

Міністерство освіти і науки, молоді та спорту України  
Чернігівський промислово-економічний коледж  
Київського національного університету технологій та дизайну

ЗАТВЕРДЖУЮ

Заступник директора з НР

\_\_\_\_\_ С.В.Бондаренко

\_\_\_\_\_ 20\_\_ р.

**Методичні вказівки щодо організації  
самостійної роботи студентів  
з дисципліни «Комп'ютерна техніка і програмування»  
спеціальності 5.05070104 «Монтаж і експлуатація  
електроустаткування підприємств і цивільних споруд»**

Уклав

Малиновська Н.Б.

Розглянуто на засіданні  
циклової комісії  
обліково-аналітичних та економічних дисциплін  
Протокол №\_\_ від \_\_\_\_\_ 20\_\_ року

Голова циклової комісії

І.В.Ренська

# Самостійна робота № 1

**Тема:** Апаратні засоби персональних комп'ютерів

**Мета:** розглянути основні групи клавіш на клавіатурі, формувати навички роботи з клавіатурою.

**Питання, що виносяться на самостійне вивчення:**

## 1.2 Клавіатура ПК

**Практичне завдання:**

ЗАВДАННЯ: набрати з клавіатури потрібні літери, слова, символи, цифри.

- 1 Завантажте програму „Блокнот”.
- 2 Користуючись алфавітними клавішами, розташованими в центральній частині клавіатури, наберіть латинській алфавіт: abcdefghijklmnopqrstuvwxyz
- 3 Перейдіть до режиму вводу великих букв. Повторіть набір літер латинського алфавіту: ABCDEFGHIJKLMNOPQRSTUVWXYZ
- 4 Введіть з клавіатури слідуєчі слова, використовуючи для набору одиночної великої літери клавішу : Shift Enter Init Esc
- 5 Вилучте набрані символи, користуючись клавішею Del або Backspace
- 6 Наберіть цифри, користуючись клавішами верхнього рядка основного меню:  
1 234567890
- 7 Наберіть символи , \. / [ ] ; ' = -
- 8 Користуючись клавішею , наберіть символи: | ? : " { } ~ ! @ # \$ % A & \* ( ) \_ + .
- 9 Наберіть цифри, користуючись клавішами додаткового цифрового блоку клавіатури. При цьому повинно бути ввімкнено режим .

**Література:**

- 1 Дибкова Л.М. Інформатика та комп'ютерна техніка – К.: Видавничий центр „Академія”, 2002. – с. 320
- 2 Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник.- К.: Каравела, 2003. – 464 с.
- 3 Інформатика / за ред. О.І.Пушкаря – К.: Видавничий центр „Академія”, 2001. – с.696

**Питання для самоконтролю:**

- 1 Які групи клавіш має клавіатура ?
- 2 Яке призначення кожної групи клавіш?
- 3 Як влаштована алфавітно-цифрова клавіатура?
- 4 Яким чином проводиться перемикання клавіатури на потрібну мову?
- 5 Де розташована числова клавіатура?
- 6 Якою клавішею вилучити (стерти) символ зліва від курсору?
- 7 Якою клавішею вилучити (стерти) символ справа від курсору?
- 8 Якою клавішею перемістити курсор на початок рядка ? В кінець рядка ?
- 9 Для чого використовується функціональна клавіатура?
- 10 Вкажіть клавішу "Увімкнути/вимкнути режим великих/малих літер".
- 11 Вкажіть клавішу "Перемикач верхнього/нижнього регістрів".
- 12 Яке призначення спеціальних клавіш Ctrl і Alt?

# 1 Клавіатура ПК

*Клавіатура* – пристрій, за допомогою якого здійснюється введення даних і команд. Розрізняється по кількості клавіш і наявності додаткових пристроїв. Стандартна клавіатура має 102 – 104 клавіші. На персональних комп'ютерах використовується стандартна (однакова для всіх країн світу) клавіатура IBM PS Клавіатура IBM PC складається, з 5 груп клавіш:

- алфавітно-цифрова клавіатура для введення літер англійського, російського та українського алфавітів, арабських цифр, розділових знаків і спеціальних символів {#, @, \$, %, &...};
- числова клавіатура для набору чисел і знаків арифметичних дій; клавіатура керування курсором і редагування;
- функціональна клавіатура F1-F12;
- спеціальна клавіатура.

## Алфавітно-цифрова клавіатура

Алфавітно-цифрова клавіатура має чотири ряди символічних клавіш, на яких можуть бути зображені чорним кольором англійські, червоним — російські, сірим - відмінні від російських українські літери (Й, І, Ї, Є), а також розділові знаки і спеціальні-символи. В нижній частині клавіатури розташована довга клавіша - "пропуск".

*Користуючись однією і тією ж клавішею, можна отримати від 2 до 6 різних символів.*

*Наприклад, за допомогою такої клавіші можна отримати 6 символів: англійські S і s; російські **БІ** і **Ы**; українські **І** і **і**.*

*Клавіатура може забезпечувати роботу англійською (En), російською (Ru) і українською (Uk) мовами.*

*Перемикання з однієї мови на іншу відбувається одночасним натисканням двох клавіш Ctrl+Shift (Alt+Shift).*

Встановити потрібну мову ще можна за допомогою панелі задач: клацнути на індикаторі мови, з'явиться список з переліком мов, у якому досить клацнути на потрібному рядку

В кожній з названих мов клавіатура може працювати у нижньому регістрі - вводити малі літери і нижні символи, або у верхньому регістрі вводити великі літери і верхні символи.

Якщо є потреба набирати текст великими літерами, натискають і відпускають клавішу Caps Lock - клавіатура переходить у верхній регістр. При цьому світиться індикатор "Caps Lock", і при натисканні на клавішу з літерою на екрані з'являється велика літера. При повторному натисканні на клавішу Caps Lock гасне індикатор "Caps Lock" і клавіатура повертається у нижній регістр (режим малих літер).

Клавіша Caps Lock вмикає/вимикає режим великих/малих літер.

Дія клавіші Caps Lock не поширюється на клавіші з цифрами і знаками, на яких верхній і нижній регістри перемикаються тільки за допомогою клавіші Shift.

Альтернативне перемикання регістрів виконується при натисканні і утриманні клавіші Shift (лівої або правої).

Якщо на клавіатурі включений нижній регістр (не світиться індикатор "Caps Lock"), то натискання і притримування клавіші Shift перемикає клавіатуру у верхній регістр.

Якщо на клавіатурі включений верхній регістр (світиться індикатор "Caps Lock"), то натискання і притримування клавіші Shift перемикає клавіатуру у нижній регістр.

Наприклад, за допомогою такої клавіші в нижньому регістрі вводиться цифра 3 як на англійській, так і на українській або російській мовах. За допомогою натиснутої клавіші Shift можна ввести символи:

# - англійська мова;

№ - українська або російська мова.

Для того, щоб повторити який-небудь символ на екрані кілька разів, не потрібно стільки ж разів натискати на клавішу. Досить її натиснути і тримати натиснутою -

комп'ютер сам почне виводити символ за символом. Після одержання потрібної кількості символів клавішу відпускають, щоб не з'являлися зайві символи.

#### Числова клавіатура

При введенні чисел можна користуватися клавішами з цифрами алфавітно-цифрової клавіатури. Але вводити числа за їх допомогою протягом значного часу, як це робить бухгалтер, економіст, банківський працівник тощо, не зручно. Тому на стандартній клавіатурі справа є додаткова клавіатура, як у калькулятора для введення цілих чисел і десяткових дробів, а також знаків арифметичних дій.

При натисканні на клавішу Num Lock числова клавіатура вмикається, при повторному натисканні - вимикається.

Клавіші Enter, + (додавання), -(віднімання), \*(множення), /(ділення) клавішею Num Lock не перелякуються.

А коли клавіша Num Lock ввімкнута і світиться індикатор "Num Lock", додаткова клавіатура працює як числова.

Наприклад: при натисканні на клавішу "7 Номе" на екрані з'явиться цифра 7. при натисканні на "(0 Ins)" - цифра 0.

Треба мати на увазі, що "з мовчазної згоди" при натискання на клавішу "крапка" числової клавіатури в англійській мові з'являється кома, в українській або російській мовах - крапка.

#### Клавіатура керування курсором і редагування

На комп'ютері при редагуванні текстів необхідно курсор перемішувати по тексті, щоб вказати місце, де виправити помилки, вилучити або вставити слова і речення. Для виконання цієї роботи зручно користуватися додатковою клавіатурою із двох груп (6 і 4) клавіш.

Дві групи клавіш керування курсором і редагування, які мають постійну дію і клавішею Num Lock не перемикаються,

Група у 6 клавіш має такі значення:

Insert (Ins) - увімкнути/вимкнути режим вставки/заміни:

Delete (Del) - вилучити символ справа від курсору;

Home - перемістити курсор на початок рядка:

End - перемістити курсор в кінець рядка:

Page Up - перейти на попередню екранну сторінку тексту:

Page Down - перейти на наступну екранну сторінку тексту.

Групу у 4 клавіші називають стрілочними клавішами і використав для пересування по тексті.

Крім того, при повторному натисканні на клавішу Num Lock числова клавіатура вимикається, гасне індикатор "Num Lock" і вмикається клавіатура керування курсором і редагування.

#### Функціональна клавіатура

Функціональна клавіатура складається з клавіш від F1 до F12. Багато програм використовують їх для виконання різноманітних дій-функцій.

Наприклад, в багатьох програмах при натисканні на клавішу F1 виводиться на екран текст допомоги (інструкція по роботі з програмою). В текстових редакторах при натисканні на клавішу F7 проводиться перевірка правопису.

Функціональні та інші клавіші використовуються також у комбінації із клавішами Ctrl, Alt, Shift для подачі різноманітних команд. Натискається і УТРИМУЄТЬСЯ, наприклад, клавіша Ctrl (Alt, Shift) і потім натискається одна з функціональних клавіш. Наприклад, комбінація клавіш Alt+F4 використовується для закриття активного вікна, Shift+F12 - збереження файлу на магнітний диск тощо.

#### Спеціальна клавіатура

Спеціальні клавіші знаходяться в різних місцях клавіатури і служать для подачі різноманітних команд керування.

Esc - відміна дії або команди, вихід із програми;

Tab - табуляція, переміщення курсору вправо на 8 символів у тексті, переміщення по комірках таблиці або перехід до наступного елементу діалогового вікна;

Caps Lock - увімкнення/вимкнення режиму великих літер (індикатор "Caps Lock" світиться - увімкнутий режим великих літер);

Shift - увімкнення альтернативного регістру, працює для всіх клавіш і на різних мовах;

Ctrl - спеціальна клавіша, використовується в комбінації з іншими клавішами;

Alt - додаткова спеціальна клавіша, використовується в комбінації з іншими клавішами;

Enter - в тексті - перехід на новий рядок, у вікнах Windows підтвердження виконання певних дій;

(Backspace, BS) - вилучити (стерти) символ зліва від курсору;

Print Screen - записати копію екрана у буфер, як малюнок;

Scroll Lock - має спеціальне призначення.

Pause - зупинка виконання програми, після натискання будь-якої клавіші програма продовжує роботу;

Ctrl+Pause - припинення роботи виконуваної програми;

Ctrl+Alt+Del - перезавантаження операційної системи (можна використовувати кнопку Reset на системному блоці).

## Самостійна робота № 2

**Тема:** Характеристика сервісних програм

**Мета:** отримати загальні відомості про комп'ютерні віруси, методи стиснення інформації; типи архівних файлів; програми-архіватори; сформувані вміння: працювати з програмами-архіваторами.

**Питання, що виносяться на самостійне вивчення:**

2.2 Дисккові сервісні програми. Захист від комп'ютерних вірусів. Стиснення інформації

**Практичне завдання:**

- 1 Проведіть тестування оперативної пам'яті, протестуйте та проведіть "лікування" від вірусів диск с:
- 2 Створити папку Архів у своїй робочій папці .
  - 2.1 Скопіювати в папку Архів по одному файлу будь-якої програми.
  - 2.2 Запустити на виконання архіватор WinRAR.
  - 2.3 Оцінити можливий ступінь і час стиснення файлів, що знаходяться в папці Архів, за допомогою різних форматів і методів стиснення , а потім розархівувати всіма способами. Архівні файли помістити в папці Архів.

**Література:**

- 1 Дибкова Л.М. Інформатика та комп'ютерна техніка – К.: Видавничий центр „Академія”, 2002. – с. 320
- 2 Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник.- К.: Каравела, 2003. – 464 с.
- 3 Інформатика / за ред. О.І.Пушкаря – К.: Видавничий центр „Академія”, 2001. – с.696

**Питання для самоконтролю:**

- 1 Поняття і сутність архівації.
- 2 Поняття архівного файлу і його структура.
- 3 Основні параметри, що характеризують програму-архіватор
- 4 Функціональні можливості програми WinZip
- 5 Інтерфейс програми WinZip
- 6 Функціональні можливості програми WinRar
- 7 Інтерфейс програми WinRar
- 8 Порівняльна характеристика програм WinRar і WinZip
- 9 Поняття комп'ютерного вірусу.
- 10 Класифікація комп'ютерних вірусів.
- 11 Структура комп'ютерного вірусу.
- 12 Схема функціонування комп'ютерного вірусу.
- 13 Ознаки появи вірусу.
- 14 Програми виявлення і захисту від вірусів.
- 15 Вибір антивірусних програм.
- 16 Практичні міри захисту від вірусів

## 1 Захист від комп'ютерних вірусів.

Спочатку небагато теорії. По-перше, вірус - це програма (Деякі до цього часу не знають). І шкодити вона може лише програмно, але ніяк не апаратно – страшні казки ходять про віруси, які вбивають і зводять з розуму користувачів за допомогою виводу на екран небезпечної для людини кольорової гами, були і будуть казками. Далі - вірус - це програма, спроможна до розмноження. Існують віруси, котрі не займаються нічим, крім розмноження.

Всі віруси можна поділити на групи :

1) **Завантажувальні віруси** – Заражають завантажуючі сектори HDD;FDD.

2) **Файлові віруси** – Заражають файли. Ця група в свою чергу поділяється на віруси, які заражають виконувальні файли (COM-, EXE-віруси); файли даних (макровіруси); віруси – супутники, які використовують імена інших програм; віруси сімейства DIR, які використовують інформацію про файлову структуру. Причому два останніх типи зовсім не модифікують файли на диску.

3) **Завантажувально-файлові віруси** – спроможні вражати, як код завантажувальних секторів, так і код файла. Віруси поділяються на резидентні та нерезидентні. Перші при отриманні керування, завантажуються в пам'ять і можуть діяти на відміну від нерезидентних не тільки під час роботи зараженого файла.

4) **STEALTH-віруси** фальсифікують інформацію, читаючи з диску так, що активна програма отримує не вірні дані. Вірус перехоплює вектор призупинення INT 13h і поставляє читаючій програмі іншу інформацію, яка показує, що на диску “все в нормі”. Ця технологія використовується як в файлових, так і в завантажувальних вірусах.

5) **Ретровірусами** називаються звичайні файлові віруси, котрі заражають антивірусні програми, знищують їх або роблять їх непрацездатними. Тому практично всі антивіруси, в першу чергу перевіряють свій розмір і контрольну суму файлів.

6) **Multipartition** – віруси можуть вражати одночасно EXE, COM, boot-сектор, MBR, FAT і директорії. Якщо вони до того ж володіють поліморфними властивостями і елементами невидимості, то стає зрозуміло, що такі віруси - одні з найбільш небезпечних.

В класифікації вірусів Dr.Solomon's присутні також “троянські програми” (TROJANS), котрі проводять шкідливі дії замість оголошених легальних функцій або наряду з ними. Вони не спроможні на самовідтворення і передаються тільки при копіюванні користувачем. Цікаве явище презентує собою поганий вірус – як результат порчі реального вірусу або просто погано написаний програмістом. Такий вірус нічого не може зробити – або “висне” при виконанні, або не може заражати інші файли. Інколи йде інший процес – вірус виконує непродумані дії, котрі ведуть до знищення інформації. Серед авторів вірусів не часто зустрічаються погані програмісти. Всього на сьогоднішній день існують тисячі вірусів, але тільки в декількох десятках із них реалізовані оригінальні ідеї, інші є лише “варіаціями на тему”.

## ОЗНАКИ ЗАРАЖЕННЯ ВІРУСОМ

1	Збільшення розміру пам'яті
2	Уповільнення роботи комп'ютера
3	Затримки при виконанні програм
4	Незрозумілі зміни в файлах
5	Зміна дати модифікації файлів без причини
6	Незрозумілі помилки Write-protection
7	Помилки при інсталяції і запуску WINDOWS
8	Відключення 32-розрядного допуску до диску
9	Неспроможність зберігати документи Word в інші каталоги, крім TEMPLATE
10	Погана робота дисків

Ранні ознаки зараження дуже важко виявити, але коли вірус переходить в активну фазу, тоді легко помітити такі зміни :

1	<b><i>Зникнення файлів</i></b>
2	Форматування HDD
3	Неспроможність завантажити комп'ютер
4	Неспроможність завантажити файли
5	Незрозумілі системні повідомлення, музикальні ефекти і т.д.

## ДЕЯКІ МІФИ ПРО ВІРУСИ

### 1) **Віруси самопоширюються**

Віруси не можуть виконувати себе. Із цього виходить, що вони не поширюються самі. Вірус не може нічого зробити, перед тим як заражені програми не завантажаться або комп'ютер не перевантажиться з зараженого диску.

### 2) **Віруси можуть поширюватися між будь-якими комп'ютерами**

В теорії можна написати вірус, котрий може функціонувати в різних ОС, але це завдання дуже важке. На практиці можна передбачити, що DOS-віруси неспроможні заразити такі комп'ютери, як, наприклад, Macintosh, Unix, Vax.

### 3) **Віруси можуть заразити захищені від запису диски**

Віруси не можуть заразити захищені від запису диски. Однак диски можуть бути заражені, коли захист виключений.

### 4) **Деякі віруси абсолютно не шкідливі**

Є віруси, котрі не знищують інформацію, але вони збільшують навантаження на процесор і змінюють програмний код без відома користувача.

### 5) **Тільки в піратських дисках знаходяться віруси**

Часто віруси знаходяться в піратських копіях, але відомі випадки, коли комерційне ПЗ мало віруси.

### 6) **Віруси можуть руйнувати комп'ютери**

Час від часу з'являються слухи про віруси, котрі руйнують монітор, або руйнують HDD, але ні разу це не підтвердилось.

## Що робити знайшовши вірус?

***Не панікувати!***

Засоби захисту від вірусів поділяються на такі групи, як детектори, фаги, ревізори, охоронці, вакцини.

