit3.clear;
end;

```

11. Запрограмуйте команду «**Закрити**»:

```

procedure TForm1. N5Click(Sender: TObject);
begin
    close
end;

```

12. Збережіть проект (*File* → *Save*). Виконайте (*Run* → *Run*).

Додаткові завдання

1. Створіть проект для обчислення значень функції

$$y = \begin{cases} 2x + 1, & \text{якщо } x > 4, \\ 2x - 1, & \text{якщо } x \leq 4. \end{cases}$$

Вказівка. Розмістіть об'єкти на формі (рис. 2.43):

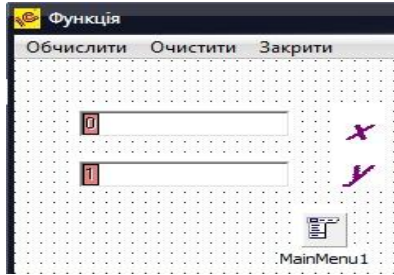


Рис. 2.43. Розміщення об'єктів на формі

Програмний код для команди «Обчислити»:

```
procedure TForm1. N1Click(Sender: TObject);
  var x,y:real;code:integer;s1:string;
begin
  val(edit1.text, x, code);
  if x>4 then y:=2*x+1 else y:=2*x-1;
  str(y:9:2,s1); edit2.text:=s1;
end;
```

2. Створіть проект для обчислення значень функції $y = \frac{x+7}{x-2} + \frac{5}{x}$. *Вказівка.* Програмний код для команди «Обчислити»:

```
procedure TForm1. N1Click(Sender: TObject);
  var x,y:real;code:integer;s1:string;
begin
  val(edit1.text,x,code);
  if (x=2) or (x=0) then edit2.text:= 'не існує'
  else begin
    y:= (x + 7)/(x - 2)+5/x;
    str(y:9:2,s1); edit2.text:= s1; end; end;
```

3. Створіть проект для розв'язання квадратного рівняння $ax^2 + bx + c = 0$, $a \neq 0$ за формулами: $D = b^2 - 4ac$;

$$x_1 = \frac{-b - \sqrt{D}}{2a}, \quad x_2 = \frac{-b + \sqrt{D}}{2a}.$$

4. Створіть проект «Пори року» (рис. 2.44):

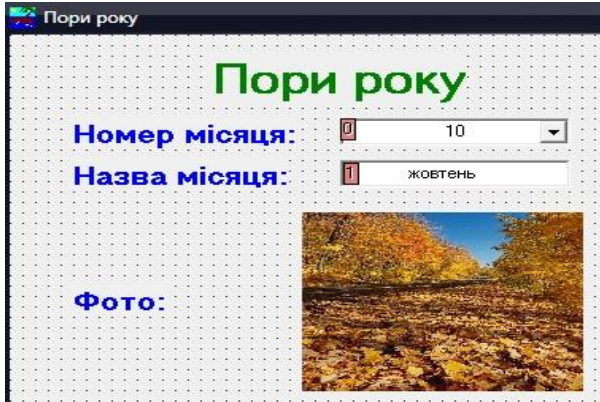


Рис. 2.44. Розміщення об'єктів на формі

ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 4

Створіть проекти для виконання завдань 1 і 2 контрольної роботи 3.

2.5. Програмування циклів

Цикл – це процес виконання певних команд деяку кількість разів. Є три види команд циклів.

1. Команда циклу з параметром (**for**). Є два види команди **for**:

а) **for** <параметр>: = <вираз 1> **to** <вираз 2> **do** <команда 1>;

Параметр - це змінна цілого, символьного, логічного або перелічуваного типу, а вирази 1 і 2 – початкове та кінцеве значення параметра.

Приклад. Нехай $s = 0$, $d = 1$. Після виконання команди циклу
for $i := 4$ to 6 do begin $s := s + i$; $d := d * i$ end;
змінна s матиме значення $0 + 4 + 5 + 6 = 15$, а змінна
 $d = 1 \cdot 4 \cdot 5 \cdot 6 = 120$.

б) **for**<параметр>:=<вираз 1>**downto**<вираз 2>**do** <команда 1>;

У цій команді крок зміни параметра дорівнює -1.

Приклад. Нехай $s = 0$, $d = 1$. Після виконання команди циклу
for $i:=7$ downto 5 do begin $s:=s+i$; $d:=d*i$ end;
змінна s матиме значення $0 + 7 + 6 + 5 = 18$, а змінна
 $d = 1 \cdot 7 \cdot 6 \cdot 5 = 210$.

2. Команда циклу з передумовою (**while**) має вигляд:

while < логічний вираз > **do** < команда 1 >;

Приклад. Нехай $s = 0$, $i = 1$, $d = 1$. Після виконання команди
while $i \leq 4$ do begin $s:=s+i$; $d:=d*i$ end;
змінна s матиме значення $0 + 1 + 2 + 3 + 4 = 10$, а змінна
 $d = 1 \cdot 2 \cdot 3 \cdot 4 = 24$.

3. Команда циклу з післяумовою (**repeat**) має вигляд:

repeat < команди > **until** < логічний вираз >;

Приклад. Нехай $s = 0$, $i = 1$, $d = 1$. Після виконання команди
repeat $s:=s+i$; $d:=d*i$ until $x > 5$;
змінна s матиме значення $0 + 1 + 2 + 3 + 4 + 5 = 15$, а змінна
 $d = 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 = 120$.

ПРАКТИЧНА РОБОТА 6

Тема. Створення проекту «Обчислення суми та добутку цілих чисел із заданого відрізка $[a;b]$ ».

Мета. Отримання навичок програмування циклів.

Об'єкти: форма, текстове поле, поле редагування, головне меню, перемикач (**RadioButton**).

Хід роботи

1. Завантажте програму Delphi.

2. Об'єкт **RadioButton** використовують для створення засобу вибору одного варіанту з кількох можливих. Він міститься у розділі «Standard» (рис. 2.45):

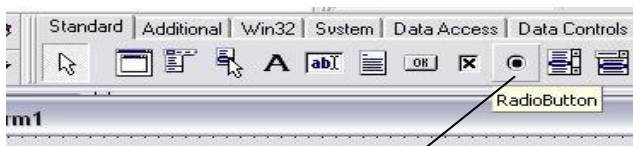


Рис. 2.45. Об'єкт **RadioButton**

3. Розмістіть об'єкти на формі (рис. 2.46):

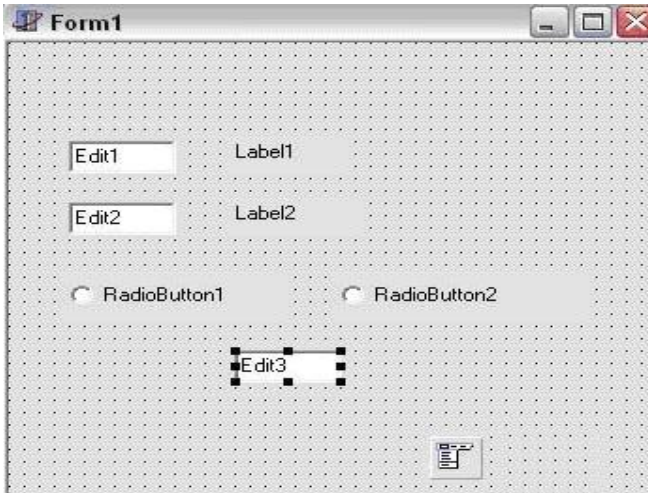


Рис. 2.46. Розміщення об'єктів на формі

4. Збережіть проект у своїй папці (*File* → *Save All*).
5. Задайте значення властивостей об'єктів:

Form	<i>Caption:</i> Сума та добуток чисел
Label1	<i>Caption:</i> a
Label2	<i>Caption:</i> b
Edit1	Очистіть поле <i>Text</i> <i>Hint:</i> Введіть лівий кінець відрізка <i>ShowHint:</i> True
Edit2	Очистіть поле <i>Text</i> <i>Hint:</i> Введіть правий кінець відрізка <i>ShowHint:</i> True
Edit3	Очистіть поле <i>Text</i> <i>ReadOnly:</i> True
Radiobutton1	<i>Checked:</i> True <i>Caption:</i> Сума
Radiobutton2	<i>Checked:</i> False <i>Caption:</i> Добуток
MainMenu1	<i>Captions:</i> Обчислити, Очистити, Закрити

Форма матиме вигляд (рис. 2.47):

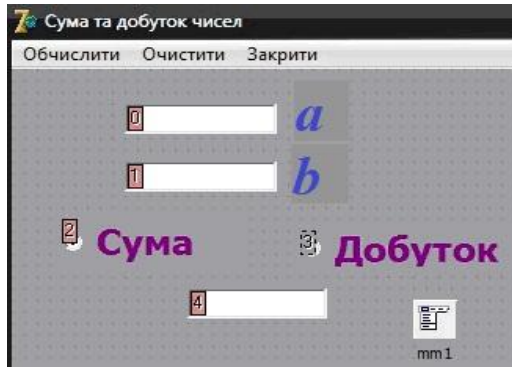


Рис. 2.47. Форма після задання властивостей об'єктів

6. Запрограмуйте команду «Закрити»:

