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: «Двумерные массивы»

Выполнил: ст. гр. ТУУ-111

Михалин. А. В. Вариант №2

27.05.2025

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

11.06.2025

(дата приёмки)

Москва – 2025 г.

1. **Цель**

Изучить и реализовать «выкройку» недиагональных элементов одной матрицы и «вклеивание» их поверх элементов другой матрицы. «Выкройка» подразумевает выставление на позициях элементов значений, равных «-1».

1. **Формулировка задачи**

Реализовать «выкройку» недиагональных элементов одной матрицы и «вклеивание» их поверх элементов другой матрицы. «Выкройка» подразумевает выставление на позициях элементов значений, равных «-1». Результаты работы программы должны быть наглядно представлены в консоли, что позволит проверить корректность выполнения операций. В процессе разработки важно учитывать обработку возможных ошибок ввода данных и обеспечивать удобство взаимодействия с пользователем.

1. **Блок-схема алгоритма**



Рисунок 3.1.1 – Блок-схема алгоритма



Рисунок 3.1.2 – Блок-схема алгоритма



Рисунок 3.1.3 – Блок-схема алгоритма



Рисунок 3.1.4 – Блок-схема алгоритма



Рисунок 3.1.5 – Блок-схема алгоритма

1. **Подбор тестовых примеров**
   1. Тестовый пример

Входные данные:

Размер матриц:3

Матрица A:

Матрица В:

Ожидаемый результат:

Матрица A после изменений:

Матрица B после изменений:

* 1. Тестовый пример

Входные данные:

Размер матрицы: g

Ожидаемый вывод:

"Ошибка! Введите целое положительное число: "

Работа программы не прекращается

1. **Листинг (код) программы**

using System;

class Program

{

static void Main()

{

// Блок объявления всех переменных

int size; // Размер квадратных матриц

double[,] userMatrix1; // Пользовательская матрица A

double[,] userMatrix2; // Пользовательская матрица B

// Основной код программы

// Ввод размера матрицы

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

while (!int.TryParse(Console.ReadLine(), out size) || size <= 0)

Console.WriteLine("Ошибка! Введите целое положительное число: ");

// Ввод элементов матрицы A

userMatrix1 = new double[size, size];

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

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

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

{

Console.Write($"Элемент [{i + 1},{j + 1}]: ");

while (!double.TryParse(Console.ReadLine(), out userMatrix1[i, j]))

Console.WriteLine("Ошибка! Введите число: ");

}

// Ввод элементов матрицы B

userMatrix2 = new double[size, size];

Console.WriteLine("Введите элементы матрицы В" +

" :");

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

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

{

Console.Write($"Элемент [{i + 1},{j + 1}]: ");

while (!double.TryParse(Console.ReadLine(), out userMatrix2[i, j]))

Console.WriteLine("Ошибка! Введите число: ");

}

Console.WriteLine(" исходная матрица A:");

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

{

Console.Write("|");

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

Console.Write($"{userMatrix1[i, j],6:F2}");

Console.WriteLine(" |");

}

Console.WriteLine("\nисходня матрица В:");

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

{

Console.Write("|");

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

Console.Write($"{userMatrix2[i, j],6:F2}");

Console.WriteLine(" |");

}

Console.ReadKey(true);

Console.WriteLine("Вычисление результатов");

// Вычисление результата матрицы А

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

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

if (i != j)

{

userMatrix2[i, j] = userMatrix1[i, j];

userMatrix1[i, j] = -1;

}

// Вывод результатов

Console.WriteLine(" матрица A новая:");

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

{

Console.Write("|");

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

Console.Write($"{userMatrix1[i, j],6:F2}");

Console.WriteLine(" |");

}

Console.WriteLine("\nМатрица В новая:");

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

{

Console.Write("|");

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

Console.Write($"{userMatrix2[i, j],6:F2}");

Console.WriteLine(" |");

}

Console.ReadKey(true);

