**ОТЧЕТ**

по лабораторной работе №4

ПРОГРАММИРОВАНИЕ МЕТОДОВ ОБРАБОТКИ ОДНОМЕРНЫХ МАССИВОВ НА ЯЗЫКЕ PASCAL

Выполнил: студент группы ИВТ/б-12о

Безверхний Вадим

Проверил: ст.преподаватель кафедры ИУТС

Захаров В.В.

1. Цель работы

Исследование циклических алгоритмов и программ, осуществляющих типичные операции над одномерными массивами. Приобретение навыков программирования и вывода массивов.

1. Задание на работу

Создать программу, реализующую обработку массива вещественных чисел с использованием двух подпрограмм: процедуры и функции. Исходные данные для варианта №8:

Таблица 1

|  |  |  |
| --- | --- | --- |
| Задания и значения исходных данных | | |
| Набор значений элементов массива и дополнительные исходные данные | Функция должна вычислять | Процедура должна выводить на экран элементы, удовлетворяющие условию |
| X = 104; -0.1\*10-4; 0.6\*10-4; -1; -10.1.  B = 3\*10-4. | Номер макси- мального элемента массива |  |

Для использования заданного массива в виде параметра процедуры и функции, создадим пользовательский тип данных – массив значений типа real с индексами в диапазоне от 1 до 5 (количество элементов массива):

**Type** massive = **array**[1..n] **of** real;

1. Схемы программ
   1. Схема основной программы

Начало

Ввод a; n

C, i от 1 до n

n – количество элементов массива

Ввод m[i]

maxim

good\_element

(a, n, m)

Конец

C, i = i + 1

Рисунок 1 – Блок-схема основной программы

* 1. Схема подпрограммы-функции, вычисляющей максимальный элемент массива

maxim

max🡨massive[1]

A, i🡨2, i<=size

Да

massiv[i] > max

Нет

max massiv[i]

A, i+1

maxim max

Конец

Рисунок 2 – Блок-схема подпрограммы-функции

* 1. Схема подпрограммы-процедуры, выводящей на экран все элементы массива удовлетворяющие условие

A, i+1

Вывод

A, i←1, i≤n

Начало

Да

Нет

Конец

Рисунок 3 – Блок-схема подпрограммы-процедуры

1. Текст программы

**program** lr4;

**const**

n=5;

**type**

massive = **array**[1..n] **of** real;

**var**

a,max:real;

i:byte;

m:massive;

**procedure** good\_element(b: real; size: byte;**const** massiv: massive);

**var**

i: byte;

**begin**

writeln('Rabota proceduri ');

**for** i:= 1 **to** size **do**

**if** abs(massiv[i]) > b **then** writeln(massiv[i]);

**end**;

**function** maxim(size: byte;**const** massiv: massive): real;

**var**

i: byte;

max: real;

**begin**

max:= massiv[1];

**for** i:= 2 **to** size **do**

**if** massiv[i] > max **then**

max := massiv[i];

maxim := max;

**end**;

**begin**

writeln('Parametr '); readln(a);

writeln('Elemets of massiv ');

**for** i := 1 **to** n **do**

read(m[i]);

max := maxim(n,m);

writeln('Rabota function ',max);

good\_element(a,n,m);

**end**.

5.Результаты работы программы

Программа дает следующий результат:

* результат работы функции: 1000;
* результат работы процедуры:1000, -1, -10.1;

6.Вывод

Использование функции оправдано в связи с необходимостью вернуть в программу единственное значение, в данном случае – вещественное число. Процедуры используются из-за необходимости выборки из массива и вывода на экран нескольких чисел-элементов массива.

Чтобы осуществить ввод массива в функцию и процедуру в виде параметра, был описан пользовательский тип данных – массив вещественных элементов с индексами от 1 до n (n – константа).

Разработанные подпрограммы успешно выполняют заданную задачу:

* Максимальный элемент массива: 1000;
* элементы, удовлетворяющие условию 1000, -1, -10.1.