

Лабораторна робота №2

Структура Pascal та VBA програм.

Ввід-вивід даних.

Лінійні обчислення математичних формул.

Мета роботи: здобуття практичних навичок підготовки та редагування текстів TurboPascal та VBA програм; використання команд вводу-виводу.

Теоретичні відомості.

2.1. Структура програми

2.1.1 Структура TP програми

Кожна TP програма складається з де-кількох обов'язкових частин.

Програма починається із заголовка за яким слідує блок описів та блок операторів - безпосередньо програма. У блоці описів міститься інформація про усі об'єкти, що будуть використовуватись, а в блоці операторів - послідовність дій з описаними об'єктами.

Заголовок програми

Заголовок програми виконує допоміжну функцію і ніяк не впливає на хід виконання програми. Заголовок, якщо він присутній, починається з ключового слова Program за яким слідує ім'я програми та обов'язковий розділювач (;). Ім'я програми повинне відповідати вимогам до ідентифікаторів у TurboPascal. Допускається порожнє ім'я програми. Не допускається використання для імені програми зарезервованих слів типу If, Procedure, Function та ін. Якщо ім'я програми співпадає з назвою однієї з вбудованих функцій, процедур чи модулів TurboPascal, то останні стають недоступними при виконанні програми.

Приклади:

Правильно
Program;
Program MyProgram;

Не правильно
Program
Program data development;
Program graph;

Блок описів

У блоці описів містяться описи міток, констант, типів, змінних, процедур та функцій. Кожен опис повинен починатись з відповідного ключового слова, а закінчуватись обов'язковим розділювачем (;). Якщо поточне описування відноситься до того ж класу об'єктів, що й попереднє, то ключове слово може бути опущене.

Блок операторів

Блок операторів або безпосередньо програма являє собою опис послідовності дій. Блок операторів завжди починається ключовим словом BEGIN, а закінчується ключовим словом END з крапкою. Таку комбінацію називають ще зовнішніми логічними дужками, на відміну від просто логічних дужок (Begin ... end;), що закінчуються (;). (TurboPascal не розрізняє великих та малих літер, тому наведені логічні дужки є ідентичними. Однак рекомендується зовнішні логічні дужки вводити саме великими літерами для полегшення читання тексту програми).

Таким чином структура TurboPascal-програми може бути умовно зображена:

```
[Program [ProgramName] ;]  
[type .... ;]  
[var .... ;]  
[const .... ;]  
[label .... ;]  
[.....;]  
BEGIN  
[.....]  
END.
```

2.1.2 Структура VBA програми

Програма створена за допомогою VBA за структурою принципово відрізняється від Pascal програми. VBA дозволяє створювати об'єктно-орієнтовані проекти, що базуються на об'єктах програм MS Office. Частіш за

все VBA програма зберігається разом з документом або шаблоном в якому створена. Для запуску такої програми на виконання використовують одну з стандартних подій: відкриття файлу, що містить програму, натискання кнопки на панелі інструментів, генерація спеціальної події, натискання комбінації гарячих клавіш тощо. На етапі освоєння VBA, запускати програму можна безпосередньо з середовища, за допомогою команди меню **Run → Run Sub/UserForm**, або просто натиснувши клавішу **F5**.

Однією з характерних для VBA програм є структура, що базується на основній формі – спеціальному вікні з органами управління, до якої приписують основний модуль. Докладніше роботу з формами буде розглянуто пізніше.

Таким чином структура VBA програми може бути досить розгалужена і складатися з окремих процедур (**Sub**) або функцій (**Function**), які викликаються одна з одної. Кожна процедура чи функція може містити опис змінних, які в ній використовуються. Однак на відміну від Pascal-програми, опис змінних є необов'язковим.

Головний модуль програми – модуль, що запускається першим, може містити додаткові вказівки. До останніх відноситься: **Option explicit** – вказівка вимагати опис змінних до їх використання та **Option base 1** – вказівка на те, що нумерація елементів масивів починається з 1 (за замовчуванням з 0). Ці додаткові вказівки VBA виносить у верхню частину модуля, відділяє їх рискою. Їх дія розповсюджується на усі модулі проекту.

Опис змінних всередині одного модуля звичайно проподять оператором **Dim**, який має наступний синтаксис:

```
Dim Var1Name as TypeName1 , Var2Name as TypeName2 ...  
Var1Name, Var2Name, ... - ідентифікатори змінних.
```

TypeName – назви стандартних типів даних VBA.

Докладніше опис даних буде розглянуто нижче.