**Детектори** (сканери). Їх метою є постановка діагнозу, лікуванням буде займатися інша антивірусна програма або професійний програміст – “вірусолог”.

**Фаги** (поліфаги). Програми спроможні знайти і знищити вірус (фаги) або декілька вірусів (поліфаги). Сучасні версії, як правило, проводять евристичний аналіз файлів – вони досліджують файли на предмет коду, характерного для віруса.

**Ревізори**. Цей тип антивірусів контролює всі (відомі на момент випуску програми) можливі способи зараження комп'ютерів. Таким чином, можливо знайти вірус, створений вже після виходу програми-ревізора.

**Охоронці**. Резидентні програми, постійно знаходяться в пам'яті комп'ютера і контролюють всі операції.

**Вакцини**. Використовуються для обробки файлів і завантажувальних секторів з метою попередження зараження відомими вірусами (в останній час цей метод використовується все частіше). Як відомо, ні один з даних типів антивірусів не забезпечує 100% захисту комп'ютера, і їх бажано використовувати в зв'язку з іншими пакетами. Вибір тільки одного, “найкращого” антивіруса вкрай помилковий.

Тепер про деякі характеристики антивірусних пакетів. Перше, на що треба звернути увагу, це кількість розпізнаючих сигнатур – послідовність символів, гарантовано виявляючих вірус. Треба помітити, що виробники використовують різні системи підрахунку сигнатур : якщо в одних різні версії або близькі по характеристиках версії вірусів рахуються за одну сигнатуру, то другі підраховують всі варіації. Найкращі із пакетів розпізнають біля 10 тисяч вірусів, що декілька менше загального числа існуючих сьогодні шкідливих програм. Другий параметр – наявність евристичного аналізатора невідомих вірусів, його присутність дуже корисна, але суттєво уповільнює час роботи програми.

Попробуємо розібратися з тими антивірусами, котрі зараз можна реально знайти на українському ринку або в INTERNET. Мова піде про комплексні антивірусні пакети, які забезпечують максимальний рівень захисту вашої інформації.

Серед російських розробників найбільш відомими є комплект від “ДіалогНауки” і AntiViral Toolkit Pro by Eugene Kaspersky від НТЦ КАМІ. Почнемо з продуктів “ДіалогНауки”, оскільки ці програми вже стали деяким стандартом, і подавляюча більшість комп'ютерів в нашій країні укомплектовано саме їх антивірусами.

# АНТИВІРУСНИЙ КОМПЛЕКТ ВІД “ДІАЛОГ-НАУКА”

## AIDSTEST

На початку 90-х достатньо було мати в себе цю програму і думати, що комп'ютер в повній безпеці : питання було лише в постійному її обновленні. Але часи змінюються, і тепер, крім Aidstest, не завадило б мати ще якісь програми.

Aidstest являється поліфагом. Це значить, що він може знаходити і знищувати відомі йому віруси. Програма розпізнає приблизно 2 тисячі вірусів. Оскільки він використовує сигнатурний пошук, то не може справлятися з поліморфними вірусами. Він не може також перевіряти упаковані файли і файли захищені вакциною, не має евристичного аналізу. “ДіалогНаука” включає Aidstest в свій антивірусний комплект, як безкоштовний додаток.

## DRWEB

Сильний антивірус з сильним алгоритмом знаходження вірусів. Він також, як і Aidstest, є поліфагом, однак, DrWEB може “читати” упаковані файли і архіви, файли даних в форматах Word і Excel, розброює поліморфні віруси, котрі в останній час, отримують все більше простору. Достатньо сказати, що епідемію дуже небезпечного віруса **OneHalf** зупинив саме DrWeb. Евристичний аналізатор DrWeb, досліджуючий програми в пошуці участків коду, характерних для вірусів, дозволяє знайти біля 90 невідомих вірусів. При завантаженні програми першим ділом DrWeb перевіряє самого себе на цілісність, після чого тестує ОЗП – в залежності від настройки, 640Кb або 1024Кb (включаючи НМА). Бажано перевіряти всю пам'ять – в цьому випадку процес перевірки триває більше, але справа в тому, що вже давно існують віруси спроможні завантажуватись в верхню пам'ять. Алгоритм роботи цього антивіруса заключається в тому, що він емулює процесор (створює програмну модель комп'ютера). Нові версії з'являються нечасто. По висновку останнього тестування журналом “Virus Bulletin” DrWeb вперше зайняв 3 місце серед 24 антивірусів.

Програма може працювати у діалоговому режимі, має дуже зручний інтерфейс, який можна налаштувати.

Для запуску програми необхідно ввести у командний рядок DOS команду :

Диск :\ Шлях \ **drweb.exe**

Після натискання клавіші ENTER на екрані з'явиться головне вікно. У верхній частині вікна зображується меню: **Dr.Web, Тест, Налаштування, Додатки, Допомога**.

Призначення меню :

**1. Dr.Web** – використовується для отримання інформації про програму, тимчасового виходу в DOS та завершення роботи програми.

**2. Тест** – дозволяє запустити програму в режимі перевірки та лікування файлів.

**3. Налаштування** – використовується для налашки інтерфейса програми та зміни режимів її роботи.

**4. Дополнення** – забезпечує підмикання зовнішніх файлів – баз даних, які мають інформацію про нові віруси.

**5. Поміць** – призначена для отримання довідкової інформації.

Режим пошуку вірусів вмикається вибором команди тестування в меню **Тест**, або натискуванням клавіші **F5**. При цьому на екрані над головним вікном з'являється діалогова панель **Путь для тестирования**. У рядку введення цієї панелі потрібно указати диск, каталог (каталоги) або групи файлів, де потрібно шукати віруси.

Тестування починається після натискування кнопки **ОК** діалогової панелі. Для тестування з лікуванням потрібно натиснути **Ctrl+F5**.

### **ANTIVIRAL TOOLKIT PRO BY EUGENE KASPERSKY**

Цей антивірус по популярності не набагато поступається комплекту від “ДіалогНаука”. AVP являється поліфагом і в процесі роботи перевіряє ОЗП, файли, в тому числі упаковані і архівні, а також системні сектори (Master Boot Record), завантажувальний сектор (Boot – сектор) і Partition Table. На відміну від DrWeb і Aidstest, AVP розпізнає біля 10000 вірусів, серед них поліморфні, stealth – і макровіруси, а також “Троянські програми”. Така різниця пояснюється тим, що “ДіалогНаука” незначні варіації одного вірусу приймає за одну сигнатуру, а КАМІ – різними вірусами. Програма має евристичний сканер, котрий, за затвердженням розробників антивіруса із КАМІ, знаходить біля 80 всіх вірусів. Нові бази антивірусів до AVP з'являються приблизно один раз в тиждень.

## **2 Стиснення інформації**

### **Архіватор**

При експлуатації комп'ютера з різних причин можливе пошкодження або втрата інформації на магнітних дисках.

Це може відбутися внаслідок як фізичного пошкодження дисків, так і пошкодження інформації вірусом. І тому для зменшення втрат в таких випадках, потрібно мати архівні копії файлів.

Для отримання копій файлів, використовують команди копіювання MS DOS. Але в цьому випадку копії будуть займати багато місця, що змушує мати велику кількість дискет. Більш доцільно використовувати для створення архівних копій спеціально розроблені програми. В чому ж полягає ефект таких програм:

Заархівовані архіватором програми займають в середньому від 20 до 90% свого початкового обсягу, що дозволяє на тій же кількості носіїв зберігати більшу кількість інформації.

Архіватор об'єднує групу програм під одним іменем, що дозволяє запобігати втраті деяких файлів.

Можливе поновлення в архіві тільки нових версій файлів.

Архіватор зберігає в архіві імена каталогів та файлів.

Надає можливість написання коментарів до архіву.

Користувач створює архівні файли.

Для архівації файлів використовують спеціальні програми, які називаються архіваторами. Перші архіватори з'явилися в 1985 році. Вони можуть об'єднувати в один файл цілу групу файлів, включаючи і каталоги.

Можливості сучасних архіваторів широкі і різноманітні. Але можна виділити функції, які є загальними. До них відносять: можливість додавання файлів в архів, поновлення архіву, перегляд файлів в архіві, знищення файлів в архіві, захист файлів від несанкціонованого доступу, вилучення файлів з архіву, перевірка цілісності архіву та ін.

Архівний файл може використовуватись тільки після того, як він буде відновлений у початковому вигляді, тобто розархівований.

До числа найвідоміших в недалекому минулому та широковикористовуваних архіваторів (розархіваторів) слід віднести: PKZIP.EXE, PKUNZIP.EXE, ARJ.EXE, PKPAK.EXE, PKUNPAK.EXE, LHA.EXE. Тепер існують багато більш нових програм.

### WinZip

Найбільш визнана серед усіх програм-архіваторів. WinZip можна знайти практично на кожному персональному комп'ютері. Вона необхідна, наприклад, для пересилки великої кількості дрібних файлів (як правило, мова йде про тексти і картинки) по електронній пошті, для стиснення об'ємних документів у форматі doc, і так далі

Користувачам подобається WinZip за простоту, зручність, ефективний алгоритм стиску і повсюдну поширеність

#### Інсталяція

**winzip\*.exe > Setup > OK > Next > Yes** > стартовий режим "Start with WinZip Classic", **Next** > цілком можна обійтися стандартною установкою "Express setup", **Next > Finish**.

Якщо користувач не має потреби в підказках, необхідно обрати відмовлення від "tips" > **Close**.

#### Робота з програмою

Припустимо, у директорії XXXX знаходяться файли X1, X2, ..., Xn. Їх треба запакувати в єдиний архів. Необхідно знайти папку XXXX, виділити файли X1, X2, ..., Xn (використовуючи ліву клавішу миші і кнопки **Ctrl** чи **Shift**). Потім після правого кліка мишею вибрати команду **Add to Zip**.

При вказівці назви архівного файлу розширення zip можна не вказувати, досить вписати тільки ім'я (у нашому випадку "x") і натиснути на кнопку **Add**. Результат - поява файлу x.zip у папці XXXX.

Необхідно звернути увагу на той факт, що стиск в архів документів Word (doc чи rtf) дає дуже значну економію дискового простору (до 95%). Більш скромний результат досягається при стиску html-файлів (до 80%). Складніше з графічними файлами, bmp можна стиснути і до 99%, а от, скажемо, при архівуванні tif, jpg, gif економія складе, відповідно, до 35, 25, 10%. Однак при відправленні файлів по електронній пошті необхідно враховувати не тільки їхній фізичний розмір, але і їхнє число, адже на обробку запиту про відправлення кожного файла потрібен час.

На користь закриття zip-ом файлів перед відправленням по e-mail говорить і той факт, що дуже багато одержувачів кореспонденції використовують всілякі антивірусні програми і firewalls, що можуть

повернути який-небудь doc-файл через його "підозрілість", а zip-файл, швидше за все, дійде до адресата.

Для того, щоб відкрити zip-файл достатньо, в умовах Windows, подвійного лівого кліка на zip-файл, і негайно піде відкриття вікна. Потім потрібно клікнути на кнопку меню **Extract**, вибрати місце файлу(ів) і підтвердити свій намір натисканням на **Extract** у новому робочому вікні.

Якщо користувач бажає створити нову папку для розпакованих файлів, необхідно скористатися кнопкою **New Folder**.

Розглянемо докладніше робочий інтерфейс програми.

Через **File** можна створити новий, чи відкрити закрити архів, добратися до таблиці обраних архівів - **Favorites** (формується користувачем), переглянути властивості поточного zip-файлу, винести іконку з посиланням на нього на робочий стіл, а також діяти як зазвичай із самим архівом (переносити, копіювати, перейменовувати, видаляти, роздруковувати і прив'язувати до поштових служб). Аматори які усюди йдуть з "майстром підказок" під рукою можуть вибрати режим **Wizard** (є також окрема кнопка).

Меню **Actions** відкриває широкий спектр можливостей (більшість з яких дублюється спеціальними кнопками): додати (**Add**), видалити (**Delete**), розпакувати (**Extract**), переглянути окремі файли в архіві (**View**), виділити усі файли в архіві (**Select All**) і скасувати це дія (**Invert Selection**). Відразу є можливість перевірити заархівовані файли на предмет виявлення вірусів (**Virus Scan**). Для того, щоб зробити архів "саморозпаковуючимся", і перетворити його в exe-файл, що виконується, необхідно обрати команду **Make .Exe File** (керування передається програмі WinZip Self-Extractor). Якщо при цьому потрібно "розкласти" архів, що саморозпаковується, по дискетах, необхідно обрати опцію **Safe spanning method**. Серед додаткових можливостей цього меню: перетворення архівного файлу в універсальний формат пересилання по електронній пошті UUE, попередня перевірка архіву на розпаковуєчість, підготовка коментаря до архіву, створення окремого меню в розділі **Start > Programs (CheckOut)**.

У **Options** спочатку потрібно звернути увагу на можливість закрити архів на пароль (**Password**). Це може стати у нагоді як для елементарної гарантії приватності переписування, так і для викладання в Мережу комерційного продукту. Просунутий користувач може дозволити собі щось поміняти в конфігурації WinZip (**Configuration**). Рядок меню **Sort** відкриває доступ до різних правил сортування файлів усередині архіву.

Графічне меню дублює деякі, найбільш популярні функції програми.

## **WinRar (російська версія)**

Конкурент програми WinZip, тому що має в наявності російську версію. На світовому ринку програмних засобів відома як програма-архіватор з одним із кращих алгоритмів стиску даних. Підтримує багатотомне архівування, відновлення "зіпсованих" архівів, блокування їх модифікації. На відміну від WinZip, припустимий розмір файлу для архівування не обмежений (для WinZip - обмеження 4 Gb). У WinRar є можливість скористатися спеціальним алгоритмом мультимедіа-стискання, оптимізуючим компресію оцифрованого звуку і графічних файлів.

Інсталяція

**wrar\*.exe > Встановити > ОК > Готово.**

Робота з програмою

Найпопулярнішим засобом звертання до програми (як і у випадку WinZip) є вибір WinRar у меню, що спливає після правого кліка мишею. Тут з'явилися нові можливості "Додати в архів..." чи "Додати в \*.rar". Вибір першої дає трохи більший простір для маневру.

Примітно, що WinRar з повагою відноситься до користувачів, що звикли до архіватора WinZip і надає можливість архівувати у форматі zip (правда, при цьому істотно обмежує функціональні можливості архівування. Користувач у такий спосіб може на наочному прикладі переконатися, що WinRar домагається істотно кращих результатів стиску в порівнянні з WinZip. Вам пропонується вибір: чи універсальність формату, чи ефективність компресії. Крім того, WinRar підтримує так зване "багатотомне" архівування, що значно перевершує по зручності метод розбивки "на дискети" використовуваний у WinZip. Утім, на користь останнього говорить не тільки широка поширеність, але і той факт, що в середньому він архівує швидше, ніж WinRar.

Для створення "exe-архіва", що саморозпаковується, необхідно обрати опцію "Створити SFX-архів" (SFX - від Self-eXtracting). Також потрібно мати на увазі, що такий exe-файл завжди буде істотно важче свого гаг-двійника .

Для створення багатотомного архіву (rar чи exe) потрібно вказати розмір тому (за замовчуванням у байтах, але є можливість і дописувати k, K, m, M, де 1k=1024 байт, 1K=1000 байт, 1m=1024k, 1M=1000K). Багатотомний rar-архів прийнято іменувати "беззупинним", тому що весь стиснутий файл розглядається, як єдин потік даних. "Беззупинні" rar-архіви варто використовувати, коли ступінь стиску істотніше швидкості процесу архівації, при цьому архів не передбачається оновляти, витягати з нього окремі файли. Проте швидкість створення багатотомного exe-архіву, що саморозпаковується, ще нижче, ніж у rar-двійника.

## Самостійна робота № 3

**Тема:** Операційна система Windows

**Мета:** набути навички роботи з об'єктами у середовищі Windows

**Питання, що виносяться на самостійне вивчення:**

3.2 Робота з об'єктами у середовищі Windows

### Література:

- 1 Дибкова Л.М. Інформатика та комп'ютерна техніка – К.: Видавничий центр „Академія”, 2002. – с. 320
- 2 Інформатика. Комп'ютерна техніка. Комп'ютерні технології: Підручник.- К.: Каравела, 2003. – 464 с.
- 3 Інформатика / за ред. О.І.Пушкаря – К.: Видавничий центр „Академія”, 2001. – с.696

**Питання для самоконтролю:**

- 1 Вікна яких типів використовуються у Windows ?
- 2 Як працювати з меню у вікнах програми?
- 3 Що розташоване на робочому столу Windows?
- 4 Як здійснюється інсталяція програмних засобів?
- 5 Які дії з вікнами можна виконувати у Windows?
- 6 Як поновити вилучені файли?
- 7 Як створити ярлик для файла?

### 3.2 Робота з об'єктами у середовищі Windows

Інтерфейс Windows розрахований для роботи з маніпулятором типу миша. Без нього робота стає практично неможливою. Однак у багатьох випадках не виключено використання клавіатури (окремих клавіш або їх комбінацій). Маніпулятор миша не замінює, а доповнює клавіатуру. Миша в багатьох випадках полегшує та прискорює роботу користувача ПК.

Мишу використовують для:

- виділення (вибору) і перетягування об'єктів;
- введення команд шляхом вибору елементів меню;
- вказування певної позиції на екрані;
- запуску додатків;
- натискування екранних кнопок;
- виконання операцій редагування тощо.

На миші може бути дві чи три кнопки і скролер. Переважно користуються тільки двома: правою крайньою та лівою крайньою, а середню не використовують. При переміщенні миші поверхнею столу, синхронно переміщується на екрані і вказівник цієї миші. Таким чином, розташувати вказівник миші можна у будь-якій точці екрана.

Вказівник миші – це Ваш електронний палець. Щоб натискувати “пальцем” на кнопки, зображені на екрані, потрібно перемістити вказівник на зображення кнопки і клацнути головною кнопкою миші (див. нижче).

Вигляд вказівки миші змінюється залежно від того, яка операція виконується з використанням миші. Наведемо для прикладу деякі:

Зовнішній вигляд	Ситуація (операція)	Описання
	Основний режим	Використовується при вказуванні на об'єкт і виконання клацання.
	Вибір справки	Після виконання клацання на довільному елементі екрана таким вказівником з'являється довідкова інформація про даний елемент.
	Система недоступна	Така форма вказівника миші показує, що програма занята і в даний момент керувати нею користувач не може.
	Виділення тексту	Таким вказівником можна вибрати позицію в тексті або виділити текст.
	Операція неможлива	У даній позиції дію виконати за допомогою миші неможливо.
	Зміна вертикальних розмірів	Використовується при зміні розмірів вікон.
	Зміна горизонтальних розмірів	Використовується при зміні розмірів вікон.
	Зміна розмірів по горизонталі	Використовується при зміні розмірів вікон.
	Переміщення	Використовується при зміні позиції вікна.

