Лабораторна робота № 11_12

Тема : Програмування з використанням підпрограм та модулів користувача

Мета роботи:

1. Навчитись визначатись з функціями (процедурами) користувача у задачі та складати програми з їх використанням.

2. Навчитись створювати модуль користувача та використовувати різні його елементи (типи, змінні, підпрограми) у програмі.

ПЛАН

1. Ознайомитись з технологією програмування з використанням функцій користувача.
2. Ознайомитись з технологією програмування з використанням процедур користувача.
3. Ознайомитись з технологією програмування з використанням модулів користувача.
4. Виконати завдання з використанням підпрограми-функції.
5. Виконати завдання з використанням підпрограми-процедури.
6. Створити власний модуль користувача.
7. Дати відповіді на контрольні запитання.
8. Оформити звіт.

ТЕОРЕТИЧНІ ВІДОМОСТІ

1. Програмування з використанням функцій

При програмуванні під функцією слід розуміти програму, яка перетворює множину значень аргументів у єдиний кінцевий результат. Результатом роботи функції – є скаляр. Функція найчастіше входить до складу оператору в якості операнду.

Слід відрізняти **стандартні функції і функції користувача**:

- перші входять до стандартного набору функцій, визначених програмним середовищем мови програмування,
- другі визначаються користувачем за його власним бажанням.

Стандартні функції, що входять до бібліотечного файлу turbo.tpl:

ABS(x) - абсолютне значення числа,
SQR(x) - піднесення числа до квадрату,
SIN(x) - синус,
COS(x) - косинус,

ARCTAN(x) - арктангенс,
EXP(x) - експонента,
LN(x) - логарифм натуральний,
SQRT(x) - визначення кореня квадратного,
A DIV B - визначення цілої частини при ділені A на B,
A MOD B - визначення залишка при ділені A на B,
TRUNC(X) - визначення цілої частини X,
ROUND(X) - округлення x в сторону ближнього цілого
ODD(X) - визначення парності (false) чи непарності (true),
ORD(x) - визначення порядкового номера елемента,
CHR(X) - визначення символа по його порядковому номеру,
SUCC(X) - знаходження наступного елемента,
PRED(X) - знаходження попереднього елемента,

Функції та процедури для роботи з **множинами**;
Функції та процедури для роботи з **рядками**;
Функції та процедури для роботи з **динамічними змінними**;
Функції та процедури для роботи з **файлами**;
Функції та процедури для роботи з **консоллю**;
Функції та процедури **графіки** описані у графічній бібліотеці ***Graph.tpu***.

Функція користувача:

Функція оформляється як окрема частина програми (допоміжний алгоритм у програмі) у формі

FUNCTION *<ім'я> (<спис. формал. параметрів>: <тип>):<тип результату>;*
¦ розділ визначення локалиних параметрів ¦
BEGIN
¦ розділ операторів додаткового алгоритму ¦
END;

Звернення до функції здійснюється по її імені з передачею значень вхідних фактичних параметрів.

Приклад : Для масива цілих чисел Ai, де i=1,2,...m знайти суму елементів від 1-го по 12-й і від 8 по 15 та добуток цих сум.

```
PRORAM PR2;
CONST m=15;
VAR a:ARRAY [1..m] OF REAL; p,j:INTEGER;
  FUNCTION summa(n,r:INTEGER):INTEGER;
   VAR i,s:INTEGER;
   BEGIN s:=0;
      FOR i:=n TO k DO S:=S+A[I];
      summa:=s;
   END;
BEGIN
  WRITELN('Введіть значення елементів масива A');
  FOR j:=1 TO m DO READ (a[j]);
```

```
   p:=summa(1,12)*summ(8,15);
   WRITELN; WRITELN('Добуток P=',P:8);
END.
```

<Перейти до плану>

2. **Програмування з використанням процедур**.

Підпрограмою називають частина програми, до якої можна звернутися із різних частин основної програми. При формуванні підпрограми бажано дотриму-ватись незалежності ідентифікаторів змінних відносно змінних основної програми.