2.2. Ввід-вивід даних

2.2.1 Ввід-вивід в TP-програмі

Операції вводу-виводу є основними засобами спілкування між оператором та комп'ютером. TurboPascal забезпечує виконання цих дій за допомогою стандартних процедур **Read**, **ReadLn**, **Write**, **WriteLn** - відповідно для вводу та виводу. Вказані процедури працюють з типізованими і нетипізованими файлами. Стандартний пристрій вводу - консоль (CON:) прирівнюється до нетипізованого файлу і сприймається без вказування.

Процедура **Read**

- для типізованого файлу, читає черговий компонент файлу і присвоює прочитане значення змінній.

Формат: **Read(f , v1 [, v2, ..., vn])**;

де **f** - змінна або константа типу **file of ...**;

v1, v2, vn - перша, друга та n-на змінні відповідно;

- для нетипізованого файлу читає одне або більше значень в одну або більше змінних.

Формат: **Read([var f: text;] v1 [, v2, ..., vn])**;

де **f** - змінна типу **text**;

Процедура **ReadLn**

- виконує процедуру **Read** і переходить до іншого рядка файлу; працює тільки з нетипізованими файлами (**text**) і стандартним пристроєм вводу-виводу (**CON:**).

Формат: **ReadLn([var f: text;] v1 [, v2, ..., vn])**;

Процедура **Write**

- для типізованих файлів, записує значення однієї або декількох змінних в компоненти файлу.

Формат: **Write(f, v1 [, v2, ..., vn])**;

- для нетипізованих файлів, записує значення однієї або декількох змінних в файл.

Формат: **Write**([var f: text;] v1 [,v2,...,vn]);

Процедура **WriteLn**

- виконує процедуру **Write** і виводить в файл маркер кінця рядка (переводить каретку на новий рядок); працює тільки з нетипізованими файлами (**text**) і стандартним пристроєм вводу-виводу (**CON:**).

Формат: **WriteLn**([var f: text;] v1 [, v2, ..., vn])

Процедури виводу **Write** і **WriteLn** надають можливість визначати ширину полів та формат виводу. Якщо змінні просто перераховані у списку операторів **Write** чи **WriteLn**, то їх значення виводяться підряд, без пропусків, причому цілочисельні, логічні, символні та рядкові значення займають стільки позицій, скільки їм необхідно. Дійсні значення за тих же умов виводяться в форматі, що нараховує 17 символів: 1-знак числа (+ або -); 2-13 перші 11 цифр мантиси числа; 14-17 степінь числа в форматі E+## або E-##. Якщо після аргументу будь-якого типу через двокрапку вказати ціле число, або вираз, що має своїм результатом ціле число то вивід буде здійснюватись у крайні праві позиції поля вказаної ширини. Вивід дійсних виразів здійснюється в тому ж форматі, але ширина поля не може бути менше 8. Якщо після дійсного виразу через двокрапки вказати два цілих числа, то в крайні праві позиції поля вказаної ширини виводиться значення в форматі з фіксованою крапкою, причому після крапки виводиться кількість (0 ... 24) цифр дробової частини, задана другим цілим числом.

Приклад:

```
a := 2.123456789
WriteLn(a);
WriteLn(a:4:6);
```

Перший оператор виведе число **+2.1234567890E+00**, в той час коли другий – число **2.123456**.

Функція **ReadKey**.

Додаткові можливості вводу надає функція **ReadKey**, що розміщується в зовнішньому модулі **CRT**. Функція **ReadKey** читає символ з клавіатури. Результат має символний тип (**Char**).

Формат: `ReadKey: Char;`

Приклад використання:

```
uses Crt;
var Ch : Char;
BEGIN
    WriteLn('Press a key');
    Ch := ReadKey;
    if Ch = #0 then
        begin
            Ch := ReadKey;
            WriteLn('You hit special key number ',
                Ord(Ch));
        end
    else
        WriteLn('You hit a key with ASCII ', 'code ', Ord(Ch));
    END.
```

2.2.2 Ввід-вивід в VBA-програмі

Принципи виконання функцій вводу-виводу в VBA-програмі суттєво відрізняються від розглянутих для Pascal-програми. Способи вводу інформації в програму і виводу програмою результатів можна умовно розділити на три групи: використання стандартних діалогових вікон, використання полів материнської програми (наприклад Excel), використання форм. Форми – являються найбільш універсальними засобами вводу-виводу, але, зважаючи на їх складність, будуть розглянуті в окремій лабораторній роботі.

