УО «Белорусский государственный университет информатики и радиоэлектроники»

Кафедра ПОИТ

Отчет по лабораторной работе №2.3

по предмету «Основы алгоритмизации и программирования»

Вариант 20

Выполнил:

Машевский Д.В

Гр. 351003

Проверил:

Данилова Г. В.

Минск 2023

**Задание:**

Сложить две треугольные матрицы порядка n, у которых только элементы над главной диагональю отличны от нуля.

**Код на Delphi:**

Program SumMatr;

{$APPTYPE CONSOLE}

Uses

SysUtils;

Type

TMatrix = Array Of Array Of Integer;

Function InputChoice(): Integer;

Const

MIN = 0;

MAX = 1;

Var

Vib: Integer;

IsCorrect: Boolean;

Begin

Vib:= 0;

Write('Введите 0, если хотите сделать ввод с консоли и 1, если хотите ввод из

файла.');

Repeat

IsCorrect := True;

Write('Выбор: ');

Try

Readln(Vib);

Except

IsCorrect := False;

Write('Ошибка ввода. ');

End;

If IsCorrect And (Vib <> MAX) And (Vib <> MIN) Then

Begin

IsCorrect := False;

Write('Число должно быть либо 0, либо 1. ');

End;

Until IsCorrect;

InputChoice := Vib;

End;

Procedure DelProg();

Begin

Writeln('Данная программа складывает две треугольные N порядка, у которых только

элементы над главной диагональю отличны от нуля.');

End;

Function EnterN(): Integer;

Var

N: Integer;

IsCorrect: Boolean;

Begin

N:= 0;

Begin

Repeat

IsCorrect := True;

Try

Readln(N);

Except

IsCorrect := False;

Write('Ошибка ввода. ');

End;

If IsCorrect And (N < 2) Then

Begin

IsCorrect := False;

Writeln('Значение порядка матрицы должно быть больше 1. ');

End;

Until IsCorrect;

EnterN := N;

End;

End;

Function CreateTriangularMatrix(N: Integer): TMatrix;

Var

Matrix: TMatrix;

I, J: Integer;

IsCorrect: Boolean;

Begin

SetLength(Matrix, N, N);

For I := 0 To N - 1 Do

Begin

For J := I To N - 1 Do

Begin

Repeat

IsCorrect := True;

Write('Введите Mas', (I + 1), (J + 1), ': ');

Try

Readln(Matrix[I][J]);

Except

IsCorrect := False;

Write('Ошибка ввода. ');

End;

Until IsCorrect;

End;

End;

CreateTriangularMatrix := Matrix;

End;

Procedure Na();

Begin

Writeln('Введите размерность матриц');

End;

Function AddTriangularMatrices(Matrix1, Matrix2: TMatrix): TMatrix;

Var

N, i, j: Integer;

ResultMatrix: TMatrix;

Begin

N := Length(Matrix1);

SetLength(ResultMatrix, N, N);

For I := 0 To N - 1 Do

Begin

For J := I To N - 1 Do

Begin

ResultMatrix[i][j] := Matrix1[I][J] + Matrix2[I][J];

End;

End;

AddTriangularMatrices:= ResultMatrix;

End;

Procedure PrintResultMatrixConsol(ResultMatrix: TMatrix);

Var

N, I, J: Integer;

Begin

N := Length(ResultMatrix);

For I := 0 To N - 1 Do

Begin

For J := 0 To N - 1 Do

Begin

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

End;

Writeln;

End;

End;

Function OutputChoice(): Integer;

Const

MIN = 0;

MAX = 1;

Var

Vib: Integer;

IsCorrect: Boolean;

Begin

Vib:= 0;

Write('Введите 0, если хотите сделать вывод в консоль; 1, если хотите вывод в

файл.');

Repeat

IsCorrect := True;

Write('Выбор: ');

Try

Readln(Vib);

Except

IsCorrect := False;

Write('Ошибка ввода. ');

End;

If IsCorrect And (Vib <> MAX) And (Vib <> MIN) Then

Begin

IsCorrect := False;

Write('Число должно быть либо 0, либо 1. ');

End;

Until IsCorrect;

OutputChoice :=Vib;

End;

Function PTF(): String;

Var

Path: String;

IsCorrect: Boolean;

Begin

Repeat

IsCorrect := True;

Write('Укажите путь к файлу: ');

Readln(Path);