При програмуванні мовою Паскаль підпрограма формується як процедура:

Procedure *<ім'я> (<список формальних параметрів>);*
¦ розділ опису констант, змінних, типів ¦
begin
¦ розділ операторів ¦
end;

Звертання до процедури виконується через її ім'я з переліком, у дужках, значень фактичних параметрів:

< ***ім'я>*** *(фактичні параметри);*

За допомогою формальних і фактичних параметрів дані передаються із програми у процедуру і навпаки. Формальний параметр вказується зі своїм типом, фактичні параметри - без типу.

Між формальними і фактичними параметрами повинна бути встановлена відповідність по їх кількості, типу, порядку слідування.

Приклад:

```
PROCEDURE SUM(a,b,c:REAL; VAR x,y:REAL);
   VAR  z:REAL;
   BEGIN
   z:=a+b+c;  x:=sqr(z); y:=sqrt(z);
   END;
```

Звернення до цієї процедури із основної програми може бути таким:

SUM(25.4,44.6,30,X,Y); *або* ***SUM(10.1,D,R+20.1,Z,T);***

Процедура повинна описуватись на початку основної програми.

Приклад : знайти суму та добуток перших n цілих чисел.

```
PROGRAM P9;
VAR n:INTEGER;
   sum,pr:REAL;
```

```
    PROCEDURE SUMA(k:INTEGER; VAR x,y:REAL);
    VAR i:INTEGER;
        BEGIN x:=0; y:=1;
        FOR i:=1 TO k DO BEGIN x:=x+i; y:=y*i; END;
        END;
{ Основна програма }
BEGIN  WRITELN('Введіть значення n'); READ(n);
     SUMA(n,sum,pr); { Звернення до процедури }
     WRITELN('Сума sum=",sum:5);
     WRITELN('Добуток pr=',pr:5);
END.
```

Результати роботи програми:

```
----------------------
¦ Введіть значення n 10 ¦
¦ Сума  sum=55          ¦
¦ Добуток pr=36288      ¦
```

<Перейти до плану>

3. Модуль користувача поділяється на декілька розділів:

- 1.Заголовок модуля;
- 2.Інтерфейсна частина;
- 3.Реалізаційна частина;
- 4.Ініціалізаційна частина.

ЗАГОЛОВОК МОДУЛЯ

UNIT Им'я модуля;
{$N+} Глобальні директиви компілятора;

ІНТЕРФЕЙСНАЯ ЧАСТИНА

INTERFACE Початок розділу об'яв;
USES Використаємі при об'яві модуля;
LABEL Підрозділ об'яви доступних глобальних міток;
CONST Підрозділ об'яви доступних глобальних констант;
TYPE Підрозділ об'яви доступних глобальних типів;
VAR Підрозділ об'яви доступних глобальних перемінних;
PROCEDURE Заголовки доступних процедур;
FUNCTION Заголовки доступних функцій;

РЕАЛІЗАЦІЙНА ЧАСТИНА

IMPLEMENTATION Початок розділа реалізації;
USES Модулі, які використовуються при реалізації
LABEL Підрозділ об'яви скритих глобальних міток;
CONST Підрозділ об'яви скритих глобальних констант;
TYPE Підрозділ об'яви скритих глобальних типів;
VAR Підрозділ об'яви скритих глобальних змінних;
PROCEDURE Тіла доступних і скритих процедур;

FUNCTION Тіла доступних і скритих функцій;

ІНІЦІАЛІЗАЦІЙНА ЧАСТИНА

BEGIN
Основной блок модуля;
END.

Ім'я модулю надається згідно за правилом надавання імені файлу MS DOS.

На диск записується текст модуля(на мові Turbo Paskal) в файл з ім'ям таким же, як ім'я модуля **UNIT** та поширеням ***.PAS.**

При компіляції (трансляція об'єктного файлу повинна направлятись на диск Compile\D\Disk) тип файлу надається компілятором: **<ім'я модуля> .TPU,** <ім'я модуля> береться із заголовка **UNIT.**

В програму, яка використовує об'єкти модуля, модуль приєднується за допомогою зарезервованого слова **USES<ім'я модуля>;**

При трансляції програми , яка використовує модуль компілятор:

1) Продивляється зміст системної бібліотеки **Turbo.TPL**;
2) Якщо не знайдено файл **<ім'я модуля>.TPU** , продовжується пошук цього файлу в поточному катологзі;
3) Якщо не знайдено, то пошук ведеться в альтернативі **Options/Directory/Unit Directory;**
4) Якщо не знайдено - повідомляє про помилку;
5) Якщо компілятор активізується не за допомогою ALT+F9,а за допомогою **F9 (F10,C/make, або F10,C/Build**, то пошук відбувається також і в текстових файлах **<ім'я модуля>.PAS**, які транслюються перед трансляцією самої програми (відмінність див в лекції).

