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

Богомолов В.Н.

Вариант №1

27.03.2025

(дата выполнения)

Проверил: к.т.н., доц. Сафронов А.И.

27.03.2025

(дата приёмки)

Москва – 2025 г.

1. **Цель**

Изучить и реализовать алгоритм поэлементного умножения матриц на языке C#, используя многомерные массивы. Особое внимание уделяется работе с единичной матрицей и пользовательским вводом данных.

1. **Формулировка задачи**

В рамках данной работы необходимо изучить и реализовать алгоритм поэлементного умножения матриц на языке программирования *C#*, используя многомерные массивы. Основное внимание уделяется работе с единичной матрицей, которая представляет собой квадратную матрицу, где все элементы на главной диагонали равны единице, а остальные элементы равны нулю. Программа должна обеспечивать корректный пользовательский ввод данных, включая размер матрицы и её элементы, а также выполнять поэлементное умножение единичной матрицы на пользовательскую матрицу. 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**

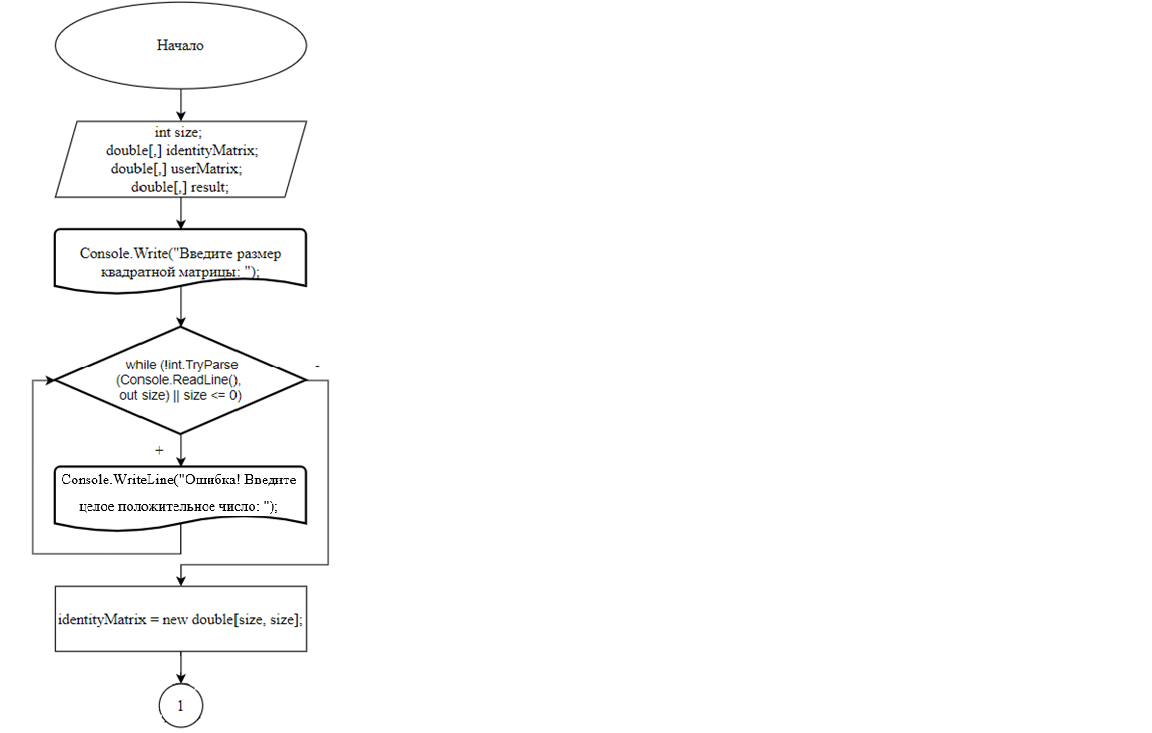


Рисунок 3.1.1 – Блок-схема алгоритма

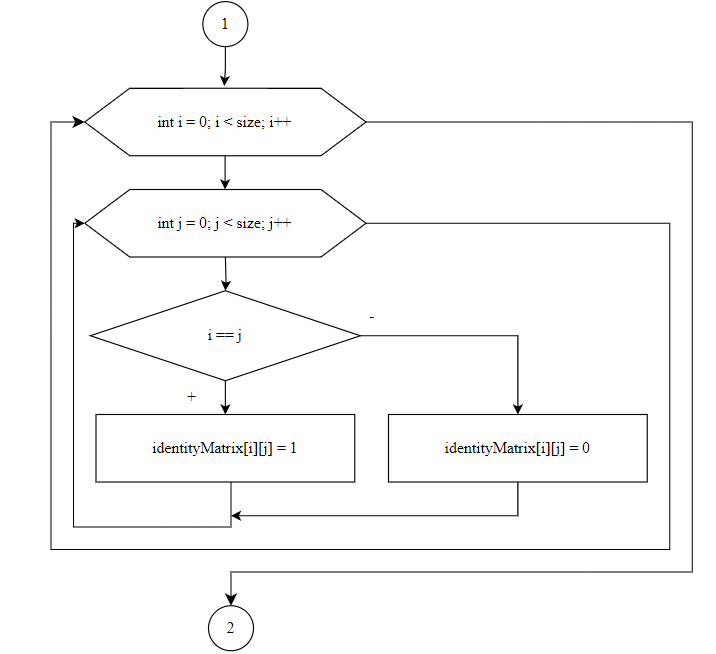


Рисунок 3.1.2 – Блок-схема алгоритма

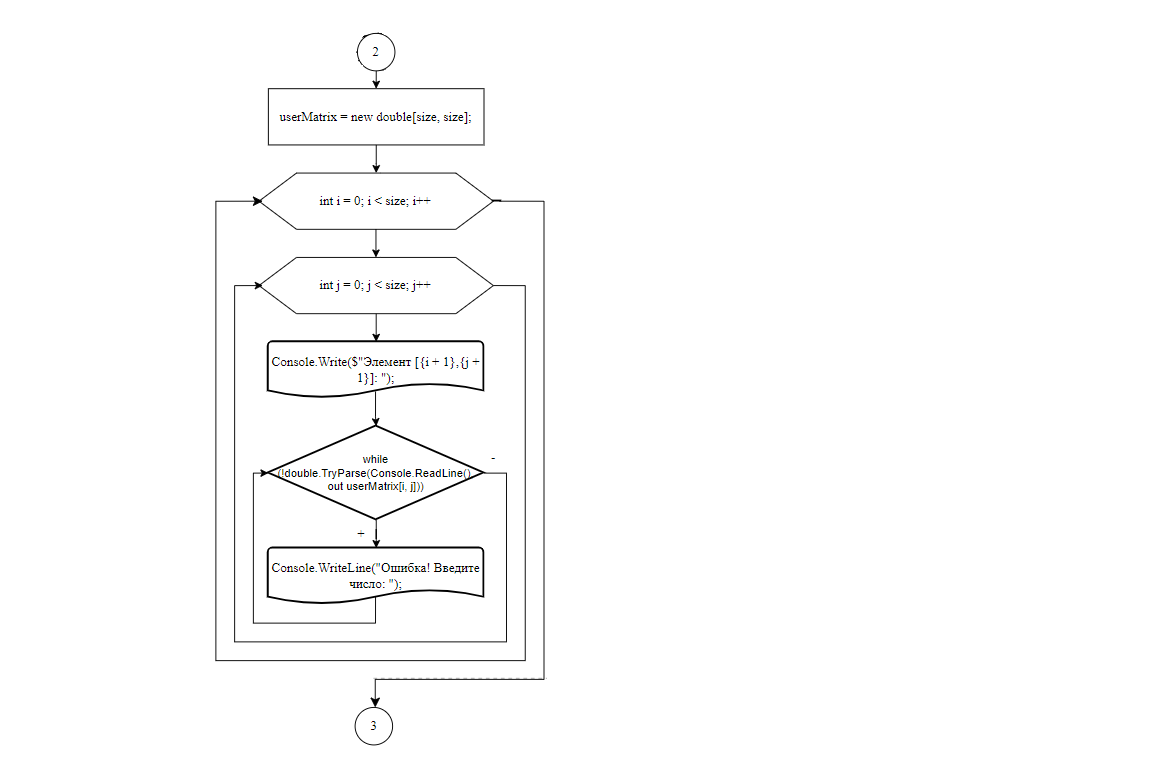


Рисунок 3.1.3 – Блок-схема алгоритма

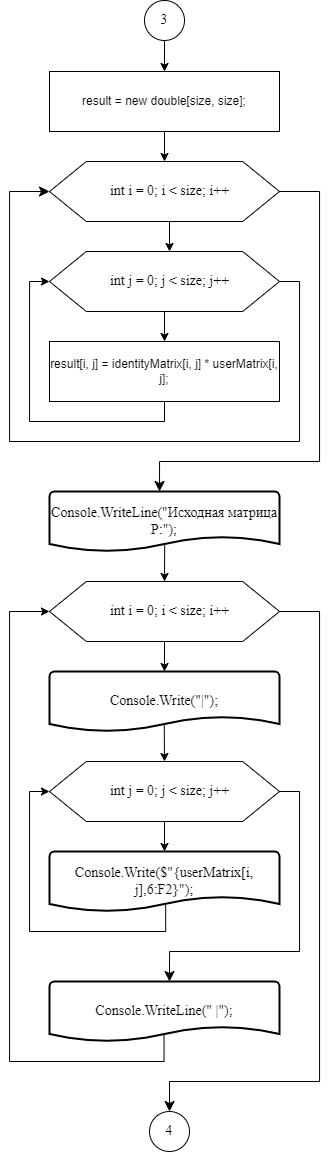


Рисунок 3.1.4 – Блок-схема алгоритма

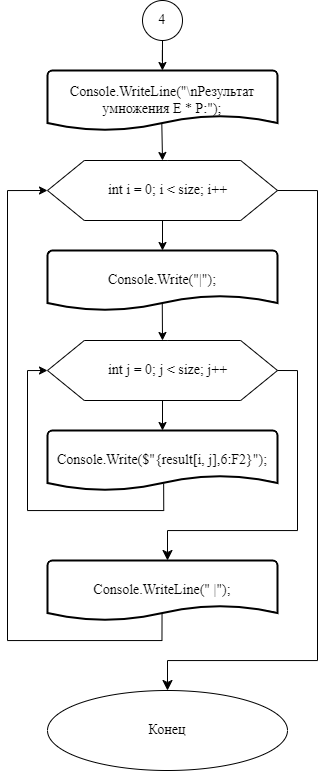


Рисунок 3.1.5 – Блок-схема алгоритма

1. **Подбор тестовых примеров**
   1. Тестовый пример №1

Входные данные:

Размер матрицы: 2

Матрица P:

Элемент [1,1]: 1

Элемент [1,2]: 2

Элемент [2,1]: 3

Элемент [2,2]: 4

Поэлементное перемножение:

