**Федеральное государственное бюджетное образовательное**

**учреждение высшего образования**

**«Саратовский государственный технический университет имени Гагарина Ю.А.»**

Кафедра **«**Прикладные информационные технологии»

**ОТЧЁТ**

**ПО ПРАКТИЧЕСКОЙ РАБОТЕ №6**

**ПРИКЛАДНЫЕ АСПЕКТЫ ТЕОРИИ ГРАФОВ**

Работу выполнил:

Студент группы б1-ИФСТ-31

Кошелева Алина Денисовна

Работу проверил преподаватель:

Дороднова Лилия Владимировна

Саратов 2023

**Разработка приложения по решению задачи теории графов**

Выполнение задания предполагает использование среды разработки, которая позволяет создавать оконные или консольные пользовательские интерфейсы в среде MS Windows. Первая часть задания, рассчитанная на 3–4 занятия, предполагает создание минимального набора функций объекта, который позволит работать с объектом «Взвешенный граф» (будем далее называть взвешенный граф сетью).

**СРОКИ ВЫПОЛНЕНИЯ ЗАДАНИЯ**

Представленное ниже задание нужно выполнить в течение 3-4-х практических занятий. Отчет по проекту должен находиться в Вашей рабочей папке (рекомендуется сетевая папка студента). Состав каждого отчета описан в конце задания.

*Предельные сроки* выполнения всего задания и подготовки отчета – последняя неделя ноября (23–29 ноября)

**ИСПОЛЬЗУЕМЫЙ ИНСТРУМЕНТАРИЙ**

Самостоятельный программный проект студента предполагает использование в качестве инструмента среду разработки, *которая удобна для студента* и позволяет ему максимально быстро и эффективно выполнить все задания. Предпочтительным вариантом инструмента разработки является среда MS Visual Studio C# (x86) доступная в компьютерном классе.

*В основной отчёт по этому заданию необходимо обосновать инструментария по следующим позициям (каждый пункт должен содержать 1–2 абзаца текста):*

1. Название среды разработки, используемая лицензия разработчика
2. Исполняемая ОС, возможность переноса приложения на другую ОС
3. Степень знакомства с инструментом разработки (раскрыть по пунктам: начальные навыки, навыки формирования интерфейса, навыки разработки обработчиков событий, описание имеющегося опыта самостоятельной разработки приложений с использованием специальных пакетов – привести примеры)
4. Вид демонстрации приложения преподавателю (с использованием компьютера в компьютерном классе, с использованием собственного ноутбука, с использованием удалённого доступа к собственному компьютеру из компьютерного класса)
5. Использование специальных библиотек (перечислить, если используются графические, графовые библиотеки, указать ссылки)

**ОПИСАНИЕ ОБЩЕГО ЗАДАНИЯ И ЗАДАНИЯ НА ПЕРВОМ ПРАКТИЧЕСКОМ ЗАНЯТИИ**

**Выбор Intelij IDEA в качестве среды для разработки приложения для построения графов можно обосновать следующими причинами:**

1. **Название среды разработки и используемая лицензия разработчика:**

* Среда разработки: Intelij IDEA.
* Лицензия разработчика: Intelij IDEA предоставляет две основные версии лицензий - Community Edition (бесплатная) и Ultimate Edition (платная). Community Edition предоставляет базовый набор функций, в то время как Ultimate Edition включает в себя дополнительные возможности и инструменты для разработки Java-приложений.

1. **Исполняемая ОС и возможность переноса приложения на другую ОС:**

* Intelij IDEA поддерживает различные операционные системы, включая Windows, macOS и Linux. Это означает, что вы можете разрабатывать приложения на Intelij IDEA на одной операционной системе и переносить их на другую без необходимости изменения кода. Однако, при переносе приложения на другую ОС, необходимо учитывать различия в файловой системе и зависимостях, которые могут потребовать некоторой настройки.

1. **Степень знакомства с инструментом разработки:**

* Начальные навыки: Intelij IDEA предоставляет интуитивно понятный интерфейс и множество функций, которые делают процесс разработки более эффективным и удобным для начинающих разработчиков.
* Навыки формирования интерфейса: Intelij IDEA обладает мощными инструментами для создания графического интерфейса приложений с использованием Swing, JavaFX и других технологий.
* Навыки разработки обработчиков событий: Intelij IDEA предоставляет удобные средства для разработки обработчиков событий, включая автодополнение кода, отладку и подсказки.
* Описание имеющегося опыта самостоятельной разработки приложений. Примеры можно привести самостоятельной разработки в области приложений с использованием фреймворка - Spring.