Використовувати кнопку – це означає натиснути на неї. Спочатку потрібно розмістити вказівник миші на екрані на потрібний об'єкт, а потім натиснути на кнопку миші один або два рази.

Мишею можна виконати такі дії:

#### 1. *Одне клацання.*

Короткочасне натискування і відпускання кнопки миші, коли її вказівник знаходиться в потрібній позиції екрана, називають **клацанням**.

Ця дія найбільш поширена. Одне клацання – це просто вказівка на об’єкт і швидке натискування і вивільнення головної кнопки миші. Головна кнопка миші найчастіше ліва. Але можна настроїти мишу так, що головною буде права кнопка.

При цій дії відбувається вибір або виділення довільного об’єкта, команди меню, натискування екранних кнопок.

Якщо ми будемо використовувати слово “вибрати”, то його можна замінити словом “клацнути”.

## 2. Подвійне клацання.

Подвійне клацання представляє собою натискування і вивільнення головної кнопки миші два рази підряд у швидкому темпі, без переміщення миші між клацаннями.

При цій дії відбувається запуск програм.

## 3. Допоміжне клацання.

Допоміжне клацання заключається в натискуванні і вивільненні кнопки, яка не вважається головною.

Така дія часто викликає контекстне меню та інші потрібні речі.

*Контекстне меню* – це меню, що вміщає групу команд, які безпосередньо відносяться до об’єкта, для якого це меню викликане.

## 4. Перетягування.

Переміщення вказівника миші, який знаходиться на виділеному об’єкті, при натиснутій кнопці миші називають перетягуванням.

Для того щоб перетягнути об’єкт, слід помістити на нього вказівник миші, потім натиснути головну кнопку миші і, утримуючи її, переміщати мишу, тим самим перетягуючи об’єкт у потрібне місце на екрані. В кінці перетягування кнопку миші потрібно відпустити.

*Отже, кнопки миші використовуються для того, щоб здійснювати якісь дії. Спочатку потрібно розмістити вказівник миші на екрані на потрібний об’єкт, наприклад, на піктограму, а потім натиснути на кнопку миші один або два рази.*

Операційна система Windows дозволяє внести зміни в параметри, які впливають на маніпулювання миші (команда Пуск/Настройка/Панель управління/Мышь). До них відносяться швидкість вказівника миші, швидкість виконання подвійного клацання, зміна головної кнопки, зміна вигляду вказівника миші.




### **Анатомія вікна та основні дії над вікном**

Слово “windows” в перекладі з англійської означає “вікна”.

*Вікно* у ОС Windows – це частина екрану, обмежена прямокутною рамкою, всередині якої оперує прикладна програма, відображається вміст папки або документа. Вікно можна переміщати, збільшувати, зменшувати, згортати, відкривати і закривати.

Одночасно може бути відкрито декілька вікон. Одне з них є активним, тобто це те вікно, з яким у даний момент можна працювати. Активне вікно автоматично виводиться на передній план.

У верхній частині кожного вікна є рядок заголовку (рис.6.2). В ньому знаходиться:

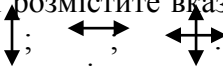
- кнопка виклику системного меню: розкриває системне меню, з допомогою якого здійснюється управління розміром і позицією вікна;
- ім’я програми, а інколи ім’я документа;
- кнопки управління вікном, які дозволяють виконати над вікном такі дії:  - згортає вікно в значок на панель задач;  - розгортає вікно до розмірів екрана або до розміру вікна прикладної програми;  - закриває вікно. Якщо дані у вікні не були збережені, то при закритті вікна на екрані з’являється запит на збереження.

Колір рядка заголовка вказує, чи є вікно активним чи ні. Для активних вікон по замовчужанню використовується насичений колір, у неактивних вікон рядок заголовку не має кольору або заглушений.

У кожному вікні є рядок меню, в якому можна знайти всі необхідні команди для управління Windows-програмою.

Панель інструментів – ряд екранних кнопок, які полегшують роботу користувача (див. нижче).

*Рядок стану* – це рядок, в якому відображається певна інформація, зміст якої залежить від програми, в якій працюємо і від дій, які ми виконуємо (див. нижче).

*Зміна розмірів вікна.* Якщо Ви розмістите вказівник миші на край вікна, то він може прийняти один з таких виглядів: . Утримуючи кнопку миші натиснутою, перетягуємо край вікна до тих пір, поки вікно не досягне потрібних розмірів

*Зміна положення вікна.* Щоб перемістити вікно, потрібно розмістити вказівник миші на рядок заголовка вікна, натиснути кнопку миші і переміщувати вказівник миші в потрібному напрямку. При відпусканні кнопки миші воно зафіксується в потрібному положенні (цю дію можна також виконати за допомогою системного меню).

Перегляд вмісту вікна з допомогою смуги прокручування

Вікна можуть мати смуги прокручування, які розміщені на правому і на нижньому краях вікна. Вони з'являються тоді, коли не весь вміст вікна в даний момент у ньому показаний.

Функції смуг прокручування такі:

- вертикальна смуга на правому краї вікна прокручує вміст вікна у вертикальному напрямку вниз або вгору.
- горизонтальна смуга на нижньому краї вікна прокручує вміст вікна у горизонтальному напрямку вліво або вправо.

На верхньому і нижньому кінцях смуги прокручування є маленькі стрілочки, з допомогою яких можна прокручувати вміст вікна. На смугі розміщений бігунок прокручування – сірий прямокутник. Він вказує позицію вікна в межах документа. Розмір бігунка залежить від того, яка частина інформації вміщується у вікні.

Прокручувати вміст інтернет-сторінки і тексту можна також за допомогою кнопки скролера на миші.

Методи перегляду вмісту вікна за допомогою смуги прокручування:

- 1) “зачіпляємо” бігунок мишкою, тобто розміщуємо вказівник миші на бігунці, натискаємо і тримаємо кнопку миші. Не відпускаючи її, тягнемо мишею бігунок у відповідному напрямку смуги, поки потрібна позиція документа не появиться у вікні;
- 2) клацаємо по смугі (але поза бігунком). При цьому прокручується документ на розмір вікна;
- 3) розміщуємо вказівник миші на стрілку прокручування (на кінці смуги), натискаємо кнопку миші і тримаємо її натисненою. Поки кнопка буде натиснутою, вміст вікна прокручуватиметься.
- 4) з допомогою кнопок зі здвоєними стрілками, розміщеними у нижній частині вертикальної смуги прокручування, виконуємо посторінкове пролистування документа.

### **Виділення об'єктів**

1. Виділення одного об'єкта:

- клацають один раз на імені об'єкта мишею;
- за допомогою клавіш управління курсором можна змінювати виділений об'єкт.

2. Виділення декількох об'єктів, які в списку знаходяться один за одним:

- виділити перший об'єкт;
- натиснути клавішу **Shift** і, утримуючи її натисненою, клацнемо на останньому об'єкті групи, що виділяється. Всі об'єкти, розміщені між цими двома, будуть виділені.

Або:

- натиснути клавішу **Shift** і, утримуючи її натисненою, натискати відповідні клавіші управління курсором.

3. Виділення декількох об'єктів, розсіяних по списку.

- виділити перший об'єкт;
- натиснути клавішу **Ctrl** і, утримуючи її натисненою, клацнути на наступному об'єкті. Обидва об'єкти будуть виділеними і т.д.

4. Виділення всіх об'єктів у списку.

- виконати команду **Правка/Выделить все** або натиснути комбінацію клавіш CTRL+A. Щоб відмінити виділення всіх об'єктів, потрібно клацнути на довільному об'єкті в списку. В результаті виділеним залишиться тільки той об'єкт, на якому клацнули мишею. Щоб відмінити виділення окремого об'єкта в групі, потрібно натиснути клавішу **Ctrl**, і утримуючи її натисненою, клацнути на відповідному об'єкті.

Щоб обернути виділення (зробити так, щоб невиділені об'єкти стали виділеними, а виділені – невиділеними), потрібно виконати команду **Правка/Обратить выделение**.

*Операції, які можна проробити з виділеною групою файлів:*

- копіювання;
- переміщення;
- знищення.

### **Перейменування об'єктів**

*1-й спосіб:*

- необхідно виділити об'єкт, який слід перейменувати;
- виконати команду **Файл/Переименовать**;
- в полі імені файла з'явиться курсор, тобто ім'я можна редагувати;
- натиснути клавішу **Enter**.

*2-й спосіб:*

- для включення режиму редагування імені об'єкта необхідно один раз клацнути по об'єкту (виділити об'єкт), а другий раз – по полю його імені;
- другий раз клацати потрібно не відразу після першого, для того, щоб операційна система не сприйняла їх як подвійне клацання;
- при цьому текст обрамляється, фон поля тексту імені стає чорним, а вказівка миші набирає вигляду вертикальної риски (як у текстовому редакторі); можна також редагувати ім'я об'єкта.

### **Вилучення об'єктів**

1. Виділити об'єкт (об'єкти), який (які) необхідно вилучити.
2. Виконати команду **Файл/Удалить** або натискувати на клавішу **Del**.
3. Після появи на екрані діалогового вікна **Подтверждение удаления папки** натискати кнопку **Да** для підтвердження вилучення або **Нет**, якщо передумали знищувати об'єкт.

Для виключення випадкового вилучення об'єктів в операційній системі Windows 95 використовують спеціальну папку, яка називається **Корзина**. В разі вилучення об'єкти не вилучаються, а вміщуються в цю папку. Це дозволяє за необхідності відновити вилучені об'єкти. Якщо очистити вміст **Корзини**, то відновлення об'єктів, які в ній знаходились, стає неможливим.

### **Пошук об'єктів**

*Приведемо перелік критеріїв, за якими здійснюється пошук об'єктів:*

- ім'я об'єкта (або його частина) та місце його розміщення;
- інтервал часу, протягом якого цей об'єкт був створений;
- вміст файла.

*Сферу пошуку можна обмежувати:*

- типом об'єктів;
- місцем пошуку;
- певним розміром об'єкта.

Щоб запустити систему пошуку об'єктів потрібно виконати команду **Сервис/Найти/Файлы и папки**.

Після виконання цієї команди з'явиться діалогове вікно **Найти: Все файлы**, яке має три вкладки.

Опис цих трьох вкладок наведено в додатку 1.

### **Властивості об'єктів**

Залежно від типу кожний об'єкт володіє власним набором властивостей і дозволяє виконати ряд певних дій.

Команди дії, які застосовуються до об'єкта, зібрані в контекстному меню об'єкта.

Властивості певного об'єкта можна переглянути, вибравши в меню **Файл** або в **контекстному** меню пункт **Свойства**. Набір вкладок, доступних у вікні **Свойства** залежить від вибраного об'єкта.

*Встановлення атрибутів файла та каталога:*

- вибрати файл (каталог) або групу файлів та каталогів;
- відкрити меню **Файл** або викликати контекстне меню і клацнути вказівкою миші по команді **Свойства**;
- після виконання команди на робочому столі з'являється діалогове вікно **Свойства**, декілька вкладок;
- на вкладці **Общие** можна прочитати інформацію про виділені об'єкти і встановити (або відмінити) потрібні атрибути. Для цього потрібно покласти галочку навпроти відповідного атрибуту (атрибутів) і натиснути кнопку **Применить**.

***9 Відміна команди***

Майже у всіх прикладних програмах операційної системи Windows остання виконана дія (а в деяких і ряд попередніх дій) може бути відмінена.

Для цього потрібно:

- 1) виконати команду **Правка/Отменить**;
- 2) натиснути комбінацію клавіш **Ctrl+Z** або **Alt+Backspace**;

## Самостійна робота № 4

**Тема:** Основи програмування мовою Pascal

**Мета:** Оволодіння практичними навиками розробки, програмування обчислювального процесу розгалуженої структури, отримання подальших навиків по від налагодженню і тестуванню програми.

**Питання, що виносяться на самостійне вивчення:**

4.2 Умовні оператори. Оператори циклу

### Література:

- 1 Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. М., Наука, 1988.
- 2 Марченко А.И., Л.А. Марченко Программирование в среде Turbo PASCAL 7.0, Базовый курс. – Киев: «Век+», 2004
- 3 Меженный О.А. Самоучитель Turbo Pascal – М.: «Диалектика», 2004
- 4 Ставровский А. Турбо Паскаль 7.0 Учебник – Киев, «Ирина», 2000

**Питання для самоконтролю:**

- 1 Які оператори використовуються для програмування розгалужень?
- 2 Як виконуються оператори переходу?
- 3 Яку з функцій: Sin(x), Abs(x), Trunc(x) можна замінити умовним оператором if  $x < 0$  then  $x := -x$ ?
- 4 Як записується і як працює оператор FOR?
- 5 Для організації яких циклів застосовуємо оператор FOR?
- 6 У чому відмінність оператора WHILE від оператора REPEAT?
- 7 Як програмуються циклічні алгоритми з явно заданим числом повторень циклу?
- 8 Як програмуються циклічні алгоритми з заздалегідь невідомим числом повторень циклу?
- 9 Напишіть оператор циклу, що не виконується жодного разу.
- 10 Напишіть оператор циклу, що виконується необмежене число раз.

## Оператори циклу

**Цикл** – це процес виконання певного набору команд деяку кількість разів. Цикл реалізують або за допомогою конструкції if-goto, або, що значно ефективніше, за допомогою команд циклу. Є три види команд циклів: з параметром, з передумовою та з післяумовою.

**1. Команда циклу з параметром (for).** Є два різновиди команди for. Розглянемо перший:

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

Тут параметр — це змінна цілого, символьного, логічного або перерахованого типу, а вирази 1 і 2 задають початкове та кінцеве значення параметра.

*Дія команди.* Параметрові циклу присвоюється значення виразу 1. Якщо це значення менше-рівне, ніж значення виразу 2, то виконується команда 1. Після виконання команди 1 значення параметра автоматично збільшується на 1 і знову порівнюється зі значенням виразу 2 і т.д. Коли значення параметра стане більшим, ніж значення виразу 2, то виконується наступна після циклу команда.

*Приклад.* Нехай  $s=0$ . Після виконання команди циклу

**for**  $i:=4$  **to** 6 **do begin**  $s:=s+i$ ;  $z:=2*i$  **end**;

змінна  $s$  набуде значення  $0+4+5+6=15$ , а змінна  $z$  — 12.

Розглянемо другий різновид команди циклу **for**:

**for** <параметр> := <вираз 1> **downto** <вираз2> **do** <команда 1>;

Ця команда діє як попередня, але крок зміни параметра є  $-1$ .

*Приклад.* Нехай  $s=0$ . Після виконання команди циклу

**for**  $i:=6$  **downto** 4 **do begin**  $s:=s+i$ ;  $z:=2*i$  **end**;

змінна  $s$  набуде значення  $0+6+5+4=15$ , а змінна  $z$  — 8.

*Зауваження.* Значення параметра в середині циклу змінювати не можна.

**Задача 4.** Обчислити значення виразу, що містить суму і добуток

$$y = \sum_{i=1}^{25} \frac{5i^2}{i!} + \prod_{i=1}^{25} i^2$$

```
program SumaDobutok;
```

```
uses Crt;
```

```
var i: integer;
```

```
    suma, dob, y: real;
```

```
    fact: longint;
```

```
begin
```

```
    clrscr;
```

*{Спочатку сума дорівнює нулю, а добуток одиниці}*

```
    suma:=0; dob:=1; fact:=1;
```

```
    for i:=1 to 25 do
```

```
        begin
```

```
            fact:=fact*i;
```

```
            suma:=suma+5*sqr(i)/fact; {Обчислюємо  $i!=1*2*3*\dots*i$ }
```

```
            dob:=dob*i*i; {Обчислюємо суму та }
```

```
        end; {добуток }
```

```
    y:=suma+dob;
```

```
    writeln('y=', y:7:2);
```

```
    readln
```

```
end.
```

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

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

*Дія команди.* Доки значення логічного виразу істинне, виконується команда 1.

Істинний логічний вираз описує умову продовження процесу виконання команди циклу.

*Приклад.* Нехай змінні  $x$ ,  $s$  мають значення  $x=4$ ,  $s=0$ . Після виконання команди

**while**  $x \leq 8$  **do begin**  $s:=s+x$ ;  $x:=x+1$  **end**;  
вони набудуть значень  $s=4+5+6+7+8=30$ ,  $x=9$ .

*Приклад.* Вивести таблицю чисел від 20 до 30, їхні квадрати та куби за допомогою команди **while** можна так:

```
i := 20;  
while i <= 30 do  
begin  
  writeln(i:4, i*i:6, i*i*i:8);  
  i := i + 1  
end;
```

**Задача 5.** Протабулювати функцію  $y = \sin x$  на проміжку  $[0; 3,1]$ , з кроком  $h = 0,1$  і обчислити середнє арифметичне ( $s1$ ) значень функції більших, ніж 0,1 і менших, ніж 0,6.

```
program FindSerednie;  
uses Crt;  
var x, y, s, s1, h, xk: real; n : integer;  
begin  
  clrscr;  
  x := 0; xk := 3.1; h := 0.1;  
  s := 0; n:=0;  
  while x <= xk + h/2 do  
    begin  
      y := sin(x);  
      writeln(x:3:1, y:6:2);  
      if (y > 0.1) and (y<0.6) then  
        begin s := s + y; n := n + 1 end;  
      x := x + h ;  
    end;  
  if n > 0 then  
    begin s1 := s / n; writeln ('Середнє =', s1) end  
  else writeln('Таких значень немає n=0');  
  readln  
end.
```

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

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

*Дія команди.* Команди виконуються в циклі, доки значення логічного виразу не стане істинним. Істинний логічний вираз задає умову виходу з циклу.

*Приклад.* Нехай змінні  $x$ ,  $y$  мають значення  $x=5$ ,  $y=0$ . У результаті виконання команди

```
repeat  $y:=y+x$ ;  $z:=2*x$ ;  $x:=x-2$  until  $x \leq 1$ ;  
змінні  $y$ ,  $z$ ,  $x$  набудуть значень  $y=0+5+3=8$ ,  $z=6$ ,  $x=1$ .
```

**Задача 6.** Протабулювати функцію  $y=\sin(x)$  на проміжку  $[-p,p]$  з кроком  $h=p/5$ . Результати обчислень вивести на екран у вигляді таблиці. Знайти максимальне ( $\max$ ) та мінімальне ( $\min$ ) значення функції на цьому проміжку.