1\*1=1

2\*0=0

3\*0=0

4\*1=0

Ожидаемый результат:

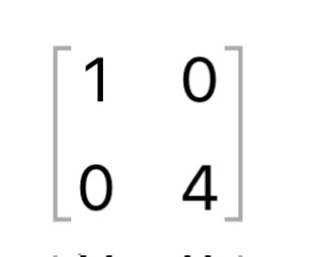


Рисунок 4.1.1 – Тестовый пример №1

* 1. Тестовый пример №2

Входные данные:

Размер матрицы: 3

Матрица P:

Элемент [1,1]: 2

Элемент [1,2]: 0

Элемент [1,3]: 5

Элемент [2,1]: 1

Элемент [2,2]: 3

Элемент [2,3]: 4

Элемент [3,1]: 6

Элемент [3,2]: 7

Элемент [3,3]: 8

Поэлементное перемножение:

2\*1=2

0\*0=0

5\*0=0

1\*0=0

3\*1=3

4\*0=0

6\*0=0

7\*0=0

8\*1=8

Ожидаемый результат:

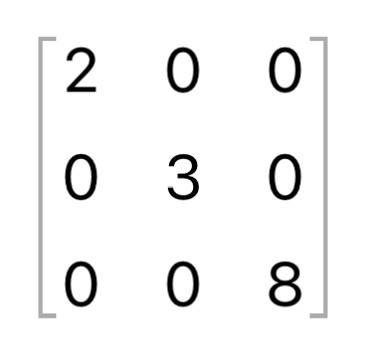


Рисунок 4.2.1 – Тестовый пример №2

* 1. Тестовый пример №3

Входные данные:

Размер матрицы: abc => затем 2 (исправление)

Элемент [1,1]: a => затем 2 (исправление)

Элемент [1,2]: 2

Элемент [2,1]: 3

Элемент [2,2]: 4

Ожидаемый результат: Программа требует корректные данные до их ввода.