Приклад модуля

```
UNIT Cmplx;
INTERFACE
  type complex = record
     re, im:real
  end;
  Procedure AddC  (x, y: complex;  var z: complex) ;
  Procedure SubC  (x, y: complex;  var z: complex) ;
  Procedure MulC  (x, y: complex;  var z: complex) ;
  Procedure DivC  (x, y: complex;  var z: complex) ;
  const          c : complex =(re :0.1; im :    -1);
IMPLEMENTATION
   Procedure AddC;
   begin
        z.re := x.re + y.re; z . im := x.im + y. im
```

```
   end {AddC};
   Procedure SubC;
   begin
       z.re := x.re - y. re ;
       z.im := x.im - y.im
   end {SubC};
   Procedure MulC;
   begin
       z.re := x.re*y.re - x.im*y. im;
       z.im := x.re*y.im + x.im*y.re
   end {MulC};
   Procedure DivC;
   var
   zz : real;
   begin
       zz := sqr(y.re) + sqr(y.im);
       z. re := (x.re * y.re + x.im * y.im) / zz;
       z.im := (x.re * y.im - x.im * y.re) / zz
   end {DivC};
end.
```

Текст цього модуля варто помістити у файл CMPLX.PAS. Його можна відкомпілювати (ALT+F9), створивши TPU-файл, після чого Вашій програмі стануть доступні процедури з нової бібліотеки. Наприклад, у наступній програмі здійснюються чотири арифметичні операції над парою комплексних чисел.

```
Uses Cmplx;
var
   a, b, c : complex;
begin
   a.re := 1; a.im := 1;
   b.re := 1; b.im := 2;
   AddC(a, b, c);
   WriteLn('Сложение: 'c.re:5:1, c.im:5:1,'i') ;
   SubC(a, b, c) ;
   WriteLn('Вычитание: 'c.re:5:1, c.im:5:1,'i');
   MulC(a, b, c);
   WriteLn('Умножение: 'c.re:5:1, c.im:5:l,'i') ;
   DivC(a, b, c);
   WriteLn('Деление: 'c.re:5:l, c.im:5:1,'i');
end.
```

Після оголошення Uses Cmplx програмі стали доступні всі об'єкти, оголошені в інтерфейсній частині модуля CMPLX. При необхідності можна перевизначити будь-який з цих об'єктів, як це відбулося, наприклад, з

оголошеною в модулі типізованою константою *C*. Перевизначення об'єкта означає, що знову оголошений об'єкт «закриває» раніше визначений у модулі однойменний об'єкт. Щоб одержати доступ до «закритого» об'єкту, потрібно скористатися складеним ім'ям: перед ім'ям об'єкта поставити ім'я модуля і крапку. Наприклад, оператор

WriteLn(cmplx.c.re:5:l, cmplx.c.im:5:1,'i');

виведе на екран уміст «закритої» типізованої константи з розглянутого прикладу.

<Перейти до плану>

Завдання до лабораторної роботи

Завдання А.

1. Виконати на ЕОМ програму, що використовує підпрограму-функцію, відповідно до номера параметрів, зазначених у таблиці.
2. Перевірити правильність виконання програми за допомогою тестового варіанту.