**4. Использование специальных библиотек:**

* Некоторые популярные библиотеки для разработки графических интерфейсов в Java включают Swing и JavaFX.
* Для работы с графами вы можете использовать библиотеки, такие как JGraphT или GraphStream.
* Для работы с базами данных вы можете использовать библиотеки, такие как Hibernate или JDBC.
* При использовании специальных библиотек важно обратить внимание на их документацию и официальные ресурсы для получения подробной информации о функциональности и использовании.

*В целом, выбор Intelij IDEA в качестве среды для разработки приложения для построения графов обусловлен ее интегрированной средой разработки, интеграцией с базами данных и поддержкой разных платформ. Это позволяет разработчикам эффективно создавать приложения для построения графов с использованием различных инструментов и технологий.*

**Общая функциональность разрабатываемого объекта**

1. *Создание пустого графа, у которого нет вершин и ребер при инициализации приложения или при выполнении соответствующей команды*

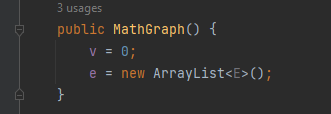


Рисунок 1 – Инициализация

1. *Загрузка данных о графе из текстового файла*

*- представляющего матрицу смежности, содержащую веса (формат текстового файла, представление весов ребер см. ниже)*

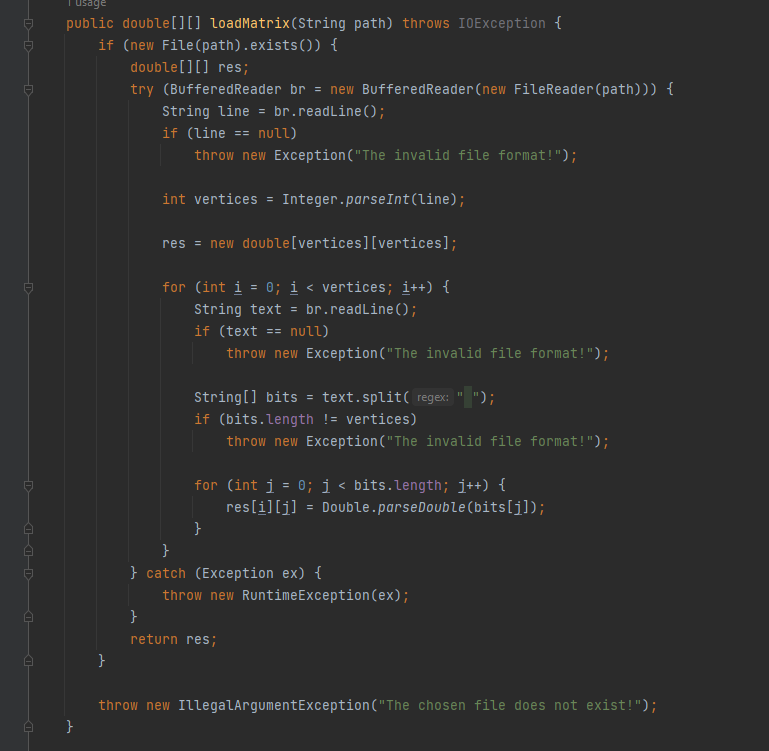


Рисунок 2 - Загрузка графа в виде матрицы

* *представляющего списки ребер (формат текстового файла, представление весов ребер см. ниже)*



