УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №2.4

по предмету «Основы алгоритмизации и программирования»

Вариант 15

Выполнила:

Городко К. Е.

Гр. 351005

Проверила:

Данилова Г. В.

Минск 2023

**ЗАДАНИЕ:**

Дана матрица A порядка n. Найти строку матрицы, в которой больше

всего нулевых элементов.

**КОД ПРОГРАММЫ DELPHI:**

Program LAB2\_4;

Uses

System.SysUtils;

Type

TMatrix = Array Of Array Of Integer;

TArrOI = Array Of Integer;

Const

MINSIZE = 2;

MAXSIZE = 10;

MIN = -1000;

MAX = 1000;

Function CheckInput(Const MINNUM, MAXNUM: Integer;

OutputMessage: String): Integer;

Var

Num: Integer;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Write(OutputMessage);

Try

Readln(Num);

Except

IsCorrect := False;

Writeln('Введенные данные не соответствуют условию. Повторите попытку.');

End;

If IsCorrect And ((Num < MINNUM) Or (Num > MAXNUM)) Then

Begin

IsCorrect := False;

Writeln('Введенные данные не соответствуют условию. Повторите попытку.');

End;

Until (IsCorrect);

CheckInput := Num;

End;

Function InputMatrix(Matrix: TMatrix): TMatrix;

Var

I, J: Integer;

IsCorrect: Boolean;

Begin

Writeln('Введите матрицу.');

For I := 0 To High(Matrix) Do

For J := 0 To High(Matrix[I]) Do

Begin

Write('Введите элемент ', I + 1, ' строки, ', J + 1,

' столбца матрицы: ');

Matrix[I][J] := CheckInput(MIN, MAX, '');

End;

Writeln;

InputMatrix := Matrix;

End;

Function FillMatrixFromConsole(): TMatrix;

Var

RowNum, ColNum: Integer;

Matrix: TMatrix;

Begin

Writeln('Введите размеры матрицы (от 2 до 10).');

RowNum := CheckInput(MINSIZE, MAXSIZE, 'Введите количество строк матрицы: ');

ColNum := CheckInput(MINSIZE, MAXSIZE, 'Введите количество столбцов матрицы: ');

SetLength(Matrix, RowNum, ColNum);

Matrix := InputMatrix(Matrix);

FillMatrixFromConsole := Matrix;

End;

Function CheckFilePath(Path: String): Boolean;

Var

IsCorrect: Boolean;

Begin

If Not FileExists(Path) Then

Begin

Writeln('Введенного файла не существует. Повторите попытку.');

IsCorrect := False;

End

Else

If ExtractFileExt(Path) <> '.txt' Then

Begin

Writeln('Введенный Вами файл не является текстовым. Повторите попытку.');

IsCorrect := False;

End;

CheckFilePath := IsCorrect;

End;

Function CheckFileInputPath(): String;

Var

Path: String;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Writeln('Введите путь к файлу, содержащему матрицу. Первой строкой должны

быть введены размеры (строки и столбцы, от 2 до 10).');

Readln(Path);

IsCorrect := CheckFilePath(Path);

Until IsCorrect;

CheckFileInputPath := Path;

End;

Function FillMatrixFromFile(Path: String; Matrix: TMatrix): TMatrix;

Var

I, J, RowNum, ColNum: Integer;

FIn: TextFile;

IsCorrect: Boolean;

Begin

AssignFile(FIn, Path);

Repeat

IsCorrect := True;

Try

Try

Reset(FIn);

Read(FIn, RowNum);

Read(FIn, ColNum);

If (IsCorrect And (RowNum < MINSIZE) Or (RowNum > MAXSIZE) Or

(ColNum < MINSIZE) Or (ColNum > MAXSIZE)) Then

Begin

Writeln('Размер в выбранном файле не соответствуют условию.

Повторите попытку.');

IsCorrect := False;

End;

SetLength(Matrix, RowNum, ColNum);

I := 0;

While ((I < RowNum) And IsCorrect) Do

Begin

J := 0;

While ((J < ColNum) And IsCorrect) Do

Begin

If Eof(FIn) Then

Begin

Writeln('Размер введенной матрицы не соответствует

заданному. Повторите попытку.');

IsCorrect := False;

End;

Read(FIn, Matrix[I][J]);

If (IsCorrect And ((Matrix[I][J] < MIN) Or

(Matrix[I][J] > MAX))) Then

Begin

Writeln('Данные выбранного файла не соответствуют

условию. Повторите попытку.');

IsCorrect := False;

End;

Inc(J);

End;

Inc(I);

End;

If (IsCorrect And (Not Eof(FIn))) Then

Begin

Writeln('Размер введенной матрицы не соответствует заданному.

Повторите попытку.');

IsCorrect := False;

End;

Finally

CloseFile(FIn);

End;

Except

Writeln('Данные выбранного файла не соответствуют условию. Повторите

попытку.');

IsCorrect := False;

End;

If Not IsCorrect Then

Path := CheckFileInputPath();

Until IsCorrect;

Writeln('Данные из файла успешно считаны.');

Writeln;

FillMatrixFromFile := Matrix;

End;

Procedure OutputMatrix(Matrix: TMatrix);

Var

I, J: Integer;

Begin

Writeln('Введенная матрица:');

For I := 0 To High(Matrix) Do

Begin

For J := 0 To High(Matrix[0]) Do

Write(Matrix[I][J], ' ');

Writeln;

End;

Writeln;

End;

Function FindZeroAmountOfEachRow(Matrix: TMatrix): TArrOI;

Var

I, J: Integer;

ZeroAmount: TArrOI;

Begin

SetLength(ZeroAmount, High(Matrix) + 1);

For I := 0 To High(Matrix) Do

Begin

ZeroAmount[I] := 0;

For J := 0 To High(Matrix[I]) Do

If Matrix[I][J] = 0 Then

Inc(ZeroAmount[I]);

End;

FindZeroAmountOfEachRow := ZeroAmount;

End;

Function FindMaxZeroAmount(ZeroAmount: TArrOI): Integer;

Var

I, MaxZeroAmount: Integer;

Begin

MaxZeroAmount := ZeroAmount[0];

For I := 1 To High(ZeroAmount) Do

If ZeroAmount[I] > MaxZeroAmount Then

MaxZeroAmount := ZeroAmount[I];