```
procedure TForm1. N5Click(Sender: TObject);  
begin  
    close  
end;
```

7. Запрограмуйте команду «Очистити»:

```
procedure TForm1. N3Click(Sender: TObject);  
begin  
    edit1.clear; edit2.clear; edit3.clear;  
end;
```

8. Запрограмуйте команду «Обчислити»:

```
procedure TForm1. N1Click(Sender: TObject);  
    var a,b,code,s,d,i:integer;s1,s2:string;  
begin  
    val(edit1.text,a,code); val(edit2.text,b,code);  
    s:=0; d:=1;  
    for i:=a to b do  
        begin s:=s+i;d:=d*i; end;  
    str(s,s1); str(d,s2);  
    if radiobutton1.checked then edit3.text:=s1  
    else edit3.text:=s2;  
end;
```

9. Ще раз збережіть проект (*File* → *Save*).
10. Виконайте проект (*Run* → *Run*).

Додаткові завдання

1. Створіть проект для обчислення кількості цілих чисел з відрізка [11; 5678], які діляться на 3.

Вказівка. Властивості «Text» об'єкта **Edit1** задайте значення 11, **Edit2** - значення 5678. Тут треба використовувати команду **while**, оскільки значення проміжної змінної *i* повинне змінюватись не на 1, а на 3. Початкове значення *i* беремо за 12, оскільки **12 – це перше число з відрізка [11; 5678], яке ділиться на 3**. Програмний код для команди «Обчислити» буде таким:

```

procedure TForm1. N1Click(Sender: TObject);
    var a,b,code,n,i:integer;s1:string;
begin
    val(edit1.text,a,code); val(edit2.text,b,code);
    n:=0;i:=12;
    while i<=b do
        begin
            n:=n+1;i:= i+3;
        end;
    str (n,s1); edit3.text:= s1;
end;

```

2. Створіть проект для обчислення суми та добутку цілих чисел з відрізка [6; 34], які діляться на 5.

2.6. Робота з масивами

Масив (array) – це набір елементів одного типу, які зберігаються в послідовних комірках оперативної пам'яті і мають спільну назву. Приклад **одномірного масиву** наведено на рис. 2.48. Якщо ***a*[0..10]** – ім'я та розмірність масиву, тоді ***a*[0] = 45, *a*[1] = 30, *a*[4] = 32** і т.д.

0	1	2	3	4	5	6	7	8	9	10
45	30	21	65	32	754	8	3	67	98	5

Рис. 2.48. Одновимірний масив

Двовимірний масив даних – це таблиця, яка має декілька рядків (рис. 2.49):

	0	1	2	3	4	5	Рядок
0	23	876	9	67	9	65	Стовпець
1	6	5	7	34	567	32	
2	76	0	2	76	34	8	
3	45	97	3	6	4	90	

Рис. 2.49. Таблиця (двовимірний масив)

Якщо $b[0..5, 0..3]$ – ім'я та розмірність масиву, тоді у нашому прикладі $b[0,0]=23$, $b[1,0]=876$, $b[1,3]=97$, $b[2,1]=7$, $b[3,1]=34$ і т.д.

Загальний вигляд конструкції опису масиву:

```
array [<розмірність>] of <назва базового типу>;
```

Описати масив можна в розділах **type**, **const** або **var**.

Наприклад:

- 1) Type abcd = array [0..10] of real;
- 2) Const massiv1:abcd=(5, 3, 4, 8, 90.4, 23, 1,0. 6,2);
- 3) Var a1,a2:integer;d:real;

ПРАКТИЧНА РОБОТА 7

Тема. Створення проекту «Обчислення кількості елементів масиву $b[0..3,0..2]$, більших від заданого числа a » (рис. 2.50).

	0	1	2	3
0				
1				
2				

Рис. 2.50. Масив $a[0..3,0..2]$

Мета. Отримання навичок складання програм з використанням масивів.

Об'єкти: форма, текстове поле, поле редагування, головне меню, *таблиця текстових рядків (StringGrid)*.

Хід роботи

1. Завантажите середовище програмування Delphi.
2. Об'єкт *StringGrid* використовують для створення двовимірної таблиці текстових рядків. Він міститься у розділі «Additional» (рис. 2.51):

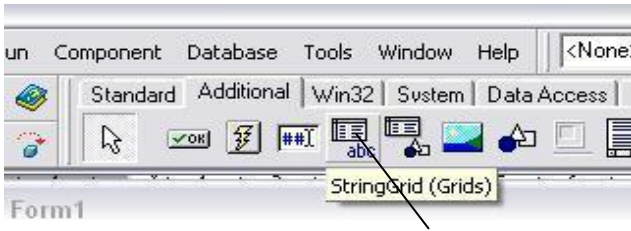


Рис. 2.51. Об'єкт *StringGrid*

3. Розмістіть об'єкти на формі (рис. 2.52):

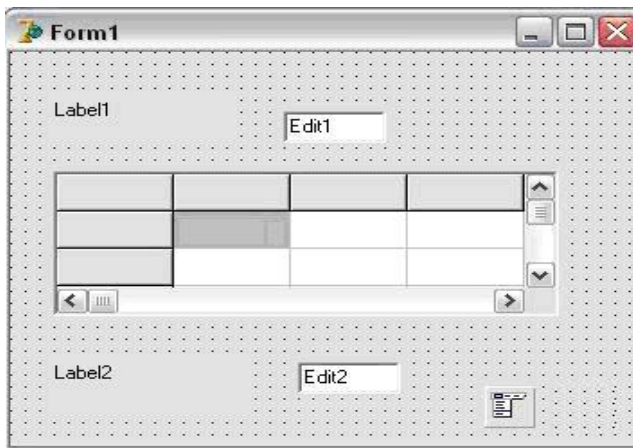


Рис. 2.52. Розміщення об'єктів на формі

4. Задайте значення властивостей об'єктів (рис. 2.53):

Об'єкт <i>Form</i>	<i>Caption</i> : Таблиця
Об'єкт <i>Label1</i>	<i>Caption</i> : Число <i>a</i>
Об'єкт <i>Label2</i>	<i>Caption</i> : Кількість чисел <i>n</i>
Об'єкти <i>Edit1, Edit2</i>	Очистіть поле <i>Text</i>
Об'єкт <i>MainMenu1</i>	Обчислити, Очистити, Закрити

Об'єкт *StringGrid1*

<i>Властивість</i>	<i>Опис властивості</i>	<i>Значення</i>
<i>ColCount</i>	Кількість стовпців таблиці	4
<i>RowCount</i>	Кількість рядків таблиці	3
<i>FixedCols</i>	Кількість стовпців таблиці, які не прокручуються зліва	0
<i>FixedRows</i>	Кількість рядків, які не прокручуються вгору	0
<i>Options (goEditing)</i>	Можливість редагування	True

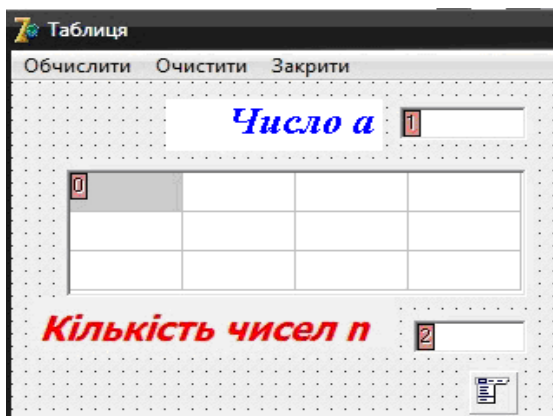


Рис. 2.53. Форма після задання властивостей об'єктів

3. Запрограмуйте команду «Закрити»:

```
procedure TForm1. N5Click(Sender: TObject);
begin close end;
```

4. Запрограмуйте команду «Очистити»:

```
procedure TForm1. N3Click(Sender: TObject);
    var i,j:integer;
begin
    edit1.clear; edit2.clear;
    for i:=0 to 3 do
        for j:=0 to 2 do
            stringgrid1.cells[i,j]:=' ';
end;
```


5. Запрограмуйте команду «Обчислити»:

```
procedure TForm1. N1Click(Sender: TObject);  
  var b:array[0..3,0..2] of integer;  
      a,code,n,i,j:integer;s1:string;  
begin  
  val(edit1.text,a,code);n:=0;  
  for i:=0 to 3 do  
    for j:=0 to 2 do  
      begin  
        val(stringgrid1.cells[i,j],b[i,j],code);  
        if b[i, j]>a then n:=n+1;  
      end;  
    str (n,s1); edit2.text:=s1;  
end;
```

6. Збережіть проект у своїй папці (*File* → *Save All*).

7. Виконайте проект (*Run* → *Run*).

Додаткові завдання

1. Створіть проект для обчислення добутку елементів масиву $k[0..2,0..1]$ цілих чисел, рівних заданому числу m .

Вказівка. Програмний код для команди «Обчислити»:

```
procedure TForm1. N1Click(Sender: TObject);  
  var k: array[0..2,0..1] of integer;  
      m,code,d,i,j:integer;s1:string;  
begin  
  val(edit1.text,m,code);d:= 1;  
  for i:=0 to 2 do  
    for j:=0 to 1 do  
      begin  
        val(stringgrid1.cells[i, j],k[i,j],code);  
        if k[i,j]=m then d:=d*k[i,j];  
      end; str(d,s1); edit2.text:= s1;  
end;
```

2. Створіть проект для обчислення суми елементів масиву $d[1..4, 1..2]$ дійсних чисел, менших від заданого числа t .

3. Створіть проект для обчислення суми, добутку та кількості елементів масиву $g[1..5, 1..3]$ цілих чисел, більших від заданого числа f .

4. Створіть проект для обчислення суми елементів головної діагоналі масиву $h[1..5, 1..5]$ цілих чисел (*елементи головної діагоналі* – це елементи, у яких номер стовпця дорівнює номеру рядка).
5. Створіть проект для обчислення кількості елементів головної діагоналі масиву $h[1..4, 1..4]$, рівних числу 0.
6. Створіть проект для обчислення добутку елементів головної діагоналі масиву $h[1..3, 1..3]$, більших від числа 7.
7. Створіть проект для обчислення суми елементів головної діагоналі масиву $g[1..3, 1..4]$, менших від числа 5.

2.7. Процедури і функції

Підпрограми призначені для оброблення окремих частин складного завдання. Є два види підпрограм – *процедури* і *функції*. *Стандартні підпрограми* містяться у модулях *System*, *Crt*, *Dos*, *Graph*. *Підпрограми користувача* створюються і описуються в основній програмі у розділах **procedure** або **function**.

1. Процедури (procedure). Загальний опис процедури:

```
procedure < назва > (< список формальних параметрів >);
    < розділи описів та оголошень процедури >;
begin
    < розділ команд процедури >
end;
```

У списку *формальних параметрів* перераховують змінні та їх тип. Розрізняють *параметри-аргументи* (*параметри-значення*) – вхідні дані для процедури, і *параметри-результати* (*параметри-змінні*), через які можна повертати результати роботи процедури в основну програму. Перед списками параметрів-результатів пишуть слово **var**.

Приклад. Розглянемо процедуру **tangens**:

```
procedure tangens (x:real; var y:real);
    begin y = (sin(x))/(cos(x)); end;
```

Тут x - це *формальний параметр-аргумент*, y - *формальний параметр-результат*.

Процедуру викликають з основної програми або іншої підпрограми за допомогою команди виклику:

```
<назва процедури>(<список фактичних параметрів>);
```

Фактичними параметрами-аргументами можуть бути константи, змінні, вирази, а параметрами-результатами – тільки змінні. Типи даних не вказують, але між фактичними та формальними параметрами має бути відповідність щодо кількості і типів.

Приклад 1. Програма для обчислення значень функції $a = tg(b) + ctg(b) + tg^2(b)$ з використанням процедури **tangens**:

```
unit Unit1;  
...  
var a,b:real;code:integer;s1:string;  
procedure tangens (x:real; var y:real);  
    begin y := (sin(x))/(cos(x)); end;  
procedure TForm1.Button1.Click(Sender: TObject);  
begin  
    val(edit1.text,b,code); tangens(b,n);  
    a := n + 1/n + n2;  
    str(a,s1); edit2.text:=s1;  
end;  
end.
```

Змінні, описані у розділі описів основної програми, називаються *глобальними*. Вони діють у всіх підпрограмах. Змінні, описані у розділі описів певної процедури, називаються *локальними*. Вони діють тільки у межах цієї процедури.

У цій програмі глобальними є змінні a , b і $s1$, оскільки вони описані у розділі *var* головної програми. Локальними є змінні x та y , бо вони описані лише в підпрограмі.

2. **Функції (function)** можуть повертати у місце виклику тільки один результат простого стандартного типу.

Загальний опис функції:

```
function <назва> (<список формальних параметрів>) : <тип функції>;  
    < розділи описів та оголошень функції>;  
begin  
    < розділ команд функції, де має бути така команда: назва := вираз >  
end;
```

Приклад. Розглянемо функцію **tangens**:

```
function tangens (x: real):real;  
    begin  
        tangens := (sin(x))/(cos(x));  
    end;
```

Виклик функції здійснюється так:

```
< назва > (< список фактичних параметрів >
```

Приклад 2. Програма для обчислення значень функції $a = tg(b) + ctg(b) + tg^2(b)$ з використанням функції **tangens**:

```
unit Unit1;  
...  
var a,b:real;code:integer;s1:string;  
function tangens (x:real):real;  
    begin  
        tangens := (sin(x))/(cos(x));  
    end;  
procedure TForm1.Button1.Click(Sender: TObject);  
begin  
    val(edit1.text,b,code);  
    a := tangens(b)+1/tangens(b)+sqr(tangens(b));  
    str(a,s1); edit2.text:=s1;  
end;  
end.
```

ЗАВДАННЯ

1. Створіть проект для обчислення значень функції $a = tg(b) + ctg(b) + tg^2(b)$ з використанням процедури обчислення тангенса.

2. Створіть проект для обчислення значень функції $m = ctg(n) + tg(n) + ctg^2(n)$ з використанням функції для обчислення котангенса.

3. Створіть проект для обчислення площі S трикутника, якщо задані координати його вершин (x_1, y_1) , (x_2, y_2) , (x_3, y_3) , використовуючи процедуру або функцію для обчислення сторони трикутника.

2.8. Робота з рядками

1. **Функція Concat. Формат:** **Concat** (R1, R2, R3).

Об'єднує рядки R1, R2, R3 в один рядок.

Наприклад: R1 := 'Мова'; R2 := 'програмування';

R3 := 'Pascal'; R := **Concat** (R1, R2, R3);

Результат: R = 'Мова програмування Pascal'.

2. **Функція Length. Формат:** **Length** (R).

Видає довжину рядка R, враховуючи пропуски.

Наприклад: R = 'Turbo Pascal'; N := **Length** (R);

Результат: N = 12.

3. **Функція Copy. Формат:** **Copy** (R, Poz, N).

Копіює фрагмент довжиною N рядка R з позиції Poz.

Наприклад: R := 'Turbo Pascal'; Poz := 7; N := 6;

Word := **Copy** (R, Poz, N);

Результат: Word = 'Pascal'.

4. **Функція Pos. Формат:** **Pos** (Word, R).

Знаходить номер позиції P, з якої починається перше входження слова Word у рядок R.

Наприклад: R := 'Севастополь'; Word := 'сто';

P := **Pos** (Word, R);

Результат: P = 5.

5. Процедура Insert. Формат: Insert(Word, R, Poz).

Вставляє слово Word у рядок R, починаючи з позиції Poz.

Наприклад: Poz := 20; R := 'Мова програмування Pascal'; Word := 'Turbo'; **Insert** (Word, R, Poz);

Результат: R = 'Мова програмування Turbo Pascal'.

6. Процедура Delete. Формат: Delete (R, Poz, N).

Видаляє слово, котре починається з позиції Poz і має задану довжину N у рядку R.

Наприклад: Poz := 1;

R := 'Мова програмування Turbo Pascal';

N := 19; **Delete** (R, Poz, N);

Результат: R = 'Turbo Pascal'.

Завдання. Створіть проект, котрий об'єднує два рядки в один (прізвище, ім'я) і знаходить довжину утвореного рядка.

Вказівка. Розмістити об'єкти на формі можна, як на рис 2.54:

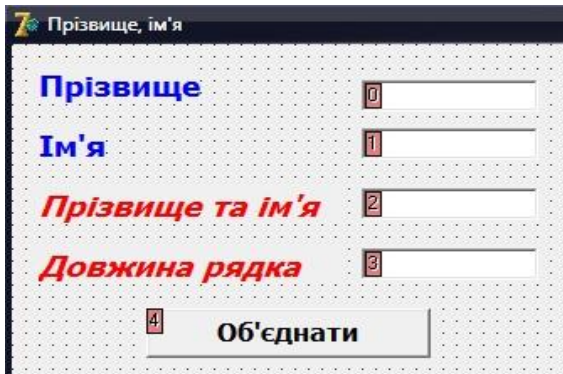


Рис. 2.54. Розміщення об'єктів на формі

Програмний код для кнопки «Об'єднати»:

```
procedure TForm1.Button1Click(Sender: TObject);  
  var d:integer;s:string;  
begin  
  edit3.text:=concat(edit1.text,' ',edit2.text);  
  d:=length(edit3.text);  
  str(d,s);  edit4.text:=s; end;
```

Приклад виконання програми (рис. 2.55):

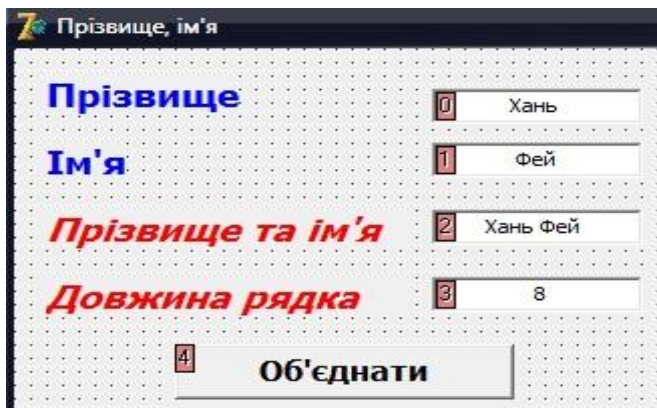


Рис. 2.55. Приклад виконання програми

ЗАВДАННЯ

1. Створіть проект для підрахунку кількості пропусків у тексті.
2. Створіть проект для заміни всіх символів 'а' на символи 'в'.
3. Створіть проект для видалення всіх символів 'с' у тексті.
4. Створіть проект для заміни у тексті одного слова іншим словом такої самої довжини.

2.9. Засоби роботи з файлами та графікою

2.9.1. Робота з файлами

Файл – це сукупність даних, розміщених на зовнішньому носії. Дані у файлі називаються *елементами*. Кількість даних, на відміну від масиву, під час описування файла не вказують. Елемент файла не має індексу (номера). Тип елемента може бути будь-яким, крім типу файл.

Файловий тип даних описують у розділі опису типів так:

type < ім'я типу > = **file of** < базовий тип >;

або у розділі оголошення змінних:

var < список змінних > : **file of** < базовий тип >;

Приклад: type myfile:file of integer;
var file1:myfile;
file2, file3:file of string;

Дії з файлами

Для того, щоб знайти потрібний елемент файла, необхідно послідовно переглянути всі попередні. Це називається *послідовним доступом до файла*.

Для оброблення файла необхідно відкрити файл, виконати необхідні дії, закрити файл.

Для визначення кінця файла існує логічна функція

eof (<ім'я файла>);

Значенням цієї функції буде **true**, якщо досягнутий кінець файла. Для роботи з файлами існують команди:

assignfile(<ім'я файла>, <зовнішнє ім'я>) – встановлює зв'язок між іменем файла та файлом на зовнішньому носії;
reset(<ім'я файла>) – відкриває файл для зчитування даних;
read(<ім'я файла>, <ім'я змінної>) – зчитує дані із файла в оперативну пам'ять;
rewrite(<ім'я файла>) – відкриває файл для запису даних;
write(<ім'я файла>, <ім'я змінної>) – записує дані у файл;
closefile(<ім'я файла>) – закриває файл.

<ім'я файла> - це ім'я *файлової змінної*. задане у розділі оголошення змінних;

<зовнішнє ім'я> - це ім'я файла даних на зовнішньому носії, взяте в лапки, наприклад, *'d:\folder1\file1.txt'*. Якщо файл міститься в тій же папці, що й проект, то можна вказати тільки ім'я файла.

Крім файлів *послідовного доступу* можна створювати і обробляти файли *прямого доступу*. Для цього перед використанням команд *read* або *write* треба забезпечити

доступ до k -го елемента (нумерація з нуля) файла за допомогою команди `seek(<ім'я файла>, k);`

Текстові файли

Типізовані файли, описані вище, кодуються комп'ютером. Їх не можна редагувати або переглядати з допомогою текстового редактора. Тому, крім типізованих, використовують *текстові файли*.

Елементами текстових файлів є рядки (послідовності символів: літер, цифр, знаків та пропусків). Такий файл можна створити і редагувати за допомогою текстового редактора. Розділювачем між елементами файла є пропуск. Введення кожного рядка закінчується натисненням клавіші введення. Для перевірки наявності символів у рядку використовують функцію `eoln(<ім'я файла>);`, яка набуває значення *true*, якщо знайдено кінець файла.

Текстові файли описують так:

```
var <список імен змінних>: textfile;
```

Дані з рядка текстового файла можна зчитати за допомогою команд:

```
read(<ім'я файла>, <список параметрів>);  
readln(<ім'я файла>, <список параметрів>);
```

Під час виконання команди `readln` зайві дані у рядку ігноруються і наступна команда `read` або `readln` буде зчитувати дані з наступного рядка.

Рядок текстового файла можна створити з допомогою звичайного текстового редактора або за допомогою команд:

```
write(<ім'я файла>, <список виразів>);  
writeln(<ім'я файла>, <список виразів>);
```

У текстовий файл можна додавати (дописувати) дані за допомогою процедури `append(<ім'я файла>);`.

Завдання. У завданні про об'єднання рядків та знаходження довжини рядка додайте кнопку «*Записати у файл*».

Вказівки для виконання

1. Розмістити об'єкти на формі можна так (рис 2.56):

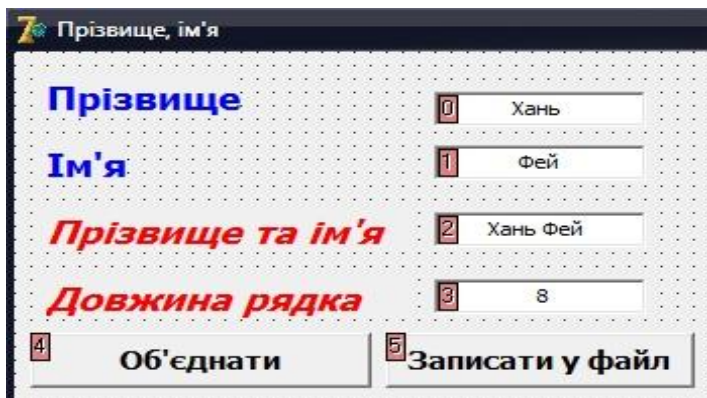


Рис. 2.56. Розміщення об'єктів на формі

2. Програмний код для кнопки «*Записати у файл*»:

```
procedure TForm1.Button2Click(Sender: TObject);  
var f:textfile;s:string;d:integer;  
begin  
  assignfile(f, 'file1.txt');  
  edit3.text:=concat(edit1.text,' ',edit2.text);  
  d:=length(edit3.text); str(d,s); edit4.text:=s;  
  rewrite(f); writeln(f, edit3.text,' ', d);  
  closefile(f);  
end;
```

3. Перевірте наявність запису у файлі *file1.txt* (рис. 2.57):

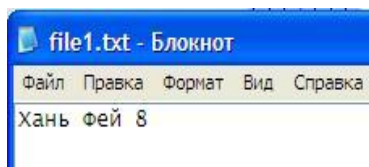


Рис. 2.57. Текстовий файл

2.9.2. Поняття про графіку у візуальному програмуванні

Вікно з канвою для побудови зображень надає компонент *TPaintBox*, який міститься у розділі «System» (рис. 2.58):



Рис. 2.58. Розділ «System»

Канву містить властивість *Canvas*, а побудову зображень виконує обробник події *OnPaint*.

Графічні інструменти

Ім'я	Опис
<i>Pen</i>	Перо для зображення простих ліній. Застосовується для функції <i>LineTo</i> або для зображення рамки для фігури
<i>Brush</i>	Пензель для заповнення області певним кольором. Застосовується у функціях <i>Rectangle</i> , <i>FillRect</i> , <i>FloodFill</i> .
<i>Font</i>	Задання шрифту, яким буде зображено текст
<i>Region</i>	Задає регіон (замкнений простір): круг, квадрат, довільна фігура, отвори у фігурах

Основні функції

Ім'я	Опис	Приклад
<i>MoveTo</i>	Переміщує точку початку побудови лінії у точку $(x; y)$.	<code>Canvas.MoveTo(50, 100)</code>
<i>LineTo</i>	Будує лінію з поточної позиції до точки $(x; y)$.	<code>Canvas.LineTo(50, 100)</code>
<i>Ellipse</i>	Будує еліпс, вписаний у невидимий квадрат з координатами верхнього лівого і правого нижнього кутів	<code>Canvas.Ellipse(0, 0, 50, 50);</code>
<i>FillRect</i>	Заповнює прямокутник кольором поточного пензля (<i>brush</i>).	<code>Canvas.FillRect(Bounds(0,0,100,100));</code>

<i>FloodFill</i>	Заповнює дану область кольором поточного пензля.	Canvas.FloodFill (10, 10, clBlack, fsBorder);
<i>Rectangle</i>	Будує прямокутник (або квадрат).	Canvas.Rectangle (Bounds(20, 20, 50, 50));
<i>RoundRect</i>	Будує прямокутник із заокругленими кутами.	Canvas.RoundRect(20, 20, 50, 50, 3, 3);
<i>TextOut</i>	Зображення рядка, починаючи з (x, y).	Canvas.TextOut(10, 10, 'Some text');

Координата (0,0) - це верхній лівий кут екрана. Перемістити точку початку побудови лінії можна за допомогою властивості *PenPos* у канвасі. **Наприклад**, "*Canvas.PenPos.x:=20*", "*Canvas.PenPos.y:=50*", або "*Canvas.PenPos:= Point(20,50)*". **Основні властивості пера** – це **ширина і колір**, наприклад, "*Canvas.Pen.Width:=4*", "*Canvas.Pen.Color:= clLime*".

Наприклад, програма для створення вікна на рис. 2.59:

```

procedure TForm1.PaintBox1Click(Sender: TObject);
  var x,y:integer;
begin
  with paintbox1.canvas do
  begin
    brush.color:=clred;
    ellipse(0,0,width,height);
    font.Name:='arial';
    font.size:=height div 5;
    font.Style:=[fsbold, fsitalic];
    font.Color:=clwhite;
    x:=(width-textwidth('Delphi')) div 2;
    y:=(height-textheight('D')) div 2;
    textout(x,y,'Delphi');
  end;
end;

```



Рис. 2.59. Приклад використання компонента TPaintBox



Рис. 2.60. Зображення для завдання

Завдання. Створіть зображення, показане на рис.2.60.

ЗАВДАННЯ ДЛЯ КОНТРОЛЬНОЇ РОБОТИ 5

Завдання 1. Створіть проект для завдання 3 контрольної роботи 3.

Завдання 2. Створіть проект для обчислення:

Варіант 1	суми елементів масиву $a[1..2, 1..3]$ цілих чисел, більших від заданого числа s .
Варіант 2	добутку елементів масиву $b[1..2, 1..4]$ цілих чисел, менших за задане число z .
Варіант 3	кількості елементів масиву $c[1..2, 1..5]$ цілих чисел, що дорівнюють заданому числу a .
Варіант 4	добутку елементів масиву $d[1..3, 1..2]$ цілих чисел, менших за задане число n .
Варіант 5	кількості елементів масиву $n[1..3, 1..3]$ цілих чисел, менших за задане число t .
Варіант 6	суми елементів масиву $m[1..3, 1..4]$ цілих чисел, що дорівнюють заданому числу h .
Варіант 7	кількості елементів масиву $p[1..3, 1..5]$ цілих чисел, більших від заданого числа f .
Варіант 8	добутку елементів масиву $k[1..4, 1..2]$ цілих чисел, менших за задане число d .
Варіант 9	кількості елементів масиву $h[1..4, 1..3]$ цілих чисел, що дорівнюють заданому числу u .
Варіант 10	добутку елементів масиву $t[1..4, 1..5]$ цілих чисел, менших за задане число w .

Варіант 11	кількості елементів масиву $w[1..5, 1..2]$ цілих чисел, менших за задане число r .
Варіант 12	суми елементів масиву $x[1..5, 1..3]$ цілих чисел, менших за задане число u .
Варіант 13	суми елементів масиву $y[1..5, 1..4]$ цілих чисел, більших від заданого числа p .
Варіант 14	добутку елементів масиву $z[1..5, 1..5]$ цілих чисел, менших за задане число a .
Варіант 15	кількості елементів масиву $g[1..2, 1..3]$ цілих чисел, що дорівнюють заданому числу d .
Варіант 16	добутку елементів масиву $s[1..2, 1..2]$ цілих чисел, менших за задане число g .
Варіант 17	кількості елементів масиву $q[1..4, 1..4]$ цілих чисел, що дорівнюють заданому числу h .
Варіант 18	суми елементів масиву $u[1..3, 1..6]$ цілих чисел, менших за задане число k .
Варіант 19	суми елементів масиву $v[1..6, 1..2]$ цілих чисел, більших від заданого числа k .
Варіант 20	добутку елементів масиву $f[1..2, 1..6]$ цілих чисел, менших за задане число n .

Завдання 3. У завданні 1 додайте кнопку для запису результатів виконання програми у текстовий файл.

Завдання 4. За допомогою текстового редактора «Блокнот» створіть текстовий файл, який містить елементи масиву із завдання 2. У завданні 2 додайте кнопку для зчитування елементів масиву зі створеного текстового файла.

ПІСЛЯМОВА

Навчальний матеріал, викладений у посібнику, є базою як для вивчення основ інформатики та обчислювальної техніки, а також суміжних та спеціальних дисциплін у вищій школі, так і для практичного використання засобів комп'ютерної техніки і сучасних інформаційних технологій у науково-пізнавальній діяльності студентів.

Викладачі фізики і хімії, математики і біології, мови та літератури, географії, історії та інших загальноосвітніх і спеціальних дисциплін дедалі ширше використовують комп'ютер у повсякденній роботі зі студентами. Студенти використовують комп'ютер як допоміжний засіб під час підготовки домашніх завдань, рефератів та інших робіт, а також самостійного вивчення навчального матеріалу.

Таким чином, застосування інформаційних технологій у системі освіти дозволяє підвищити якість навчального процесу у вищому навчальному закладі. Застосування комп'ютерів у процесі вивчення майже всіх дисциплін поступово переходить зі статусу експерименту у режим постійного функціонування як невід'ємний елемент освітньої системи.

СПИСОК ЛІТЕРАТУРИ

1. *Навчальні програми (довузівська підготовка іноземних громадян).* – К.: Політехніка, 2005. – 48-59 с.
2. *Бедренко В.И.* Основы информатики и вычислительной техники: Учебное пособие для иностранных студентов. – К.: НАУ, 2009. – 324 с.
3. *Информатика: Базовый курс / С.В.Симонович и др.*- СПб.: Питер, 2001.- 640 с.
4. *Симонович С.В., Евсеев Г.А., Алексеев А.Г.* Общая информатика. – М.: АСТ-ПРЕСС Книга, 2004. – 592 с.
5. *Фабричев В.А., Труш О.І., Чижевський Й.Ф.* Основы информатики: Навч. посібник. – К.: Книжкове видавництво НАУ, 2006. – 352 с.
6. *Фаронов В.* Delphi 6. Учебный курс. – С.-П: Питер, 2002.- 507 с.
7. *Глинський Я.М., Анохін В.Є., Ряжська В.А.* Паскаль. Turbo Pascal і Delphi.- Львів: Деол, 2002. – 142 с.
8. *Следзінський І.Ф., Василенко Я.П.* Основы информатики. Посібник для студентів. – Тернопіль: Навчальна книга – Богдан, 2003. – 160 с.
9. *Єфименко В.В., Онищенко С.М.* Опрацювання табличних даних засобами Microsoft Excel. Лабораторний практикум: Навчальний посібник. – К.: Логос, 2005. - 167 с.
10. *Кравченко С.М., Єфименко В.В., Онищенко С.М.* Операційна система Microsoft Windows. Лабораторний практикум: Навчальний посібник. – К.: Логос, 2005. - 76 с.
11. *Зарецька І.Т., Колодяжний Б.Г., Гурджій А.М., Соколов О.Ю.* Информатика 10-11. – К.: Навчальна книга, 2002. – 495 с.
12. *Глинський Я.М.* Практикум з інформатики. – Львів: Деол, 2002. – 224 с.
13. *Глинський Я.М.* Информатика: В 2-х ч. – Львів: Деол, 2002. – 255 с.
14. *Симонович С., Евсеев Г.* Практическая информатика. – М.: АСТ-ПРЕСС, 2001. – 480 с.

Навчальне видання

КОРЧУК Олена Юріївна,
КОСЯК Валентина Іванівна

ОСНОВИ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Навчальний посібник

Редактор Р.М. Шульженко
Технічний редактор А.І. Лавринович
Коректор О.О. Крусь
Комп'ютерна верстка Н.С. Ахроменко

Підп. до друку 26.03.2018. Формат 60×84/16. Папір офс.
Офс. друк. Ум. друк. арк. 9,30. Обл.-вид. арк. 10,0.
Тираж 100 пр. Замовлення № 37-1.

Видавець і виготівник
Національний авіаційний університет
03680. Київ – 58, проспект Космонавта Комарова, 1.

Свідоцтво про внесення до Державного реєстру ДК №977 від 05.07.2002