Министерство транспорта Российской Федерации

Федеральное государственное автономное образовательное

учреждение высшего образования

«Российский университет транспорта»

(ФГАОУ ВО РУТ(МИИТ), РУТ (МИИТ)

Институт транспортной техники и систем управления

Кафедра «Управление и защита информации»

Лабораторная работа № 10

по дисциплине: «Программирование и основы алгоритмизации»

на тему: «Многомерные массивы»

Выполнил: ст. гр. ТУУ-111

Плотников В.Р.

Вариант №3

08.06.2025

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

\_\_\_\_\_\_\_\_\_\_\_

(дата приёмки)

Москва – 2025 г.

**1. Цель работы**

Решить поставленную задачу программирования по разделу «Многомерные массивы». Продумать организацию удобного ввода элементов многомерного массива. Продумать схему корректного вывода элементов многомерного массива.

**2. Формулировка задачи**

Вычислить определитель (детерминант) любой матрицы размерностью большей, чем [3x3]. Учесть ограничение на размерность матрицы при указании соответствующего значения в процессе исполнения программы.

**3. Схема алгоритма**

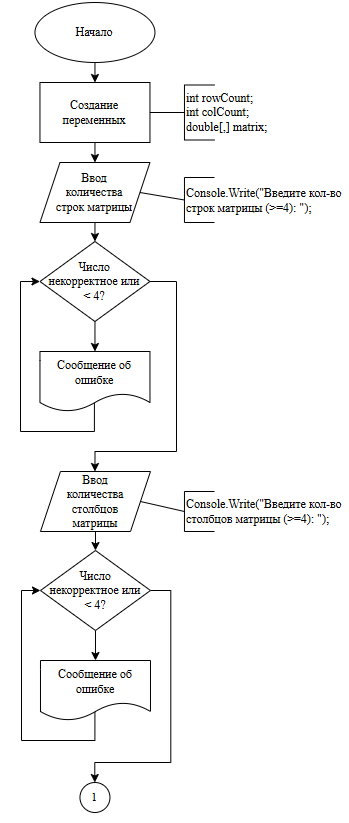


Рисунок 3.1 – Блок-схема алгоритма

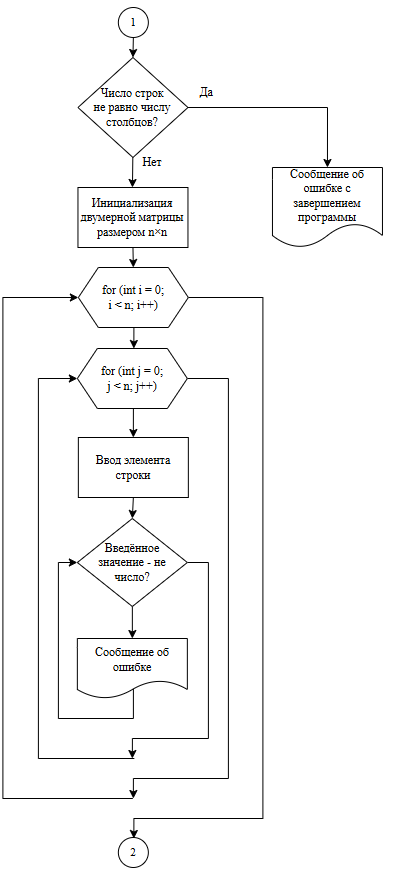


Рисунок 3.2 – Блок-схема алгоритма

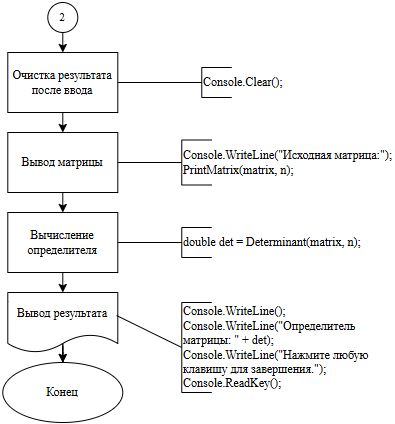


Рисунок 3.3 – Блок-схема алгоритма

**4. Подбор тестовых примеров**

4.1 Матрица с целыми числами

Ввод:

5

5

1 -2 3 0 4

0 5 -1 2 -3

4 0 1 -2 1

-1 3 0 2 2

2 -3 1 1 0

Ожидаемый результат:

Определитель матрицы: 461

4.2 Матрица с вещественными числами

Ввод:

4

4

1.7 -2.3 0.5 4.1

0.0 3.3 2.1 1.6

0.0 0.0 -1.5 2.0

0.0 0.0 0.0 3.0

Ожидаемый результат:

Определитель матрицы: -25.245

4.3 Ввод некорректного значения

Ввод:

4

4

1 2 Три

Ожидаемый результат:

Ошибка! Введите корректное числовое значение:

4.4 Неквадратная матрица

Ввод:

4

6

Ожидаемый результат:

Для определителя нужна квадратная матрица. Программа завершена.

**5. Код программы**

using System;

namespace Lab10

{

class Program

{

static void Main(string[] args)

{

int rowCount;

int colCount;

double[,] matrix;

Console.BackgroundColor = ConsoleColor.White;

Console.ForegroundColor = ConsoleColor.Black;

Console.Clear();

Console.Write("Введите кол-во строк матрицы (>=4): ");

while (!int.TryParse(Console.ReadLine(), out rowCount) || rowCount < 4)

{

Console.Write("Ошибка! Введите целое число, не меньше 4: ");

}

Console.Write("Введите кол-во столбцов матрицы (>=4): ");

while (!int.TryParse(Console.ReadLine(), out colCount) || colCount < 4)

{

Console.Write("Ошибка! Введите целое число, не меньше 4: ");

}

if (rowCount != colCount)

{

Console.WriteLine("Для определителя нужна квадратная матрица. Программа завершена.");

Console.ReadKey();

return;

}

int n = rowCount;

matrix = new double[n, n];

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

{

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

{

Console.Write($"Введите {j + 1}-й элемент {i + 1}-й строки: ");

while (!double.TryParse(Console.ReadLine(), out matrix[i, j]))

{

Console.Write("Ошибка! Введите корректное числовое значение: ");

}

}

}

Console.Clear();

Console.WriteLine("Исходная матрица:");

PrintMatrix(matrix, n);

double det = Determinant(matrix, n);

Console.WriteLine();

Console.WriteLine("Определитель матрицы: " + det);

Console.WriteLine("Нажмите любую клавишу для завершения.");

Console.ReadKey();

}

static void PrintMatrix(double[,] matrix, int n)

{

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

{

Console.Write("|");

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

{

Console.Write($"{matrix[i, j],8:0.##}");

}

Console.WriteLine(" |");

}

}

static double Determinant(double[,] matrix, int n)

{

if (n == 1) return matrix[0, 0];

if (n == 2) return matrix[0, 0] \* matrix[1, 1] - matrix[0, 1] \* matrix[1, 0];

double det = 0;

double[,] minor = new double[n - 1, n - 1];

for (int k = 0; k < n; k++)

{

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

{

int col = 0;

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

{

if (j == k) continue;

minor[i - 1, col] = matrix[i, j];

col++;

}

}

det += (k % 2 == 0 ? 1 : -1) \* matrix[0, k] \* Determinant(minor, n - 1);

}

return det;

}

}

}

**6. Тестирование**

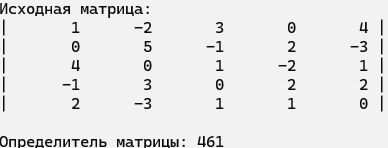


Рисунок 6.1 – Тестирование программы



Рисунок 6.2 – Тестирование программы

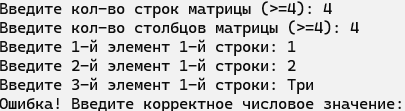


Рисунок 6.3 – Тестирование программы



Рисунок 6.4 – Тестирование программы

**7. Вывод**

В ходе выполнения работы была разработана программа, реализующая ввод, вывод и обработку квадратной вещественной матрицы размером не менее 4×4. Программа обеспечивает контроль исходных данных и корректно вычисляет определитель матрицы методом рекурсивного разложения по строке. Все требования методических указаний соблюдены.