FindMaxZeroAmount := MaxZeroAmount;

End;

Function FindNumOfMaxZeroRows(MaxZeroAmount: Integer; ZeroAmount: TArrOI): Integer;

Var

I, MaxZeroRows: Integer;

Begin

MaxZeroRows := 0;

For I := 0 To High(ZeroAmount) Do

If ZeroAmount[I] = MaxZeroAmount Then

Inc(MaxZeroRows);

FindNumOfMaxZeroRows := MaxZeroRows;

End;

Function FindMaxZeroRowIndex(MaxZeroRows, MaxZeroAmount: Integer;

ZeroAmount: TArrOI): Integer;

Var

I, Index: Integer;

Begin

For I := 0 To High(ZeroAmount) Do

If ZeroAmount[I] = MaxZeroAmount Then

Index := I;

FindMaxZeroRowIndex := Index;

End;

Function FindMaxZeroRowIndexArray(MaxZeroRows, MaxZeroAmount: Integer;

ZeroAmount: TArrOI): TArrOI;

Var

I, J: Integer;

IndexArray: TArrOI;

Begin

SetLength(IndexArray, MaxZeroRows);

J := 0;

For I := 0 To High(ZeroAmount) Do

If ZeroAmount[I] = MaxZeroAmount Then

Begin

IndexArray[J] := I;

Inc(J);

End;

FindMaxZeroRowIndexArray := IndexArray;

End;

Function CheckFileOutputPath(): String;

Var

Path: String;

IsCorrect: Boolean;

Begin

Writeln;

Repeat

IsCorrect := True;

Writeln('Введите путь к файлу, в который нужно записать результат.');

Readln(Path);

IsCorrect := CheckFilePath(Path);

If IsCorrect And FileIsReadOnly(Path) Then

Begin

Writeln('Введенный Вами файл доступен только для чтения. Повторите

попытку.');

IsCorrect := False;

End;

Until IsCorrect;

CheckFileOutputPath := Path;

End;

Procedure WriteResultIntoFile(MaxZeroAmount, MaxZeroRows, Index: Integer;

IndexArray: TArrOI; Matrix: TMatrix; Path: String);

Var

I, J: Integer;

IsCorrect: Boolean;

FOut: TextFile;

Begin

Repeat

IsCorrect := True;

Assign(FOut, Path);

Try

Try

Rewrite(FOut);

If MaxZeroAmount = 0 Then

Writeln(FOut, 'В матрице нет строк с нулевыми элементами.')

Else

If MaxZeroRows = 1 Then

Begin

Write(FOut, 'Строка с максимальным количеством нулевых

элементов: ');

For J := 0 To High(Matrix[0]) Do

Write(FOut, Matrix[Index][J], ' ');

End

Else

If (MaxZeroRows = (High(Matrix) + 1)) And

(MaxZeroAmount = (High(Matrix[0]) + 1)) Then

Writeln(FOut, 'Все элементы матрицы - нулевые элементы.')

Else

Begin

Writeln(FOut, 'Строки с максимальным количеством нулевых

элементов: ');

For I := 0 To High(IndexArray) Do

Begin

For J := 0 To High(Matrix[0]) Do

Write(FOut, Matrix[IndexArray[I]][J], ' ');

Writeln(FOut);

End;

End;

Finally

CloseFile(FOut);

End;

Except

Writeln('Произошла ошибка. Повторите попытку.');

IsCorrect := False;

Path := CheckFileOutputPath();

End;

Until IsCorrect;

Writeln('Результат записан.');

End;

Procedure OutputMaxZeroRow(Index: Integer; Matrix: TMatrix);

Var

J: Integer;

Begin

Write('Строка с максимальным количеством нулевых элементов: ');

For J := 0 To High(Matrix[0]) Do

Write(Matrix[Index][J], ' ');

End;

Procedure OutputMaxZeroRows(IndexArray: TArrOI; Matrix: TMatrix);

Var

I, J: Integer;

Begin

Writeln('Строки с максимальным количеством нулевых элементов: ');

For I := 0 To High(IndexArray) Do

Begin

For J := 0 To High(Matrix[0]) Do

Write(Matrix[IndexArray[I]][J], ' ');

Writeln;

End;

End;

Procedure OutputResult(MaxZeroAmount, MaxZeroRows, Index: Integer;

IndexArray: TArrOI; Matrix: TMatrix);

Begin

If MaxZeroAmount = 0 Then

Writeln('В матрице нет строк с нулевыми элементами.')

Else

If MaxZeroRows = 1 Then

OutputMaxZeroRow(Index, Matrix)

Else

If (MaxZeroRows = (High(Matrix) + 1)) And

(MaxZeroAmount = (High(Matrix[0]) + 1)) Then

Writeln('Все элементы матрицы - нулевые элементы.')

Else

OutputMaxZeroRows(IndexArray, Matrix);

End;

Function CheckChoiceInput(OutputMessage: String): Integer;

Var

Num: Integer;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Writeln(OutputMessage);

Try

Readln(Num);

Except

IsCorrect := False;

Writeln('Введенные данные не соответствуют условию. Повторите попытку.');

End;

If IsCorrect And ((Num <> 0) And (Num <> 1)) Then

Begin

IsCorrect := False;

Writeln('Введенные данные не соответствуют условию. Повторите попытку.');

End;

Until (IsCorrect);

Writeln;

CheckChoiceInput := Num;

End;

Procedure WriteCondition();

Begin

Writeln('Данная программа находит строку матрицы, в которой больше всего нулевых

элементов.');

Writeln('Элементы матрицы - целые числа от -1000 до 1000.');

End;

Var

MaxZeroAmount, MaxZeroRows, Index, Choice: Integer;

ZeroAmount, IndexArray: TArrOI;

Matrix, MaxZero: TMatrix;

FInPath, FOutPath: String;

Begin

WriteCondition();

Choice := CheckChoiceInput ('Если Вы хотите вводить данные в консоль, введите 0.