Вар-т Завд.	Умови задачі	Примітки
1	Обчислити великі корені кв. рівнянь x^2-ax+b=0 cy^2-dy-f=0	Усі корені дійсні
2	Підрахувати число точок, що знаходяться усередині кола радіусом r з центром на початку координат; координати задані масивами X(100), Y(100)	Відстань точки від початку координат обчислювати в підпрограмі
3	➢ Визначити периметри енкутників, координати їх вершин задані в масивах X, Y.	Довжину сторони енкутників обчислювати в підпрограмі
4	Підрахувати число точок, що знаходяться усередині кола радіусом r з центром у точці з координатами (1,1); координати задані масивами X(80), Y(80)	Відстань точки від центра кола визначати в підпрограмі
5	Обчислити z=(v_1+v_2+v_3)/3 , де v_1 ,v_2,v_3, - об’єми куль з радіусами r_1 , r_2 і r_3 відповідно	V_i обчислювати в підпрограмі
6	Обчислити суми позитивних елементів масивів X(N), Y(M), Z(K)	N<=60 M<=60 K<=70
7	Обчислити середнє арифметичне позитивних елементів для масивів A(N1), B(N2), C(N3)	N1<=100 N2<=100 N3<=100
8	Підрахувати кількість елементів матриць X(10,15) і Y(20,12), що задовольняють	

	умовам $0<=x_{ij}<=1$ і $0<=y_{ij}<=1$	
9	Обчислити суми позитивних елементів кожного рядка для матриць A(10,12) і B(15,10)	
10	Обчислити $z=(x_{m1}+x_{m2})/2$, де x_{m1} і x_{m2} – найменші елементи масивів X1(70), X2(80)	
11	Обчислити суми елементів головних діагоналей матриць A(N,N) B(M,M)	M<=20 N<=30
12	Обчислити $z=(s_1+s_2)/2$, де s_1- сума позитивних елементів масиву X(50); s_2- сума негативних елементів масиву Y(60)	Обидві суми обчислювати в одній підпрограмі
13	Підрахувати число нульових елементів для матриць A(N,M) і B(M,N)	M<=20 N<=20
14	Обчислити суми елементів нижніх трикутних матриць для матриць A(15,15) і B(20,20)	
15	Визначити число позитивних елементів до першого негативного в масивах X(40), Y(50), Z(N)	N<=50

<Перейти до плану>

Завдання Б.

1. Виконати на ЕОМ програму, що використовує підпрограму-процедуру відповідно до номерів, зазначених у таблиці.
2. Перевірити правильність виконання програми за допомогою тестового варіанту.

Варіант Завдання	Умова задачі	Примітки
1	Обчислити $z = (s_1+s_2)/k_1k_2$,де s_1 і k_1 – сума і кількість позитивних елементів масиву X(N); s_2 і k_2 – сума і кількість позитивних елементів масиву Y(M).	M<=100 N<=100
2	Обчислити $\mathbf{z = (e^{s1}+e^{s2})/k_1k_2}$, де s_1 і k_1 – сума і кількість позитивних елементів масиву X(100); s_2 і k_2 – сума і кількість позитивних елементів масиву Y(80).	Обидві суми обчислювати в одній підпрограмі
3	Обчислити і запам’ятати суми позитивних елементів кожного рядка A(10,20), B(15,10).	
4	Обчислити $z = (x_1+y_1)/(x_2-y_2)$, де x_1 і x_2 – корені рівняння $2x^2+x-4 = 0$, y_1 і y_2 – корені рівняння $ay^2+2y-1 = 0$.	Усі корені дійсні
5	Знайти найбільші елементи і їхні порядкові	N<=80 M<=70

	номери масивів X(N) і Y(M)	
6	Переписати позитивні елементи масиву X(100) і Y(80) у масив Z підряд	Запис у масив Z здійснювати в підпрограмі
7	Знайти найменші елементи і номери рядків і стовпців, у яких вони розташовані, для матриць A(10,15) і B(15,12)	
8	Вивести на друк елементи **цілочисельних** матриць N(5,8) і M(10,6), кратні трьом	
9	Обчислити z, де x_i і y_i – задані масивами $$z = \frac{\sum_{j=1}^{40} \sin x_i + \sum_{i=1}^{50} \cos y_i}{\sum_{i=1}^{40} \lvert x_1 \rvert}$$	Усі суми обчислювати в одній підпрограмі
10	Обчислити $z = (x_{max}-y_{min})/2$, де x_{max} – максимальний елемент масиву X(50); y_{min} – мінімальний елемент масиву Y(40)	x_{max} і y_{min} обчислювати в одній підпрограмі
11	Обчислити і запам'ятати кількість негативних елементів кожного стовпця для матриць A(10,10), B(15,20)	
12	Обчислити суми елементів верхньої трикутної матриці для матриць A(10,10), B(15,15)	
13	Знайти середні значення і стандартні відхилення для елементів масивів X(N), Y(M)	N<=100 M<=100
14	Обчислити суми і кількості елементів, що знаходяться в інтервалі від a до b для матриць X(10,8) і Y(10,12)	
15	Перетворити масиви X(50) і Y(60), розташувавши в них підряд тільки позитивні елементи. Замість інших елементів записати нулі	

<Перейти до плану>

Завдання В

1) Створити модуль користувача **UNIT** , до складу якого включити як мінімум 2 типи користувача, функцію (задача А) та процедуру (задача Б) користувача .