Максимальне чи мінімальне значення функції чи значення, яке задовольняє деякій іншій умові пошуку, визначають методом сканування (перегляду) і порівняння всіх підряд значень функції з деяким еталоном.

```
program Tabulfuncii;  
uses Crt;  
var h, x, y, max, min: real;  
begin clrscr;  
  h:=pi/5; x:=-pi;  
  max:=sin(x); min:=sin(x);  
  writeln(' x y');
```

{Пропускаємо, що max і min}  
{знаходяться в першій точці}

```
repeat  
  y:=sin(x); writeln(x:7:2, y:7:2);  
  if y>max then max:=y;  
  if y<min then min:=y;  
  x:=x+h;  
until x>pi+h/2;  
  writeln('max=', max:5:2, ' min=', min:5:2);  
readln  
end.
```

*{Визначаємо максимум функції}*  
*{Визначаємо мінімум функції}*  
*{Збільшуємо x на h}*  
*{Перевіряємо умову виходу з циклу}*

## Самостійна робота № 5

**Тема:** Основи програмування мовою Pascal

**Мета:** вивчити суть методу послідовної деталізації (послідовного уточнення), призначення підпрограм у мові Паскаль; правила запису, механізм виклику і виконання функцій і процедур користувача; поняття локальних і глобальних величин, формальних і фактичних параметрів; відмінності між процедурами і функціями.

**Питання, що виносяться на самостійне вивчення:**

5.2 Процедури і функції. Масиви

### Література:

- 1 Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. М., Наука, 1988.
- 2 Марченко А.И., Л.А. Марченко Программирование в среде Turbo PASCAL 7.0, Базовый курс. – Киев: «Век+», 2004
- 3 Меженный О.А. Самоучитель Turbo Pascal – М.: «Диалектика», 2004
- 4 Ставровский А. Турбо Паскаль 7.0 Учебник – Киев, «Ирина», 2000

**Питання для самоконтролю:**

- 1 У чому відмінність між процедурою і функцією?
- 2 Чим відрізняються формальні і фактичні параметри?
- 3 Чим відрізняються параметри-значення і параметри-перемінні?
- 4 Як з'являються глобальні і локальні змінні? Яке правило видимості цих змінних?
- 5 Що являє собою масив як структура даних?
- 6 Яким чином виводяться елементи масива на екран?
- 7 Які дані можуть виступати як індекси елементів масивів?
- 8 У яких розділах можна описати масив?

## 5.2 Процедури і функції. Масиви

Підпрограми призначені для реалізації алгоритмів опрацювання окремих частин деякої складної задачі. Вони дають змогу реалізовувати концепцію структурного програмування, суть якого полягає в розкладанні складної задачі на послідовність простих підзадач і в складанні для алгоритмів розв'язування кожної підзадачі відповідних підпрограм. Розрізняють два види підпрограм — підпрограми-процедури та підпрограми-функції. Підпрограми поділяються на стандартні та підпрограми користувача. Стандартні підпрограми створювати не потрібно — вони містяться у стандартних модулях System, Crt, Dos, Graph тощо. **Підпрограма користувача** — це поійменована група команд, яку створюють і описують в основній програмі в розділах **procedure** або **function** і до якої звертаються з будь-якого місця програми потрібну кількість разів.

### 1 Процедури (procedure)

Загальний опис процедури:

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

У списку *формальних* параметрів перераховують *змінні разом із зазначенням їхніх типів*. Розрізняють *параметри-аргументи* (інший термін: параметри-значення) — вхідні дані для процедури, і *параметри-результати* (інший термін: параметри-змінні), через які можна повертати результати роботи процедури в основну програму. Перед списками параметрів-результатів *кожного* типу записують слово **var**. Зауважимо, що масиви фіксованих розмірів у списках формальних параметрів не можна описувати за допомогою слова **array** (див. зразки програм).

Розділи описів і оголошень у підпрограмах мають таку саму структуру як і в основній програмі.

*Приклад.* Розглянемо процедуру з назвою Cina, яка визначає  $c$  — вартість  $k$  хвилин телефонної розмови з похвилинною оплатою 0.6 грн. + 20% ПДВ.

```
procedure Cina(k:integer; var c:real);  
begin  
  c:=k*0.6; c:=c+0.2*c;  
end;
```

У наведеному прикладі  $k$  є формальним параметром-аргументом,  $c$  — формальним параметром-результатом.

До процедури звертаються з розділу команд основної програми або іншої підпрограми. Процедуру викликають за допомогою команди виклику

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

Параметри, які записують у команді виклику процедури, називаються *фактичними*. Фактичними параметрами-аргументами можуть бути сталі, змінні, вирази, а параметрами-результатами — лише змінні. Типи даних тут не зазначають.

Між фактичними і формальними параметрами має бути відповідність за кількістю й типами. Зверніть увагу, відповідні фактичні та формальні параметри можуть мати різні імена.

Команда виклику функціонує так: значення фактичних параметрів присвоюються відповідним формальним параметрам процедури, виконується процедура, визначаються параметри-результати, значення яких надаються (повертаються) відповідним фактичним параметрам у команді виклику.

Змінні, описані в розділі описів основної програми, називаються *глобальними*. Вони діють у всіх підпрограмах, з яких складається програма. Змінні, описані в розділі описів конкретної процедури, називаються *локальними*. Вони діють тільки в межах даної

процедури.

Процедури можуть отримувати і повертати значення не тільки через параметри-результати, але й через глобальні змінні. Тому списків параметрів у процедурі може і не бути.

**Задача 1.** Розв'язати задачу про кількість викликів на АТС з попереднього параграфу, використовуючи три процедури: 1) для визначення кількості викликів за кожну секунду (надамо їй назву Kіlvykl); 2) для обчислення суми викликів за перші 10 секунд (Sumavykl); 3) для визначення найбільшої кількості викликів за деяку секунду (Maxkilykl). Використати функцію random.

```
program ATS1;
uses Crt;
type vyklyk= array[1..10] of integer;
var y:vyklyk; max, s:integer;
procedure Kilyykl(var y:vyklyk);
var i:integer;
begin
    {Процедура Kilyykl визначає}
    {кількість викликів кожної секунди}
    for i:=1 to 10 do
        begin
            y[i]:=random(i);
            writeln('y(' , i , ')=' , y[i]:5);
        end;
    end;
procedure Sumavykl(y:vyklyk; var
s:integer);
var i:integer;
begin
    {Процедура обчислює суму викликів за}
    {перші 10 секунд}
    s:=0; for i:=1 to 10 do s:=s+y[i];
    writeln('Сума викликів S=' , s:3);
end;
procedure Maxkilykl(y:vyklyk; var
max:integer);
var i:integer;
begin
    {Процедура Maxkilykl визначає}
    {найбільшу кількість викликів}
    {за деяку секунду}
    max:=y[1];
    for i:=2 to 10 do
        if max<y[i] then max:=y[i];
    write('Максимальна кількість викликів
за одну ');
    writeln('секунду дорівнює ' , max:3)
end;
begin
    clrscr;
    randomize;
    Kilyykl(y);
    Sumavykl(y, s);
    Maxkilykl(y, max);
    readln
end.
```

### Функції (function).

Функція, на відміну від процедури, може повертати в місце виклику лише один результат простого стандартного типу.

Загальний опис функції:

```
function <назва>(<список формальних параметрів>) : <тип функції>;
<розділи описів і оголошень функції>;
```

**begin**

<розділ команд функції, де має бути така команда: назва:=вираз>

**end;**

У розділі команд функції має бути команда присвоєння значення деякого виразу назві функції. Результат функції повертається в основну програму через її назву (як і випадку використання стандартних функцій, таких як sin, cos). Виклик функції здійснюється лише з виразів

<назва> (<список фактичних параметрів>).

*Приклад.* Створимо функцію для обчислення  $\text{tg}(x)$  та обчислимо значення виразу  $\text{tg}(x)+\text{ctg}(x)+\text{tg}^2(x)$ .

```

program Myfunc;
uses Crt;
var x,y:real;
function tg(x:real):real;
begin
  tg:=sin(x)/cos(x)
end;
begin clrscr;
  writeln('Введіть x');
  readln(x);
  y:=tg(x)+1/tg(x)+sqr(tg(x));
  writeln('y=', y:5:2); readln
end.

```

**Задача 2.** Кондитерська фабрика для виготовлення п'яти сортів цукерок використовує п'ять видів сировини. Нехай норми затрат  $a_{ij}$  кожного виду сировини  $i$  на виробництво 1т цукерок сорту  $j$  задані формулою  $a_{ij} = 2|\text{Sin}(i)|+j$ ,  $i,j = \overline{1,5}$ . Вивести на екран таблицю затрат сировини. Визначити для якого сорту цукерок ( $\text{imin}$ ) потрібно найменше сировини ( $\text{min}$ ) третього виду.

```

program Fabryka1;
uses Crt;
const n=5;
type vytraty = array [1..n, 1..n] of real;
var imin:integer; a:vytraty;
function func(i,j:integer):real;
begin
  func:=2*abs(sin(i))+j;
end;
procedure Table(var a:vytraty);
var i,j:integer;
begin
  writeln(' Вид сировини');
  writeln(' 1 2 3 4 5');
  for i:=1 to n do {Утворимо таблицю затрат}
  begin
    write(i, ' сорт');
    for j:=1 to n do
    begin
      a[i,j]:=func(i,j); {Використаємо створену функцію}
      write( a[i,j]:7:2); {Роздрукуємо елементи i-го рядка}
    end;
    writeln {Перейдемо на новий рядок}
  end;
end;
procedure MinSyrov(a:vytraty; var imin:integer);

```

```
var i,j:integer; min:real;
begin
  imin:=1; min:=a[1,3];
  for i:=2 to n do
    if a[i,3]<min then
      begin
        min:=a[i,3]; imin:=i
      end;
  writeln('Найменше сировини третього виду ');
  writeln('необхідно для цукерок ', imin, ' сорту')
end;
begin
  clrscr;
  Table(a);
  MinSyrov(a, imin);
  readln
end.
```

*{Виклик процедури Table}*

*{Виклик процедури MinSyrov}*

## Самостійна робота № 6

**Тема:** Основи програмування мовою Pascal

**Мета:** вивчити поняття "стринг" і виробити навички роботи із символічною інформацією в мові програмування Pascal, навчитися використовувати рядка символів і безлічі при рішенні задач.

**Питання, що виносяться на самостійне вивчення:**

### 5.3 Рядки

#### Література:

- 1 Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. М., Наука, 1988.
- 2 Марченко А.И., Л.А. Марченко Программирование в среде Turbo PASCAL 7.0, Базовый курс. – Киев: «Век+», 2004
- 3 Меженный О.А. Самоучитель Turbo Pascal – М.: «Диалектика», 2004
- 4 Ставровский А. Турбо Паскаль 7.0 Учебник – Киев, «Ирина», 2000

**Питання для самоконтролю:**

- 1 Як описуються строкові змінні?
- 2 Яка максимальна довжина рядка припустима в Pascal?
- 3 Які операції припустимі над строковими даними?
- 4 У чому відмінність строкової змінної від масиву символів?
- 5 Які стандартні процедури і функції для роботи з рядками ви знаєте?
- 6 Що виведе функція  $\text{Cory}(x, \text{Pos}(' ', x) + 1, 18)$ , якщо  $x = \text{'Сила є - розуму не треба'}$ ?
- 7 Чому дорівнює значення  $x[0]$  після присвоювання  $x := \text{'питання'}$ ?

### 5.3 Рядки

**Рядки та дії з ними.** Дане типу рядок — це послідовність довільних символів (тобто елементів типу char). Сталі типу рядок записують за допомогою двох штрих-символів, які охоплюють текст. Рядок може містити від 0 до 255 символів. Наприклад, 'Україна', 'Львівська політехніка', " – порожній рядок нульової довжини, ' ' – рядок, що містить один символ-пропуск.

Змінну типу рядок оголошують за допомогою слова string так:

```
var <змінна>: string[n];
```

де  $n$  – довжина рядка,  $n < 256$ . Довжину рядка можна не зазначати.

*Приклад.*

```
const slovo='University';  
var fraza1: string[45]; fraza2: string;
```

Над змінними типу рядок визначені операції з'єднання (+) та порівняння (<, <=, >, >=, =, <>). Порівняння двох рядків здійснюється зліва направо до перших різних символів, причому 'A'<'B', 'B'<'C' тощо. “Більшим” вважається символ, який розташований в алфавіті далі. Числовий код символу дає функція ord, наприклад, ord('B')=66, ord('A')=65. Зворотню дію виконує функція chr: chr(66) дає 'B'.

*Приклад.* Нехай t1='New', t2=' Year'. Тоді з'єднанням цих рядків буде s:=t1+t2 (s матиме значення 'New Year'). Тут s > t1.

*Приклад.* Вивести на екран малі літери латинського алфавіту та їхні коди можна так:  
for v:= 'a' to 'z' do writeln(v, ord(v):5).

**Функції та процедури для дій з рядками.** Над даними типу рядок визначені такі стандартні функції:

**length**(<рядок>) — визначає кількість символів у рядку;

**copy**(r, m, n) — дає n символів рядка r, починаючи з символу з номером m;

**concat**(r1, r2, ..., rn) — з'єднує рядки r1, ..., rn в один рядок;

**pos**(r1, r2) — визначає номер символу, з якого починається входження рядка r1 у рядок r2.

та процедури:

**insert**(r1, <змінна>, n) — вставляє рядок r1 у рядок, заданий змінною, починаючи з позиції n;

**delete**(<змінна>, m, n) — вилучає n символів з рядка, заданого змінною, починаючи з позиції m;

**str**(<число>, <змінна>) — переводить числове дане в дане типу рядок;

де зазначена змінна посилає в процедуру вхідне дане типу рядок і отримує назад інший рядок – результат виконання процедури.

*Приклад.* Нехай змінна Lviv має значення 'Львівська політехніка'. Розглянемо приклади функцій та їхні значення:

<u>Функція</u>	<u>Значення</u>
length(Lviv)	21;
copy(Lviv, 15, 11)	техніка;
concat(Lviv, ' – 2000')	Львівська політехніка – 2000;
pos('т', Lviv)	15.

Наступні процедури нададуть змінній Lviv таких значень:

<u>Процедура</u>	<u>Значення змінної Lviv</u>
insert('НУ ', Lviv, 1)	'НУ Львівська політехніка'
delete(Lviv, 6, 16)	'Львів'
str(2000, Lviv)	'2000'
val('1256', Lviv1, Ozn)	Lviv1=1256, Ozn=0

Є два способи опрацювання даних типу string. Перший — можна опрацювати весь рядок як єдине ціле, застосовуючи до нього функції та процедури, другий — можна розглядати рядок як масив, складений з елементів-символів, і опрацювати його за правилами роботи з елементами масиву.

*Приклад.* Деякі значення змінним fraza1 та fraza2 з попереднього прикладу можна надати, а потім вивести тексти на екран так:

```
fraza1 := 'Ви любите канікули?';  
for i := 1 to 19 do read(fraza2[i]); {Вводимо текст з клавіатури}  
writeln(fraza1); writeln(fraza2).
```

*Задача 1.* Кодування інформації. Вилучити з фрази а пропуски, коми і крапки, інші символи продублювати. Вивести результат.

```
program MyCode;  
uses Crt;  
var a, b, c : string; i : integer;  
begin  
  clrscr;  
  write('Введіть текст: ');  
  readln(a);  
  b := ''; {Вводимо будь-яку фразу}  
  for i := 1 to Length(a) do  
    begin  
      c := Copy(a, i, 1); {Зверніть увагу на коментар унизу:}  
      if (c <> ',') and (c <> '.') and (c <> ' ') { або так: c := a[i]; }  
        then b := b + c + c  
    end;  
  writeln(b);  
  readln  
end.
```

*Задача 2.* Скласти програму, яка скрізь у заданому тексті mytext замінить деяке слово іншим словом такої ж довжини (word1 на word2).

```
program FindAndChange;  
uses Crt;  
var mytext, word1, word2 : string; i, k : integer;  
begin  
  clrscr;  
  write('Введіть текст: '); readln(mytext);  
  write('Введіть шукане слово: '); readln(word1);  
  write('Введіть інше слово: '); readln(word2);  
  k := length(word1);  
  for i := 1 to length(mytext) - k do  
    if copy(mytext, i, k) = word1 then  
      begin  
        delete(mytext, i, k);  
        insert(word2, mytext, i)  
      end;  
  writeln(mytext);  
  readln  
end.
```

*Задача 3.* Нехай задано рядок 'Я люблю інформатику'. Визначити довжину рядка. Вивести на екран друге слово цього рядка.

```
program Ryadok;  
uses Crt;  
const r1: string = 'Я люблю  
інформатику';
```

```

var i,k,m,n1,n2: integer;
begin
  clrscr;
  m:=0; {Визначаємо довжину рядка}
  k:=length(r1);
  writeln('Довжина рядка k=', k); {Переглядаємо всі символи рядка}
  for i:=1 to k do {Шукаємо пропуск}
    if r1[i]=' ' then
      begin
        m:=m+1; {Визначаємо номер першого та другого пропусків}
        if m=1 then n1:=i;
        if m=2 then n2:=i
      end;
    for i:=n1+1 to n2-1 do write(r1[i]); {Виводимо слово між двома пропусками}
  readln
end.

```

## Самостійна робота № 7

**Тема:** Основи програмування мовою Pascal

**Мета:** Познайти з поняттям файлового типу даних (типізовані, текстові і нетипізовані файли); виробити навички роботи з файловим типом даних у мові програмування Pascal. навчитися зчитувати інформацію з файлів, записувати інформацію у файл; навчитися вирішувати задачі з використанням файлів

**Питання, що виносяться на самостійне вивчення:**

5.4 Файли

### Література:

- 1 Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. М., Наука, 1988.
- 2 Марченко А.И., Л.А. Марченко Программирование в среде Turbo PASCAL 7.0, Базовый курс. – Киев: «Век+», 2004
- 3 Меженный О.А. Самоучитель Turbo Pascal – М.: «Диалектика», 2004
- 4 Ставровский А. Турбо Паскаль 7.0 Учебник – Киев, «Ирина», 2000

**Питання для самоконтролю:**

- 1 Що таке файл? Які існують види файлів?
- 2 Які стандартні процедури і функції містить Borland Pascal для роботи з файлами?
- 3 Яким повинен бути зміст програми по створенню файлу і його коректуванню (заміна елементів, додавання елементів, видалення елементів)?
- 4 Які особливості роботи з текстовими файлами?
- 5 Які особливості роботи з типізованими файлами?
- 6 Як підрахувати число рядків у текстовому файлі?

## 5.4 Файли

**Опис типів та оголошення типованих файлів.** Часто виникає потреба опрацювати інформацію, розташовану на зовнішніх носіях (на дисках). Прикладами таких даних є інформація про успішність студентів, розклад руху транспорту, наявність товарів у магазинах, адресні книжки.