Если использовать файл, введите 1.');

If Choice = 0 Then

Matrix := FillMatrixFromConsole()

Else

Begin

FInPath := CheckFileInputPath();

Matrix := FillMatrixFromFile(FInPath, Matrix);

End;

OutputMatrix(Matrix);

ZeroAmount := FindZeroAmountOfEachRow(Matrix);

MaxZeroAmount := FindMaxZeroAmount(ZeroAmount);

MaxZeroRows := FindNumOfMaxZeroRows(MaxZeroAmount, ZeroAmount);

If MaxZeroRows = 1 Then

Index := FindMaxZeroRowIndex(MaxZeroRows, MaxZeroAmount, ZeroAmount)

Else

IndexArray := FindMaxZeroRowIndexArray(MaxZeroRows, MaxZeroAmount,

ZeroAmount);

Choice := CheckChoiceInput('Если Вы хотите вывести результат в консоль,

введите 0. Если в файл, введите 1.');

If Choice = 0 Then

OutputResult(MaxZeroAmount, MaxZeroRows, Index, IndexArray, Matrix)

Else

Begin

FOutPath := CheckFileOutputPath();

WriteResultIntoFile(MaxZeroAmount, MaxZeroRows, Index, IndexArray,

Matrix, FOutPath);

End;

End.

**КОД ПРОГРАММЫ С++:**

#include <iostream>

#include <fstream>

using namespace std;

const int MINSIZE = 2;

const int MAXSIZE = 10;

const int MIN = -1000;

const int MAX = 1000;

int checkInput(const int MINNUM, const int MAXNUM, string outputMessage) {

int num = 0;

bool isIncorrect;

do {

isIncorrect = false;

cout << outputMessage;

cin >> num;

if (cin.get() != '\n') {

cin.clear();

while (cin.get() != '\n');

isIncorrect = true;

cout << "Введенные данные не соответствуют условию. Повторите попытку."

<< endl;

}

if (!isIncorrect && ((num < MINNUM) || (num > MAXNUM))) {

isIncorrect = true;

cout << "Введенные данные не соответствуют условию. Повторите попытку."

<< endl;

}

} while (isIncorrect);

return num;

}

int\*\* createMatrix(int rowNum, int colNum) {

int\*\* matrix = new int\* [rowNum];

for (int i = 0; i < rowNum; i++)

matrix[i] = new int[colNum];

return matrix;

}

int\*\* inputMatrix(int rowNum, int colNum) {

bool isIncorrect;

int\*\* matrix = createMatrix(rowNum, colNum);

for (int i = 0; i < rowNum; i++)

for (int j = 0; j < colNum; j++) {

cout << "Введите элемент " << i + 1 << " строки, " << j + 1 << " столбца

матрицы: ";

matrix[i][j] = checkInput(MIN, MAX, "");

}

cout << endl;

return matrix;

}

int\*\* fillMatrixFromConsole(int& rowNum, int& colNum) {

cout << "Введите размеры матрицы (от 2 до 10)." << endl;

rowNum = checkInput(MINSIZE, MAXSIZE, "Введите количество строк матрицы : ");

colNum = checkInput(MINSIZE, MAXSIZE, "Введите количество столбцов матрицы : ");

int\*\* matrix = inputMatrix(rowNum, colNum);

return matrix;

}

bool checkFileExtension(string path) {

bool isIncorrect = true;

if (path.substr(path.size() - 4) == ".txt")

isIncorrect = false;

else

cout << "Введенный Вами файл не является текстовым. Повторите попытку."

<< endl;

return isIncorrect;

}

string checkFileInputPath() {

string path;

bool isIncorrect;

do {

isIncorrect = false;

cout << "Введите путь к файлу, содержащему матрицу. Первой строкой должны

быть введены размеры (строки и столбцы, от 2 до 10)." << endl;

cin >> path;

ifstream fin(path);

fin.open(path);

if (!fin.is\_open()) {

cout << "Введенного файла не существует. Повторите попытку." << endl;

isIncorrect = true;

}

if (!isIncorrect)

isIncorrect = checkFileExtension(path);

fin.close();

} while (isIncorrect);

return path;

}

int\*\* fillMatrixFromFile(string path, int& rowNum, int& colNum) {

bool isIncorrect;

int\*\* matrix = 0;

do {

isIncorrect = false;

ifstream fin(path);

try {

fin >> rowNum;

fin >> colNum;

if (!isIncorrect && (rowNum < MIN || rowNum > MAX || colNum < MIN ||

colNum > MAX)) {

cout << "Размер в выбранном файле не соответствуют условию. Повторите

попытку." << endl;

isIncorrect = true;

}

matrix = createMatrix(rowNum, colNum);

for (int i = 0; i < rowNum; i++)

for (int j = 0; j < colNum; j++) {

fin >> matrix[i][j];

if (!isIncorrect && ((matrix[i][j] < MIN) ||

(matrix[i][j] > MAX))) {

cout << "Данные выбранного файла не соответствуют условию.

Повторите попытку." << endl;

isIncorrect = true;

}

}

if (!isIncorrect && !fin.eof()) {

cout << "Размер введенной матрицы не соответствует заданному.

Повторите попытку." << endl;

isIncorrect = true;

}

}

catch (string errorMessage) {

cout << "Данные выбранного файла не соответствуют условию. Повторите

попытку." << endl;

isIncorrect = true;

}

fin.close();

if (isIncorrect)

path = checkFileInputPath();

} while (isIncorrect);

cout << "Данные из файла успешно считаны." << endl;

cout << endl;

return matrix;

}

void outputMatrix(int rowNum, int colNum, int\*\* matrix) {

cout << "Введенная матрица:" << endl;

for (int i = 0; i < rowNum; i++) {

for (int j = 0; j < colNum; j++)

cout << matrix[i][j] << " ";

cout << endl;

}

cout << endl;

}

int\* findZeroAmountOfEachRow(int rowNum, int colNum, int\*\* matrix) {

int\* zeroAmount = new int[rowNum];

for (int i = 0; i < rowNum; i++) {

zeroAmount[i] = 0;

for (int j = 0; j < colNum; j++)

if (matrix[i][j] == 0)

zeroAmount[i]++;

}

return zeroAmount;

}

int findMaxZeroAmount(int rowNum, int\* zeroAmount) {

int maxZeroAmount = zeroAmount[0];

for (int i = 1; i < rowNum; i++)

if (zeroAmount[i] > maxZeroAmount)

maxZeroAmount = zeroAmount[i];

return maxZeroAmount;

}

int findNumOfMaxZeroRows(int rowNum, int maxZeroAmount, int\* zeroAmount) {

int maxZeroRows = 0;

for (int i = 0; i < rowNum; i++)

if (zeroAmount[i] == maxZeroAmount)

maxZeroRows++;

return maxZeroRows;