Іншою особливістю вводу-виводу в VBA являється те, що інформація, представлена на екрані являється послідовністю символів, а не числом, чи іншою інформацією з відомою структурою. Тому з'являється необхідність перетворення послідовності знаків в число, чи числа в послідовність знаків, використовуючи стандартні функції перетворення.

Ввід через стандартне вікно вводу

Стандартне вікно вводу створюється функцією `InputBox`, у вікні відображається підказка і поле для вводу інформації, а також набір кнопок.

Синтаксис:

InputBox(prompt[, title][, default][, xpos][, ypos][, helpfile, context]), де

prompt – обов'язковий текстовий вираз, що служить підказкою. Максимальна довжина виразу складає 1024 символи. Якщо необхідно вивести підказку в декілька рядків, використовують символи „повернення каретки” **Chr(13)** або/і „переведення рядка” **Chr(10)**.

title – необов'язковий текстовий вираз, що відображається як заголовок вікна. Якщо вираз пропущено, то в якості заголовка вікна використовується назва материнської програми

default – необов'язковий текстовий вираз, що відображається в полі вводу, як відповідь за замовчуванням. Якщо його пропустити, то поле вводу відображається порожнім.

xpos, ypos – необов'язкові числові вирази, що дозволяють розмістити вікно вводу на екрані, означають відстань від лівого краю екрану до лівого краю вікна та відстань від верхнього краю екрану до верхнього краю вікна відповідно. Відстані вимірюються в одиницях **twip**, що складають 1/20 пункта, або 1/1440 умовного дюйма на екрані. Якщо параметри пропущено, то діалогове вікно розташовується приблизно по центру екрану.

helpfile, context – необов'язкові параметри, що використовуються у тих випадках, коли необхідно створити засоби виклику розширеної підказки. Частіше всього наведену функцію використовують як праву частину виразу присвоювання, в лівій частині якого знаходиться змінна, для якої зчитується значення. Наприклад:

```
a = InputBox(" A = ")
```

При виконанні цього виразу в програмі на екрані з'явиться вікно, наведене на рисунку 2.1.



Рисунок 2.1. Зовнішній вигляд простого вікна вводу.

Вивід через стандартне вікно виводу

Стандартне вікно виводу створюється функцією **MsgBox**. У вікні відображається інформація, призначена для виводу та набір кнопок.

Синтаксис:

MsgBox (prompt [, buttons] [, title] [, helpfile, context]), де **prompt** – обов'язковий текстовий вираз, що являється предметом виводу.

На цей вираз поширюються правила та обмеження, наведені вище, в описі функції **InputBox**. При потребі, рекомендується текстовий вираз створювати раніше, включаючи в нього постійні та змінні рядкові величини, перетворені в послідовість символів числові значення, тощо.

buttons – необов'язковий числовий вираз, що являється сумою значень, що описують кількість і вид кнопок в стандартному вікні виводу. Деякі можливі значення наведено в таблиці 2.1, докладніше з можливими значеннями можна ознайомитись з системи електронної підказки VBA. Якщо значення пропущено, то використовується 0.

Таблиця 2.1 Окремі значення констант для змінної **buttons**

Константа	Значення	Опис
vbOkOnly	0	Відображає тільки кнопку OK
vbOkCancel	1	Відображає кнопки OK та Cancel
vbYesNo	4	Відображає кнопки Yes та No
vbMsgBoxHelpButton	16384	Додає кнопку Help

title – необов'язковий текстовий вираз, що відображається як заголовок вікна. Якщо вираз пропущено, то в якості заголовка вікна використовується назва материнської програми

helpfile, **context** – необов'язкові параметри, що використовуються у тих випадках, коли необхідно створити засоби виклику розширеної підказки. У випадку, якщо функція містить тільки один аргумент, вона може використовуватись окремо і тільки для виводу інформації. Наявність аргументів крім першого перетворює **MsgBox** з функції виводу на функцію вводу-виводу і передбачає її використання, як правої частини операторів присвоювання. Змінній, що являється лівою частиною такого оператора, передається значення, що відповідає коду натиснутої кнопки.

Приклад застосування в простому форматі:

MsgBox ("Система на має розв'язків")

При виконанні цього виразу в програмі на екрані з'явиться вікно, наведене на рисунку 2.2.



Рисунок 2.2. Зовнішній вигляд простого вікна виводу.

Ввід-вивід через поля материнської програми.

Використання полів даних материнських програм є досить гнучким інструментом вводу-виводу. В якості таких полів можуть виступати, наприклад, клітинки електронного бланку Excel.

