Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра информационных технологий и автоматизированных систем

Дискретная математика

Лабораторная работа № 2

Тема: «Проверка свойств матрицы»

Выполнил: студент группы ИВТ-22-1б

Игошев Матвей Иванович

Проверил: Ст. Преподаватель кафедры ИТАС

Рустамханова Г.И.

г. Пермь – 2023

Оглавление

Цель работы 3

Задачи работы 4

Этапы выполнения 5

Заключение 6

Список используемой литературы 7

# Цель работы

Создать консольное приложение, которое будет проверять, вводимую из файла, матрицу на выполнения основных бинарных отношений.

# Задачи работы

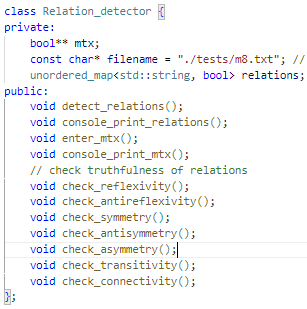
1. Реализовать ввод матрицы из файла
2. Продумать класс обработчика матрицы
3. Реализовать методы для определения выполнения свойств
4. Протестировать приложение

# Этапы выполнения

# **Объяснение работы кода программы**

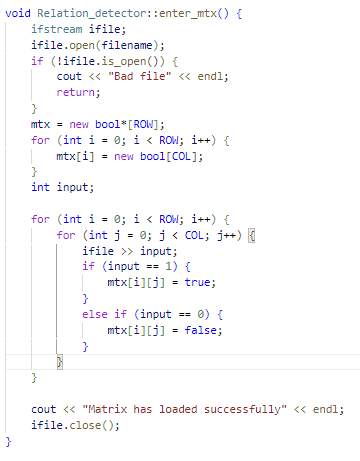
Реализация приложения может быть представлена через класс, точнее весь функционал можно упаковать в класс. Класс объединит в себе все нужные переменные: данные матрицы, информация о том выполняются ли бинарные отношения и название файла.

Класс выглядит следующим образом:



Теперь осталось лишь реализовать, все указанные функции. Как видно из определения класса матрица храниться в виде динамического массива булевых переменных. С помощью поля filename класс обращается к файлу. Поле relations хранит в себе пары ключ значение в виде «название отношения » : «логическая переменная». Функции, у которых в начале имени прописано «check», очевидно проверяют выполняется то или иное отношение в матрице. Функция detect\_relations выполняет все функции c припиской «check».

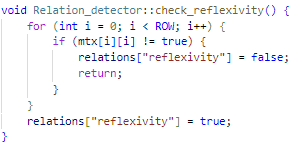
Одной из самых главных функций является функция enter\_mtx(), которая обеспечивает ввод матрицы в программу из файла. Выглядит ее реализация таким образом:



В данной функции по имени файла, программа получает массив. Константы ROW и COL равны 6 и являются количеством строк и столбцов соответственно. После из массива получается массив былевых переменных.(важно отметить что значения должны быть равны либо 1, либо 0) . Если все завершилось хорошо, то программа выводит сообщение в консоль.

Получив матрицу, теперь необходимо проверить ее на выполнение отношений. Для этого предусмотрена функция detect\_relations и функции с припиской «check».

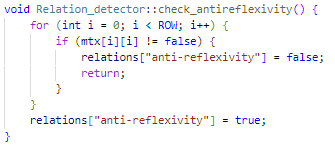
На примере функции check\_reflexivity, рассмотрим то как определяется выполнение отношения у матрицы. Код функции выглядит таким образом:



Проверка в матрице построена на том, что индексы матрицы это как бы разные элементы и им соответствуют 1 и 0, показывая их принадлежность к какому-либо отношению P, которое может соответствовать нескольким бинарным отношениям.

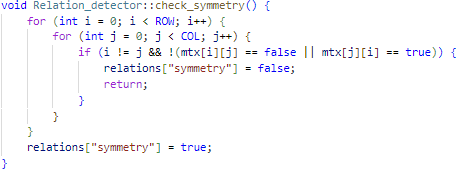
Суть проверки на выполнения отношения в том, что если не выполняется условие отношения на каком-то участке матрицы, то значит отношение не верно и можно завершить выполнение функции, иначе цикл закончится и таким образом подтвердится выполнение отношения рефлексивности в данном случае. Все остальные функции написаны аналогичным образом.

Таким образом выглядит проверка на антирефлексивность:



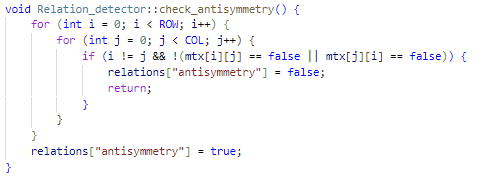
В данном случае ясно, что антирефлексивность выполнится, если каждый элемент главной диагонали равен 0.