If Not FileExists(Path) Then

Begin

Write('Такой файл не найден. ');

IsCorrect := False;

End;

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

Begin

IsCorrect := False;

Write('У файла должно быть расширение txt. ');

End;

Until IsCorrect;

PTF := Path;

End;

Function ReadSizeFile(Path: String): Integer;

Var

N: Integer;

IsCorrect: Boolean;

InputFile: TextFile;

Begin

N:= 0;

AssignFile(InputFile, Path);

Reset(InputFile);

IsCorrect := True;

Try

Readln(InputFile, N);

Except

IsCorrect := False;

Writeln('Ошибка при считывании порядка матрицы. ');

End;

If IsCorrect And (N < 2) Then

Begin

IsCorrect := False;

Writeln('Значение порядка матрицы в файле должно быть больше 1. ');

End;

Close(InputFile);

If IsCorrect Then

ReadSizeFile := N

Else

Write('Введите порядок матрицы с консоли. ');

ReadSizeFile := (EnterN);

End;

Function ReadWrongElement(I: Integer; J: Integer):Integer;

Var

Elem: Integer;

IsCorrect: Boolean;

Begin

Elem:= 0;

Repeat

IsCorrect := True;

Write('Введите а', (I + 1), (J + 1), ' с консоли: ');

Try

Readln(Elem);

Except

IsCorrect := False;

Write('Ошибка ввода. ');

End;

Until IsCorrect;

ReadWrongElement := Elem;

End;

Function MatrixFile(N: Integer; Path: String): TMatrix;

Var

Matrix: TMatrix;

IsCorrect: Boolean;

InputFile: TextFile;

I, J: Integer;

Begin

N:= 0;

AssignFile(InputFile, Path);

Reset(InputFile);

Readln(InputFile);

SetLength(Matrix, N, N);

For I := 0 To N - 1 Do

Begin

For J := I To N - 1 Do

Begin

IsCorrect := True;

Try

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

Except

IsCorrect := False;

Write('Ошибка при считывании a', (I + 1), (J + 1), ' с файла. ');

End;

If Not IsCorrect Then

Matrix[I][J] := ReadWrongElement(I, J);

End;

End;

Close(InputFile);

MatrixFile := Matrix;

End;

Procedure PrintResultMatrixFile(ResultMatrix: TMatrix);

Var

Sum: Real;

N, I, J: Integer;

Matrix1, Matrix2: TMatrix;

IsCorrect: Boolean;

Path: String;

OutputFile: TextFile;

Begin

N:= 0;

Path := PTF();

AssignFile(OutputFile, Path);

Reset(OutputFile);

Rewrite(OutputFile);

For I := 0 To N - 1 Do

Begin

For J := I To N - 1 Do

Begin

Write(OutputFile, ResultMatrix[I][J], ' ')

End;

End;

Close(OutputFile);

End;

Function MatrixRead(Matrix: TMatrix; Vib: Integer): TMatrix;

Var

N: Integer;

InputFile: TextFile;

Path: String;

Begin

Na();

If Vib = 0 Then

Begin

N := EnterN();

Matrix := CreateTriangularMatrix(N);

MatrixRead := Matrix;

End

Else

Begin

Path := PTF();

AssignFile(InputFile, Path);

Reset(InputFile);

N := ReadSizeFile(Path);

MatrixRead := MatrixFile(N, Path);

End;

End;

Procedure VivodSum(Vib: Integer; ResultMatrix: TMatrix);

Begin

If Vib = 0 Then

PrintResultMatrixConsol(ResultMatrix)

Else

Begin

PrintResultMatrixFile(ResultMatrix);

End;

End;

Var

Matrix, Matrix1, Matrix2, ResultMatrix: TMatrix;

Vib1, Vib2: Integer;

Begin

DelProg();

Vib1 := InputChoice();

Matrix1 := MatrixRead(Matrix, Vib1);

Matrix2 := MatrixRead(Matrix, Vib1);

ResultMatrix := AddTriangularMatrices(Matrix1, Matrix2);

Vib2 := OutputChoice();

VivodSum(Vib2, ResultMatrix);

End.