}

int findMaxZeroRowIndex(int rowNum, int maxZeroRows, int maxZeroAmount, int\* zeroAmount) {

int index = 0;

for (int i = 0; i < rowNum; i++)

if (zeroAmount[i] == maxZeroAmount)

index = i;

return index;

}

int\* findMaxZeroRowIndexArray(int rowNum, int maxZeroRows, int maxZeroAmount, int\* zeroAmount) {

int\* indexArray = new int[maxZeroRows];

int j = 0;

for (int i = 0; i < rowNum; i++)

if (zeroAmount[i] == maxZeroAmount) {

indexArray[j] = i;

j++;

}

return indexArray;

}

string checkFileOutputPath() {

string path;

bool isIncorrect;

cout << endl;

do {

isIncorrect = false;

cout << "Введите путь к файлу, в который нужно записать результат." << endl;

cin >> path;

ofstream fout(path);

fout.open(path);

if (!fout.is\_open()) {

cout << "Произошла ошибка. Повторите попытку." << endl;

isIncorrect = true;

}

if (!isIncorrect)

isIncorrect = checkFileExtension(path);

fout.close();

} while (isIncorrect);

cout << endl;

return path;

}

void writeResultIntoFile(int rowNum, int colNum, int maxZeroAmount, int maxZeroRows,

int index, int\* indexArray, int\*\* matrix, string path) {

bool isIncorrect;

do {

isIncorrect = false;

ofstream fout(path);

try {

if (maxZeroAmount == 0)

fout << "В матрице нет строк с нулевыми элементами.";

else

if (maxZeroRows == 1) {

fout << "Строка с максимальным количеством нулевых элементов: ";

for (int j = 0; j < colNum; j++)

fout << matrix[index][j] << " ";

}

else

if ((maxZeroRows == rowNum) && (maxZeroAmount == colNum))

fout << "Все элементы матрицы - нулевые элементы.";

else {

fout << "Строки с максимальным количеством нулевых элементов: "

<< endl;

for (int i = 0; i < maxZeroRows; i++) {

for (int j = 0; j < colNum; j++)

fout << matrix[indexArray[i]][j] << " ";

fout << endl;

}

}

}

catch (string errorMessage) {

cout << "Произошла ошибка. Повторите попытку." << endl;

isIncorrect = true;

}

fout.close();

if (isIncorrect)

path = checkFileOutputPath();

} while (isIncorrect);

cout << "Результат записан.";

}

void outputMaxZeroRow(int colNum, int index, int\*\* matrix) {

cout << "Строка с максимальным количеством нулевых элементов: ";

for (int j = 0; j < colNum; j++)

cout << matrix[index][j] << " ";

}

void outputMaxZeroRows(int colNum, int maxZeroRows, int\* indexArray, int\*\* matrix) {

cout << "Строки с максимальным количеством нулевых элементов: " << endl;

for (int i = 0; i < maxZeroRows; i++) {

for (int j = 0; j < colNum; j++)

cout << matrix[indexArray[i]][j] << " ";

cout << endl;

}

}

void outputResult(int rowNum, int colNum, int maxZeroAmount, int maxZeroRows,

int index, int\* indexArray, int\*\* matrix) {

if (maxZeroAmount == 0)

cout << "В матрице нет строк с нулевыми элементами.";

else

if (maxZeroRows == 1)

outputMaxZeroRow(colNum, index, matrix);

else

if ((maxZeroRows == rowNum) && (maxZeroAmount == colNum))

cout << "Все элементы матрицы - нулевые элементы.";

else

outputMaxZeroRows(colNum, maxZeroRows, indexArray, matrix);

}

int checkChoiceInput(string outputMessage) {

int num = 0;

bool isIncorrect;

do {

isIncorrect = false;

cout << outputMessage << endl;

cin >> num;

if (cin.get() != '\n') {

cin.clear();

while (cin.get() != '\n');

isIncorrect = true;

cout << "Введенные данные не соответствуют условию. Повторите попытку."

<< endl;

}

if (!isIncorrect && (num != 0) && (num != 1)) {

isIncorrect = true;

cout << "Введенные данные не соответствуют условию. Повторите попытку."

<< endl;

}

} while (isIncorrect);

cout << endl;

return num;

}

void writeCondition() {

cout << "Данная программа находит строку матрицы, в которой больше всего нулевых

элементов." << endl;

cout << "Элементы матрицы - целые числа от -1000 до 1000." << endl;

}

int main() {

setlocale(LC\_ALL, "Russian");

int choice = 0;

int rowNum = 0;

int colNum = 0;

int maxZeroAmount = 0;

int maxZeroRows = 0;

int index = 0;

int\* indexArray = new int;

int\*\* matrix;

writeCondition();

choice = checkChoiceInput("Если Вы хотите вводить данные в консоль, введите 0.

Если использовать файл, введите 1.");

if (choice == 0)

matrix = fillMatrixFromConsole(rowNum, colNum);

else {

string finPath = checkFileInputPath();

matrix = fillMatrixFromFile(finPath, rowNum, colNum);

}

outputMatrix(rowNum, colNum, matrix);

int\* zeroAmount = findZeroAmountOfEachRow(rowNum, colNum, matrix);

maxZeroAmount = findMaxZeroAmount(rowNum, zeroAmount);

maxZeroRows = findNumOfMaxZeroRows(rowNum, maxZeroAmount, zeroAmount);

if (maxZeroRows == 1)

index = findMaxZeroRowIndex(rowNum, maxZeroRows, maxZeroAmount, zeroAmount);

else

indexArray = findMaxZeroRowIndexArray(rowNum, maxZeroRows, maxZeroAmount,

zeroAmount);

choice = checkChoiceInput("Если Вы хотите вывести результат в консоль, введите 0.

