

Тема №2 .

СЕРЕДОВИЩЕ BORLAND PASCAL. СТРУКТУРА ПРОГРАМИ. ОРГАНІЗАЦІЯ ВВОДУ/ВИВОДУ

Система програмування Borland (Turbo) Pascal 7.0 являє собою інтегроване середовище і включає в себе екранний редактор, компілятор, редактор зв'язків, налагоджувач.

Інтегроване середовище Turbo Pascal запускається командою **turbo** [необов'язкове ім'я файлу] або **bp** [необов'язкове ім'я файлу] для середовища Borland Pascal.

Робочий екран інтегрованого середовища Borland Pascal складається з трьох різних частин – рядок меню в верхній частині екрану, рядок стану внизу, та робоча область в центрі.

Основні функції інтегрованого середовища Borland (Turbo) Pascal наведено в додатку 2.

Структура програми

В загальному випадку структуру Pascal-програми можна записати у такому вигляді:

Розділ оголошень та узгоджень

PROGRAM	Заголовок програми
{ \$... }	Глобальні директиви компілятора
USES	Бібліотеки, що підключаються
LABEL	Оголошення глобальних міток
CONST	Оголошення глобальних констант
TYPE	Оголошення глобальних типів
VAR	Оголошення глобальних змінних

Розділ текстів процедур і функцій

PROCEDURE/FUNCTION	Заголовок процедури/функції
LABEL	Оголошення локальних міток
CONST	Оголошення локальних констант
TYPE	Оголошення локальних типів

VAR	Оголошення локальних змінних
BEGIN	Початок блоку процедури/функції
END ;	Кінець блоку процедури/функції
Розділ основного блоку програми	
BEGIN	Початок основного блоку програми
END .	Кінець основного блоку програми

Організація вводу/виводу

Для реалізації зчитування значень з клавіатури використовуються оператори – *read(x)*, *readln(x)*. Ідентифікатор може бути один або декілька, розділених комою. Програма, зустрівши такого оператора, буде чекати вводу даних з клавіатури та натиснення клавіші *Enter*. Наприклад,

readln(x, y) ;

Такий оператор зупинить виконання програми і буде чекати доки користувач не введе з клавіатури значення, що буде присвоєне змінній *x*, натисне пропуск («пробіл»), введе значення, що буде присвоєне змінній *y* та натисне *Enter*. Після чого виконання програми буде продовжено.

Різниця між оператором *read* та *readln* полягає в тому, що після виконання зчитування даних оператор *read* залишає курсор (поточне положення) в тому рядку, де відбувалося зчитування, а оператор *readln* переводить курсор на початок наступного рядка. Така сама різниця і між операторами виводу даних – *write*, *writeln*. Оператори виводу, на відміну від операторів вводу, не зупиняють виконання програми, а лише виводять дані на екран. Після виконання будь-якого з цих операторів, наступне введення/виведення даних буде виконуватися в тому рядку (якщо використовувався *read* чи *write*) або ж на початку наступного рядка (*readln* чи *writeln*).

Оператор *readln* можна використовувати без ідентифікаторів та дужок. За цим він зупиняє виконання програми до натиснення клавіші *Enter*, у такому випадку зчитування даних не відбудеться і користувачеві необхідно поясни-

ти що йому робити для продовження виконання програми (натиснути *Enter*).

Завдання

Ознайомитися з головним меню та вікнами середовища, відпрацювати основні прийоми роботи з редактором коду.

Скласти блок-схему алгоритму та написати програму знаходження суми двох чисел. Спочатку створити програму із заданням чисел у програмі (у вигляді змінних та констант), потім змінити її так, щоб значення зчитувалися з клавіатури. Доповнити програму інформаційними повідомленнями та коментарями.

Навчитися виконувати прогін та відстеження створеної програми. Ознайомитися зі звітом про виконану програму та правилами його оформлення.

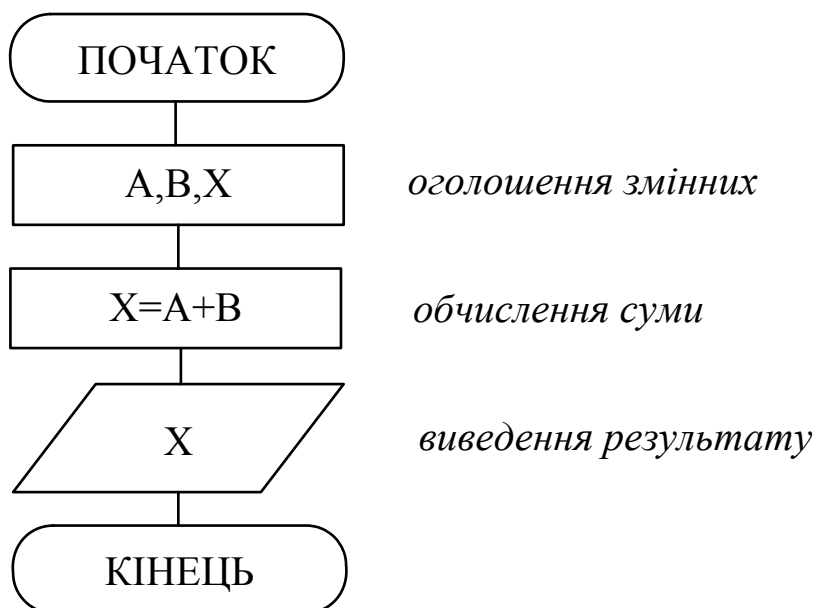
Методичні рекомендації

1. Завантажити інтегроване середовище Borland Pascal за допомогою файлу запуску `bp3` (або за допомогою попередньо встановленого ярлика програми).
2. Детально ознайомитися з середовищем програмування та пунктами головного меню (додаток 2).
3. Скласти блок-схему алгоритму задачі обчислення суми двох чисел, попередньо проаналізувавши технічну постановку задачі.