**Код на С++:**

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

void delProg(){

cout << "Данная программа складывает две треугольные N порядка, у которых только

элементы над главной диагональю отличны от нуля.";

}

int inputChoice() {

int vib;

bool isIncorrect;

cout << "Введите 0, если хотите сделать ввод с консоли; 1, если хотите ввод из

файла. ";

do

{

isIncorrect = false;

cout << "Выбор: ";

cin >> vib;

if (cin.fail() || cin.peek() != '\n')

{

isIncorrect = true;

cout << "Ошибка ввода. ";

cin.clear();

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

}

if (!(isIncorrect) && (vib != 1) && (vib != 0))

{

isIncorrect = true;

cout << "Число должно быть либо 0, либо 1. ";

cin.clear();

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

}

}

while (isIncorrect);

return vib;

}

int enterN() {

int n;

bool isIncorrect;

do

{

isIncorrect = false;

cout << "Введите порядок матрицы: ";

cin >> n;

if (cin.fail() || cin.peek() != '\n')

{

isIncorrect = true;

cout << "Ошибка ввода. ";

cin.clear();

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

}

if (!(isIncorrect) && (n < 2))

{

isIncorrect = true;

cout << "Порядок матрицы должен быть числом, большим 1. ";

cin.clear();

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

}

} while (isIncorrect);

return n;

}

double\*\* createMatrixA(int n){

double\*\* a = new double\* [n];

int i;

for (i = 0; i < n; i++)

{

a[i] = new double[n];

}

return a;

}

double\*\* createTriangularMatrix(double\*\* a, int n){

bool isIncorrect;

int i;

int j;

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

if (j >= i)

{

do {

isIncorrect = false;

cout << "Введите a" << (i + 1) << (j + 1) << ": ";

cin >> a[i][j];

if (cin.fail() || cin.peek() != '\n')

{

isIncorrect = true;

cout << "Ошибка ввода. ";

cin.clear();

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

}

} while (isIncorrect);

}

else

{

a[i][j] = 0;

}

}

}

return a;

}

void outputPathInputCondition()

{

cout << "Введите путь к файлу, содержащему массив:\n";

}

bool isFileCorrect(string path)

{

bool isIncorrect = false;

string line;

fstream f(path);

if (!f.is\_open())

{

isIncorrect = true;

cout << "Нет файла по заданному пути, либо же он отсутствует или недоступен

для открытия. ";

}

else if ((path[path.length() - 1] != 't') &&

(path[path.length() - 2] != 'x') &&

(path[path.length() - 3] != 't') &&

(path[path.length() - 4] != '.'))

{

isIncorrect = true;

cout << "Недопустимое расширение файла. ";

}

f.close();

return isIncorrect;

}

string pTF()

{

bool isIncorrect;

string path;

do

{

cout << "Укажите путь к файлу:\n";

isIncorrect = false;

cin >> path;

isIncorrect = isFileCorrect(path);

if (isIncorrect)

{

cout << "Повторите ввод пути к файлу:\n";

}

} while (isIncorrect);

return path;

}

int readSizeFile(string path){

int n, i;

i = 0;

bool isIncorrect;

ifstream fin(path);

isIncorrect = false;

if (!fin.eof())

{

fin >> n;

if (fin.fail() || fin.peek() != '\n')

{

cout << "Ошибка считывания размера с файла. \n";

isIncorrect = true;

}

if (!isIncorrect && (n < 2))

{

cout << "Порядок матрицы должен быть числом, большим 1. \n";

isIncorrect = true;

}

}

fin.close();

if (isIncorrect)

{

n = enterN();

}

return n;

}

double\*\* matrixFile(string path, double\*\* a, int n){

ifstream fin(path);

fin.seekg(2, ios::beg);

int i;

int j;

for (i = 0; i < n; i++)

{

for (j = 0; j < n; j++)

{

fin >> a[i][j];

}

}

fin.close();

return a;

}

double\*\* matrixRead(int ch, int& n) {

double\*\* a;

string path;

if (ch == 0)

{

n = enterN();

a = createMatrixA(n);

a = createTriangularMatrix(a, n);

}

else

{

path = pTF();

n = readSizeFile(path);

a = createMatrixA(n);

outputPathInputCondition();

a = matrixFile(path, a, n);

}

return a;

}

int outputChoice() {

int vib;

bool isIncorrect;

cout << "Введите 0, если хотите сделать вывод в консоль; 1, если хотите вывод в

файл. ";

do {

isIncorrect = false;

cout << "Выбор: ";

cin >> vib;

if (cin.fail() || cin.peek() != '\n')

