Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего образования

ПЕРМСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

(ФГАОУ ВО ПНИПУ)

УДК 004.652.4

ОТЧЕТ

О ЛАБОРАТОРНОЙ РАБОТЕ

По дисциплине «Дискретная математика» на тему:

РАЗРАБОТКА ПРОГРАММЫ ДЛЯ ПРОВЕРКИ СВОЙСТВ МАТРИЦ

Студент 2 курса группы РИС-24-3Б,  
специальности   
09.03.04 Программная инженерия \_\_\_\_\_\_\_\_\_\_ Егор Михайлович Носков  
  
  
Старший преподаватель

кафедры ИТАС \_\_\_\_\_\_\_\_\_\_ Рустамханова Гульшат Ильдаровна

Пермь 2025

# Постановка задачи

Разработать программу для анализа свойств бинарных отношений, представленных в виде матриц смежности. Программа должна выполнять следующие последовательные действия:

1. Позволять пользователю ввести квадратную матрицу бинарного отношения;
2. Проверяет матрицу на основные свойства бинарных отношений:

* Рефлексивность;
* Симметричность;
* Антисимметричность;
* Транзитивность;
* Связанность графа (для любой пары различных вершин существует хотя бы одно ребро между ними).

1. Классифицирует отношение по типам:

* Отношение эквивалентности
* Отношение порядка
* Отношение строгого порядка

1. Выводит результаты проверки всех свойств в наглядной форме.

# Описание классов и методов программы

Программа анализа свойств матриц была разработана на языке программирования C# как консольное приложение.

Задача: Управление основным потоком выполнения программы и взаимодействием с пользователем.

Основные функции:

- Ввод размерности матрицы

- Построчный ввод элементов матрицы

- Вызов методов проверки свойств и классификации отношений

- Вывод результатов анализа

Основные методы класса:

- Main() - точка входа в программу

- IsReflexive() - проверка рефлексивности

- IsSymmetric() - проверка симметричности

- IsAntiSymmetric() - проверка антисимметричности

- IsTransitive() - проверка транзитивности

- IsConnect() - проверка связанности графа

- IsEquivalenceRelation() - проверка на отношение эквивалентности

- IsOrderRelation() - проверка на отношение порядка

- IsStrictOrderRelation() - проверка на отношение строгого порядка

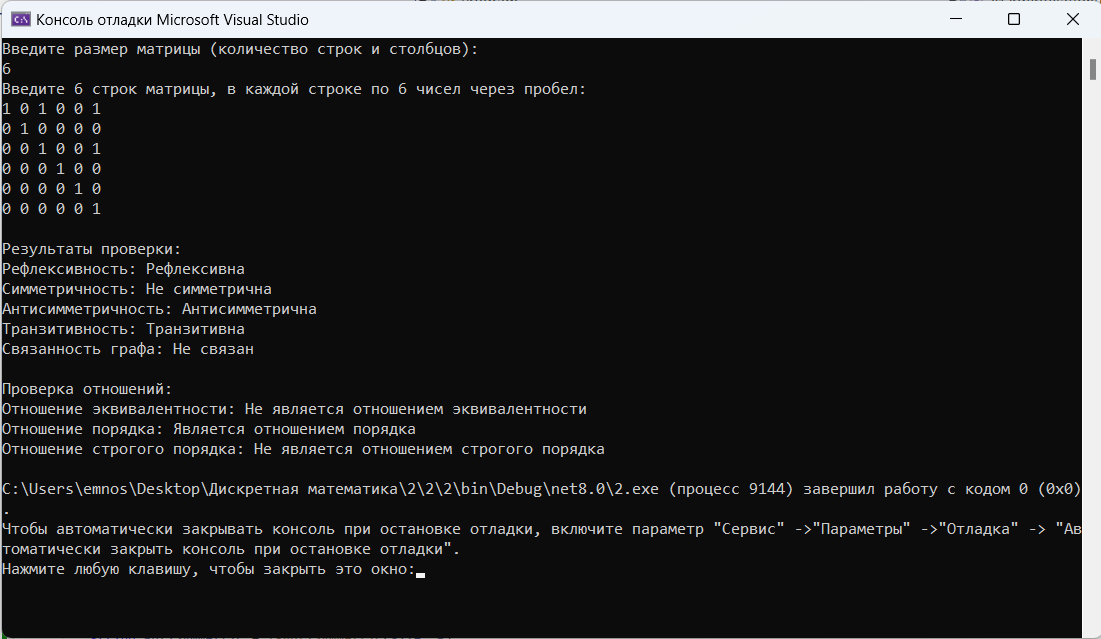
# Результат работы программы

После запуска программы появляется консольное окно с последовательным запросом данных (Изображение 1):

1. Ввод размерности матрицы;

2. Построчный ввод элементов матрицы через пробел;

3. Автоматический анализ и вывод результатов проверки всех свойств;



Изображение 1 – Процесс работы программы

# Приложение А

using System;

class Program

{

static void Main()

{

Console.WriteLine("Введите размер матрицы (количество строк и столбцов):");

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

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

Console.WriteLine($"Введите {size} строк матрицы, в каждой строке по {size} чисел через пробел:");

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

{

string[] input = Console.ReadLine().Split(' ');

matrix[i] = new int[size];

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

{

matrix[i][j] = int.Parse(input[j]);

}

}

Console.WriteLine("\nРезультаты проверки:");

Console.WriteLine($"Рефлексивность: {IsReflexive(size, matrix)}");

Console.WriteLine($"Симметричность: {IsSymmetric(size, matrix)}");

Console.WriteLine($"Антисимметричность: {IsAntiSymmetric(size, matrix)}");

Console.WriteLine($"Транзитивность: {IsTransitive(size, matrix)}");

Console.WriteLine($"Связанность графа: {IsConnect(size, matrix)}");

}

static string IsReflexive(int size, int[][] a)

{

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

{

if (a[i][i] == 0)

return "Не рефлексивна";

}

return "Рефлексивна";

}

static string IsSymmetric(int size, int[][] a)

{

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

{

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

{

if (a[i][j] != a[j][i])

return "Не симметрична";

}

}

return "Симметрична";

}

static string IsAntiSymmetric(int size, int[][] a)

{

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

{

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

{

if (i != j && a[i][j] == 1 && a[j][i] == 1)

return "Не антисимметрична";

}

}

return "Антисимметрична";

}

static string IsTransitive(int size, int[][] a)

{

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

{

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

{

if (a[i][j] == 1)

{

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

{

if (a[j][k] == 1 && a[i][k] == 0)

return "Не транзитивна";

}

}

}

}

return "Транзитивна";

}

static string IsConnect(int size, int[][] a)

{

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

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

if (i != j && a[i][j] == 0 && a[j][i] == 0)

return "Не связан";

return "Связан";

}

}

# ССЫЛКА НА GITHUB

