**Министерство науки и высшего образования Российской Федерации**

федеральное государственное автономное образовательное учреждение

высшего образования

**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ**

**ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

Инженерная школа природных ресурсов

Направление подготовки Химическая технология

Отделение химической инженерии

**СОСТАВЛЕНИЕ ПРОГРАММ С ИСПОЛЬЗОВАНИЕМ ДВУМЕРНЫХ МАССИВОВ. ФАЙЛЫ**

**Лабораторная работа по дисциплине «Углубленный курс информатики»**

Выполнил студент гр. А.В.Исаева

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Отчет принят:

Преподаватель

доцент ОХИ ИШПР, к.т.н. В.А. Чузлов

(Подпись)

\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ 2020 г.

Томск 2020 г.

**Цель работы:** **составление программ с использованием двумерных массивов**

**Теоретическая часть**

Двумерные массивы Двумерные массивы в Паскале трактуется как одномерный массив, тип элементов которого также является массивом (массив массивов). Положение элементов в двумерных массивах Паскаля описывается двумя индексами. Их можно представить в виде прямоугольной таблицы или матрицы.

Описание двумерного массива Паскаля.

Существует несколько способов объявления двумерного массива Паскаля.

Мы уже умеем описывать одномерные массивы, элементы которых могут иметь любой тип, а, следовательно, и сами элементы могут быть массивами. Рассмотрим следующее описание типов и переменных:

Пример описания двумерного массива Паскаля

Program lab5;

Type

Vector = array [1..5] of <тип\_элементов>;

Matrix= array [1..10] of vector;

Var m: matrix;

Мы объявили двумерный массив Паскаля m, состоящий из 10 строк, в каждой из которых 5 столбцов. При этом к каждой i -й строке можно обращаться m [ i ], а каждому j -му элементу внутри i -й строки – m [ i , j ].

Определение типов для двумерных массивов Паскаля можно задавать и в одной строке:

Program lab5;

Type

Matrix= array [1..5] of array [1..10] of < тип элементов >;

или еще проще:

Program lab5;

type

matrix = array [1..5, 1..10] of <тип элементов>;

Обращение к элементам двумерного массива имеет вид: M [ i , j ]. Это означает, что мы хотим получить элемент, расположенный в i -й строке и j -м столбце. Тут главное не перепутать строки со столбцами, а то мы можем снова получить обращение к несуществующему элементу. Например, обращение к элементу M [10, 5] имеет правильную форму записи, но может вызвать ошибку в работе программы.

**Практическая часть**

**Задание 1**

Заполнить матрицу а(3, 3) случайными числами от -5 до 5. Найти произведение минимального элемента матрицы на сумму ее положительных элементов. Значения элементов матрицы и результат расчета вывести в файл.

**Программная реализация**

**program** LB6\_1;

**var**

i, j: integer;

min, s, p: real;

a: **array** [1..3, 1..3] **of** real;

f: text;

**begin**

assign(f, 'result.txt');

rewrite(f);

p := 0;

**for** i := 1 **to** 3 **do**

**begin**

**for** j := 1 **to** 3 **do**

**begin**

a[i, j] := random(11) - 5;

write(f, a[i, j]:5);

**end**;

writeln(f);

**end**;

min := a[1, 1];

**for** i := 1 **to** 3 **do**

**begin**

**for** j := 1 **to** 3 **do**

**if** a[i, j] < min **then**

**begin**

min := a[i, j];

**end**;

**end**;

s := 0;

**for** i := 1 **to** 3 **do**

**begin**

**for** j := 1 **to** 3 **do**

**if** a[i, j] > 0 **then**

s := s + a[i, j];

**end**;

p := s \* min;

writeln(f, 'Произведение = ', p);

close(f)

**end**.

**Ответ**

-4 -4 0

4 5 2

-4 4 -2

Произведение = -60

**Задание 2**

Заполнить матрицу а(4, 4) случайными целыми числами от   
1 до 100. Вывести значения ее элементов в файл. Найти максимальный элемент в каждой строке. Среди максимальных элементов каждой строки найти минимальный. Результаты вывести в файл.