Если в файл, введите 1.");

if (choice == 0)

outputResult(rowNum, colNum, maxZeroAmount, maxZeroRows, index, indexArray,

matrix);

else {

string foutPath = checkFileOutputPath();

writeResultIntoFile(rowNum, colNum, maxZeroAmount, maxZeroRows, index,

indexArray, matrix, foutPath);

}

delete[] indexArray;

delete[] matrix;

delete[] zeroAmount;

return 0;

}

**КОД ПРОГРАММЫ JAVA:**

import java.util.Scanner;

import java.io.\*;

public class main {

static final int MINSIZE = 2;

static final int MAXSIZE = 10;

static final int MIN = -1000;

static final int MAX = 1000;

private static int checkInput(final int MINNUM, final int MAXNUM,

String outputMessage, Scanner scan) {

int num = 0;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.print(outputMessage);

try {

num = Integer.parseInt(scan.nextLine());

}

catch (NumberFormatException e) {

System.out.println("Введенные данные не соответствуют условию.

Повторите попытку.");

isIncorrect = true;

}

if (!isIncorrect && ((num < MINNUM) || (num > MAXNUM))) {

isIncorrect = true;

System.out.println("Введенные данные не соответствуют условию.

Повторите попытку.");

}

} while (isIncorrect);

return num;

}

private static int[][] inputMatrix(int rowNum, int colNum, Scanner scan) {

boolean isIncorrect;

int[][] matrix = new int[rowNum][colNum];

for (int i = 0; i < rowNum; i++)

for (int j = 0; j < colNum; j++) {

System.out.print("Введите элемент " + (i + 1) + " строки, " + (j + 1)

+ " столбца матрицы : ");

matrix[i][j] = checkInput(MIN, MAX, "", scan);

}

System.out.println();

return matrix;

}

private static int[][] fillMatrixFromConsole(Scanner scan) {

int rowNum = 0;

int colNum = 0;

System.out.println("Введите размеры матрицы (от 2 до 10).");

rowNum = checkInput(MINSIZE, MAXSIZE, "Введите количество строк матрицы: ",

scan);

colNum = checkInput(MINSIZE, MAXSIZE, "Введите количество столбцов

матрицы: ", scan);

int[][] matrix = inputMatrix(rowNum, colNum, scan);

return matrix;

}

private static boolean checkFilePath(String path, File file) {

boolean isIncorrect = false;

if (!file.exists()) {

System.out.println("Введенного файла не существует. Повторите попытку.");

isIncorrect = true;

}

else

if (!path.endsWith(".txt")) {

System.out.println("Введенный Вами файл не является текстовым.

Повторите попытку.");

isIncorrect = true;

}

return isIncorrect;

}

private static String checkFileInputPath(Scanner scan) {

String path;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.println("Введите путь к файлу, содержащему матрицу. Первой

строкой должны быть введены размеры (строки и

столбцы, от 2 до 10).");

path = scan.nextLine();

File file = new File(path);

isIncorrect = checkFilePath(path, file);

} while (isIncorrect);

return path;

}

private static int[][] fillMatrixFromFile(String path, Scanner scan) {

int rowNum = 0;

int colNum = 0;

boolean isIncorrect;

int[][] matrix = new int[rowNum][colNum];

do {

isIncorrect = false;

try (Scanner scanFile = new Scanner(new FileReader(path))) {

rowNum = scanFile.nextInt();

colNum = scanFile.nextInt();

if (!isIncorrect && (rowNum < MIN || rowNum > MAX || colNum < MIN ||

colNum > MAX)) {

System.out.println("Размер в выбранном файле не соответствуют

условию. Повторите попытку.");

isIncorrect = true;

}

matrix = new int[rowNum][colNum];

for (int i = 0; i < rowNum; i++)

for (int j = 0; j < colNum; j++) {

matrix[i][j] = scanFile.nextInt();

if (!isIncorrect && ((matrix[i][j] < MIN) ||

(matrix[i][j] > MAX))) {

System.out.println("Данные выбранного файла не

соответствуют условию. Повторите

попытку.");

isIncorrect = true;

}

}

if (!isIncorrect && scanFile.hasNext()) {

System.out.println("Размер введенной матрицы не соответствует

заданному. Повторите попытку.");

isIncorrect = true;

}

} catch (Exception e) {

System.out.println("Данные выбранного файла не соответствуют условию.

Повторите попытку.");

isIncorrect = true;

}

if (isIncorrect)

path = checkFileInputPath(scan);

} while (isIncorrect);

System.out.println("Данные из файла успешно считаны.");

System.out.println();

return matrix;

}

private static void outputMatrix(int[][] matrix) {

System.out.println("Введенная матрица:");

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[0].length; j++)

System.out.print(matrix[i][j] + " ");

System.out.println();

}

System.out.println();

}

private static int[] findZeroAmountOfEachRow(int[][] matrix) {

int[] zeroAmount = new int[matrix.length];

for (int i = 0; i < matrix.length; i++) {

zeroAmount[i] = 0;

for (int j = 0; j < matrix[0].length; j++)

if (matrix[i][j] == 0)

zeroAmount[i]++;

}

return zeroAmount;

}

private static int findMaxZeroAmount(int[] zeroAmount) {

int maxZeroAmount = zeroAmount[0];

for (int i = 1; i < zeroAmount.length; i++)

if (zeroAmount[i] > maxZeroAmount)

maxZeroAmount = zeroAmount[i];

return maxZeroAmount;

}

private static int findNumOfMaxZeroRows(int maxZeroAmount, int[] zeroAmount) {

int maxZeroRows = 0;

for (int i = 0; i < zeroAmount.length; i++)

if (zeroAmount[i] == maxZeroAmount)

maxZeroRows++;

return maxZeroRows;

}

private static int findMaxZeroRowIndex(int maxZeroAmount, int[] zeroAmount) {

int index = 0;

for (int i = 0; i < zeroAmount.length; i++)

if (zeroAmount[i] == maxZeroAmount)

index = i;

return index;

}

private static int[] findMaxZeroRowIndexArray(int maxZeroRows, int maxZeroAmount,

int[] zeroAmount) {

int[] indexArray = new int[maxZeroRows];

int j = 0;

for (int i = 0; i < zeroAmount.length; i++)

if (zeroAmount[i] == maxZeroAmount) {

indexArray[j] = i;

j++;

}

return indexArray;

}

private static String checkFileOutputPath(Scanner scan) {

String path;

boolean isIncorrect;

System.out.println();

do {

isIncorrect = false;

System.out.println("Введите путь к файлу, в который нужно записать

результат.");

path = String.format(scan.nextLine());

File file = new File(path);

isIncorrect = checkFilePath(path, file);

if (!isIncorrect & !file.canWrite()) {

System.out.println("Введенный Вами файл доступен только для чтения.

