Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«**Пермский национальный исследовательский политехнический университет»**

Кафедра «Информационные технологии и автоматизированные системы»

**ОТЧЕТ**

Дисциплина: «Дискретная математика»

# Тема: Свойства отношений

# Семестр 3

Выполнил работу

Студент группы РИС-22-1Б

Зырянов Р.А.

Проверила

Рустамханова Г.И.

Г. Пермь-2023

**Проектирование программы**

Целью разрабатываемого приложения является анализ входных данных на соответствие заданным свойствам бинарных отношений с последующим информированием пользователя о результатах.

Для достижения этой цели программа будет выполнять следующие функции:

* Импорт исходных данных из текстового документа
* Преобразование полученной текстовой информации в матричный формат
* Поэтапная проверка матрицы на наличие таких характеристик как рефлексивность, симметричность, транзитивность и т.д.
* Формирование итогового вердикта о соответствии или несоответствии входных данных заданным критериям
* Вывод сформированных результатов анализа пользователю

Работа программы будет заключаться в последовательном выполнении следующих операций:

* Получение данных путем считывания информации из текстового файла
* Подготовка данных: удаление лишних символов и преобразование в числовой формат
* Формирование матрицы чисел на основе подготовленных данных
* Организация циклической проверки матрицы по заданным критериям
* Запись результатов проверки в отдельный массив
* Преобразование числовых результатов в текстовые сообщения
* Вывод полученных текстовых сообщений в консоль

Таким образом, разрабатываемое приложение автоматизирует процесс анализа данных на предмет соответствия указанным свойствам.