**Программная реализация**

**program** LB5;

**var**

i, j: integer;

max, min: real;

a: **array** [1..4, 1..4] **of** real;

f: text;

**begin**

assign(f, 'result.txt');

rewrite(f);

**for** i := 1 **to** 4 **do**

**begin**

**for** j := 1 **to** 4 **do**

**begin**

a[i, j] := random(100)+1;

write(f, a[i, j]:5);

**end**;

writeln(f);

**end**;

**for** i := 1 **to** 4 **do**

**begin**

max := a[i, 1];

**for** j := 2 **to** 4 **do**

**if** a[i, j] > max **then**

max := a[i, j];

**if** i = 1 **then** min := max

**else**

**if** max < min **then**

min := max

**end**;

writeln(f);

writeln(f, 'Минимальный среди максимальных ', min);

close(f)

**end**.

**Ответ**

22 47 35 5

67 23 8 69

5 3 64 82

14 49 26 22

Минимальный среди максимальных 47

**Задание 3**

Заполнить матрицу а(5, 10) случайными числами от 0 до 9. Найти столбец матрицы с максимальной суммой элементов. Значения элементов матрицы и результат расчета вывести в файл.

**Программная реализация**

**program** LB6\_3;

**var**

i, j: integer;

s, sm, p: real;

a: **array** [1..5, 1..10] **of** real;

f: text;

**begin**

assign(f, 'result.txt');

rewrite(f);

**for** i := 1 **to** 5 **do**

**begin**

**for** j := 1 **to** 10 **do**

**begin**

a[i, j] := random(10);

write(f, a[i, j]:5);

**end**;

writeln(f);

**end**;

sm := -1;

p := 0;

**for** j := 1 **to** 10 **do**

**begin**

s := 0;

**for** i := 1 **to** 5 **do**

s := s + a[i, j];

**if** s > sm **then**

**begin**

sm := s;

p := j;

**end**;

**end**;

writeln(f, 'Столбец ', p, ' с суммой= ', sm);

close(f)