Повторите попытку.");

isIncorrect = true;

}

} while (isIncorrect);

System.out.println();

return path;

}

private static void writeResultIntoFile(int maxZeroAmount, int maxZeroRows,

int index, int[] indexArray, int[][]

matrix, String path, Scanner scan) {

boolean isIncorrect;

do {

isIncorrect = false;

try (FileWriter fw = new FileWriter(new File(path))) {

if (maxZeroAmount == 0)

fw.write("В матрице нет строк с нулевыми элементами.");

else

if (maxZeroRows == 1) {

fw.write("Строка с максимальным количеством нулевых

элементов: ");

for (int j = 0; j < matrix.length; j++)

fw.write(matrix[index][j] + " ");

}

else

if ((maxZeroRows == matrix.length) && (maxZeroAmount ==

matrix[0].length))

fw.write("Все элементы матрицы - нулевые элементы.");

else {

fw.write("Строки с максимальным количеством нулевых

элементов: \n");

for (int i = 0; i < maxZeroRows; i++) {

for (int j = 0; j < matrix[0].length; j++)

fw.write(matrix[indexArray[i]][j] + " ");

fw.write("\n");

}

}

} catch (Exception e) {

System.out.println("Произошла ошибка. Повторите попытку.");

isIncorrect = true;

}

if (isIncorrect)

path = checkFileOutputPath(scan);

} while (isIncorrect);

System.out.println("Результат записан.");

}

private static void outputMaxZeroRow(int index, int[][] matrix) {

System.out.print("Строка с максимальным количеством нулевых элементов: ");

for (int j = 0; j < matrix[0].length; j++)

System.out.print(matrix[index][j] + " ");

}

private static void outputMaxZeroRows(int maxZeroRows, int[] indexArray,

int[][] matrix) {

System.out.println("Строки с максимальным количеством нулевых элементов:");

for (int i = 0; i < maxZeroRows; i++) {

for (int j = 0; j < matrix[0].length; j++)

System.out.print(matrix[indexArray[i]][j] + " ");

System.out.println();

}

}

private static void outputResult(int maxZeroAmount, int maxZeroRows, int index,

int[] indexArray, int[][] matrix) {

if (maxZeroAmount == 0)

System.out.println("В матрице нет строк с нулевыми элементами.");

else

if (maxZeroRows == 1)

outputMaxZeroRow(index, matrix);

else

if ((maxZeroRows == matrix.length) && (maxZeroAmount ==

matrix[0].length))

System.out.println("Все элементы матрицы - нулевые элементы.");

else

outputMaxZeroRows(maxZeroRows, indexArray, matrix);

}

private static int checkChoiceInput(String outputMessage, Scanner scan) {

int num = 0;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.println(outputMessage);

try {

num = Integer.parseInt(scan.nextLine());

}

catch (NumberFormatException e){

System.out.println("Введенные данные не соответствуют условию.

Повторите попытку.");

isIncorrect = true;

}

if (!isIncorrect && (num != 0) && (num != 1)) {

isIncorrect = true;

System.out.println("Введенные данные не соответствуют условию.

Повторите попытку.");

}

} while (isIncorrect);

return num;

}

private static void writeCondition() {

System.out.println("Данная программа находит строку матрицы, в которой больше

всего нулевых элементов.");

System.out.println("Элементы матрицы - целые числа от -1000 до 1000.");

}

public static void main(String[] args) {

Scanner scan = new Scanner(System.in);

int choice = 0;

int maxZeroAmount = 0;

int maxZeroRows = 0;

int index = 0;

int[] indexArray = new int[0];

int[][] matrix;

writeCondition();

choice = checkChoiceInput("Если Вы хотите вводить данные в консоль, введите

0. Если использовать файл, введите 1.", scan);

if (choice == 0)

matrix = fillMatrixFromConsole(scan);

else {

String finPath = checkFileInputPath(scan);

matrix = fillMatrixFromFile(finPath, scan);

}

outputMatrix(matrix);

int[] zeroAmount = findZeroAmountOfEachRow(matrix);

maxZeroAmount = findMaxZeroAmount(zeroAmount);

maxZeroRows = findNumOfMaxZeroRows(maxZeroAmount, zeroAmount);

if (maxZeroRows == 1)

index = findMaxZeroRowIndex(maxZeroAmount, zeroAmount);

else

indexArray = findMaxZeroRowIndexArray(maxZeroRows, maxZeroAmount,

zeroAmount);

choice = checkChoiceInput("Если Вы хотите вывести результат в консоль,

введите 0. Если в файл, введите 1.", scan);

if (choice == 0)

outputResult(maxZeroAmount, maxZeroRows, index, indexArray, matrix);

else {

String foutPath = checkFileOutputPath(scan);

writeResultIntoFile(maxZeroAmount, maxZeroRows, index, indexArray,

matrix, foutPath, scan);

}

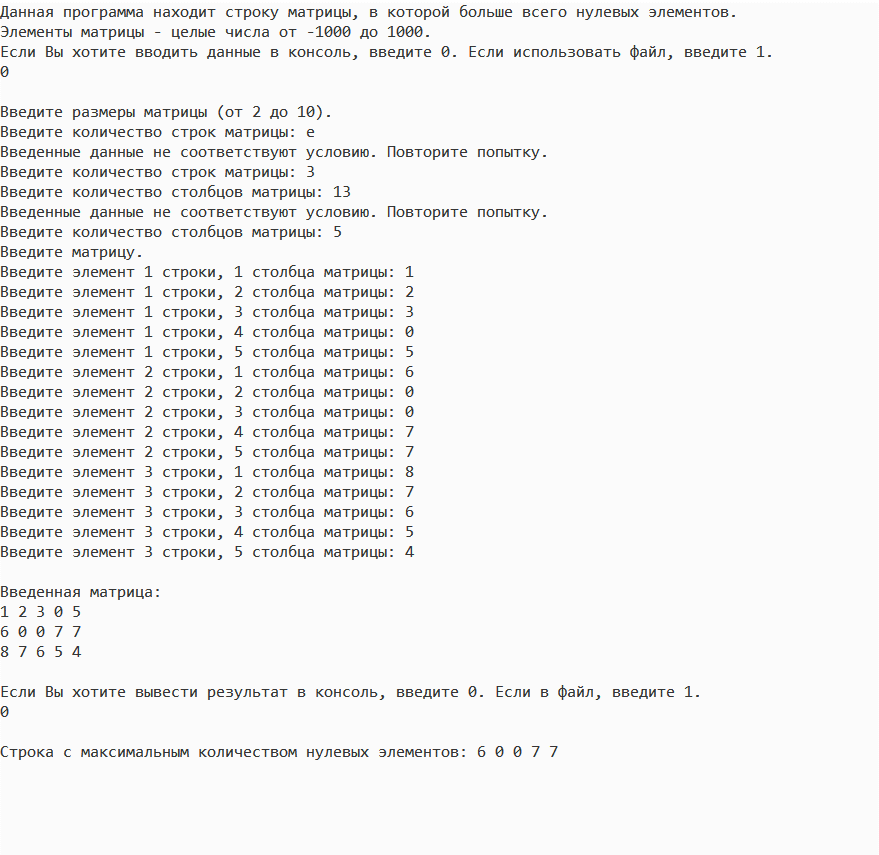
scan.close();