**Тестирование программы**

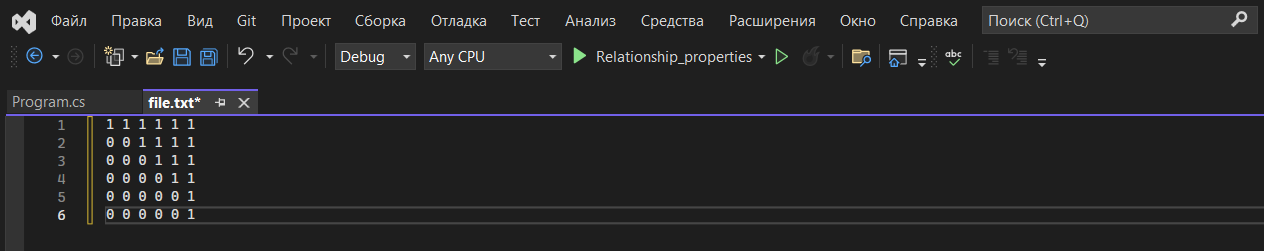


Рисунок 1 – используемая матрица

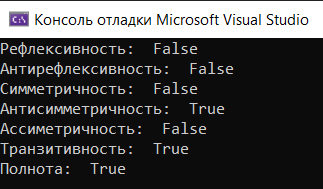


Рисунок 2 – результат работы программы

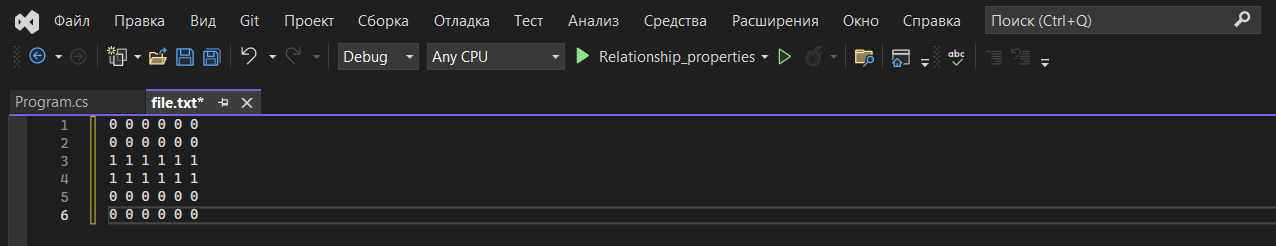


Рисунок 3 – используемая матрица

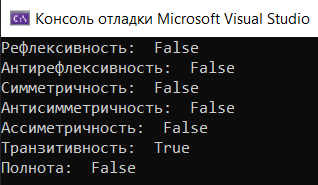


Рисунок 4 – результат работы программы

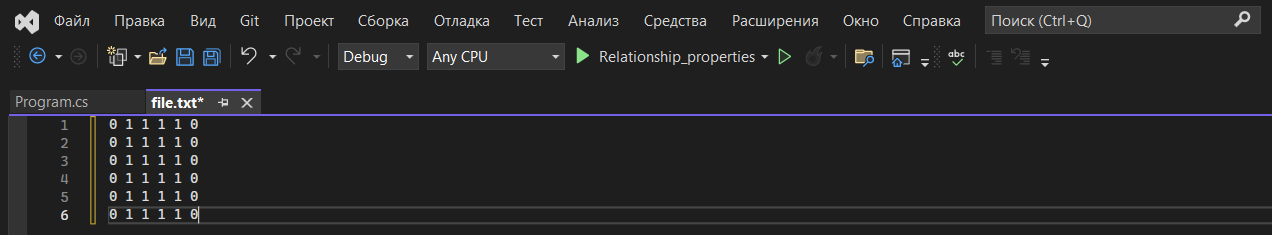


Рисунок 5 – используемая матрица

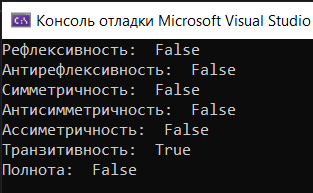


Рисунок 6 – результат работы программы

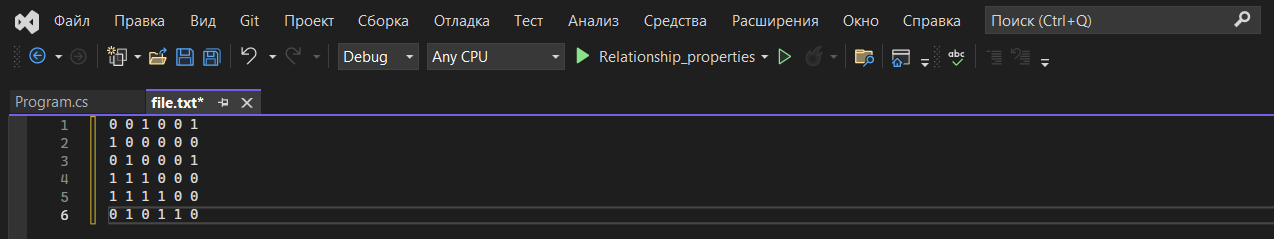


Рисунок 7 – используемая матрица

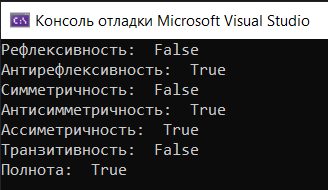


Рисунок 8 – результат работы программы

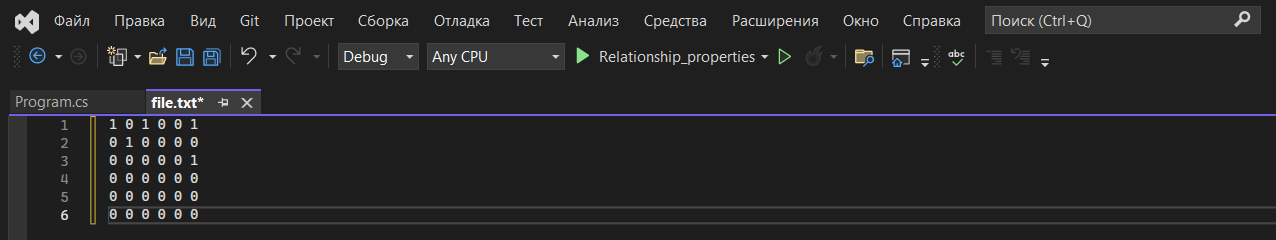


Рисунок 9 – используемая матрица

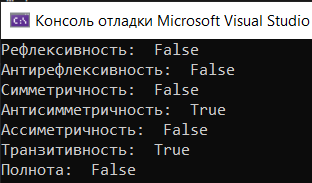


Рисунок 10 – результат работы программы

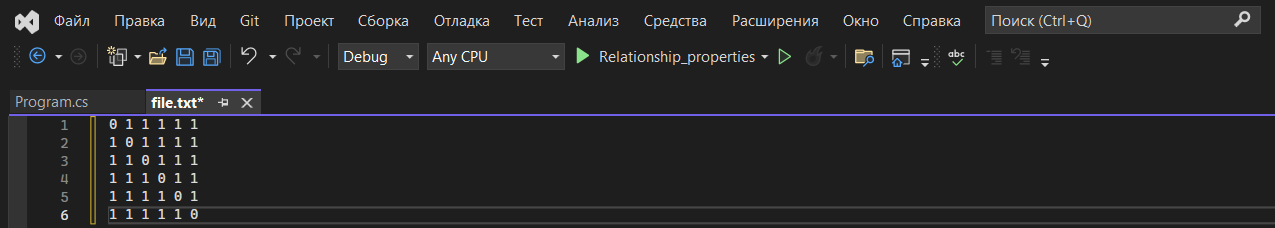


Рисунок 11 – используемая матрица

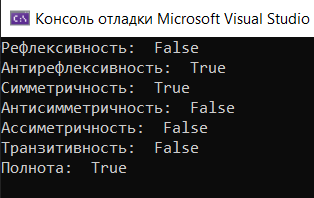


Рисунок 12 – результат работы программы

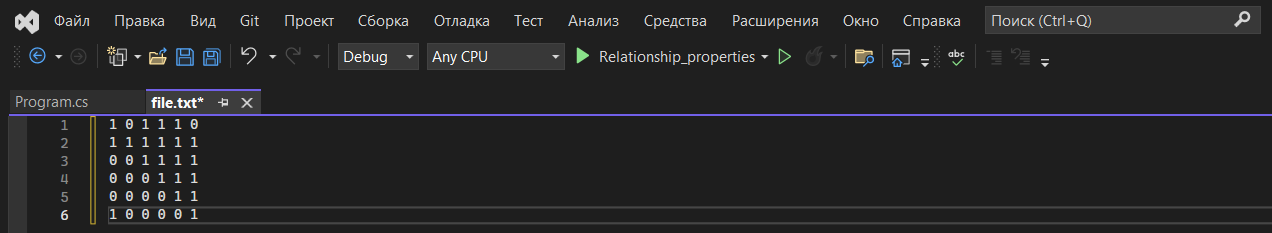


Рисунок 13 – используемая матрица

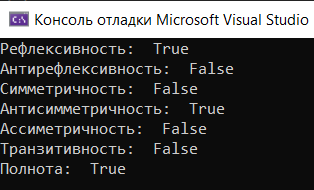


Рисунок 14 – результат работы программы

**Код программы**

using System.IO;

using System.Numerics;

using System.Reflection.Metadata.Ecma335;

using System.Runtime.Intrinsics;

string[] strs;

string path = "C://Users//Владимир//Desktop//Политех//Дискретная математика//Лабы//Свойства отношений//Relationship\_properties//file.txt";

strs = File.ReadAllLines(path);

for (int i = 0; i < strs.Length; i++)

strs[i] = strs[i].Replace(" ", "");

int[,] matrix = new int[6, 6];

for (int i = 0; i < strs.Length; i++)

{

for (int j = 0; j < strs[i].Length; j++)