**Файл** — це сукупність даних, які розташовані на зовнішньому носії. Дані у файлі називаються елементами. Кількість даних, на відміну від масиву, під час описування файлу не зазначають. Елемент файлу не має індекса. Тип елементів може бути як простим, так і складеним, але не може бути файлом.

Файловий тип даних описують у розділі описів типів так:

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

або безпосередньо у розділі оголошення змінних

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

*Приклад.*

```
type myfile = file of integer;  
list = file of string[20];  
var file1, file2: myfile; druzi: list;  
komanda: file of boolean;
```

**Дії з файлами.** Для того, щоб знайти потрібний елемент файлу, необхідно послідовно переглянути всі попередні. Це називається послідовним доступом до файлу.

Для опрацювання файлу його необхідно відкрити, виконати необхідні дії та закрити.

Для визначення кінця файлу є стандартна логічна функція

**eof** (<ім'я файлу>).

Значення цієї функції буде **true**, якщо досягнуто кінець файлу.

Файл можна відкрити лише або для читання, або для записування в нього інформації.

Для роботи з файлами є такі команди (процедури модуля System):

```
assign(<ім'я файлу>,<зовнішнє ім'я>) — налагоджує зв'язок між іменем файлу і  
файлом на зовнішньому носії;  
reset(<ім'я файлу>) — відкриває файл для читання з нього даних;  
read(<ім'я файлу>,<ім'я змінної>) — читає (вводить) дане з файлу в оперативну  
пам'ять;  
close(<ім'я файлу>) — закриває файл;  
rewrite(<ім'я файлу>) — відкриває файл для записування в нього даних;  
write(<ім'я файлу>,<ім'я змінної>) — записує (виводить) дане у файл.
```

Тут <ім'я файлу> — ім'я файлової змінної, задане у розділі оголошення змінних, <зовнішнє ім'я> — ім'я файлу даних на зовнішньому носії, взяте у лапки, наприклад, 'd:\grup1\lab1.pas'.

**Задача 1.** Створити файл записів, який містить інформацію про комп'ютери: марку, обсяг вінчестера (hdd), обсяг оперативної пам'яті (ram) та швидкодію (speed). Вивести вміст файлу на екран.

```
program Computer1;  
uses Crt;  
type comp=record {Опишемо тип запису, який містить}  
marka:string[15]; {характеристики комп'ютера}  
hdd, ram:real;  
speed:integer;  
end;  
myfile=file of comp;  
var f1:myfile; i,n:integer;  
c1:comp;  
begin  
clrscr;  
writeln('Уведіть кількість комп'ютерів');  
readln(n);  
assign(f1, 'd:\computer'); {Відкриємо f1 для записування в нього
```

```

rewrite(f1);
for i:=1 to n do
  begin
    writeln('Уведіть марку комп'ютера:');
readln(c1.marka);
    writeln('Уведіть обсяги його HDD та
RAM:');
    readln(c1.hdd, c1.ram);
    writeln('Введіть швидкодію:');
readln(c1.speed);
    write(f1, c1);
  end;
close(f1);
writeln(' Марка Обсяг вінчестера RAM
Швидкодія');
reset(f1);
for i:=1 to n do
  begin
    read(f1, c1);
    writeln(c1.marka:15, c1.hdd:10, c1.ram:7,
c1.speed:8);
  end;
readln
end.

```

*даних}*

*{Запишемо введені дані у файл f1}*

*{Закриємо файл f1}*

*{Відкриємо файл для зчитування даних}*

*{Виведемо зміст файлу на екран}*

*{Зчитуємо чергове дане з файлу}*

**Задача 2.** Використовуючи файл записів, утворений попередньою програмою, вивести на екран інформацію про комп'ютери, швидкодія яких більша, ніж 166 Мг

```

program Computer;
uses Crt;
type comp=record
  marka:string[15];
  hdd, ram:real;
  speed:integer;
end;
myfile=file of comp;
var f1:myfile; i,n:integer; c1:comp;
begin
  clrscr;
  assign(f1, 'd:\computer');
  reset(f1);
  while not eof(f1) do
    begin
      read(f1, c1);
      if c1.speed>166 then
        writeln(c1.marka:15, c1.hdd:8:2,
c1.ram:8:2);
    end;
  readln
end.

```

*{Опишемо тип, який визначає}*

*{характеристики комп'ютера}*

*{Відкриємо файл для зчитування даних}*

*{Виконується цикл, доки не досягнуто}*

*{кінець файлу}*

*{Зчитуємо дане з файлу}*

**Зауваження.** Окрім файлів послідовного доступу можна створювати і опрацьовувати файли прямого доступу. Відмінність така: перед використанням команд read чи write слід забезпечити доступ до k-го елемента (нумерація від нуля) файлу за допомогою команди

```
seek(<назва файлу>,k);
```

## Види файлів

**Текстові файли.** Дані у типованих файлах, описаних вище, певним чином кодуються комп'ютером. Ці файли не можна редагувати або переглядати за допомогою звичайного текстового редактора. Тому, окрім типованих, використовують **текстові файли**, які не мають такого недоліку.

Елементами текстових файлів є рядки (послідовності символів: букв, цифр, знаків та пропусків). Такий файл можна створювати та редагувати за допомогою текстового редактора. Розмежовувачем між елементами файлу є *пропуск*. Уведення кожного рядка закінчується натисканням на клавішу вводу. Для перевірки чи є символи у рядку використовують функцію

```
eoln(<ім'я файлу>),
```

яка приймає значення **true**, якщо знайдено кінець рядка.

Текстові файли описують у розділі опису змінних так:

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

Дані з рядка текстового файлу можна зчитати за допомогою команд

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

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

дія яких була описана вище. Різниця між цими командами така: під час виконання команди *readln* зайві дані у рядку ігноруються і наступна команда *read* чи *readln* буде зчитувати дані з наступного рядка.

Рядок текстового файлу можна створити за допомогою звичайного текстового редактора або програмним способом за допомогою команд

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

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

На відміну від звичайних файлів у текстовий файл можна додавати (дописувати) дані. Для цього замість процедури *gewrite* використовують процедуру

```
append(<ім'я файлу>);
```

**Задача 3.** Нехай деяким текстовим редактором створено файл *riku.txt*, який містить інформацію про довжини річок у кілометрах та їхні басейни у квадратних кілометрах:

Ніл	6671 2870000
Міссісіпі	6420 3238000
Амазонка	6280 6915000
Об	5410 2990000
Амур	4416 1855000
Лена	4400 2490000
Конго	4320 3690000
Нігер	4160 2092000
Волга	3531 1360000

Виведіть усі дані з файлу на екран, визначте назви найдовшої річки та річки з найбільшим басейном.

```
program Textfile;
```

```
uses Crt;
```

```
type Rika=record
```

```
  name:string[10];
```

```
  dov:integer;
```

```
  pl:longint
```

```
end;
```

```
var myfile:text; myzap:rika;
```

```
  namemax, nameplmax:string[11];
```

```
  dovmax:integer; plmax:longint;
```

```
begin
```

```
  clrscr;
```

```
  dovmax:=0; plmax:=0;
```

```
  writeln('Назва ріки Довжина(км) Басейн (кв.км)');
```

```

assign(myfile,'d:\riky.txt');
reset(myfile);
while not eof(myfile) do
  with myzap do
    begin
      readln(myfile, name, dov, pl);
      writeln(name:10, dov:10, pl:15);
      if dov>dovmax then
        begin
          dovmax:=dov; namemax:=name
        end;
      if plmax<pl then
        begin
          plmax:=pl; nameplmax:=name
        end;
      end;
      writeln('Найдовшою річкою є ', namemax);
      writeln('Найбільший басейн має річка ', nameplmax);
      readln
    end.

```

Компонентний або типізований файл - це файл з оголошеним типом його компонентів. Компонентні файли складаються з машинних представлень значень змінних, вони зберігають дані в тому ж вигляді, що і пам'ять ЕОМ.

Компонентами файлу можуть бути всі скалярні типи, а зі структурованих - масиви, безлічі, записи. Практично у всіх конкретних реалізаціях мови ПАСКАЛЬ конструкція "файл файлів" неприпустима.

Всі операції над компонентними файлами виробляються за допомогою стандартних процедур:

Reset, Rewrite, Read, Write, Close.

Для введення - виведення використовуються процедури:

```

Read(f,X);
Write(f,X);

```

де f - ім'я логічного файлу, X - або змінна, або масив, або рядок, або безліч, або запис з таким же описом, яке має компонента файлу.

Безтипові файли дозволяють записувати на диск довільні ділянки пам'яті ЕОМ і зчитувати їх з диска в пам'ять. Операції обміну з безтиповими файлами здійснюється за допомогою процедур BlokRead і BlockWrite. Крім того, вводиться розширена форма процедур Reset і Rewrite. В іншому принципі роботи залишаються такими ж, як і з компонентними файлами.

## Самостійна робота № 8

**Тема:** Основи програмування мовою Pascal

**Мета:** Вивчити режими, кольори і заповнення, координати і вікна, текст у графічному режимі, вивід рухливого зображення.

**Питання, що виносяться на самостійне вивчення:**

5.5 TURBO PASCAL і комп'ютерна графіка

### Література:

- 1 Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. М., Наука, 1988.
- 2 Марченко А.И., Л.А. Марченко Программирование в среде Turbo PASCAL 7.0, Базовый курс. – Киев: «Век+», 2004
- 3 Меженный О.А. Самоучитель Turbo Pascal – М.: «Диалектика», 2004
- 4 Ставровский А. Турбо Паскаль 7.0 Учебник – Киев, «Ирина», 2000

**Питання для самоконтролю:**

- 1 Які ви знаєте процедури і функції графічного режиму Модуля Graph Турбо Паскаля
- 2 Як задати координати графічних об'єктів?
- 3 Призначення і порядок запису процедур PutPixel та SetLineStyle.
- 4 Як заповнити області, прямокутник, еліпс?

## 5.5 TURBO PASCAL і комп'ютерна графіка

### Режими. Кольори і заповнення

Модуль Graph Турбо Паскаля містить біля п'ятдесятьох різних процедур і функцій, призначених для роботи з графічним екраном. У цьому ж модулі деякі вбудовані константи, що можуть бути використані в графічних програмах. Для того щоб скористатися всіма можливостями модуля Graph, на початку програми (після заголовка) необхідно помістити оператор використання

```
Uses Graph;
```

Основну частину модуля складають процедури введення базових графічних елементів, таких як крапки, відрізки прямих ліній, дуги і цілі окружності й ін.

Функція GraphErrorMsg. Повертає значення типу String, у якому по зазначеному коду помилки дається відповідне текстове повідомлення. Заголовок:

```
Functional GraphErrorMsg(Code: Integer): String;
```

Тут Code - код помилки, що повертається функцією GraphResult.

Функція CloseGraph. Завершує роботу адаптера в графічному режимі і відновлює текстовий режим роботи екрана.

Процедура SetColor. Установлює поточний колір для ліній, що вводяться, і символів. Заголовок:

```
Procedure SetColor(Color: Word);
```

Тут Color - поточний колір.

Процедура SetBkColor. Установлює колір фону. Заголовок:

```
Procedure SetBkColor (Color: Word);
```

Тут Color - колір тла.

На відміну від текстового режиму, у якому колір фону може бути тільки темного відтінку, у графічному режимі він може бути будь-яким. Установка нового кольору фону, негайно змінює колір графічного екрана. Це означає, що не можна створити зображення, дві ділянки якого мали б різний колір фону. Для GGA-адаптера в режимі високого дозволу установка кольору фону змінює колір активних пікселів. Зверну увагу, що після заміни кольору фону на будь-який, відмінний від 0 (Black) колір. Ви не зможете більш використовувати колір 0 як чорний, він буде замінитися на колір фону, тому що процедури модуля Graph інтерпретують колір з номером 0 як колір фону. Це означає, зокрема, що Ви вже не зможете повернути фону чорний колір!

Якщо Ваш ПК оснащений кольоровим екраном, наступна програма продемонструє роботу процедури SetBkColor. Програма виводить десять вкладених одне в одне прямокутників, після чого циклічно змінює колір фону. Для виходу з програми досить натиснути на будь-яку клавішу.

Процедура SetFillStyle. Установлює стиль (тип і колір). Заголовок:

```
Procedure SetFillStile(Fill, Color: Word);
```

Тут Fill - тип заповнення; Color - колір заповнення.

За допомогою заповнення можна покривати які-небудь фрагменти зображення періодично повторюваним візерунком.

Процедура FloodFill. Заповнює довільну замкнуту фігуру, використовуючи поточний стиль заповнення (візерунок і колір). Заголовок:

```
Procedure FloodFill(X, Y: Integer; Border: Word);
```

Тут X, Y - координати будь-якої крапки усередині замкнутої фігури; Border - колір граничної лінії. Якщо фігура не замкнута, заповнення «розіллється» по всьому екрані.

### Координати і вікна. Крапки і лінії

Будь-яке зображення формується з досить простих геометричних фігур. Це крапки, відрізки прямих, окружності і т.д. З геометрії відомо, що положення геометричного об'єкта і його форма задаються координатами його крапок.

Отже, для того щоб запрограмувати графічний висновок, треба навчитися задавати координати графічних об'єктів.

Багато графічних процедур і функції використовують покажчик поточної позиції на екрані, що на відміну від текстового курсору невидим. Положення цього покажчика, як і взагалі будь-яка координата на графічному екрані, задається щодо лівого верхнього кута,

що, у свою чергу, має координати 0,0. Таким чином, горизонтальна координата екрана збільшується з ліворуч на праворуч, а вертикальна - зверху до низу.

Функції GetMax і GetMax.

Повертають значення типу Word, що містять максимальні координати екрана в поточному режимі роботи відповідно по горизонталі і вертикалі.

Процедура SetViewPort. Установлює прямокутне вікно на графічному екрані.  
Заголовок:

Procedure SetViewPort(X1, Y1, X2, Y2: Integer; ClipOn: Boolean);

Тут X1..Y2 - координати лівого верхнього(X1,Y1) і правого нижнього(X2,Y2) кутів вікна; ClipOn - вираження типу Boolean, що визначає "відсічення" не уміщаються у вікні елементів зображення.

Процедура ClearDevice. Очищає графічний екран. Після звертання до процедури покажчик встановлюється в лівий верхній кут, а сам екран заповнюється кольором фону, заданим процедурою SetBkColor.

### ***Крапки і лінії***

Процедура PutPixel. Виводить заданим кольором крапку по зазначених координатах.  
Заголовок:

Procedure PutPixel(X, Y: Integer; Color: word);

Тут X, Y - координати крапки; Color - колір крапки.

Процедура Line. Вичерчує лінію з зазначеними координатами початку і кінця.  
Заголовок:

Procedure Line(X1, Y1, X2, Y2: Integer);

Тут X1..Y1 - координати початку(X1,Y1) і кінця(X2,Y2) лінії. Лінія викреслюється поточним стилем і поточним кольором.

Процедура SetLineStyle. Установлює новий стиль ліній, що викреслюється.  
Заголовок:

Procedure SetLineStyle(Type, Pattern, Thick: Word);

Тут Type, Pattern, Thick - відповідно тип, зразок і товщина лінії. Тип лінії може бути створений за допомогою однієї з наступних констант:

Const

SolidLn= 0; (Суцільна лінія)

DottedLn= 1; (Крапкова лінія)

CenterLn= 2; (Штрих-пунктирна лінія)

DashedLn= 3; (Пунктирна лінія)

UserBitLn= 4; (Візерунок лінії обумовлений користувачем)

### ***Незаповнені фігури***

Процедура Rectangle. Вичерчує прямокутник із зазначеними координатами кутів.  
Заголовок:

Procedure Rectangle(X1, Y1, X2, Y2: Integer);

Тут X1..Y2 - координати лівого верхнього кута(X1,Y1) і правого нижнього(X2,Y2) кутів прямокутника. Прямокутник викреслюється з використанням поточного кольору і поточного стилю ліній.

### ***Дуги, окружності, еліпси***

Процедура Circle. Вичерчує окружність. Заголовок:

Procedure Circle(X, Y: Integer; R: Word);

Тут X,Y - координати центра; R - радіус у пікселях.

Окружність виводиться поточним кольором. Товщина лінії встановлюється поточним стилем, вид лінії завжди SolidLn(Суцільна). Процедура вичерчує правильну окружність з урахуванням зміни лінійного розміру радіуса в залежності від його напрямку щодо сторін графічного екрана, тобто з урахуванням коефіцієнта GetAspectRatio. У зв'язку з цим параметр R визначає кількість пікселів у горизонтальному напрямку.

Процедура Arc. Креслить дугу окружності. Заголовок:

Procedure Arc(X, Y: Integer; Beg, End, R: Word);

Тут X, Y - координати центра; Beg, End - відповідно початковий і кінцевий кути дуги; R - радіус.

Кути описуються проти годинникової стрілки і вказуються в градусах. Нульовий кут відповідає горизонтальному напрямкові вектора з ліва на право. Якщо задати значення початкового кута 0 і кінцевого - 359, то буде виведена повна окружність. При кресленні дуги окружності використовуються ті ж угоди щодо лінії радіуса, що й у процедурі Circle.

Процедура Ellipse. Викреслює еліпсну дугу. Заголовок:

Procedure Ellipse(X, Y: Integer; Beg, End, Rx, RY: Word);

Тут X,Y - координати центра; Beg, End - відповідно початковий і кінцевий кути дуги; RX, RY - горизонтальний і вертикальний радіуси еліпса в пікселях.

При кресленні дуги еліпса використовуються ті ж угоди щодо лінії, що й у процедурі Circle, й ті ж угоди щодо кутів, що й у процедурі Arc. Якщо радіуси погодити з урахуванням масштабного коефіцієнта GetAspectRatio, буде накреслена правильна окружність.

### **Заповнення фігур**

Крім незаповнених, засоби модуля GRAPH дозволяють виводити на екран також заповнені фігури. Мова йде про такі фігури, як прямокутники, паралелепіпеди, багатокутники, еліпси і сектори.

#### **Заповнення області**

Заповнити (суцільним кольором або візерунком) деяку замкнуту фігуру вам допоможе процедура FloodFill.

Процедура FloodFill заповнює замкнуту область, використовуючи поточний стиль заповнення (шаблон і колір). Заголовок:

procedure FloodFill (x,y: Integer;Border:Word);

де x и y - координати довільної крапки в області, що потрібно заповнити;

Border - код кольору лінії, що обмежує заповнювану область.

Потрібний стиль заповнення повинний бути попередньо заданий процедурою SetFillStyle або SetFillPattern.

Якщо крапка (x, y) знаходиться усередині замкнутої області, ця область буде заповнена. Якщо крапка знаходиться зовні замкнутої області, заповнюється весь простір поза областю. Якщо ж область виявиться незамкнута, заповненням виявиться весь екран.

