**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | Информационных технологий |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | Информационные системы и технологии |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | Очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

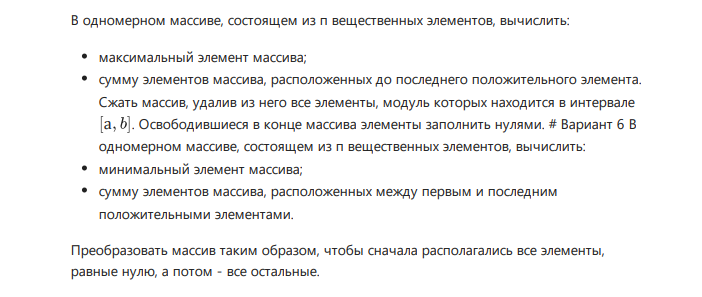
**Отчет по практической работе №5**

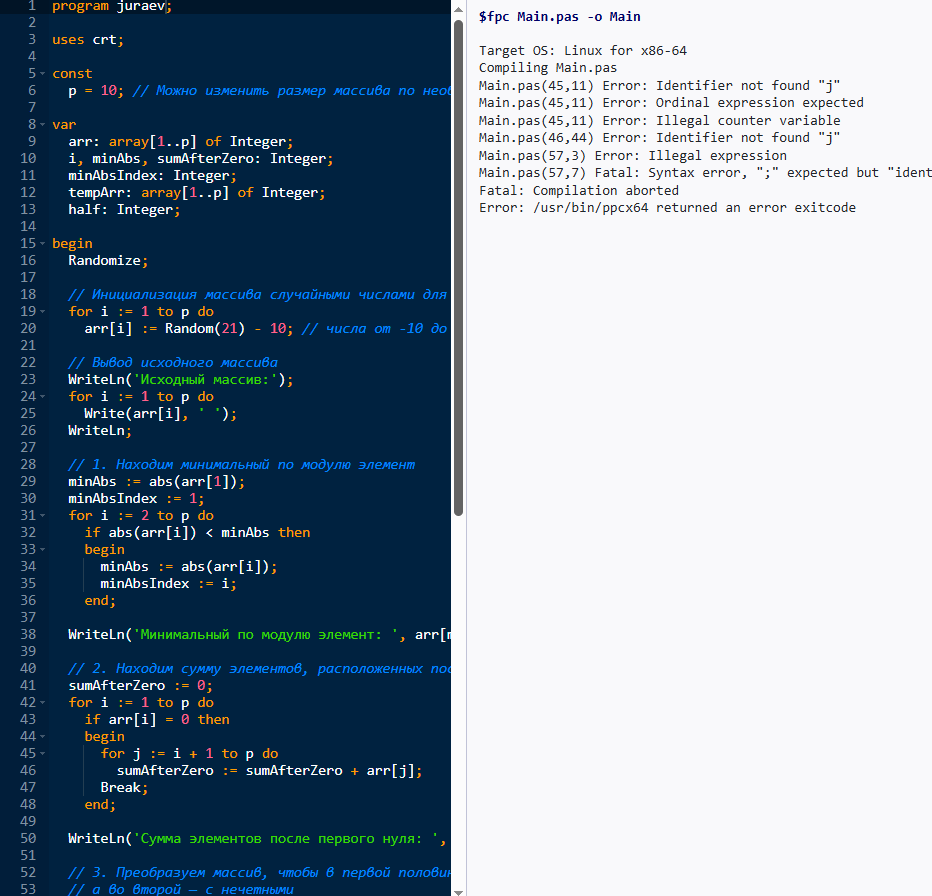
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Одномерные массивы | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | **Разработка программных модулей** |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Джураев Екубшох |  | Не требуется |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | Дкип 111 прог |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев И.В |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г.**





program juraev;

uses crt;

const

p = 10; // Можно изменить размер массива по необходимости

var

arr: array[1..p] of Integer;

i, minAbs, sumAfterZero: Integer;

minAbsIndex: Integer;

tempArr: array[1..p] of Integer;

half: Integer;

begin

Randomize;

