*Федеральное государственное бюджетное образовательное учреждение высшего профессионального образования*

|  |  |  |  |
| --- | --- | --- | --- |
| Gerb-BMSTU_01 | **«Московский государственный технический университет имени Н.Э. Баумана»** | |  |
| Факультет: | | ИУ «Информатика и системы управления» |
| Кафедра: | | ИУ-6 «Компьютерные системы и сети» |

**Отчет**

**по домашнему заданию № 9**

**«Подпрограммы. Средства отладки Delphi.»**

вариант №4

**Дисциплина: Основы программирования**

|  |  |  |
| --- | --- | --- |
| Студент группы ИУ6-51 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Милков В. В. |
|  | (Подпись, дата) | (Фамилия И.О.) |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Гукетлев Д. В. |
|  | (Подпись, дата) | (Фамилия И.О.) |

*Москва, 2016*

**Содержание**

Задание 3

Блок-схема алгоритма 4

[Код программы 5](#h.3znysh7)

[Скриншоты выполнения программы 7](#h.2et92p0)

[Вывод 8](#h.3dy6vkm)

**Задание**

Решить задачу, используя процедуру или функцию. Выбор обосновать. На примере полученной программы продемонстрировать умение:

1) назначать точку останова;

2) пошаговое выполнение программы с заходом в процедуры и без захода;

3) определять значения переменных.

Дана матрица A(10,10). Преобразовать ее, упорядочив каждую строку по неубыванию элементов.

**Блок-схема алгоритма**



**Код программы**

**program main;**

**const DIMENSION = 10;**

**procedure bubblesort(var numbers: array of integer);**

**var**

**i, j, tmp: integer;**

**begin**

**for i := length(numbers) - 1 downto 1 do**

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

**if (numbers[j-1] > numbers[j]) then begin**

**tmp := numbers[j-1];**

**numbers[j-1] := numbers[j];**

**numbers[j] := tmp;**

**end;**

**end;**

**var**

**i, j: integer;**

**matrix: array[1..DIMENSION] of array[1..DIMENSION] of integer;**

**begin**

**for i:=1 to length(matrix) - 1 do begin**

**for j:=1 to length(matrix[i]) do begin**

**matrix[i][j] := random(10);**

**end;**

**end;**

**writeln('The incoming matrix:'); writeln;**

**for i:=1 to length(matrix) - 1 do begin**

**for j:=1 to length(matrix[i]) do begin**

**write(matrix[i][j], ' ');**

**end;**

**writeln;**

**end;**

**writeln;**

**writeln('↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓');**

**writeln;**

**for i:=1 to length(matrix) - 1 do begin**

**bubblesort(matrix[i]);**

**for j:=1 to length(matrix[i]) do begin**

**write(matrix[i][j], ' ');**

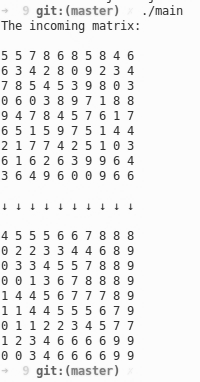
**end;**

**writeln;**

**end;**

**end.**

**Скриншоты выполнения программы**



**Вывод**

В языке Паскаль, как и в большинстве языков программирования, предусмотрены средства, позволяющие оформлять вспомогательный алгоритм как подпрограмму. Это бывает необходимо тогда, когда какой-либо подалгоритм неоднократно повторяется в программе или имеется возможность использовать некоторые фрагменты уже разработанных ранее алгоритмов. Кроме того, подпрограммы применяются для разбиения крупных программ на отдельные смысловые части в соответствии с модульным принципом в программировании.

Для использования подалгоритма в качестве подпрограммы ему необходимо присвоить имя и описать алгоритм по правилам языка Паскаль. В дальнейшем, при необходимости вызвать его в программе, делают вызов подпрограммы упоминанием в нужном месте имени соответствующего подалгоритма со списком входных и выходных данных. Такое упоминание приводит к выполнению входящих в подпрограмму операторов, работающих с указанными данными. После выполнения подпрограммы работа продолжается с той команды, которая непосредственно следует за вызовом подпрограммы.

В языке Паскаль имеется два вида подпрограмм - процедуры и функции.

Процедуры и функции помещаются в раздел описаний программы. Для обмена информацией между процедурами и функциями и другими блоками программы существует механизм входных и выходных параметров. Входными параметрами называют величины, передающиеся из вызывающего блока в подпрограмму (исходные данные для подпрограммы), а выходными - передающиеся из подрограммы в вызывающий блок (результаты работы подпрограммы).

Одна и та же подпрограмма может вызываться неоднократно, выполняя одни и те же действия с разными наборами входных данных. Параметры, использующиеся при записи текста подпрограммы в разделе описаний, называют формальными, а те, что используются при ее вызове - фактическими.

В решении данной задачи используется процедура, на вход которой приходит массив чисел. Эта процедура сортирует элементы этого массива по неубыванию.