Створити одну, чи де-кілька програм, в яких скористатись типами, функцією та процедурою створеного модуля.

о

Порядок створення модуля

ЧАСТИНА 1

1. В редакторі TurboPaskal створити текстовий файл згідно структурі (див. структура ТР модуля) при цьому ім'я модуля повинно бути обов'язково (згідно з правилами надання імені файлу в MS DOS) наприклад **Unit My_lib**; (В інтерфейсну частину модуля помістити опис 2-х типів користувача, та заголовки процедури користувача та функції користувача. Тексти підпрограм описати в реалізаційній частині).
2. . Записати текст модуля в файл з ім'ям <ім'я модуля>.PAS. Наприклад **S2/ My_Lib.PAS**
3. Встановити компіляцію на диск **F10,C/ Distination/ Disk**.
4. Відкомпілювати програму **Alt+F9 ;** або **F9**.

ЧАСТИНА 2

1. В редакторі створити програму в якій використати об'єкти модуля Unit, **Program US_LIB**. Наприклад **USES MY_LIB**;
2. Записати програму на диск **F2/ US_LIB.PAS**.;
3. Відтранслювати програму **F9** ;
4. Запустити програму на виконання, переписати отримані результати.
5. Продемонструвати програму викладачеві.

<Перейти до плану>

Контрольні запитання

1. Що таке підпрограма? Вказати, за яких умов доцільне використання підпрограм, які переваги вони надають користувачеві.
2. Які види підпрограм ви знаєте? Що являє собою функція? Що являє собою процедура?
3. Навести приклади стандартних процедур і функцій?
4. Вказати, у чому відмінність процедур від функцій користувача.
5. Вказати способи звертання до процедур і функцій користувача.
6. Що таке глобальні та локальні параметри? Різниця між ними.
7. Вказати способи передачі параметрів у підпрограму.
8. Вказати, як організовувати підпрограму без параметрів.
9. Перелічити, як погоджуються формальні і фактичні параметри.
10. Вказати конструкції, що можуть бути формальними і фактичними параметрами.
11. Коли використовується в описі параметрів процедури службове слово VAR?
12. Пояснити принципову різницю між параметрами –значеннями та параметрами–змінними.
13. Пояснити, як і куди здійснюється вихід з підпрограми.
14. Чим відрізняється модуль від звичайної програми ?

15. Коли доречно застосовувати модулі та включати до них процедури або функції?
16. Що мається на увазі під блочною структурою програми?
17. Які позитивні якості надає програмі використання модулів користувача?
18. Чим відрізняється компіляція програми з використанням модуля користувача за допомогою команд <ALT><F9> від <F9> та F10,C/Build?
19. Яким чином система знаходить місце зберігання файлу un.tpu під час його підключення до програми?
20. Що вміщує розділ INTERFACE ?
21. Що вміщує розділ Implementation ?
22. Чим відрізняються наприклад змінні або типи даних, які описані у розділі Interface від відповідних елементів, описаних у розділі Implementation?
23. Які функції під час використання модуля виконують елементи, які розміщені у ініціалізаційній його частині?
24. Чи може ім'я модуля UNIT складатись з 10 символів ?

<Перейти до плану>

Вимоги до оформлення звіту

1. Титульний лист : № лаб. роботи, назва теми роботи , Мета роботи, завдання до роботи, ПІПб виконавця та № варіанту, дата виконання та дата оформлення звіту.
2. Формалізація змісту задачі та Блок-схема алгоритму;
3. Листинг програми (роздруківка)
4. Протокол виконання програми та висновки.
5. Письмові відповіді на запитання по формі:

№ Запитання , запитання	Коротка, але змістовна відповідь

Власний підпис