#### **Заповнений прямокутник**

Процедура Bar дозволяє відображати на екрані заповнені (але не обведені) прямокутники. Подібні прямокутники можуть знадобитися при створенні (на екрані) різного роду стовпчастих діаграм.

Процедура Bar відображає необведений прямокутник, заповнений з використанням поточного шаблону і кольору. Заголовок:

procedure Bar(x1,y1, x2, y2: Integer);

де {x1, y1} і {x2, y2} - координати відповідно верхніх лівих і нижнього правого кутів прямокутника.

Для заповнення використовуються шаблон і колір, визначені за допомогою процедур SetFillStyle і SetFillPattern.

#### **Заповнений еліпс**

Для відображення заповненого еліпса в Turbo Pascal мається процедура FillEllipse, подібна до процедури Ellipse, з яким ми познайомилися вище.

Процедура FillEllipse відображає заповнений еліпс. Заголовок:

procedure FillEllipse{x, y: Integer; XRad, YRad : Word)

де (x, y) - координати центра еліпса; XRad і YRad - відповідно горизонтальний і вертикальний радіуси.

**Задача 1.** Нарисувати різними кольорами десять концентричних кіл, які мають спільний центр по середині екрана, тобто в точці з графічними координатами (320; 240), і описати навколо кіл червоний прямокутник.

```
program Circle10;
```

```
uses Crt, Graph;
```

```

var driver, mode, r : integer;
begin clrscr;
  driver := detect;
  initgraph(driver, mode, "");
  r := 10;  {Радіус першого кола 10 пікселів}
  while r <= 100 do
    begin
      setcolor(r div 10);
      circle(320, 240, r);
      r := r + 10
    end;
  setcolor(red);
  rectangle(220, 140, 420,340); readln
end.

```

**Задача 2.** Нарисувати емблему. У верхній лівій частині графічного екрана на чорному фоні нарисувати блакитний квадрат, а в ньому – чорне коло, зафарбоване жовтим кольором. У центрі емблеми чорними літерами написати слово “Львів”.



```

program Emblema;
uses Crt, Graph;
var driver, mode, i, x1 , y1: integer; x, y: real;
begin clrscr;
  driver:=detect;
  initgraph(driver,mode,"");
  setbkcolor(0);
  setcolor(3);
  rectangle(100,0,300,200);
  setfillstyle(1,3);
  floodfill(200,100,3);
  setcolor(14);
  circle(200,100,100);
  setfillstyle(1,14);
  floodfill(200,100,14);
  setcolor(0);
  circle(200,100,100);
  settextstyle(0,0,3); outtextxy(135,95, 'Львів'); readln
end.

```

**Задача 3.** Нарисувати графік функції  $y=2\sin(2x)+1$  на проміжку  $[0; 2\pi]$ , табулюючи функцію з кроком  $h=0,1$ . У результаті експериментів, міняючи значення амплітуди в пікселях (за допомогою множника M), зобразити графік на екрані якнайкраще.

```

program GrafikSin;
uses Crt, Graph;
const a=0; b=2*pi; h=0.1; h1=5; x0=60; y0=240; M=50;
var driver, mode,x1,y1:integer; x,y:real;
function f(x:real):real;
begin
  f:=2*sin(2*x)+1
end;

```



**begin**

```

clrscr; driver := detect; initgraph(driver,mode,"");
setcolor(14); setbkcolor(1);
line(20,y0,600,y0); line (x0,440,x0,20);    {Координатні осі}
x:=a; x1:= x0+trunc(M*x);    {x,y – математичні координати}
y:=f(x); y1:=y0-trunc(M*y);    {x1,y1 – графічні координати}
moveto(x1,y1);setcolor(15);
while x<=b do
  begin
    y:= f(x); y1:=y0-trunc(M*y); lineto(x1,y1);
    x1:=x1+h1;x:=x+h
  end;
settextstyle(0,0,1);
outtextxy(60,245,'0');outtextxy(360,245,'6.3');
settextstyle(0,0,2);
outtextxy(200,380,'Графік функції y=2sin2x+1');readln

```

**end.**

## Самостійна робота №9

**Тема:** Текстовий редактор Word для Windows

**Мета:** Ознайомитись із стандартними елементами вікна програми текстового процесора Microsoft Word, навчитися основним прийомам введення та редагування тексту

**Питання, що виносяться на самостійне вивчення:**

6.2 Вікна документів. Введення і редагування тексту

### Література:

1 Вережка П. Word 97 для Windows для “чайників”. Учебный курс. 2-е издание.: Пер. с англ. – К.: Диалектика, 1997. – 272 с.

2 Глинський Я.М. Практикум з інформатики: Навч. посібник. – Львів: „Підприємство Деол”, 1998.- 168 с.

3 Інформатика / за ред. О.І.Пушкаря – К.: Видавничий центр „Академія”, 2001. – с.696

4 Крепкий Ю.О. Методичні вказівки для вивчення текстового процесора Microsoft Word 2000 – Чернігів, 2001. – 116 с.

**Питання для самоконтролю:**

- 1 Як створити новий текстовий документ?
- 2 Які є панелі інструментів?
- 3 Які кнопки є на панелі інструментів форматування?
- 4 Як виділити фрагмент тексту курсором?
- 5 Як запустити програму Word?
- 6 Як увімкнути панель форматування?
- 7 Як почати новий абзац тексту?
- 8 Як розділити один абзац на два?

## 6.2 Вікна документів

Вікно редактора Word має декілька стандартних елементів. Одні з них постійно присутні на екрані, інші можна викликати за бажанням користувача. Розглянемо призначення цих елементів.

**Рядок заголовка.** Верхній рядок екрана є рядком заголовка, стандартного для Windows. В ньому виведено ім'я програми (в даному випадку Microsoft Word). Крім цього, в рядку заголовка є чотири кнопки: одна з лівого краю і три — з правого. Ліва кнопка — це кнопка виклику управляючого меню. Управляюче меню є типовим для будь-якого вікна Windows. Перша з правих кнопок згортає вікно до піктограми, друга — відновлює нормальний розмір вікна, третя — закриває вікно.

**Рядок меню.** Під рядком заголовка у вікні розміщується рядок меню, який містить такі пункти:

**Файл** — робота з файлами документів;

**Правка** — редагування документів;

**Вид** — перегляд документів;

**Вставка** — вставка в документ малюнків, діаграм, поточної дати і часу, формул та інших об'єктів;

**Формат** — форматування документів (встановлення шрифтів, параметрів абзацу);

**Сервіс** — сервісні функції (перевірка орфографії, встановлення параметрів настроювання Word);

**Таблиця** — робота з таблицями;

**Окно** — робота з вікнами документів;

**?** — довідкова інформація про Word.

Кожний пункт меню має відповідне підменю. Для відкриття меню слід натиснути клавішу [Alt] або [F10]. Після цього один з пунктів меню виділиться інверсним кольором. Для виділення потрібного пункту меню слід користуватись клавішами горизонтального переміщення курсора. Для відкриття виділеного пункту меню слід натиснути клавішу [Enter]. Відкрити меню зручніше за допомогою миші, встановивши курсор на потрібному пункті меню і натиснувши ліву кнопку.

В підменю потрібний пункт може бути вибрано або за допомогою миші (встановити курсор миші на потрібний пункт і натиснути ліву кнопку), або за допомогою клавіатури (клавішами вертикального переміщення курсора вибрати потрібний пункт і натиснути клавішу [Enter]).

В назві пунктів меню і підменю є підкреслена літера. Це дає можливість одразу вибрати пункт меню або підменю, натиснувши комбінацію клавіш [Alt — підкреслена літера меню — підкреслена літера підменю].

Деякі пункти підменю праворуч від назви пункту містять у собі позначення комбінації клавіш, за допомогою яких можна вибрати відповідний пункт підменю.

При виборі пункту підменю в нижньому рядку екрана роз'яснюється його призначення.

Слід зазначити, що назви деяких пунктів підменю мають сірий колір. Це означає, що такі пункти в даний момент недоступні (наприклад, не можна редагувати таблицю, якщо вона не існує).

В Word існує ще один спосіб виклику команд. Клацання правою кнопкою миші на виділеному тексті, слові призводить до виведення на екран контекстного меню. Це меню містить команди, які можна застосувати до виділеного об'єкту.

Користувач має змогу відмітити останню введену команду, виконавши команду **Правка/Отменить**.

**Панелі інструментів.** Під рядком меню розміщуються звичайно панелі інструментів. Панелі інструментів — це рядок кнопок, при натискуванні на які виконується певна дія. Для натискування кнопки слід клацнути мишою по кнопці. При фіксації курсора миші на кнопці під нею з'являється її назва, а в рядку стану — коротка довідка про призначення кнопки. Ряд кнопок дублюють відповідні команди меню. Однак користуватись кнопками панелі значно швидше і зручніше. Word забезпечує користувача декількома панелями інструментів.

Для вибору потрібної панелі слід скористатися командою **Вид/Панели**

інструментов. При цьому на екрані з'явиться вікно діалогу **Панели інструментов**, у списку якого можна вибрати необхідні панелі. По замовчуванню Word виводить на екран панелі інструментів **Стандартная і Форматирование**. Деякі панелі інструментів виводяться на екран автоматично при виконанні певних дій (так, наприклад, панель інструментів Рисование виводиться при побудові малюнків). Виведені на екран панелі можна перемістити типовим для середовища Windows-95 способом.

Вікно діалогу. Для виконання деяких команд потрібно вводити допоміжну інформацію. Так, для виконання команди **Файл/Открыть** необхідно вказати дисковод, каталог та ім'я файла. Для введення такої інформації використовуються вікна діалогу.

**Вікно діалогу** містить ряд елементів: кнопки, списки, прапорці, перемикачі, рядки введення. Ці елементи розміщуються за тематичними групами, які називають полями. Групи мають заголовки, що закінчуються двокрапкою. Перехід від групи до групи здійснюється або за допомогою миші, або при натискуванні клавіші [TAB]. Ім'я групи можна також виділити, натиснувши комбінацію клавіш [Alt — підкреслена літера в імені поля]. Переміщення всередині групи здійснюють за допомогою клавіш переміщення курсора.

В разі введення допоміжної інформації у вікно діалогу здійснюється встановлення прапорців і перемикачів, вибір елементів із списку, введення і редагування тексту в полях введення.

Прапорці являють собою невеличкі квадрати, в яких в разі їх ввімкнення з'являється галочка. Прапорці встановлюються незалежно один від одного.

Перемикачі (зображуються у вигляді кола) використовують тоді, коли необхідно вибрати одну з декількох опцій. Вибраний перемикач відрізняється від інших темною крапкою всередині кола.

В рядки введення вводиться текстова інформація. Наприклад, при відкритті якого-небудь файла необхідно вказати його ім'я. Останнє можна вибрати із списку імен файла або набрати в рядку введення. Рядок введення редагувати.

Списки використовують для вибору одного з декількох варіантів (наприклад, вибір шрифту). Поки маркер знаходиться всередині цього списку, його можна гортати за допомогою клавіш переміщення курсора. Елемент списку виділяється натискуванням лівої кнопки миші (клавіші [Enter]). Особливою формою списку є однорядкові списки, в яких показано тільки перший елемент. Такі списки мають праворуч стрілку, направлену вниз. Для розкриття такого списку слід встановити курсор миші на стрілку і натиснути ліву клавішу миші (натиснути комбінацію клавіш [Alt — стрілка керування курсором]). Після розкриття списку і вибору елемента список знову закривається. Деякі однорядкові списки, елементом яких є число, мають справа дві стрілки, направлені вгору та вниз. При клацанні мишею на стрілці, направлений вниз, значення елемента зменшується, а по стрілці, направлений вгору, — збільшується.

В правій частині або внизу вікна розміщені кнопки управління діалогом. Кнопка **ОК** (клавіша [Enter]) закінчує діалог з підтвердженням усіх змін, після цього Word виконує команду.

Кнопка **ОТМЕНА** (клавіша [Esc]) анулює всі зміни; діалог закінчується, але відповідна команда не виконується.

Крім кнопки **ОК** і **ОТМЕНА** в цьому вікні залежно від призначення конкретного вікна можуть бути й інші кнопки управління діалогом.

Багато вікон діалогу мають таку кількість полів, що їх не можна вивести одночасно. В цьому випадку діалог організовується за сторінками-вкладниками. Кожна вкладника має у верхній частині вікна ім'я. Для відкриття вкладники слід встановити курсор миші на імені і натиснути ліву кнопку миші.

Вікна діалогу також мають поля, назви яких закінчуються трикрапкою. При вибиранні такого поля розкривається наступне вікно, в якому також можна встановити певні параметри.

### **Вікна документів**

Word є багатовіконним редактором. Всередині вікна редактора може існувати декілька вікон документів. Користувач може встановлювати розмір і положення кожного

вікна традиційними для Windows засобами. В кожному вікні може редагуватись свій текст.

У верхньому рядку розміщується заголовок вікна, який включає ім'я файлу, що редагується, кнопку виклику управляючого меню, кнопку згортання меню до піктограми, кнопку відновлення нормального розміру і кнопку закриття. Якщо у вікно не завантажено файл, то в заголовку вказується ім'я Документ. Праворуч і знизу розміщені смуги вертикальної і горизонтальної прокрутки. Смуги мають маркери, які показують, в якому місці документа знаходиться в даний момент користувач. Перемістивши маркер на потрібну позицію полоси прокрутки, можна перейти в будь-яке місце документа.

Під рядком заголовка розміщується горизонтальна лінійка, на якій розміщені маркери відступу рядків, абзаців, позицій табуляції. Детальніше призначення елементів горизонтальної лінійки буде розглянуто далі. Горизонтальну лінійку можна вилучити з екрана за допомогою команди **Вид/Лінійка**, а потім за допомогою цієї самої команди повернути на екран.

Одне з вікон є активним. Активне вікно зображується на передньому плані і може закривати інші вікна. В активному вікні знаходиться текстовий курсор (мерехтливий вертикальний штрих) і горизонтальна риска. Текстовий курсор казує місце, куди можна вводити символи. Горизонтальна риска визначає кінець тексту.

В нижньому рядку вікна редактора Word виводиться рядок стану. Він містить інформацію щодо активного вікна:

**Стр 1** — курсор знаходиться на і-й сторінці;

**Разд в** — курсор знаходиться у в-му розділі;

**к/р** — від початку документа до курсора — к сторінок, весь документ містить р сторінок;

**на 5 см** — відстань від курсора до верхньої межі сторінки становить 5 сантиметрів;

**Ст t** — курсор знаходиться в t-рядку поточної сторінки;

**Кол W** — курсор знаходиться в W-й колонці;

Робота з вікнами здійснюється за допомогою меню Окна. Команди цього Меню дозволяють відкрити нове вікно, міняти розташування вікон, зробити активним будь-яке вікно.

### **Введення і редагування тексту**

Уводити і редагувати можна тільки текст активного вікна. **Уведення символів.** Перед введенням символів слід вибрати шрифт, його розмір, формат. Основні можливості форматування будуть розглянуті далі. Символи клавіатури вводяться в позицію текстового курсора (мерехтливий вертикальний штрих). Пересунути текстовий курсор можна за допомогою клавіш керування курсором або за допомогою миші (покажчик миші перевести в потрібну позицію і натиснути ліву кнопку). Символи можуть вводиться у режимі заміни або вставки. У першому випадку введений символ заміщає той, на якому знаходився курсор. У режимі вставки частина рядка, що розміщена праворуч від курсора, зсувається на одну позицію, і символ вводиться на звільнене місце. Переключення між режимами здійснюється клавішею Ins. У режимі заміни індикатор **ЗАМ** рядка стану має чорний колір, у режимі вставки — сірий. Після введення символу курсор переміщається на одну позицію праворуч. Для вилучення символу ліворуч від курсора слід тиснути клавішу Backspace, а символу в позиції праворуч від курсора — клавішу Del. При цьому курсор переміщується на одну позицію ліворуч. Коли курсор стоїть у кінці рядка, то слово, яке не вміщується в даному рядку, переноситься в наступний. У редакторі Word є можливість переносити слова командою меню **Сервіс/Язык/Расстановка переносов**. На екрані з'явиться вікно діалогу. В цьому вікні можна встановити дві опції: автоматичне перенесення слів у документі; переносити слова з великих букв. Звичайно Word розділяє для перенесення тільки слова з малих букв або слова, які починаються з великої букви. Встановлення останньої опції дає змогу переносити слова, написані великими буквами. Якщо в будь-якій позиції рядка натиснути клавішу Enter, то редактор переходить на новий рядок з абзацу. Для переходу в наступний рядок без створення абзацу слід натиснути комбінацію клавіш Shift + Enter. Службові символи (закінчення рядка, закінчення абзацу та ін.), як правило, на екран не виводяться. Користувач може включати виведення цих символів відповідною кнопкою панелі інструментів **Стандартная**. Поруч з автоматичною

версткою рядків існує й автоматична верстка сторінок. Як тільки рядки тексту не помістяться на одній сторінці, вони автоматично перемістяться на наступну. На екрані між сторінками буде видно штрихову розподільну лінію. Автоматична верстка сторінок функціонує тільки тоді, коли у вікні діалогу **Параметри** вкладинка **Общие** встановлено прапорець **Фоновая разбивка на страницы**. Користувач може примусово ввести роздільник сторінок. Для цього слід підвести курсор до рядка, з якого має починатися наступна сторінка, і натиснути комбінацію клавіш **Ctrl + Enter**. На екрані з'явиться новий розділ сторінок — штрихова лінія з написом **Разрыв страницы**. Ці розподільні лінії легко вилучити як звичайні рядки тексту. **Переміщення за текстом.** Переміщуватися за текстом можна за допомогою миші, використовуючи смуги прокрутки. Для переміщення є можливість використовувати і такі клавіші: **PgUp**, **PgDn** — на розмір вікна вгору, вниз; **Home**, **End** — на початок, закінчення рядка; **Ctrl + Home**, **Ctrl + End** — на початок, закінчення тексту.

**Виділення тексту.** Виділення тексту є однією з важливих операцій, оскільки редагування можна виконувати тільки з виділеним фрагментом тексту. Для виділення тексту мишею слід встановити курсор миші на початок фрагмента, натиснути на ліву кнопку і, не відпускаючи її, перемістити курсор до кінця фрагмента. При цьому колір виділеного фрагмента інвертуватиметься. Подвійне натискування лівої кнопки миші виділяє слово, на якому встановлено курсор миші. Натискування лівої кнопки миші, коли курсор знаходиться біля лівої межі рядка, виділяє рядок, а подвійне натискування — весь абзац. Весь текст можна виділити, виконавши команду меню **Правка/Выделить все**. Фрагмент залишається виділеним, поки не буде виділений інший фрагмент. Для зняття виділення слід встановити курсор миші в будь-яке місце поза виділеним фрагментом і натиснути ліву кнопку миші. Фрагмент можна виділити і за допомогою комбінацій клавіш: **Shift + →**, **Shift + ←** — виділення символу праворуч, ліворуч від курсора; **Shift + Home**, **Shift + End** — виділення тексту від курсора до початку, кінця рядка; **Shift + PgUp**, **Shift + PgDn** — виділення фрагмента від курсора до початку, кінця тексту.