1. **Листинг (код) программы**

using System;

class Program

{

static void Main()

{

// Блок объявления всех переменных

int size; // Размер квадратной матрицы

double[,] identityMatrix; // Единичная матрица E

double[,] userMatrix; // Пользовательская матрица P

double[,] result; // Результат умножения E \* P

// 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: ");

}

// Инициализация единичной матрицы

identityMatrix = new double[size, size];

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

{

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

{

identityMatrix[i, j] = (i == j) ? 1 : 0;

}

}

// Ввод элементов матрицы P

userMatrix = new double[size, size];

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

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 userMatrix[i, j]))

{

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

}

}

}

// Вычисление результата

result = new double[size, size];

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

{

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

{

result[i, j] = identityMatrix[i, j] \* userMatrix[i, j];

}

}

// Вывод результатов

Console.WriteLine("Исходная матрица P:");

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

{

Console.Write("|");

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

{

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

}

Console.WriteLine(" |");

}

Console.WriteLine("\nРезультат умножения E \* P:");

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

{

Console.Write("|");

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

{

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

}

Console.WriteLine(" |");

}

Console.ReadKey(true);

}

}

1. **Расчет тестовых примеров на ПК**

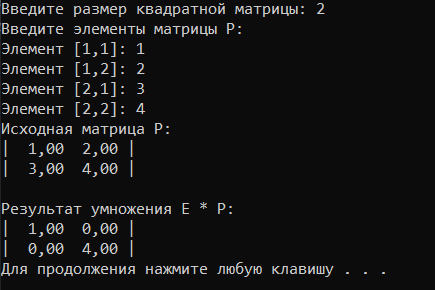
****

Рисунок 6.1.1 – Тестовый пример №1

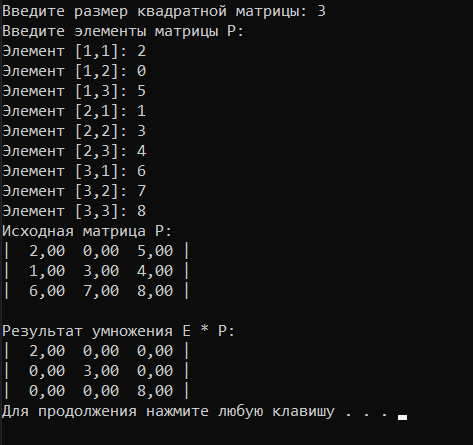
****

Рисунок 6.1.2 – Тестовый пример №2

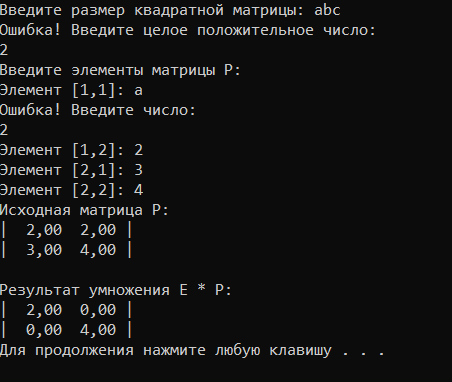


Рисунок 6.1.3 – Тестовый пример №3

1. **Вывод**

В ходе выполнения лабораторной работы была разработана программа на языке *C#*, реализующая алгоритм поэлементного умножения единичной матрицы на пользовательскую. В рамках задачи успешно выполнены следующие этапы: генерация единичной матрицы, элементы которой на главной диагонали равны единице, а остальные — нулю; организация корректного ввода данных с клавиатуры, включая проверку на целочисленный размер матрицы и числовые значения её элементов; наглядный вывод исходной и результирующей матриц с визуальным разделением элементов символом |. Программа прошла тестирование на нескольких примерах, подтвердив корректность работы алгоритма. В процессе разработки были закреплены навыки работы с многомерными массивами, а также освоены методы обработки пользовательского ввода и форматированного вывода данных в консольном приложении.