// Инициализация массива случайными числами для примера

for i := 1 to p do

arr[i] := Random(21) - 10; // числа от -10 до 10

// Вывод исходного массива

WriteLn('Исходный массив:');

for i := 1 to p do

Write(arr[i], ' ');

WriteLn;

// 1. Находим минимальный по модулю элемент

minAbs := abs(arr[1]);

minAbsIndex := 1;

for i := 2 to p do

if abs(arr[i]) < minAbs then

begin

minAbs := abs(arr[i]);

minAbsIndex := i;

end;

WriteLn('Минимальный по модулю элемент: ', arr[minAbsIndex], ' (позиция ', minAbsIndex, ')');

// 2. Находим сумму элементов, расположенных после первого нулевого элемента

sumAfterZero := 0;

for i := 1 to p do

if arr[i] = 0 then

begin

for j := i + 1 to p do

sumAfterZero := sumAfterZero + arr[j];

Break;

end;

WriteLn('Сумма элементов после первого нуля: ', sumAfterZero);

// 3. Преобразуем массив, чтобы в первой половине были элементы с четными позициями,

// а во второй — с нечетными

half := p div 2;

// Создаем два массива: для четных и нечетных позиций

var evenPosArr: array of Integer;

var oddPosArr: array of Integer;

SetLength(evenPosArr, 0);

SetLength(oddPosArr, 0);

for i := 1 to p do

begin

if (i mod 2) = 0 then

begin

SetLength(evenPosArr, Length(evenPosArr) + 1);

evenPosArr[High(evenPosArr)] := arr[i];

end

else

begin

SetLength(oddPosArr, Length(oddPosArr) + 1);

oddPosArr[High(oddPosArr)] := arr[i];

end;

end;

// Заполняем первую половину массива четными элементами

for i := 1 to half do

if i <= Length(evenPosArr) then

arr[i] := evenPosArr[i - 1];

// Заполняем вторую половину массива нечетными элементами

for i := half + 1 to p do

if (i - half) <= Length(oddPosArr) then

arr[i] := oddPosArr[i - half - 1];

// Вывод преобразованного массива

WriteLn('Преобразованный массив:');

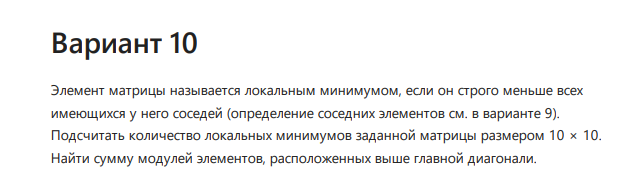
for i := 1 to p do

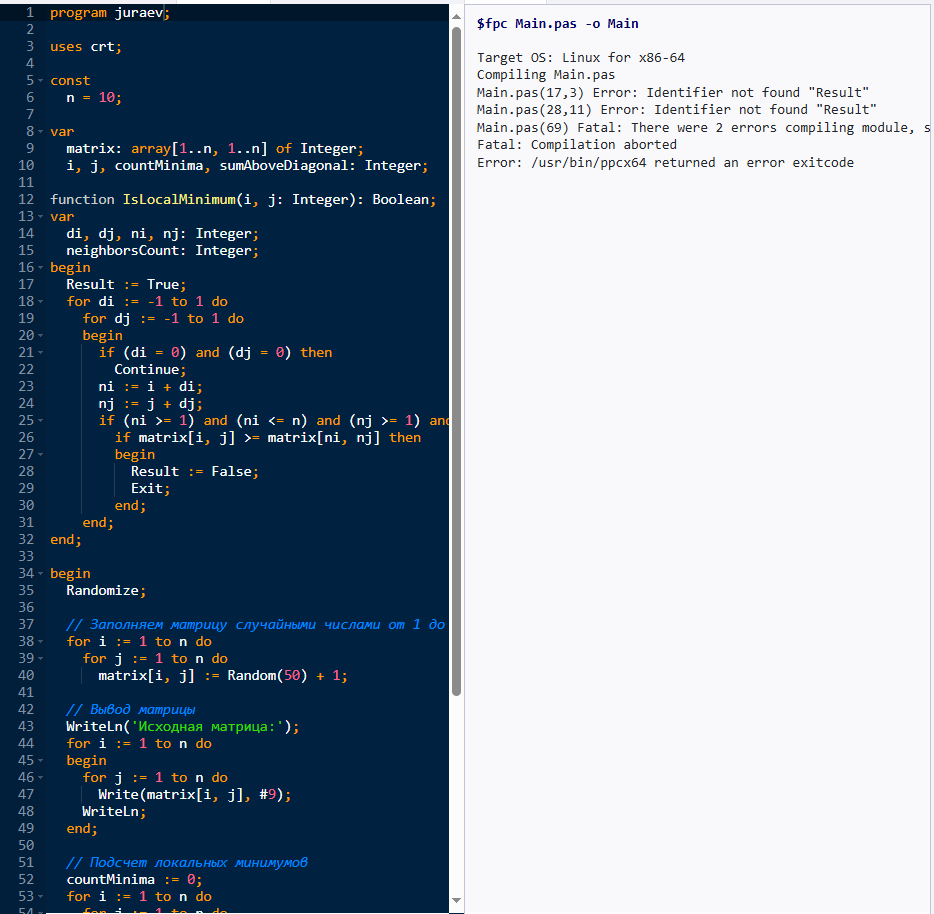
Write(arr[i], ' ');