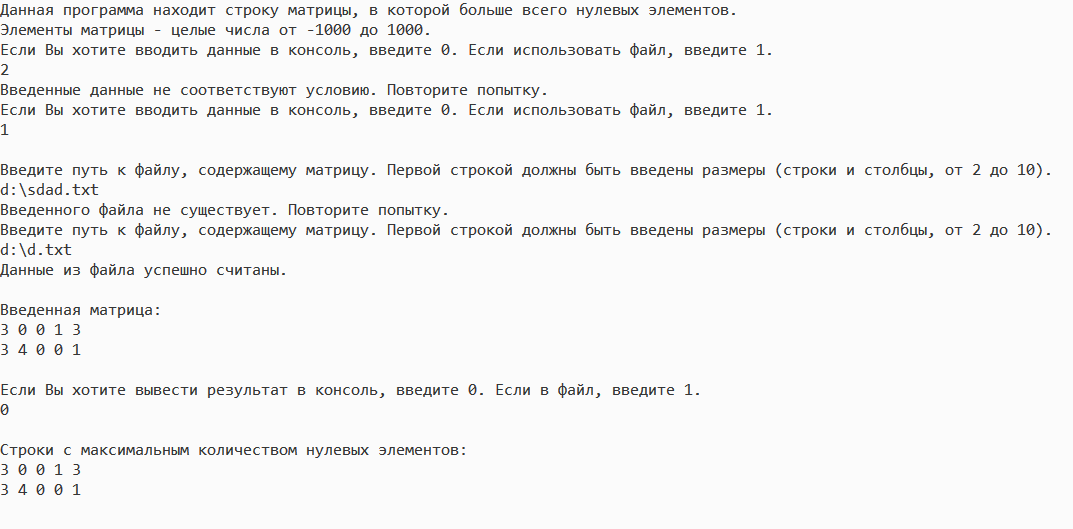
}

}

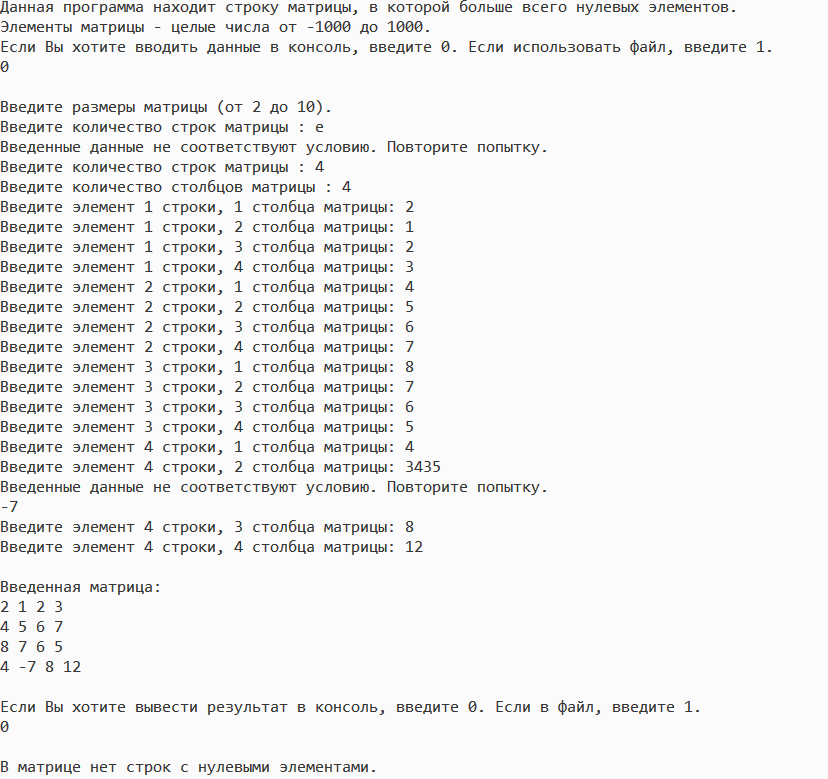
**СКРИНШОТЫ:**

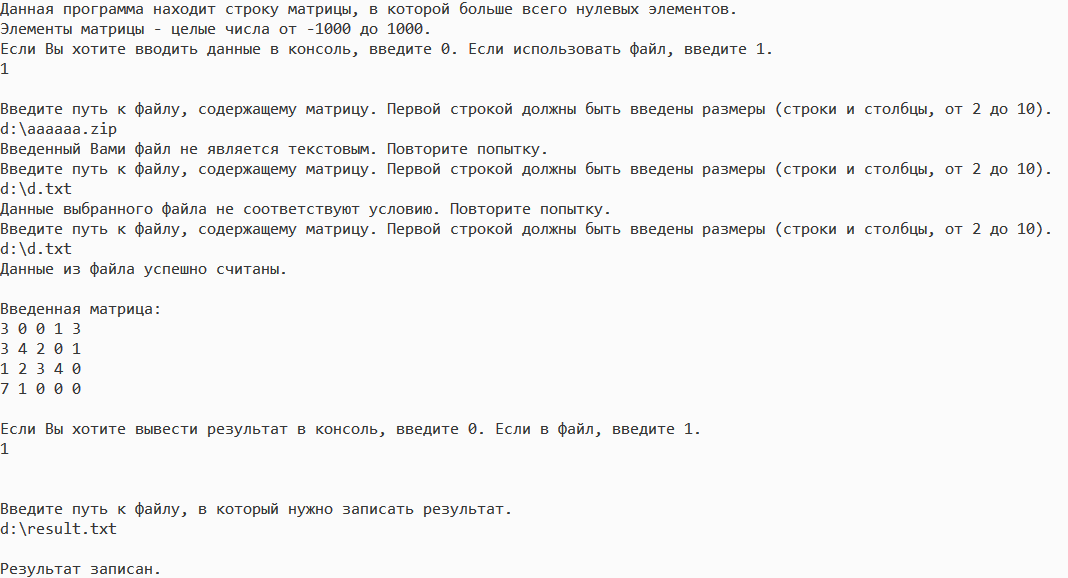