{

isIncorrect = true;

cout << "\nОшибка ввода. ";

cin.clear();

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

}

if (!(isIncorrect) && (vib != 1) && (vib != 0))

{

isIncorrect = true;

cout << "\nЧисло должно быть либо 0, либо 1. ";

cin.clear();

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

}

} while (isIncorrect);

return vib;

}

void printResultMatrixFile(double\*\* b, int n) {

bool positiveInCol;

string path;

path = pTF();

int j, i;

i = 0;

ofstream fout(path);

for (j = 0; j < n; j++)

{

positiveInCol = false;

for (i = 0; i < n; i++)

{

fout << b[i][j] << " ";

}

}

fout.close();

}

double\*\* createResultMatrix(double\*\* matrix1, double\*\* matrix2, int n) {

double\*\* b = new double\* [n];

int i;

int j;

for (i = 0; i < n; i++)

{

b[i] = new double[n];

for (j = i; j < n; j++)

{

b[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

return b;

}

void printResultMatrix(double\*\* b, int n) {

int i;

int j;

for (i = 0; i < n; i++)

{

for (j = i; j < n; j++)

{

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

}

cout << endl;

}

}

void vivodSum(int vib, int n, double\*\* b) {

if (vib == 0)

{

printResultMatrix(b, n);

}

else {

printResultMatrixFile(b, n);

}

}

int main() {

setlocale(LC\_ALL, "");

delProg();

int vib1, vib2, n;

double\*\* matrix1;

double\*\* matrix2;

double\*\* resultMatrix;

vib1 = inputChoice();

matrix1 = matrixRead(vib1, n);

matrix2 = matrixRead(vib1, n);

resultMatrix = createResultMatrix(matrix1, matrix2, n);

vib2 = outputChoice();

vivodSum(vib2, n, resultMatrix);

delete[] matrix1;

delete[] matrix2;

delete[] resultMatrix;

return 0;

}

**Код на Java:**

import java.util.Scanner;

import java.io.\*;

public class Main {

static Scanner scanner = new Scanner(System.in);

public static int inputChoice() {

int vib;

boolean isIncorrect;

vib = 0;

System.out.println("Введите 0, если хотите сделать ввод с консоли; 1, если

хотите ввод из файла. ");

do {

isIncorrect = false;

System.out.print("Выбор: ");

try {

vib = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("Ошибка ввода. ");

}

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

isIncorrect = true;

System.out.println("Число должно быть либо 0, либо 1. ");

}

} while (isIncorrect);

return vib;

}

public static int enterN() {

int n;

boolean isIncorrect;

n = 0;

do {

isIncorrect = false;

System.out.print("Введите порядок матрицы: ");

try {

n = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("Ошибка ввода. ");

}

if (!isIncorrect && (n < 2)) {

isIncorrect = true;

System.out.println("Порядок матрицы должен быть числом, большим 1.

");

}

} while (isIncorrect);

return n;

}

public static double[][] createTriangularMatrix(int n) {

double[][] a = new double[n][n];

boolean isIncorrect;

int i;

int j;

for (i = 0; i < n; i++) {

for (j = i ; j < n; j++) {

do {

isIncorrect = false;

System.out.println("Enter a[" + (i + 1) + "][" + (j + 1) + "]:

");

try {

a[i][j] = Double.parseDouble(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("Input error");

}

} while (isIncorrect);

}

}

return a;

}

public static String pTF() {

String path;

boolean isIncorrect;

do {

isIncorrect = false;

System.out.println("Укажите путь к файлу: ");

path = scanner.nextLine();

File file = new File(path);

if (!file.exists()) {

isIncorrect = true;

System.out.println("Такой файл не найден. ");

}

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

isIncorrect = true;

System.out.println("У файла должно быть расширение txt. ");

}

} while (isIncorrect);

return path;

}

public static int readSizeFile(String path) {

int n;

boolean isIncorrect;

n = 0;

isIncorrect = false;

try {

Scanner scannerFile = new Scanner(new File(path));

n = scannerFile.nextInt();

scannerFile.close();

} catch (Exception e) {

isIncorrect = true;

System.out.println("Ошибка при считывании порядка матрицы. ");

}

if (!isIncorrect && n < 2) {

isIncorrect = true;

System.out.println("Значение порядка матрицы в файле должно быть больше

1. ");