WriteLn;

end.

**Лаба 6**





program juraev;

uses crt;

const

n = 10;

var

matrix: array[1..n, 1..n] of Integer;

i, j, countMinima, sumAboveDiagonal: Integer;

function IsLocalMinimum(i, j: Integer): Boolean;

var

di, dj, ni, nj: Integer;

neighborsCount: Integer;

begin

Result := True;

for di := -1 to 1 do

for dj := -1 to 1 do

begin

if (di = 0) and (dj = 0) then

Continue;

ni := i + di;

nj := j + dj;

if (ni >= 1) and (ni <= n) and (nj >= 1) and (nj <= n) then

if matrix[i, j] >= matrix[ni, nj] then

begin

Result := False;

Exit;

end;

end;

end;

begin

Randomize;

// Заполняем матрицу случайными числами от 1 до 50

for i := 1 to n do

for j := 1 to n do

matrix[i, j] := Random(50) + 1;

// Вывод матрицы

WriteLn('Исходная матрица:');

for i := 1 to n do

begin

for j := 1 to n do

Write(matrix[i, j], #9);

WriteLn;

end;

// Подсчет локальных минимумов

countMinima := 0;

for i := 1 to n do

for j := 1 to n do

if IsLocalMinimum(i, j) then

Inc(countMinima);

WriteLn('Количество локальных минимумов: ', countMinima);

// Сумма элементов выше главной диагонали

sumAboveDiagonal := 0;

