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

Рундквист Е.А

Вариант №2

Проверил: к.т.н., доц. Сафронов А.И.

Москва – 2025 г.

1.Цель работы

Изучить и реализовать алгоритм обработки многомерных массивов использованием интегрированной среды разработки (*IDE*) *Microsoft Visual Studio* и языка программирования *Visual C.*

2. Формулировка задачи

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

3. Блок-схема алгоритма

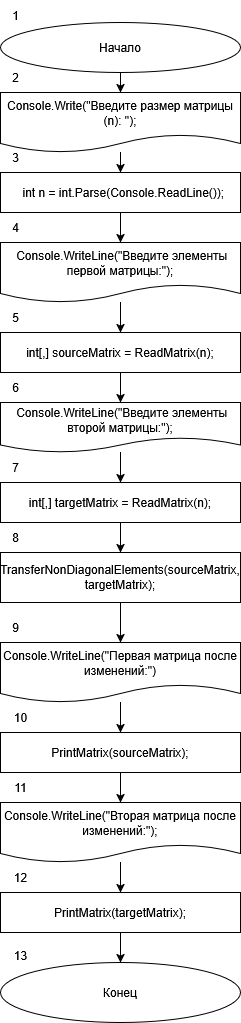


Рисунок 3.1 – Блок-схема алгоритма основной функции

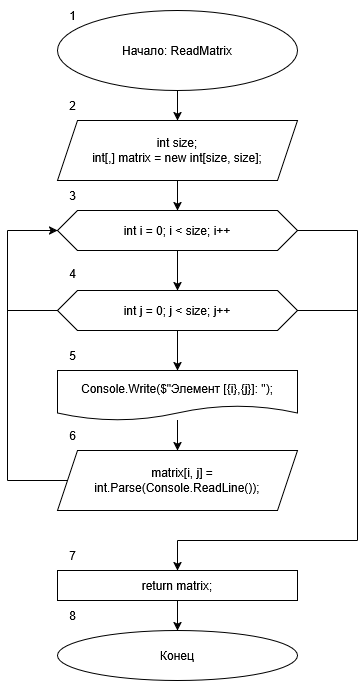


Рисунок 3.2 – Блок-схема алгоритма для получения изначальной матрицы

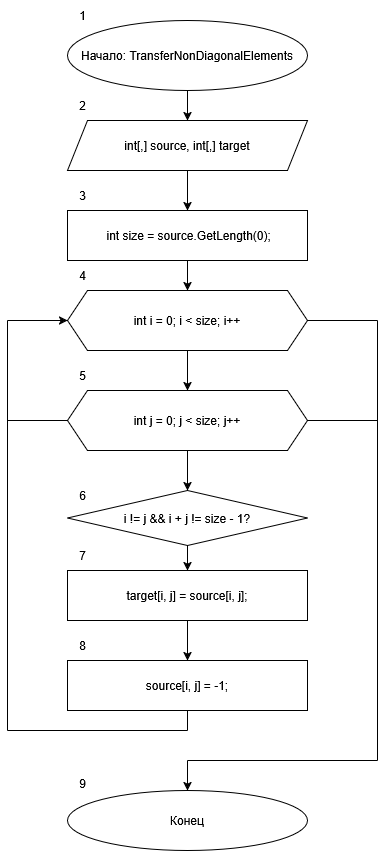


Рисунок 3.3 – Блок-схема алгоритма для преобразования матриц

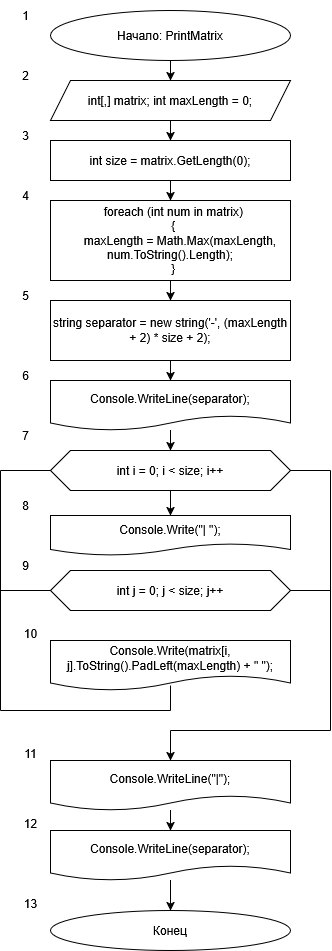


Рисунок 3.4 – Блок-схема алгоритма функции для вывода матрицы

4. Подбор тестовых примеров

1. Обычный пример матрицы 3 на 3 со случайно подобранными числами.

