МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

Лабораторна робота №7

з дисципліни «Основи програмування»

на тему: *«Робота з бібліотеками»*

Виконав:

студент гр.ІО-31

Долинний О.В.

Перевірив:

Саверченко В.Г.

Київ 2013

1. ***Теоретичні відомості.***

Модуль — це оформлена за деякими правилами сукупність визначень типів, констант, змінних та підпрограм. Спочатку складається текст модуля, потім він компілюється і записується на диск у вигляді tpu-файла. Такий файл неможливо викона­ти, як самостійну програму. Його використовують, як допоміжний файл у процесі розробки головної програми.

Виділяють два напрямки вживання модулів. Перший – зв’язаний з формуванням бібліотек під­програм повністю готових до роботи. Це зручно в рамках будь-якої технології програмування. Достатньо підключити модуль до програми і можна використовувати його елементи в цій програмі.

Другий напрямок - конструювання програм великого розміру. В зв'язку з сегментною організацією пам'яті персональних ЕОМ існують обмеження на розмір програми (не більше як 64К). Очевидно, що цього замало для розв'язання реальних виробничих задач.

Для виконання програми на Pascal в оперативній пам'яті комп’ютера виділяється місце після пам'яті зайнятої системними програмами MS-DOS. Вся доступна пам'ять розподіляється на сегменти, в кожному з яких розміщується певна інфор­мація. В кодовому сегменті розміщена головна про­грама зі своїми підпрограмами. Всі дані, описані в головній програмі, займають окремий сегмент да­них. Кожен з модулів, що підключається, також займає один сегмент (незалежно від того, чи є цe модуль системної бібліотеки Turbo.tpl чи особистої біб­ліотеки).

На етапі проектування великої програми треба розподілити всі підпрограми між головною програмою та модулями. Частіше в модуль об’єднують логічно зв'язані підпрограми, наприклад, всі додаткові процедури та функції для роботи з рядками.

***Структура модуля***

***Unit****ім'я;  {заголовок модуля}*

***Interface***

*{Інтерфейсна частина}*

***Implementation***

*{Тексти підпрограм}*

***Begin****{Блок ініціалізації модуля{*

***End.***

Ім'я модуля вибирається довільно, але воно по­винно бути унікальним. Зразу після заголовка розміщують слово Interface. В інтерфейсній частині модуля розміщують оператор Uses, оператори описування глобальних типів, констант, змінних, які будуть передані в програму при підключенні модуля. Також, записують заголовки всіх підпрограм, які входять до модулю.

У розділі Implementation приводяться тексти всіх підпрограм у тій послідовності, в якій вони були описані в розділі Interface. В заголовках підпрограм можна не вказувати списки параметрів. Перед підпрограмами можуть бути описані локальні типи та змінні для свого блоку ініціалізації.

В блок ініціалізації звичайно включають операто­ри визначення початкових значень, відкриття файлів, перевірки їх наявності тощо. Цей блок може бути пустим, у такому разі не пишуть словоBegin.

Оператори з блоків ініціалізації всіх модулів вико­нуються на початку роботи головної програми.

1. ***Завдання роботи***

**Варіант 11**

**Текст основної програми Exec(a,n,min)**

**Min – процедура пошуку мінімального елементу в векторі a з розміром n**

1. ***Програмний код основної програми***

program lab8;

uses

Unit1;

var

a:unit1.Tvector;

n:unit1.TIndex;

min:unit1.minproc;

begin

exec(a,n,min);

end.

1. ***Програмний код модуля***

unit unit1;

interface

type

TIndex=1..10;

TVector=array[tindex] of integer;

minproc=procedure(n1:TIndex; a1:Tvector;var m1:integer) ;

procedure Input(var n0:TIndex; var a0:Tvector);

procedure Output(n0:TIndex;a0:Tvector);

procedure min0(n0:TIndex; a0:Tvector;var m0:integer);

procedure exec(a1:TVector;n1:TIndex;min1:minproc);

implementation

procedure Input(var n0:TIndex; var a0:Tvector);

var i0:TIndex;

begin

write('input the size of array ');

readln(n0);

write('input the array ');

for i0:=1 to n0 do

read(a0[i0]);

end;

procedure Output(n0:TIndex;a0:Tvector);

var i0:TIndex;

begin

for i0:=1 to n0 do

write(' ',a0[i0]);

end;

procedure min0(n0:TIndex; a0:Tvector;var m0:integer);

var i0:TIndex;

begin

m0:=a0[1];

for i0:=2 to n0 do

if a0[i0]<m0 then

m0:=a0[i0];

writeln;

writeln('min=',m0);

end;

procedure exec(a1:TVector;n1:TIndex;min1:minproc);

var m1:integer;

i0:Tindex;

begin

input(n1,a1);

output(n1,a1);

min1:=min0;

min1(n1,a1,m1);

end;

end.

1. ***Висновки.***

За допомогою підключення модуля було зменшено розмір програмного коду основної програми.

Модулі є універсальними і можуть бути підключені до будь-якої програми.