matrix[i,j] = int.Parse(strs[i][j].ToString());

}

bool[] properties = new bool[7] { true, true, true, true, true, true, true };

for (int i = 0; i < strs.Length; i++)

{

for (int j = 0; j < strs[i].Length; j++)

{

if (!((i == j && matrix[i, j] == 1) || (i != j && (matrix[i, j] == 0 || matrix[i, j] == 1))))

properties[0] = false; //рефлексивное

if (!((i == j && matrix[i, j] == 0) || (i != j && (matrix[i, j] == 1 || matrix[i, j] == 0))))

properties[1] = false; //антирефлексивное

if (!(matrix[i,j] == matrix[j, i]))

properties[2] = false; //симметричное

if (!((i == j && (matrix[i, j] == 1 || matrix[i, j] == 0)) || (i != j && (matrix[i, j] != matrix[j, i] || matrix[i, j] == 0 && matrix[j, i] == 0))))

properties[3] = false; //антисимметричное

if (!((i == j && matrix[i, j] == 0) || (i != j && (matrix[i, j] != matrix[j, i] || matrix[i, j] == 0 && matrix[j, i] == 0))))

properties[4] = false; //асимметричное

if (matrix[i, j] == 1)

{

for (int k = 0; k < 6; k++)

{

if (matrix[j, k] == 1 && matrix[i, k] != 1)

properties[5] = false; //транзитивность

}

}

if (!((i == j && (matrix[i, j] == 0 || matrix[i, j] == 1)) || (i != j && (matrix[i, j] == 1 || matrix[j, i] == 1))))

properties[6] = false; //полнота

}

}

string numberToWord(int i)

{

switch (i)

{

case 0:

return "Рефлексивность: ";

case 1:

return "Антирефлексивность: ";

case 2:

return "Симметричность: ";

case 3:

return "Антисимметричность: ";

case 4:

return "Ассиметричность: ";

case 5:

return "Транзитивность: ";

case 6:

return "Полнота: ";

default:

return "";

}

}

for (int i = 0; i < 7; i++)

Console.WriteLine($"{numberToWord(i)} {properties[i]}");