**Delphi:**



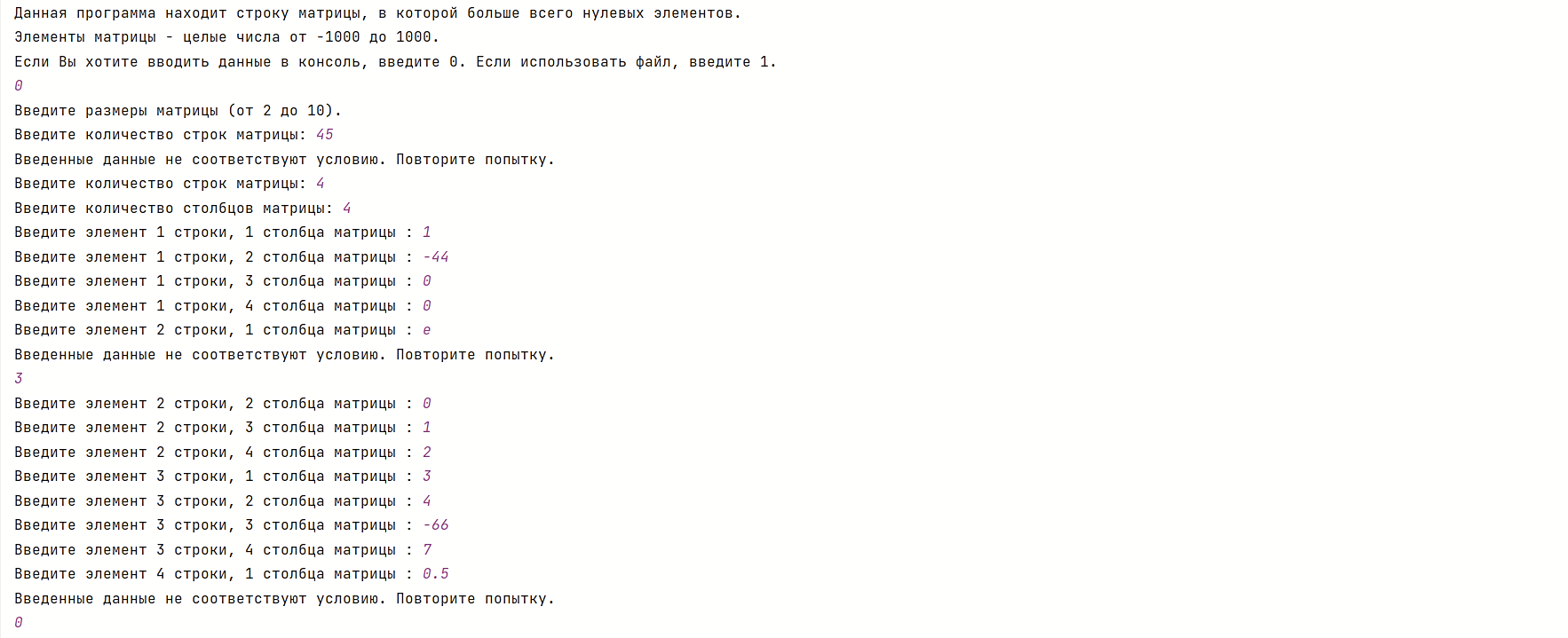


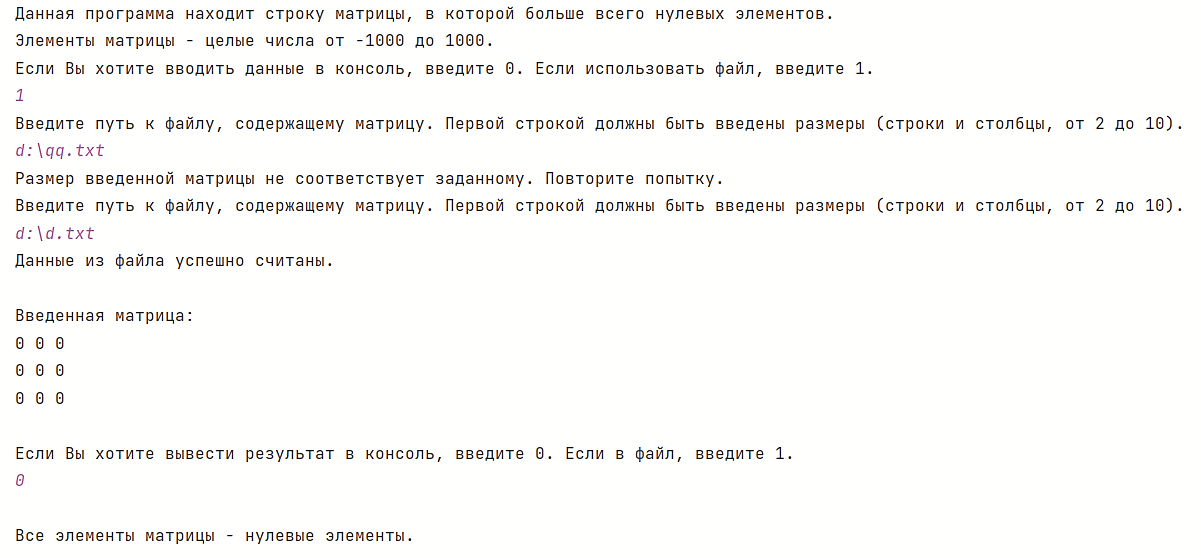
**C++:**





**Java:**





**БЛОК-СХЕМА:**