}

if (isIncorrect) {

System.out.println("Введите порядок матрицы с консоли. ");

n = enterN();

}

return n;

}

public static double[][] matrixFile(String path, int n) {

double a[][];

a = new double[n][n];

int i;

int j;

try {

Scanner scannerFile = new Scanner(new File(path));

for (i = 0; i < n; i++) {

for (j = i; j < n; j++) {

a[i][j] = scannerFile.nextDouble();

}

}

scannerFile.close();

} catch (Exception e) {

System.out.println("Ошибка при считывании файла. ");

}

return a;

}

public static double[][] matrixRead(int vib) {

double[][] a;

int n;

String path;

if (vib == 0) {

n = enterN();

a = createTriangularMatrix(n);

} else {

path = pTF();

n = readSizeFile(path);

a = matrixFile(path, n);

}

return a;

}

public static int outputChoice() {

int vib;

boolean isIncorrect;

vib = 0;

System.out.println("Введите 0, если хотите сделать вывод в консоль; 1, если

хотите вывод в файл. ");

do {

isIncorrect = false;

System.out.print("Выбор: ");

try {

vib = Integer.parseInt(scanner.nextLine());

} catch (Exception e) {

isIncorrect = true;

System.out.println("\nОшибка ввода. ");

}

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

isIncorrect = true;

System.out.println("Число должно быть либо 0, либо 1. ");

}

} while (isIncorrect);

return vib;

}

public static double[][] addTriangularMatrices(double[][] matrix1, double[][] matrix2) {

int n = matrix1.length;

double[][] resultMatrix = new double[n][n];

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

for (int j = i; j < n; j++) {

resultMatrix[i][j] = matrix1[i][j] + matrix2[i][j];

}

}

return resultMatrix;

}

public static void printResultMatrix(double[][] resultMatrix) {

int n = resultMatrix.length;

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

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

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

}

System.out.println();

}

}

public static void printResultMatrixFile(double[][] resultMatrix) {

int i;

int j;

String path;

path = pTF();

try (PrintWriter zapis = new PrintWriter(new FileWriter(path))) {

for (i = 0; i < resultMatrix[0].length; i++) {

for (j = 0; i < resultMatrix[0].length; i++) {

zapis.write(resultMatrix[i][j] + " ");

}

}

} catch (Exception e) {

System.out.println("Не удалось записать данные в файл. ");

}

}

public static void vivodSum(int vib, double[][] resultMatrix) {

if (vib == 0) {

printResultMatrix(resultMatrix);

}

else {

printResultMatrixFile(resultMatrix);

}

}

public static void main(String[] args) {

int vib1;

int vib2;

double[][] matrix1;

double[][] matrix2;

double[][] resultMatrix;

vib1 = inputChoice();

matrix1 = matrixRead(vib1);

matrix2 = matrixRead(vib1);

resultMatrix = addTriangularMatrices(matrix1, matrix2);

vib2 = outputChoice();