Таким образом выглядит проверка на симметрию:



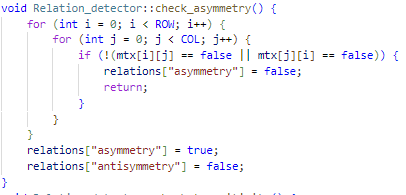
В данном случае не рассматривается элемент главной диагонали, поскольку очевидно, что он равен самому себе. Выражение в скобках получено из импликации(x отношение y -> y -> отношение x).

Проверка на антисимметрию:



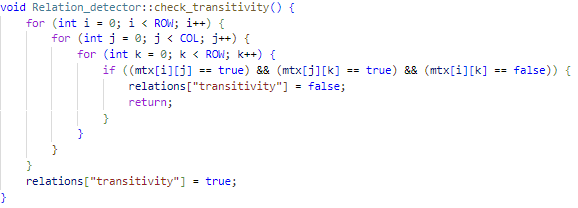
Здесь тоже самое, что и в случае с симметрией.

Проверка на асимметрию:



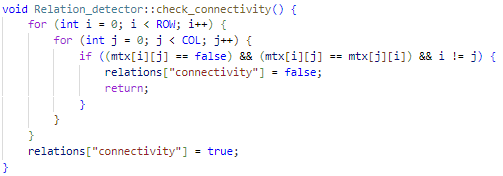
В этом случае уже необходимо учитывать и главную диагональ.

Проверка на транзитивность:

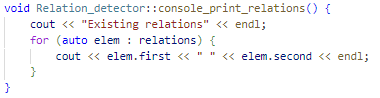


Транзитивность выглядит иначе поскольку необходимо учитывать третий элемент.

Проверка на связность:



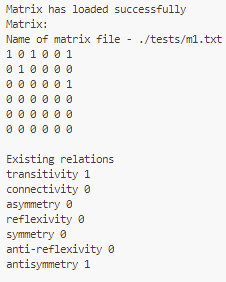
Осталось вывести все эти отношения. В программе этим занимается функция console\_print\_relations. Поскольку информация представлена в виде словаря, то для вывода нужно лишь перебрать значение с помощью цикла.



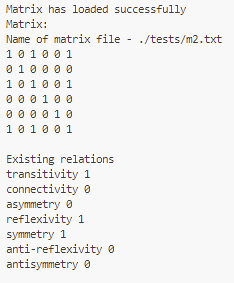
**Тестирование программы**

Для тестирования были предоставлены готовые матрицы, с заранее известными отношениями, для проверки работы программы.

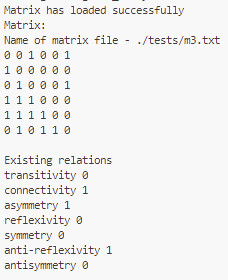
Результат тестирования первой матрицы:



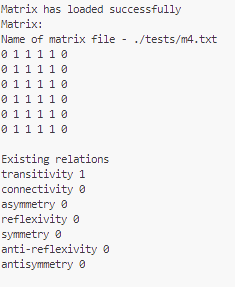
Результат тестирования второй матрицы:



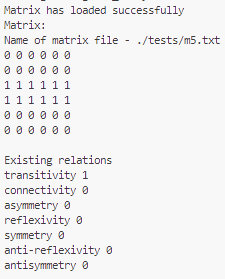
Результат тестирования третьей матрицы:



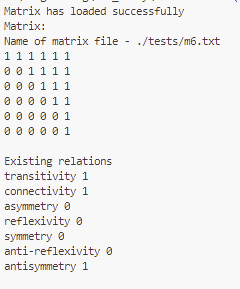
Результат тестирования четвертой матрицы:



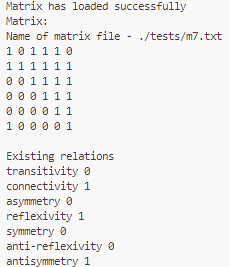
Результат тестирования пятой матрицы:



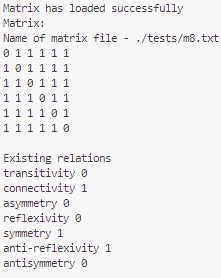
Результат тестирования шестой матрицы:



Результат тестирования седьмой матрицы:



Результат тестирования восьмой матрицы:



Как видно из результатов тестирования, программа работает верно.

# Заключение

В ходе работы были закреплены знания о бинарных отношениях и об их матричном представлении. Были получены навыки реализации систем подобной той, что была в задании.

# Список используемой литературы

1. <https://studfile.net/preview/9215001/page:3/>
2. <https://ru.wikipedia.org/wiki/%D0%91%D0%B8%D0%BD%D0%B0%D1%80%D0%BD%D0%BE%D0%B5_%D0%BE%D1%82%D0%BD%D0%BE%D1%88%D0%B5%D0%BD%D0%B8%D0%B5>