Внесення інформації безпосередньо в клітинку таблиці для наступної обробки програмою на VBA не відрізняється від простого внесення даних. Читання даних з клітинки таблиці здійснюється оператором присвоювання, в лівій частині якого розташовано ідентифікатор змінної відповідного типу.

Наприклад:

a = Sheets ("Sheet1") .Cells (1,1) .Value

Вираз, що знаходиться справа від знаку присвоювання (=) являється властивістю **Value** (значення) об'єкту **Sheets ("Sheet1") .Cells (1,1)**. Фактично вказується спосіб вибрати необхідний лист з колекції листів поточної робочої книги (за іменем

Sheet1) та спосіб вибрати необхідну клітинку з колекції клітинок вибраного робочого листа за номером рядка та стовпчика.

Вивід здійснюється за допомогою дзеркального запису:

```
Sheets ("Sheet1") .Cells (1,2) .Value = x1
```

2.3. Виконання лінійних обчислень математичних формул.

2.3.1. Математичні обчислення в Pascal-програмі

Математичні обчислення в Pascal-програмі виконуються за допомогою оператора присвоювання (assignment) (**:=**). З лівого боку від знаку **:=** знаходиться змінна, якій буде присвоєно нове значення. З правого боку - вираз, значення якого буде присвоєно змінній. Змінна та Вираз повинні бути сумісних типів.

Вираз будується зі змінних, констант та функцій, з'єднаних між собою знаками: '+' додати, '-' відняти, '*' помножити, '/' розділити, 'div' розділити цілочисельно, 'mod' взяти залишок від цілочисельного ділення. Функції, що найбільш часто використовуються в математичних обчисленнях наведено в табл. 2.2.

Аргументом функції може бути змінна, константа або функція, з типом, що відповідає типу аргументу.

Зауваження: Довжина рядка Pascal-програми не повинна перевищувати 128 символів, включаючи відступ на початку рядка. Це може викликати необхідність обчислення довгих математичних формул в декілька прийомів. Для тимчасового зберігання проміжних результатів можна використовувати додаткові змінні, але зручніше зберігати проміжний результат в тій же змінній, якій буде присвоєно кінцевий результат.

Таблиця 2.2 – Вбудовані математичні функції Pascal

Функція	Тип аргументу	Тип функції	Значення
Алгебраїчні функції			
Abs (X)	Real, Integer	Той же	Модуль числа
Exp (X)	Real, Integer	Real	Експонента ех
Ln (X)	Real, Integer (не від'ємне)	Real	Натуральний логарифм
Sqr (X)	Real, Integer	Той же	Квадрат числа x^2
Sqrt (X)	Real, Integer (не від'ємне)	Real	Квадратний корінь
Тригонометричні функції			
ArcTan (X)	Real	Real (в радіанах.)	Арктангенс
Cos (X)	Real (в радіанах)	Real	Косинус
Sin (X)		Real	Синус

Приклад:

$$a = \ln \left| y - \sqrt{|x|} \cdot \left(x - \frac{y}{z + \frac{x^2}{4}} \right) \right|$$

вираз

може бути записаний:

```
a := x-y / (z+sqr (X) / 4) ;  
a := ln (abs (y-sqrt (abs (x) ) * a) )
```

2.3.2. Математичні обчислення в VBA-програмі

Запис математичних обчислень рідко залежить від мови програмування. Найчастіше відмінності зводяться до різного позначення оператора присвоювання та індивідуального набору математичних операцій та функцій для кожної з мов. У випадку VBA оператор присвоювання позначається одиночним символом (=). Арифметичні операції в VBA здійснюються за допомогою знаків '+' додати, '-' відняти, '^' підняти до степені, '*' помножити, '/' розділити, '\' розділити цілочисельно, 'mod' взяти залишок від цілочисельного ділення.

Список математичних функцій VBA в цілому збігається зі списком функцій Pascaal. Можна вказати наступні відмінності:

- в VBA відсутня функція „квадрат”, а нотація **SQR (X)** – означає квадратний корінь від X;
- в VBA функція натурального логарифму записується як **LOG (X)** ;
- в VBA додатково існує тригонометрична функція **TAN (X)** – тангенс;
- функція арктангенс в VBA позначається як **ATN (X)** .

Завдання на лабораторну роботу.

Скласти та відлагодити програму для вводу даних, обчислення значення невідомих та виводу результатів обчислень за завданням викладача, використовуючи останній приклад. Перевірити результати роботи програми при використанні різних команд вводу-виводу і в різних середовищах програмування. Зробити висновки.

Рекомендована література: [2], [3], [5], [6] , [7], [8]