for i := 1 to n do

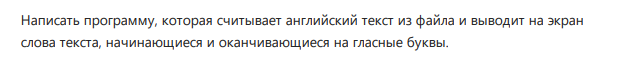
for j := i + 1 to n do

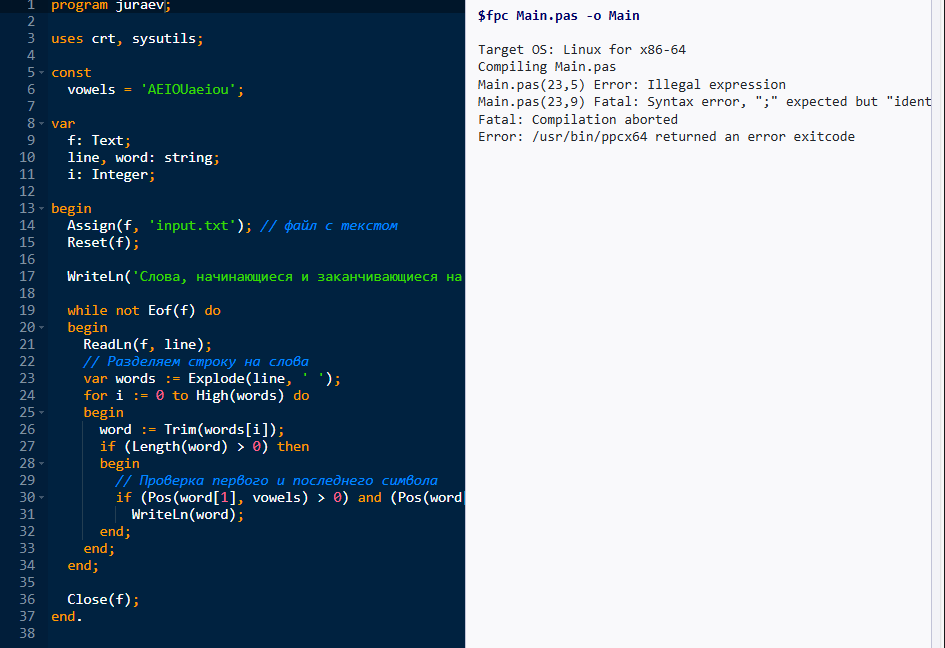
sumAboveDiagonal := sumAboveDiagonal + matrix[i, j];

WriteLn('Сумма элементов выше главной диагонали: ', sumAboveDiagonal);

end.