}

}

1. **Расчет тестовых примеров на ПК**

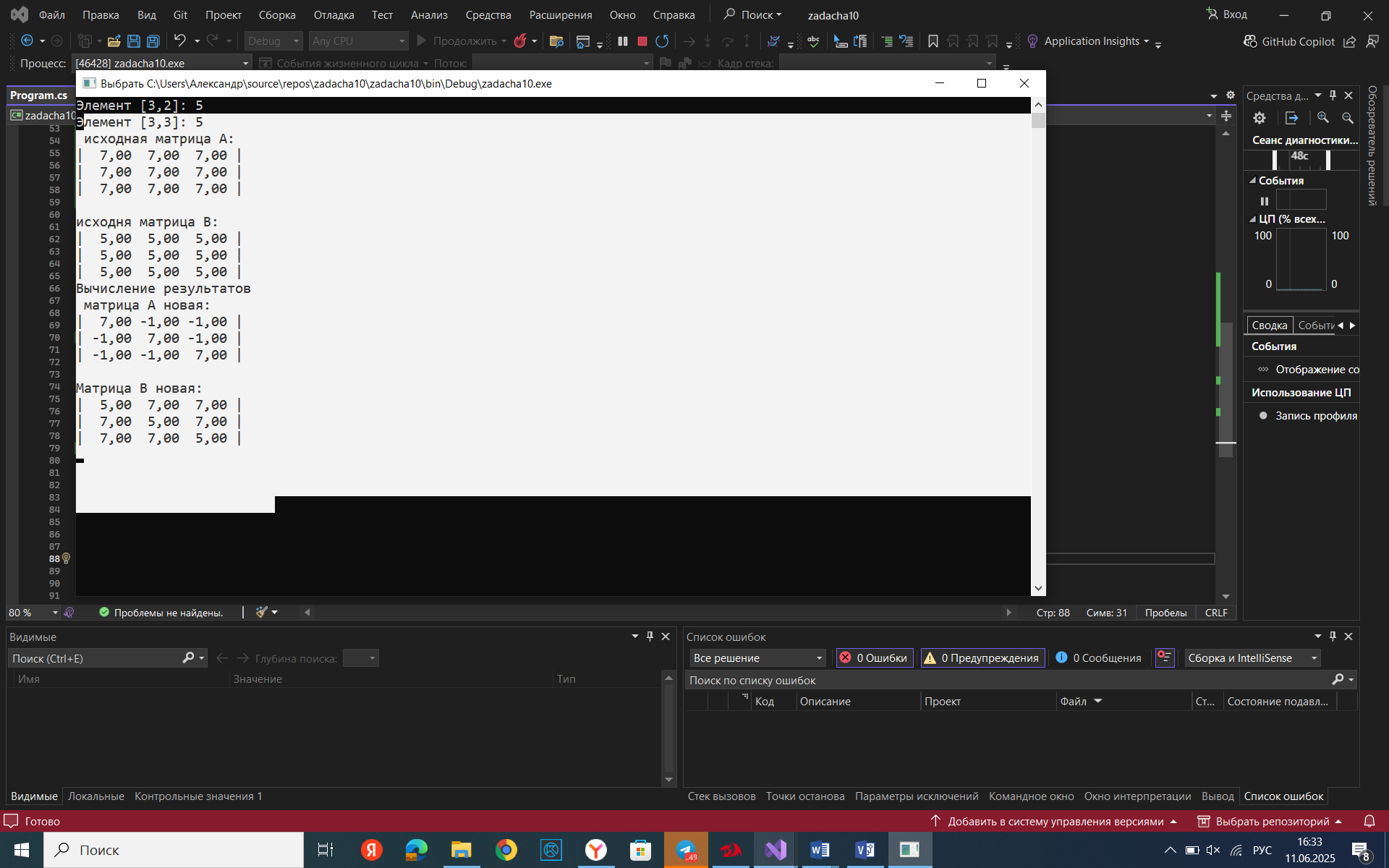
****

Рисунок 6.1.1 – Тестовый пример №1

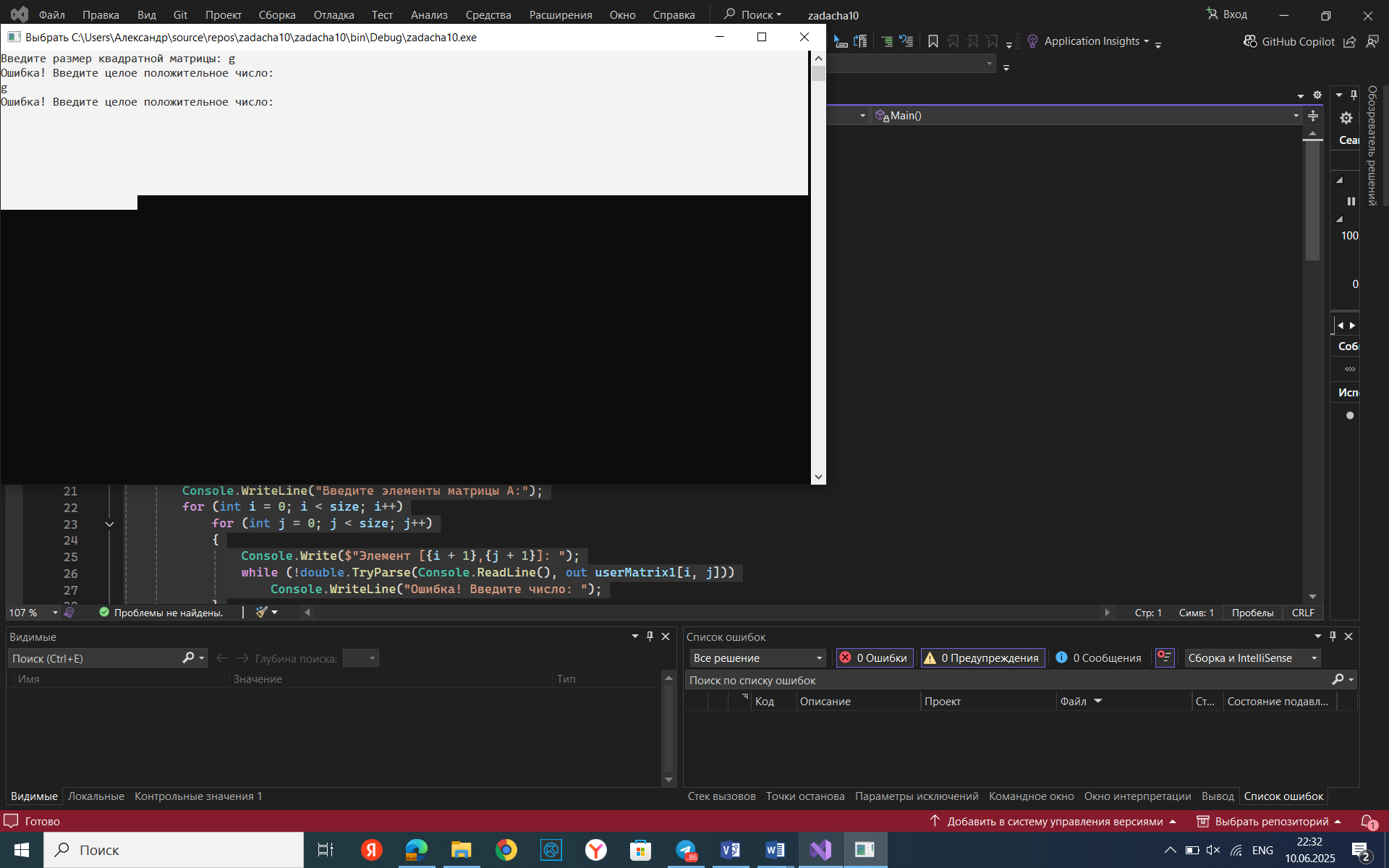


Рисунок 6.1.2 – Тестовый пример №2

1. **Вывод**

Была изучена и реализована «выкройка» недиагональных элементов одной матрицы и «вклеивание» их поверх элементов другой матрицы. Получены новые навыки работы в интегрированной среде разработки *Microsoft Visual Studio.*