**Редагування виділеного фрагмента.** Виділений фрагмент можна вилучити, перемістити, скопіювати. За таких операцій часто використовують буфер обміну **Word**. Через цей буфер редактор **Word** може обмінюватись інформацією з іншими програмами, що працюють у середовищі **Windows**. Вилучити виділений фрагмент можна за допомогою команди **Правка/Вырезать** або кнопки **Вырезать** панелі інструментів **Стандартная**. Фрагмент вилучається з тексту і поміщається в буфер обміну. Текст з буфера обміну можна багаторазово читати. Цей текст зберігається в буфері доти, доки в нього не буде занесено новий фрагмент (це може зробити й будь-яка інша програма, що працює в середовищі **Windows**). Вилучити виділений фрагмент можна і за допомогою клавіші **Del**, але при цьому фрагмент у буфер обміну не заноситься. Вставка фрагмента з буфера обміну здійснюється командою **Правка/Вставить** або кнопки **Вставить** панелі інструментів **Стандартная**. Фрагмент вставляється в позицію текстового курсора. Перенести фрагмент можна за допомогою послідовно виконаних двох команд: **Правка/Вырезать** і **Правка/Вставить**. Досить просто можна перемістити фрагмент за допомогою миші. Для цього слід встановити курсор миші на виділеному фрагменті і, не відпускаючи натиснутої лівої кнопки, перетягнути фрагмент на нове місце. Якщо тепер відпустити ліву кнопку, то фрагмент буде переміщено. Скопіювати фрагмент можна за допомогою послідовно виконаних двох команд: **Правка/Копировать** і **Правка/Вставить**. Під час виконання першої команди виділений фрагмент переноситься в буфер обміну, але з тексту не вилучається. Копіювання за допомогою миші аналогічне переміщенню, але при цьому повинна бути додатково натиснута клавіша **Ctrl**. Команди редагування діють і під час роботи з документами в різних вікнах. Це дає змогу обмінюватися фрагментами тексту між різними документами. **Поля.** Поля — це спеціальні фрагменти тексту, які розміщуються в документі і забезпечують автоматичне внесення в документ певної інформації (дати, часу, назви документа і т. ін.). Для внесення в документ поля слід встановити курсор у потрібну позицію і виконати команду **Вставка/Поле**, при цьому відкривається вікно діалогу **Поле**. У списку **Категории** вікна перераховані групи, в які об'єднані всі поля. У списку **Поля** наведені назви полів, що відповідають групі, вибраній у списку **Категории**. Відповідно до вибраного поля

змінюється зміст решти елементів вікна діалогу. Кнопка **Параметры** дає змогу перейти до вікна діалогу **Параметры** і вибрати належний формат. Деякі поля (наприклад, дату) необхідно оновлювати. Для оновлення фрагмента, який включає необхідні поля, потрібно виділити фрагмент і натиснути клавішу F9. Для зручності роботи з полями користувач може замінити поля. Вигляд тіні встановлюється у вікні діалогу **Параметры**, вкладка **Вид**. Це вікно викликається командою **Сервис/Параметры**. За допомогою комбінації клавіш Alt + F9 можна переключати з перегляду інструкцій поля на режим перегляду результату їх виконання, і навпаки. **Пошук і заміна**. Режим пошуку вказаного фрагмента тексту здійснюється командою **Правка/Найти**. Ця команда відкриває доступ до вікна **Найти**. У полі **Найти** цього вікна необхідно ввести розшукуване слово або фразу. В списку **Направление** потрібно вказати, в якому напрямку слід проводити пошук: вперед (від курсора до початку тексту), назад (від курсора до кінця тексту) чи за всім текстом. Для ігнорування у процесі пошуку різниці між великими та малими буквами необхідно встановити опцію **С учетом регистра**. Word дає змогу під час пошуку використовувати шаблони. В шаблоні може використовувати символи «?» і «\*». Символ «?» у процесі пошуку розгадається як будь-яка буква, символ «\*» — як довільне число будь-яких букв. Так, якщо для пошуку зазначається слово «до?», то можуть бути знайдені слова «док», «дог», «дот», а якщо «до\*», то можуть бути знайдені слова «дот» «док», «доданок», «доза» тощо. Для використання шаблонів слід встановити прапорець **Символы шаблона**. Після встановлення всіх опцій слід натиснути кнопку **Найти далее**, і Word почне пошук. Для продовження пошуку слід знову натиснути кнопку **Найти далее**. Заміна тексту здійснюється командою **Правка/Заменить**. Вікно діалогу **Заменить** подібне до вікна діалогу **Найти**, але має ряд додаткових елементів. У поле **Заменить на** вводиться текст заміни. Заміна може здійснюватись автоматично або за участі користувача. Для ввімкнення автоматичного режиму заміни слід встановити опцію **Заменить все**. Якщо ця опція вимкнута, то для кожної заміни Word питатиме дозволу на заміну. Процес заміни починається з натискування кнопки **Найти далее**. В неавтоматичному режимі після знаходження замінюваного тексту процес зупиняється. Якщо натиснути клавішу **Найти далее**, то заміна для даного фрагмента не відбувається і шукається наступний фрагмент, який потрібно замінити. Натиснувши клавішу **Заменить**, проводимо заміну, і для продовження процесу слід натиснути клавішу **Найти далее**.

## Самостійна робота №10

**Тема:** Текстовий редактор Word для Windows

**Мета:** ознайомитись з можливостями форматування тексту, створенням і збереженням документів.

**Питання, що виносяться на самостійне вивчення:**

6.3 Форматування тексту. Створення і збереження документів

### Література:

1 Вережка П. Word 97 для Windows для “чайників”. Учебный курс. 2-е издание.: Пер. с англ. – К.: Диалектика, 1997. – 272 с.

2 Глинський Я.М. Практикум з інформатики: Навч. посібник. – Львів: „Підприємство Деол”, 1998.- 168 с.

3 Інформатика / за ред. О.І.Пушкаря – К.: Видавничий центр „Академія”, 2001. – с.696

4 Крепкий Ю.О. Методичні вказівки для вивчення текстового процесора Microsoft Word 2000 – Чернігів, 2001. – 116 с.

**Питання для самоконтролю:**

- 1 Як можна виділяти символи, слова чи фрагменти тексту?
- 2 Як почати новий абзац тексту?
- 3 Як змінити тип шрифту в одному абзаці?
- 4 Як змінити розмір літер в одному реченні?
- 5 Як вирівняти фрагмент тексту з обох боків?
- 6 Як виправити неправильний символ у тексті?
- 7 Як текст взяти в рамку?
- 8 Як з'єднати два абзаци в один?
- 9 Як розділити один абзац на два?
- 10 Яким чином можна зкопіювати фрагмент тексту (знищити фрагмент)?

## **Форматування тексту**

Під форматуванням розуміють операції, які зв'язані з оформленням тексту і зміною його зовнішнього вигляду. Операції форматування дійсні тільки для виділених фрагментів.

Розрізняють три основні операції форматування: форматування символів; форматування абзаців; форматування сторінок.

Форматування символів. При форматуванні символів можна змінити шрифт, його розмір. Форматування здійснюється командою **Формат/Шрифт**. При цьому на екрані з'явиться вікно діалогу **Шрифт**, в якому можна встановити такі параметри:

- шрифт (вибирається із списку шрифтів);
- стиль (можна вибрати звичайний стиль, курсив, напівжирний);
- розмір шрифту;
- підкреслення (вибір різних варіантів підкреслення);
- колір символів;
- ефекти (верхній і нижній індекси, перекреслений).

У полі **Образец** цього вікна показується зовнішній вигляд тексту при вибраних параметрах.

Ряд параметрів можна встановити і за допомогою кнопок панелі інструментів **Форматирование**.

Форматування абзаців. Під абзацем у Word розуміють частину документа, яка закінчується натискуванням клавіші **Enter**. Процес форматування абзацу містить у собі:

- вирівнювання абзаців;
- встановлення відступів абзаців;
- встановлення відстані між рядками і абзацами.

Для форматування абзацу використовують команду **Формат/Абзац**, яка відкриває вікно діалогу **Абзац** зі вставкою **Отступы и интервалы**.

За замовчуванням Word вирівнює абзаци ліворуч. За допомогою опції **Выравнивание** можна встановити вирівнювання праворуч, центрування (рядки розміщуються симетрично відносно уявної вертикальної лінії, що проходить через середину сторінки) або блочне вирівнювання (рядки вирівнюються як ліворуч, так і праворуч).

Опція **Отступ** дозволяє встановити відступи зліва і справа рядків виділеного абзаца, а також окремо лівий відступ першого рядка абзацу.

Для встановлення відступів можна використовувати і горизонтальну лінійку. На ній є маркери відступу ліворуч (нижній лівий), відступу праворуч (нижній правий) і відступу першого рядка абзацу (верхній лівий). В разі зміни положення цих маркерів за допомогою миші виділений абзац буде автоматично переформатовуватися.

Відстань між рядками встановлює опція **Межстрочный** вікна діалогу **Абзац**, а між абзацами – опція **Интервал**.

Форматування сторінок. При введенні і друкуванні тексту Word використовує встановлені за замовчуванням розміри поля сторінки. Змінити ці параметри можна за допомогою команди **Файл/Параметры страницы**. При цьому відкривається вікно діалогу **Параметры страницы**, яке має ряд вкладинок.

Вкладка **Поля** дозволяє встановити розміри верхнього, нижнього, лівого і правого полів.

Вкладка **Размер бумаги** дозволяє вибрати стандартні розміри із списку форматів, які підтримує Word. Якщо ні один із стандартних розмірів не підходить, то слід вибрати в списку елементів **Специальный** і для нього вказати ширину і висоту. В полі **Ориентация** потрібно вказати орієнтацію документа **Книжная** або **Альбомная**.

## **Створення і збереження документів**

Створюючи новий документ, слід подати команду **Файл/Создать** або натиснути відповідну кнопку панелі інструментів **Стандартная**. При цьому відкривається вікно діалогу **Создание документа**. Створюючи документ, на вкладинці **Общие** у полі **Создать** слід встановити опцію **Документ**. Як правило, текстовий редактор використовують для підготовки типових документів. Word пропонує користувачу стандартні форми таких

документів та інструменти для їх використання. Основними інструментами є шаблони, майстри. Шаблон — це порожній бланк, у якому користувач заповнює потрібні поля і своєю інформацією. Шаблон може попередньо включати деякі заповнені поля, які є спільними для всіх документів даного типу. Майстер базується на шаблонах, вони дають можливість ще більше спростити процес введення даних у документ. Якщо, заповнюючи шаблон, користувач самостійно переміщується по документу, розміщує в ньому інформацію, то за допомогою майстрів дані заносяться в процесі діалогу майстра з користувачем. Після натиснення кнопки **ОК** вікна діалогу **Создать** редактор відповідно до вибраного шаблону створить вікно документа з іменем **Документ**. Переглянути шаблони, приєднані до активного документа, можна за допомогою команди **Файл/Создать/Шаблоны**. Якщо користувача не задовольняють наявні шаблони та майстри, він може створити власний шаблон або змінити один з існуючих. Для зміни шаблону слід відкрити файл шаблону (файли стандартних шаблонів мають розширення dot., вони розміщуються в каталозі **MS Office/Шаблоны**), відредагувати його як звичайний документ і зберегти. Для створення нового шаблону слід подати команду **Файл/ Создать** і вибрати **Новый документ** на вкладинці **Общие** вікна діалогу **Создание документа** встановити опцію **Шаблон**. Після натиснення кнопки **ОК** на екран виведеться вікно шаблону. Від вікна документа воно відрізняється лише назвою рядка заголовка — **Шаблон** замість **Документ**. Далі слід ввести необхідні дані (параметри форматування, текстові поля) і зберегти шаблон. Щоб зберегти документ як шаблон, слід подати команду **Файл/Сохранить как**, а потім у полі **Тип файла** вибрати елемент **Шаблон документа**.

**Збереження документів.** Для збереження тексту документа в каталозі на магнітному диску слід виконати команду **Файл/Сохранить** або натиснути відповідну кнопку панелі інструментів **Стандартная**. Якщо документ новий і команда виконується для нього вперше, то відкривається вікно діалогу **Сохранение документа**. У цьому вікні можна вказати дисковод, папку, ім'я і розширення, під яким документ буде збережений. За замовчуванням документам Word присвоюється розширення DOC. Після збереження тексту в заголовку вікна документа ім'я **Документ** змінюється на ім'я, під яким збережено документ. У вікні діалогу є однорядковий список **Тип файла**. Він дає можливість інвертувати документ у формати, відмінні від Word (наприклад, формат MS DOS, формати інших текстових редакторів). Для цього слід розкрити список і обрати потрібний формат. Якщо команда **Файл/Сохранить** виконується не вперше або документ завантажувався з магнітного диска, то за командою **Файл/Сохранить** документ записується на диск під старим іменем без діалогу з користувачем. Якщо ж потрібно зберегти файл під новим іменем, в іншій папці чи з іншим форматом, то потрібно виконати команду **Файл/Сохранить как**. При цьому відкривається вікно діалогу **Сохранение документа**, й усі дії користувача аналогічні діям у разі збереження нового документа. Будь-який документ можна автоматично записувати на диск через певний інтервал часу. Для цього слід встановити режим автоматичного збереження, виконавши команду **Сервис/Параметры** і вибравши вкладнику **Сохранение**, — на екрані з'явиться відповідне вікно діалогу. В цьому вікні основний інтерес являють три опції: 1) **Всегда создавать резервную копию**. У разі встановлення цієї опції Word завжди зберігатиме останню версію документа, а також його попередню резервну копію з тим самим іменем, що й основний документ, але з розширенням ВАК; 2) **Разрешить быстрое сохранение**. У цьому режимі зберігаються тільки зміни в документі; 3) **Автосохранение каждые ... мин.** Word зберігатиме редагований документ через зазначені інтервали без команд користувача. документ, з яким працюєте, треба зберігати часто. Як ми вже говорили, ваш документ під час роботи перебуває в оперативній пам'яті і не гарантований від випадковості, одна з яких — зникнення живлення в електричній мережі — в одну мить може знищити кількогодінну працю, тож не забувайте систематично зберігатися. Це не сповільнить вашої роботи, від якої вам не хотілося б відволікатися. Процедура збереження досить проста і забирає небагато часу. Встановіть покажчик миші на кнопці із зображенням дискети і натисніть кнопку миші або виберіть у меню **Файл/Сохранить**, чи **Сохранить как**. Тільки тоді ваша робота буде в безпеці, оскільки вона буде записана на жорсткий диск. Часте збереження, що виконується регулярно, також скорочує час підсумкового збереження документа на диску в кінці сеансу роботи. Таким чином, ви ще й

заощаджує час, що також важливо. До того ж, цей спосіб зручний ще й тим, що Word автоматично приєднує новий збережений фрагмент до матеріалу, який уже є за замовчуванням. Для збереження документа потрібно дати ім'я файлу, а також вирішити, в якій папці і на якому диску його розмістити. Якщо необхідно зберегти результати роботи у вигляді різних варіантів, можна давати ім'я файлу, кожен раз дещо змінюючи його — наприклад, присвоюючи йому номер, або доповнюючи ім'я іншою літерою абетки. Таким чином, якщо є необхідність, ви завжди можете повернутися до більш ранньої версії та порівняти варіанти.

## Самостійна робота №11

**Тема:** Текстовий редактор Word для Windows

**Мета:** Ознайомитися із способами створення таблиць у текстовому процесорі MS Word. Навчитися вводити текст у таблиці, редагувати та форматувати його.

**Питання, що виносяться на самостійне вивчення:**

6.4 Створення таблиць та робота з ними

### Література:

1 Вєревка П. Word 97 для Windows для “чайників”. Учебный курс. 2-е издание.: Пер. с англ. – К.: Диалектика, 1997. – 272 с.

2 Глинський Я.М. Практикум з інформатики: Навч. посібник. – Львів: „Підприємство Деол”, 1998.- 168 с.

3 Інформатика / за ред. О.І.Пушкаря – К.: Видавничий центр „Академія”, 2001. – с.696

4 Крепкий Ю.О. Методичні вказівки для вивчення текстового процесора Microsoft Word 2000 – Чернігів, 2001. – 116 с.

**Питання для самоконтролю:**

- 1 Що таке таблиця?
- 2 Які є способи створення таблиць?
- 3 Як створити таблицю командою Вставити таблицю?
- 4 Як ввести дані у клітинку?
- 5 Як об'єднати клітинки в одну?
- 6 Як відцентрувати дані по вертикалі?
- 7 Як зберегти текстовий документ з таблицею на диску?
- 8 Як вставити рядок у таблицю?
- 9 Як убрати границю таблиці?
- 10 Які правила сортування даних для таблиць MS Word?

#### 6.4 Створення таблиць та робота з ними


Робота з таблицями виконується за допомогою п.м. “Таблица”. Робота з текстом у таблиці виконується аналогічно як із будь-яким фрагментом документа. Для створення таблиці використовується п.м. Таблица – Добавить таблицу. Таблица буде вставлена в те місце документа де знаходився курсор. Також таблицю можна намалювати за допомогою п.м. Таблица – нарисовать таблицу. Під час створення таблиці, або вже після створення її можна задати стиль таблиці або його змінити. Пункт меню Таблица – автоформат таблицы відкриває вікно-запит, у якому пропонується список стилів форматів. Редагуванні таблиці виконується за допомогою п.м. Таблица – добавить строку (столбец). Вставка рядка (стовбчика) виконується в поточне місце.

За допомогою п.м. Таблица – выделить строку (выделить колонку, выделить таблицу) можна виділяти елементи таблиці для подальшого форматування. Виділення буде виконано того рядка або стовпчика де знаходиться курсор тобто поточного.



Змінити розміри рядка або колонки можна за допомогою лінійки форматування. На лінійки будуть відображені розміри усіх рядків або стовпчиків. Мишею перетягувати бігунки і таким чином змінювати розміри. Інший шлях – це підвести мишку до межі необхідного рядка або стовпчика, натиснути ліву кнопку миши і перетягнути межу розділу між комітками. Якщо в цей час буде виділено декілька комірок, а не весь рядок (стовпчик) розміри зміняться тільки до виділеного діапазона, а не на весь рядок (стовпчик).