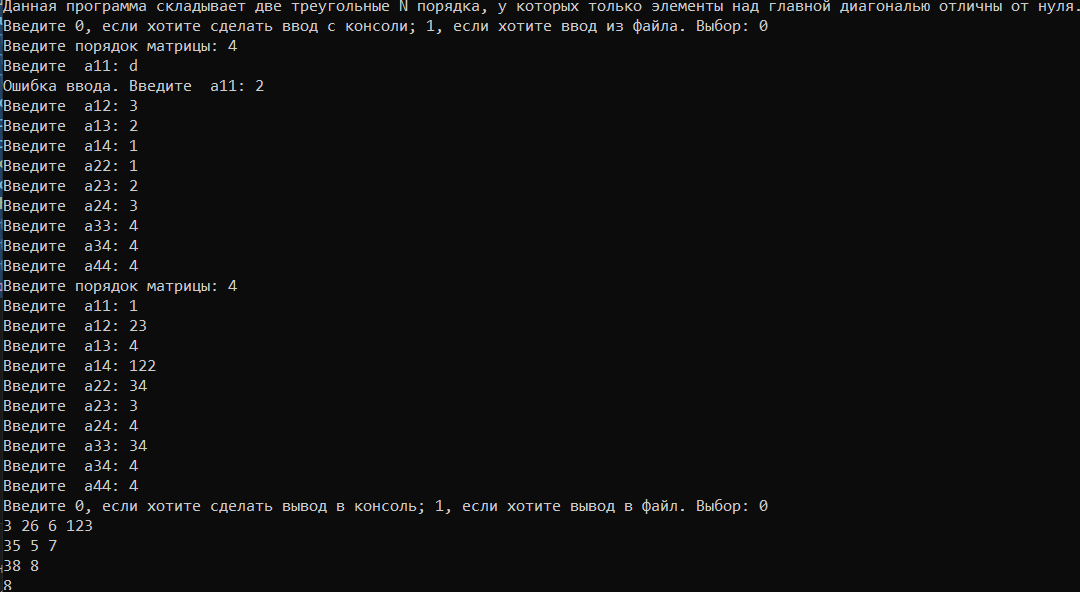
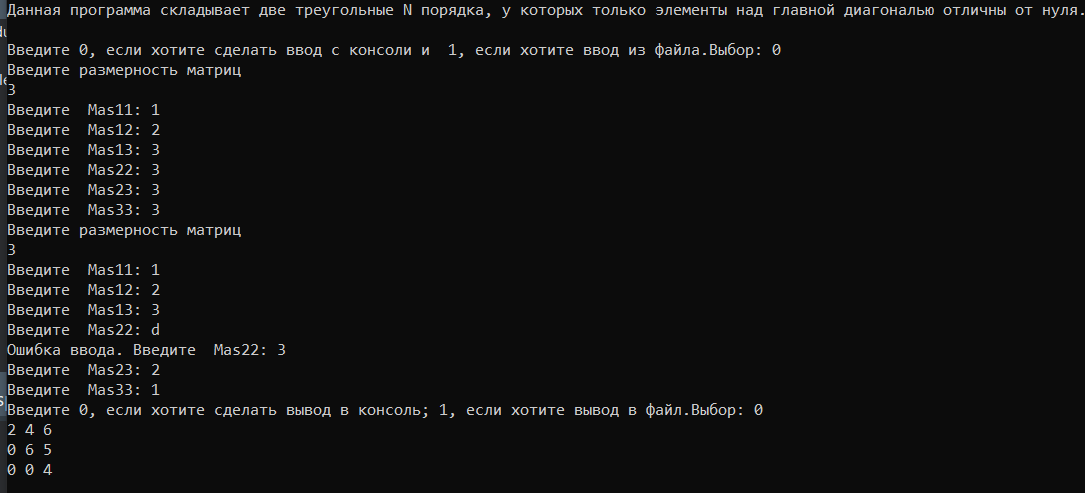
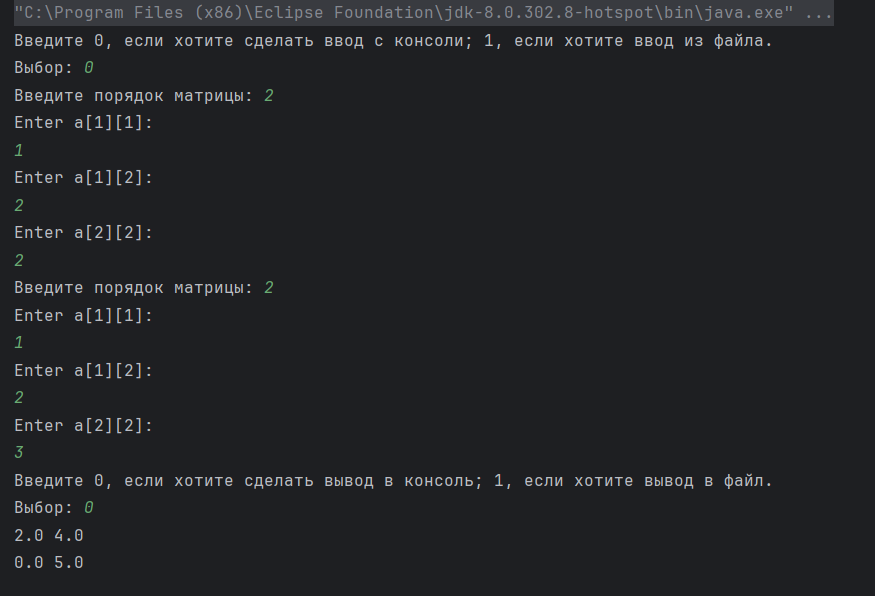
scanner.close();

vivodSum(vib2, resultMatrix);

}

}

**Скриншоты:**



**Блок-схема:**