**Лаба 7**

****

****

**program juraev;**

**uses crt, sysutils;**

**const**

**vowels = 'AEIOUaeiou';**

**var**

**f: Text;**

**line, word: string;**

**i: Integer;**

**begin**

**Assign(f, 'input.txt'); // файл с текстом**

**Reset(f);**

**WriteLn('Слова, начинающиеся и заканчивающиеся на гласные:');**

**while not Eof(f) do**

**begin**

**ReadLn(f, line);**

**// Разделяем строку на слова**

**var words := Explode(line, ' ');**

**for i := 0 to High(words) do**

**begin**

**word := Trim(words[i]);**

**if (Length(word) > 0) then**

**begin**

**// Проверка первого и последнего символа**

**if (Pos(word[1], vowels) > 0) and (Pos(word[Length(word)], vowels) > 0) then**

**WriteLn(word);**

**end;**

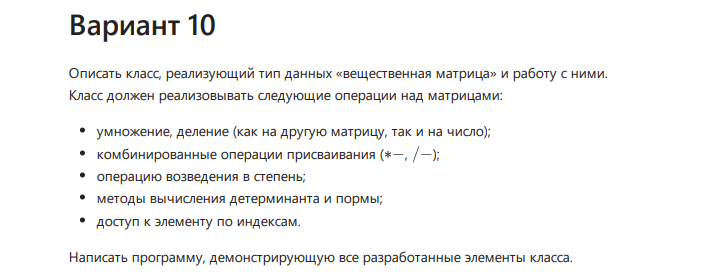
**end;**

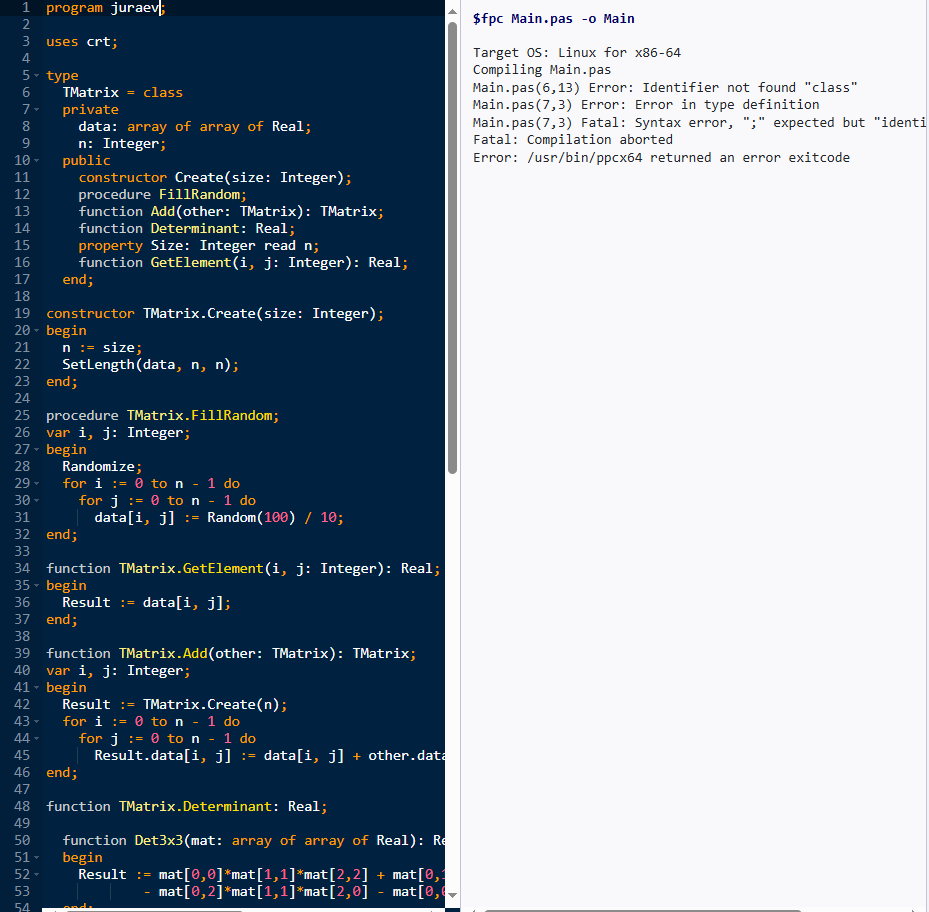
**end;**

**Close(f);**

**end.**

**Лаба 8**

****



program juraev;

uses crt;

type

TMatrix = class

private

data: array of array of Real;

n: Integer;

public

constructor Create(size: Integer);

procedure FillRandom;

function Add(other: TMatrix): TMatrix;

function Determinant: Real;

property Size: Integer read n;

function GetElement(i, j: Integer): Real;

end;

constructor TMatrix.Create(size: Integer);

begin

n := size;

SetLength(data, n, n);

end;

procedure TMatrix.FillRandom;

var i, j: Integer;

begin

Randomize;

for i := 0 to n - 1 do

for j := 0 to n - 1 do

data[i, j] := Random(100) / 10;

end;

function TMatrix.GetElement(i, j: Integer): Real;

begin

Result := data[i, j];

end;

function TMatrix.Add(other: TMatrix): TMatrix;

var i, j: Integer;

begin

Result := TMatrix.Create(n);

for i := 0 to n - 1 do

for j := 0 to n - 1 do

Result.data[i, j] := data[i, j] + other.data[i, j];

end;

function TMatrix.Determinant: Real;

function Det3x3(mat: array of array of Real): Real;

begin

Result := mat[0,0]\*mat[1,1]\*mat[2,2] + mat[0,1]\*mat[1,2]\*mat[2,0] + mat[0,2]\*mat[1,0]\*mat[2,1]

- mat[0,2]\*mat[1,1]\*mat[2,0] - mat[0,0]\*mat[1,2]\*mat[2,1] - mat[0,1]\*mat[1,0]\*mat[2,2];

end;

begin

if n = 3 then

Result := Det3x3(data)

else

Result := 0; // Для простоты считаем только 3x3

end;

var

A, B, C: TMatrix;

begin

ClrScr;

A := TMatrix.Create(3);

B := TMatrix.Create(3);

A.FillRandom;

B.FillRandom;

// Вывод матриц

WriteLn('Матрица A:');

for var i := 0 to 2 do

begin

for var j := 0 to 2 do

Write(A.GetElement(i,j):6:2,' ');

WriteLn;

end;

WriteLn('Матрица B:');

for var i := 0 to 2 do

begin

for var j := 0 to 2 do

Write(B.GetElement(i,j):6:2,' ');

WriteLn;

end;

// Сложение

C := A.Add(B);

WriteLn('A + B:');

for var i := 0 to 2 do

begin

for var j := 0 to 2 do

Write(C.GetElement(i,j):6:2,' ');

WriteLn;

end;

// Определитель

WriteLn('det(A) = ', A.Determinant:8:2);

end.