2.Пример матрица 2 на 2 где отсутствует разделение на диагональные и не диагональные элементы со случайно подобранными числами.

3. Много случайно подобранных многозначных чисел, чтобы показать корректность вывода матриц.

5. Листинг (код) программы

using System;

class Program

{

static void Main()

{

Console.Write("Введите размер матрицы (n): ");

int n = int.Parse(Console.ReadLine());

Console.WriteLine("Введите элементы первой матрицы:");

int[,] sourceMatrix = ReadMatrix(n);

Console.WriteLine("Введите элементы второй матрицы:");

int[,] targetMatrix = ReadMatrix(n);

TransferNonDiagonalElements(sourceMatrix, targetMatrix);

Console.WriteLine("Первая матрица после изменений:");

PrintMatrix(sourceMatrix);

Console.WriteLine("Вторая матрица после изменений:");

PrintMatrix(targetMatrix);

}

static int[,] ReadMatrix(int size)

{

int[,] matrix = new int[size, size];

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

{

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

{

Console.Write($"Элемент [{i},{j}]: ");

matrix[i, j] = int.Parse(Console.ReadLine());

}

}

return matrix;

}

static void TransferNonDiagonalElements(int[,] source, int[,] target)

{

int size = source.GetLength(0);

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

{

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

{

if (i != j && i + j != size - 1)

{

target[i, j] = source[i, j];

source[i, j] = -1;

}

}

}

}

static void PrintMatrix(int[,] matrix)

{

int size = matrix.GetLength(0);

int maxLength = 0;

foreach (int num in matrix)

{

maxLength = Math.Max(maxLength, num.ToString().Length);

}

string separator = new string('-', (maxLength + 2) \* size + 2);

Console.WriteLine(separator);

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

{

Console.Write("| ");

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

{

Console.Write(matrix[i, j].ToString().PadLeft(maxLength) + " ");

}

Console.WriteLine("|");

}

Console.WriteLine(separator);

}

}

6. Расчёт тестовых примеров на ПК

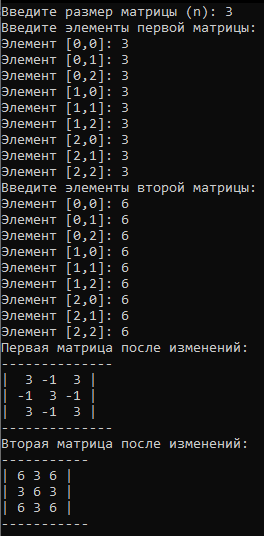


Рисунок 6.1 – Тестовый пример №1

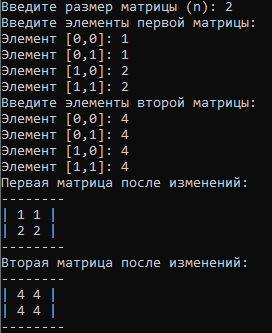


Рисунок 6.1 – Тестовый пример №2

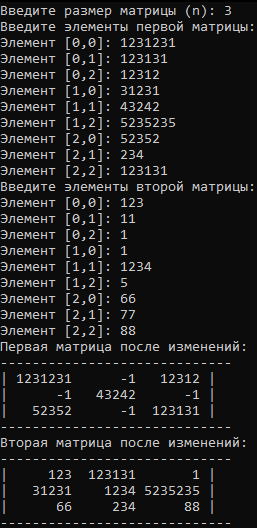


Рисунок 6.1 – Тестовый пример №3

7. Вывод по работе

В ходе выполнения лабораторной работы было изучено использование многомерных массивов в языке программирования *Visual C*. Были закреплены навыки работы с интегрированной средой разработки *Microsoft Visual Studio* и применения структурированных данных для решения практических задач.