При аналізі задачі необхідно вияснити, які дані повинні бути на вході, які – на виході, та які дії повинні виконуватися над даними в програмі. Для задачі, що розглядається, вхідні дані (два значення) задаються безпосередньо в програмі. Програма виконує операцію додавання і результат виводиться на екран.

³ Якщо в наявності середовище Turbo Pascal, запуск відбувається за допомогою файлу запуску `turbo` (див. короткі теоретичні відомості до роботи)

Знаючи всі вимоги, які ставляться до програми, можна скласти блок схему алгоритму:



4. За складеним алгоритмом у вигляді блок-схеми записати програму на Pascal.

Для цього в робочій області інтегрованого середовища Borland Pascal вводимо наступний текст програми:

```

PROGRAM SumTwice;
VAR
  a,b,x: integer;
BEGIN
  a:=5;
  b:=7;
  x:=a+b;      {обчислення}
  write(x);   {виведення результату}
END.
  
```

5. Виконати компіляцію створеної програми за допомогою пункту меню *Compile*⇒*Compile* або за допомогою гарячих клавіш *ALT+F9*. При виявленні помилок – виправити їх та перекомпілювати програму.
6. Запустити програму на виконання *Run*⇒*Run* (*CTRL+F9*).

7. Переглянути результат виконання програми за допомогою переходу у вікно користувача *Debug*⇒*User screen* (*ALT+F5*). А також здійснити перегляд результату разом з кодом програми *Debug*⇒*Output*.
8. Змінити програму таким чином, щоб спочатку на екран виводилася інформація щодо програми та її розробника, потім – значення змінних, що додаються і лише потім – результат додавання.
9. Здійснити покрокове виконання програми *Run*⇒*Step over* (*F8*) та *Run*⇒*Trace into* (*F7*). Розглянути можливість виконання програми до поточного положення курсору *Run*⇒*Go to cursor* (*F4*).
10. Створити подібну програму з різницею в тому, що числа, які додаються, задаються у вигляді констант.
11. Змінити програму таким чином, щоб початкові значення зчитувалися з клавіатури, а не задавалися програмно.
12. Дописати в програму команду зупинки для можливості перегляду виведених програмою результатів.
13. Записати коментарі до програми та зберегти її в персональній папці студента.
14. Оформити звіт про виконану роботу (зразок оформлення звіту в додатку 3).

Індивідуальні завдання:

Здійснити зчитування необхідних даних з клавіатури, виконати обчислення та вивести результат на екран.

Варіант 1.

Обчислити площу кола.

Варіант 2.

Обчислити довжину кола.

Варіант 3.

Обчислити площу прямокутника.

Варіант 4.

Обчислити площу квадрата.

Варіант 5.

Обчислити швидкість руху автомобіля.

Варіант 6.

Обчислити середнє арифметичне значення трьох чисел.

Варіант 7.

Розділити порівну зчитану суму грошей між п'ятьма працівниками.

Варіант 8.

Перевести в кілометри відстань, задану в милях.

Варіант 9.

За відомим значенням ваги в кілограмах вивести еквівалент в грамах і тонах.

Варіант 10.

Розв'язати рівняння $ax + b = c$ (при $a \neq 0$).

Варіант 11.

Перевести зчитане значення температури в градусах Цельсія в градуси Фаренгейта.

Варіант 12.

Визначити об'єм циліндричного бака.

Варіант 13.

Визначити виручку від продажу n товарів по m гривень.

Варіант 14.

Обчислити об'єм кубу.

Варіант 15.

Визначити суму до сплати за використану електроенергію при відомих тарифах (за лічильником).

Контрольні запитання:

- 1) Що таке змінна та константа? Чим відрізняється глобальна змінна/константа від локальної?
- 2) Які типи даних мови Pascal Вам відомі? Які групи серед них можна виділити?
- 3) Назвіть всі можливі варіанти налагодження програми в середовищі Borland Pascal.
- 4) Яка структура програми на мові Pascal?
- 5) Що означає ключове слово VAR?
- 6) Що записують після слова USES?
- 7) Що розуміють під вводом/виводом? Наведіть практичні приклади.
- 8) Для чого використовують функції та процедури користувача?
- 9) Які методи організації вводу/виводу можна використувати при написанні програм?
- 10) Як можна відлагодити програму рядок за рядком?
- 11) Як зупинити виконання програми на визначеному рядку?
- 12) Чим відрізняється TRACE INTO від STEP OVER?
- 13) Як працювати з декількома програмами в середовищі Borland Pascal одночасно?
- 14) Як здійснити одночасний перегляд результатів роботи програми і покрокове її виконання?
- 15) Розкрийте послідовність запису програми та її відкриття.