Щоб вилучити комірку (рядок, стовпчик) можна використати п.м. Таблица – удалить строку, удалить колонку (столбец), або Таблица – объединить ячейки.

Існує можливість створити у документі порожню таблицю і потім заповнити її комірки або перетворити існуючий текст в таблицю. Можна намалювати більш складну за структурою таблицю або створити таблицю із існуючих зовнішніх вихідних даних, наприклад, електронної таблиці MS Excel або бази даних MS Access.


Для створення порожньої таблиці необхідно встановити курсор в те місце документа, де повинна бути таблиця, та натиснути кнопку Добавить таблицу стандартної панелі інструментів . Потім за допомогою протяжки мишки слід вибрати потрібну кількість рядків і стовпчиків.

Також для вставлення таблиці до документу використовується вказівка Таблица/Вставить таблицу, після виклику якої в діалоговому вікні Вставка таблицы потрібно ввести кількість рядків та стовпчиків таблиці.

Більш складну за структурою таблицю можна намалювати за допомогою вказівки Таблица/Нарисовать таблицу або скористатися відповідною кнопкою панелі інструментів  Таблицы и границы, яка викликається на екран або за допомогою кнопки  або звичайним чином вказівкою Вид/Панели инструментов.

Кожна комірка може містити текст або графіку, яку можна формувати будь-яким способом. В окремій комірці таблиці може міститися один або кілька абзаців тексту, причому Word самостійно слідкує за перенесенням вмісту рядків в комітках. Коли текст переноситься до нового рядка, Word при необхідності автоматично збільшує його висоту.

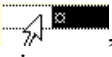

Таблиця Word може містити до 31 стовпчика. При створенні таблиці Word регулює розмір кожного стовпчика так, щоб таблиця помістилась між встановленими лівим та правим полями документа. Всі стовпчики спочатку мають однаковий розмір, але розмір кожного окремого стовпчика (а також ширину таблиці цілком) можна регулювати вручну. Кількість рядків у таблиці Word не обмежена.


Маркер окремої комірці таблиці має вигляд , його можна трактувати як еквівалент маркера абзацу для комірок. Маркер комірці відображається у режимі виведення на екран **Непечатаемых символов**.

Для переміщення від активної комірці таблиці до наступної слід натиснути клавішу **Tab**. Якщо курсор знаходиться в останній комірці останнього рядка таблиці, то натиснення цієї клавіші додає в таблицю новий рядок. Для переходу до попередньої комірці слід натиснути клавіші **Shift+Tab**. Для переміщення до попереднього або наступного рядка використовуються клавіші **↑, ↓**.

Переміщення до першої комірки рядка здійснюється за допомогою комбінації клавіш **Alt+Home** або **Alt+7** на цифровій клавіатурі. Для переміщення до останньої комірки рядка слід натиснути клавіші **Alt+End** або **Alt+1** на цифровій клавіатурі. Натиснення комбінації клавіш **Alt+Page up** або **Alt+9** на цифровій клавіатурі дозволяє переміститися до першої комірки стовпчика. Для переміщення до останньої комірки стовпчика необхідно натиснути клавіші **Alt+Page down** або **Alt+3** на цифровій клавіатурі. На початок нового абзацу в окремій комірці можна перейти за допомогою натиснення клавіші **Enter**.

### Виділення елементів таблиці

Для виділення окремої комірки разом з її вмістом необхідно встановити вказівник на лівий край комірки, курсор при цьому приймає вигляд , та натиснути ліву кнопку мишки. Для виділення рядка необхідно встановити вказівник зліва від рядка, курсор при цьому приймає вигляд , та натиснути ліву кнопку мишки. Для виділення стовпчика необхідно встановити вказівник на верхній лінії сітки стовпчика, курсор при

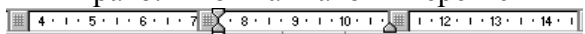
цьому приймає вигляд , та натиснути ліву кнопку мишки. Для виділення кількох комірок, рядків або стовпчиків слід, утримуючи ліву кнопку мишки, перемістити вказівник по комірках, рядках чи стовпчиках або виділити одну комірку, рядок або стовпчик, а потім, утримуючи клавішу **Shift**, виділити наступну комірку, рядок або стовпчик. Для виділення наступної комірки слід натиснути клавішу **Tab**. Для виділення вмісту попередньої комірки слід натиснути клавіші **Shift+Tab**. Крім того, виділити рядок, стовпчик чи всю таблицю можна за допомогою вказівки **Таблиця**, коли таблиця є активною, тобто вказівник введення тексту встановлено в будь-якому місці таблиці. Вказівка **Таблиця** при цьому містить такі вказівки: **Выделить строку**, **Выделить столбец**, **Выделить таблицу**.

Для виділення всієї таблиці слід перейти до таблиці, тобто зробити її активною, а потім натиснути клавіші **Alt+5** на цифровій клавіатурі при виключеному режимі **Num Lock**.

Крім того, для виділення всієї таблиці, можна скористатися вказівкою **Таблиця/Выделить таблицу**.

### Зміна розмірів стовпчиків

При початковому створенні таблиці всі стовпчики мають однакову ширину. Можна вручну змінити ширину окремого стовпчика або дозволити системі автоматично вибрати ширину кожного стовпчика в залежності від його вмісту. Кожний окремий стовпчик можна розширити або звужити за допомогою переміщення його границі (лінії сітки) вліво чи вправо. Можна також перемістити маркер правої границі стовпчика на лінійці



В будь-якому випадку розмір стовпчиків, що розташовані праворуч від того, ширина якого регулюється, налагоджується так, що ширина таблиці в цілому лишається незмінною. Якщо при налагодженні ширини стовпчика клавішу **Alt** утримувати в натиснутому стані, то значення ширини кожного стовпчика буде відображатися на лінійці.

Кнопка Автоподбор дозволяє автоматично відрегулювати ширину кожного стовпчика до максимально можливої величини відносно параметрів сторінки, на якій знаходиться активна таблиця.

Вставка та знищення елементів таблиці (рядків, стовпчиків, комірок) здійснюється за допомогою відповідних вказівок меню **Таблиця**.

Для виділення всієї таблиці слід перейти до таблиці, тобто зробити її активною, а потім натиснути клавіші **Alt+5** на цифровій клавіатурі при виключеному режимі **Num Lock**.

Крім того, для виділення всієї таблиці, можна скористатися вказівкою **Таблиця/Выделить таблицу**.

### Об'єднання комірок та розбиття на кілька частин


Існує можливість об'єднати дві або більше комірок таблиці. Наприклад, якщо потрібно об'єднати кілька комірок в одну для створення заголовка таблиці, який перетинає кілька стовпчиків.

Для об'єднання комірок слід виділити ці комірки та вибрати вказівку **Таблица/Объединить ячейки**.

Для розбиття комірки на кілька частин слід виділити її та вибрати вказівку **Таблица/Разбить ячейки**. Потім необхідно вказати кількість стовпчиків та рядків, на яку слід розбити кожен комірку.

При цьому текст комірки розбивається в залежності від кількості абзаців, які в ній знаходяться. Якщо в комірці міститься тільки один символ абзацу, то текст повністю розміщується до самої лівої з нових комірок, а останні лишаються порожніми. Якщо в комірці міститься більше ніж один символ абзацу, то абзаци розподіляються рівномірно між комірками.

### Панель інструментів Таблицы и границы

Панель інструментів Таблицы и границы можна викликати на екран за допомогою відповідної кнопки стандартної панелі  або за допомогою вказівки Вид/Панели инструментов, а далі у списку вибрати опцію Таблицы и границы. Вона містить кнопки, що викликають вказівки для вставки таблиці, її рядків, стовпців чи клітин, зміни їх параметрів, обрамлення таблиці та окремих її частин, впорядкування елементів таблиці за зростанням чи спаданням, автоматична вставка формули для обчислення суми значень групи клітин.

### Перетворення тексту в таблицю

Для перетворення тексту в таблицю слід визначити місця тексту, в яких текст повинен розділитися на рядки та стовпчики. Такі місця можна визначити за допомогою вставлення до тексту розділових символів. Наприклад, для відокремлення стовпчиків можна використати символи табуляції, а для відокремлення рядків - символи абзацу. Потім слід виділити текст, який необхідно перетворити в таблицю, та вибрати вказівку **Таблица/Преобразовать в таблицу**.

Залишається тільки задати потрібні параметри: вказати кількість та ширину стовпчиків, відмітити, який символ вважати за роздільник - символ абзацу, табуляції, крапку з комою чи інший.

### Підписи до таблиць

Word може автоматично створювати підписи до таблиць за допомогою вказівки **Вставка/Ссылка/Название**, при цьому таблиці присвоюється черговий номер, який змінюється при вставленні нових таблиць чи вилученні існуючих. На екран викликається діалогове вікно Название.

В полі **Постоянная часть** необхідно вибрати **Таблица**, в полі **Название** вибрати або вставити потрібну назву таблиці, в полі **Положение** вибрати місце розташування назви та клацнути на кнопці **Ок**.

Кнопка **Автоназвание** дозволяє встановити режим, в якому автоматично будуть добавлятися назви до всіх вставлених до документу об'єктів. Слід тільки вибрати типи об'єктів, при вставленні яких повинні автоматично створюватися їх назви.

## Самостійна робота №12

**Тема:** Текстовий редактор Word для Windows

**Мета:** Навчитися використовувати готові та створювати власні графічні об'єкти, оформляти матеріали художнього змісту та зображення, набути навички для побудови математичних формул і виразів.

**Питання, що виносяться на самостійне вивчення:**

6.5 Використання графіки. Введення математичних формул і рівнянь

### Література:

1 Вережка П. Word 97 для Windows для “чайників”. Учебный курс. 2-е издание.: Пер. с англ. – К.: Диалектика, 1997. – 272 с.

2 Глинський Я.М. Практикум з інформатики: Навч. посібник. – Львів: „Підприємство Деол”, 1998.- 168 с.

3 Інформатика / за ред. О.І.Пушкаря – К.: Видавничий центр „Академія”, 2001. – с.696

4 Крепкий Ю.О. Методичні вказівки для вивчення текстового процесора Microsoft Word 2000 – Чернігів, 2001. – 116 с.

**Питання для самоконтролю:**

- 1 Які об'єкти може містити текстовий документ?
- 2 Яким чином можна вставити готові об'єкти у документ?
- 3 Як виділити об'єкт (групу об'єктів)?
- 4 Як вставити об'єкт WordArt? Охарактеризуйте меню «Текст – Фігура» панелі **WordArt**
- 5 Як повернути об'єкт на 180°?
- 6 Як змінити розмір об'єкта, зберігаючи його пропорції?
- 7 У чому полягає операція групування і розгрупування?
- 8 Як створити та включити в текст документа формулу?
- 9 Як вибрати колір об'єкта в документі?

## 6.5 Використання графіки

Робота з графікою в Word представлена у вигляді кількох розділів.

- Використання готових малюнків в тексті
- Безпосереднє малювання в Word (застосування вбудованої програми Рисование)
- Діаграми та графіки

### I. Використання готових малюнків

Програма **Word** дозволяє легко вставляти готові малюнки в текст. При цьому підтримується три варіанти вставки:

- З файла
- З бібліотеки Clip Gallery
- З інших програм

Остання, в свою чергу, підтримує такі вставки:

- З інших програм через буфер обміну
- З інших програм як об'єкт

#### 1. Вставка малюнка з файла

Для цього необхідно вибрати вказівку **Вставка/Рисунок**, а потім вказівку **Из файла**. Малюнок вставляється в текст з позиції курсору і подальше його розміщення в тексті залежить від параметрів малюнка. Щоб налагодити відповідні параметри, треба в контекстному меню малюнка вибрати вказівку **Формат рисунка** або в меню **Формат** вибрати вказівку **Рисунок**.

На вкладці **Положение** можна вибрати різні варіанти обтікання (**в тексте, вокруг рамки, по контуру, за текстом, поверх текста**). Якщо необхідно прив'язати малюнок до тексту, доцільно включити опцію **В тексте** – в цьому режимі малюнок розташований між рядками тексту, а при виділенні натисненням миші навколо малюнка з'являється чорна рамка. Такий малюнок поводить себе як велика літера і переміщується в документі тими ж засобами – кнопками вирівнювання, пропусками і т.д. Всі інші опції дозволяють налаштувати різні способи обтікання малюнка текстом, і при виділенні навколо такого малюнка з'являються порожні білі маркери, переміщення по тексту можна здійснювати перетягуванням мишкою.

На вкладці **Рисунок** можна задати значення для обрізання малюнка, встановити параметри зображення – колір, яскравість, контрастність...

При виділенні малюнка з'являється панель інструментів **Настройка изображения**, за допомогою якої можна швидко встановити параметри малюнка, вставити новий малюнок.

#### 2. Вставка малюнка з бібліотеки Clip Gallery

Для цього треба скористатися командою **Вставка/Рисунок/Картинки**, вибрати необхідну картинку і вставити її за допомогою буфера обміну, параметри малюнка налагоджуються аналогічно п.1.

#### 3. Вставка малюнка з іншої програми

Можна вставити малюнок з будь-якої програми, що підтримує роботу з буфером обміну, скориставшись копіюванням та вставкою з буфера. Після цього всі параметри малюнка встановлюються аналогічно.

#### 4. Вставка и редактирование копий экрана

При натисненні на клавіатурі клавіші **Print Screen** "фотографія" екрану потрапляє в буфер обміну. Щоб отримати лише певну частину цього зображення, можна скористатись редактором **Paint (Пуск/Программы/Стандартные/MS Paint)**, вставити в нього копію екрану з буфера обміну, виділити та скопіювати в буфер потрібний фрагмент, а потім його вставити в Word.

При натисненні комбінації клавіш **Alt+Print Screen** в буфер обміну потрапляє копія активного вікна, таким чином можна отримати копії діалогових вікон або будь-якого одного вікна. При вставці такого зображення в Word ми працюємо з ним як з малюнком і всі параметри малюнка встановлюються аналогічно.

Розглянуті способи дозволяють вставити зображення як малюнок, для них є спільними команди в контекстному меню та параметри форматування малюнка. При подвійному кліку мишкою на малюнку відкривається діалогове вікно **Формат рисунка**.

#### 3. Вставка малюнка як об'єкта з іншої програми

При використанні вказівки **Вставка/Объект** можна скористатись трома способами вставки малюнка як об'єкта.

- Якщо вибрати вкладку **Создание** і в запропонованому списку вибрати тип об'єкта **Точечный рисунок**, то можна створити малюнок засобами Paint. При кліку мишкою за межами даного вбудованого об'єкта малюнок фіксується в тексті і після цього подвійним кліком на малюнку викликається редактор Paint, що дозволяє внести зміни в обраний малюнок.
- На вкладці **Создание из файла** задати ім'я потрібного файла, але прапорець **Связь с файлом** залишити вимкненим. В цьому випадку ми отримуємо також вбудований об'єкт, при подвійному кліку мишкою на якому викликатиметься Paint і зміни в нього можна вносити безпосередньо знаходячись в Word.
- На вкладці **Создание из файла** задати ім'я потрібного файла та ввімкнути прапорець **Связь с файлом**. При цьому ми отримуємо зв'язаний об'єкт, який відображає зміни, що відбулись і вихідному файлі після його використання. В контекстному меню такого об'єкта з'являється команда **Обновить связь**, що дозволяє відобразити зміни, що відбулися у файлі-джерелі.

Щоб змінити розміри й положення малюнка, слід натиснути на ньому мишею, після чого навколо нього з'являться маркери розміру. Пересуваючи кутові маркери мишею, можна змінювати розміри малюнка при зберіганні його пропорцій. При пересуванні інших маркерів буде змінюватися ширина або довжина малюнка. Перемістити малюнок можна за допомогою миші. При переміщенні малюнка за межі видимості екран просунеться в тому ж напрямку.

Для знищення малюнка його слід виділити й натиснути клавішу **Delete**.

## II. Рисование в Word

Малювання в текстовому документі виконується за допомогою готових графічних об'єктів, які викликаються кнопкою **Рисование** на панелі інструментів, при цьому з'являється панель інструментів **Рисование**. Вбудовані засоби малювання можна використовувати тільки в режимі **Разметка страницы**.

При створенні малюнка навколо нього відображається полотно, яке дозволяє впорядковувати та змінювати розміри об'єктів малюнка.

За допомогою кнопок панелі малювання можна зображувати лінії, стрілки, еліпси, прямокутники, кола, дуги, сектори й різні криві. Графічний об'єкт можна залити кольором або візерунком, змінити форму, дзеркально відбити або повернути, змінити колір і тип його ліній, додати до них стрілки. Кнопка Автофігуры відкриває малювання груп об'єктів. Декілька об'єктів можна згрупувати. Групою називають набір об'єктів, які Word розглядає як об'єкт. Якщо об'єднати кілька простих об'єктів до одного, то потім його можна переміщувати та копіювати. Для того, щоб створити групу, слід спочатку виділити об'єкти, які треба об'єднати (для виділення кількох об'єктів використовується клавіша **Shift** або мишкою окреслюється частина малюнка, до якої входять потрібні об'єкти), потім клацнути на кнопці **Действия** панелі інструментів **Рисование** та вибрати потрібну команду - **Группировать, Разгруппировать, Перегруппировать**.

Для підпису окремих частин малюнку доцільно скористатись елементами **Выноска**, які викликаються за допомогою кнопки, потім у списку вибрати **Выноски**, після чого можна вибрати потрібний тип виноски.






Для підпису окремої частини малюнку необхідно клацнути в тому місці малюнка, до якого повинна відноситись виноска. Далі слід перетягнути мишку до того місця, де повинен знаходитись текст виноски. Після цього на екрані з'являється поле тексту виноски, яке можна при необхідності змінити. Автоматично на екран викликається панель інструментів **Надпись**. За допомогою вказівки **Формат автофігуры** можна змінити формат виноски.

### Спецефекти, вбудовані за допомогою WordArt.

Програма Microsoft WordArt призначена для створення різних логотипів, які включають витягнутий та зогнутий текст або текст з тінню.

Для того, щоб створити такий логотип необхідно викликати вказівку **Вставка/Объект** із списку **Тип объекта** діалогового вікна **Вставка объекта** слід вибрати рядок Microsoft



Інструмент	Шаблони, які доступні при використанні
	Шаблони різних інтегралів.
	Шаблони з лініями під та над текстом.
	Шаблони для малювання стрілок під та над текстом.
	Шаблони для позначення різних множин.
	Шаблони матриць.

Вставлена до документу формула є об'єктом, для зміни її розміру слід виділити та скористатися маркерами розміру, як і для зміни розмірів та місця розташування звичайних графічних об'єктів. При збільшенні чи зменшенні формули в документі, шрифти на екрані можуть змінюватися не пропорційно. Однак при виведенні документа на друк цього не виникає.