**end**.

**Ответ**

4 6 2 8 3 6 0 0 6 7

2 8 3 0 1 2 5 2 8 9

7 9 1 1 8 5 0 8 8 7

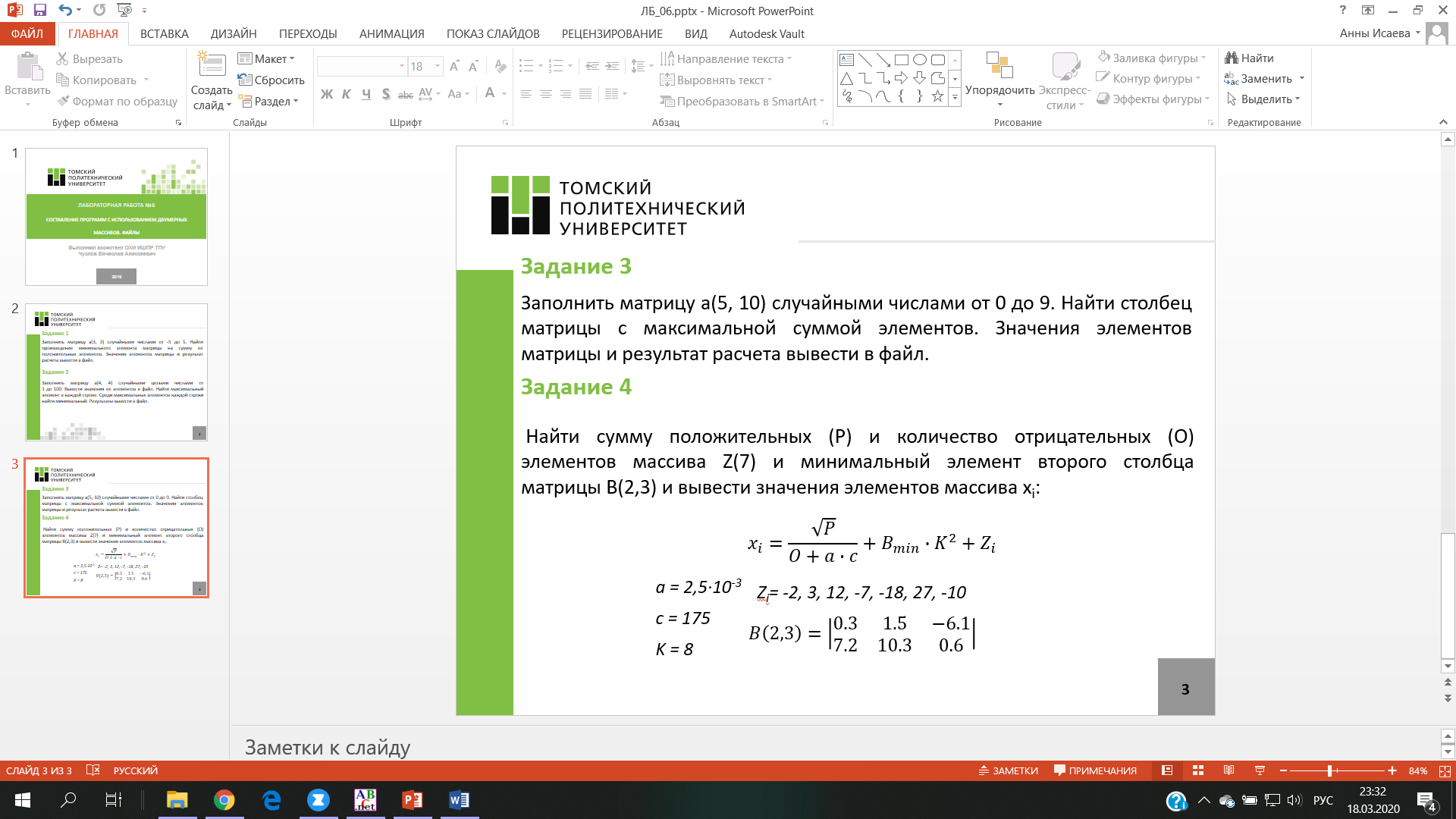
8 4 1 1 7 1 8 1 8 4

4 9 6 2 0 0 6 3 0 7

Столбец 2 с суммой= 36

**Задание 4**

Найти сумму положительных (P) и количество отрицательных (О) элементов массива Z(7) и минимальный элемент второго столбца матрицы В(2,3) и вывести значения элементов массива xi:



**Программная реализация**

**program** LB6\_1;

**const**

a = 2.5 \* exp((-3) \* ln(10));

c = 175;

k = 8;

**var**

b: **array** [1..2, 1..3] **of** real = ((0.3, 1.5, -6.1), (7.2, 10.3, 0.6));

z: **array** [1..7] **of** real = (-2, 3, 12, -7, -18, 27, -10);

x: **array** [1..7] **of** real;

P, O, min: real;

i: integer;

**begin**

P := 0;

O := 0;

min := B[1, 2];

**for** i := 1 **to** 7 **do**

**begin**

**if** z[i] > 0 **then**

P := P + z[i];

**if** z[i] < 0 **then**

O := O + 1;

**end**;

**for** i := 1 **to** 2 **do**

**begin**

**if** B[i, 2] < min **then**

min := B[i, 2];

**end**;

**for** i := 1 **to** 7 **do**

x[i] := (sqrt(P) / (O + a \* c)) + (min \* K \* K) + Z[i];

write('x: [', x[1], ' ', x[2], ' ', x[3], ' ', x[4], ' ', x[5], ' ', x[6], ' ', x[7], ']');

**end**.

**Ответ**

x: [95.4604486080919 100.460448608092 109.460448608092 90.4604486080919 79.4604486080919 124.460448608092 87.4604486080919]

**Выводы**

В ходе работы были составлены программы с использованием двухмерных массивов.