Рисунок 3 - Загрузка графа через список ребер

*3. Сохранение графа в текстовый файл с определенным пользователем именем*

* *в виде матрицы*

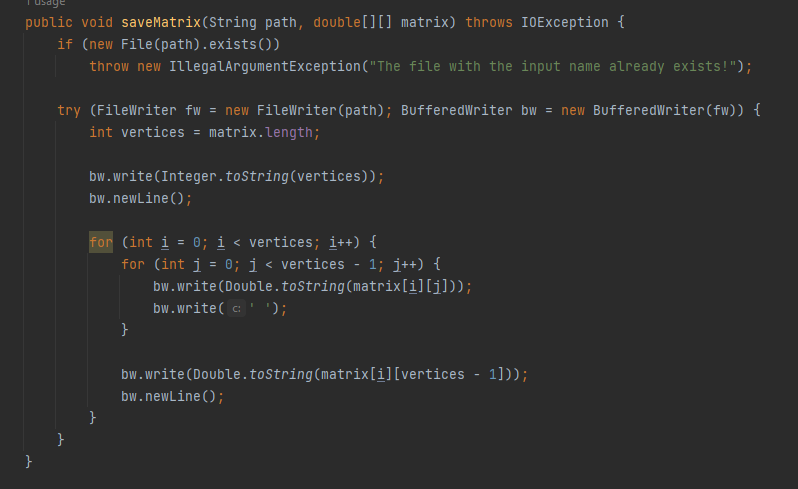


Рисунок 4 - Сохранение графа в виде матрицы

- *в виде списков ребер*

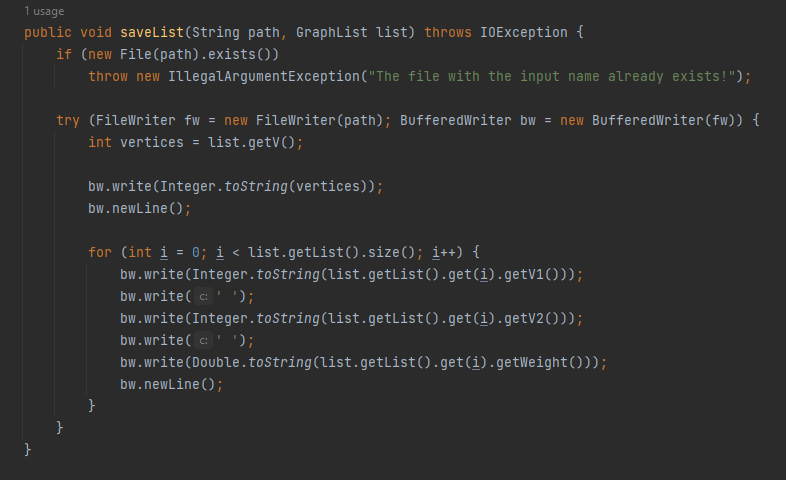


Рисунок 5 - Сохранение в виде списка

*4. Добавление вершины в граф*

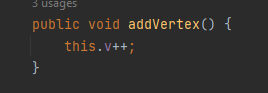


Рисунок 6 - Добавление вершины

*5. Добавление ребра в граф, заданного парой вершин и имеющего заданный вес*

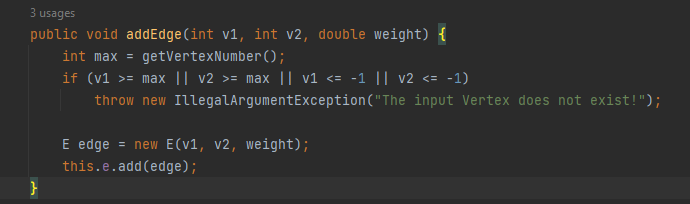


Рисунок 7 - Добавление ребра

1. *Удаление ребра из графа, заданного парой вершин*

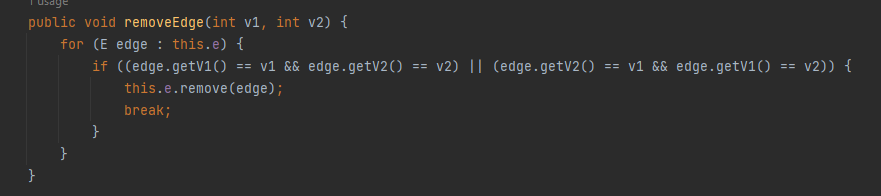


Рисунок 8 - Удаление ребра

1. *Функция (integer), возвращающая количество вершин графа*

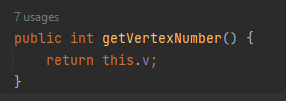


Рисунок 9 - Количество вершин

1. *Функция (integer), возвращающая количество ребер графа*

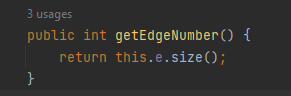


Рисунок 10 – Функция, возвращающая количество ребер

1. *Функция (boolean) с аргументами в виде заданных двух вершин, возвращающая true, если эти две вершины смежны и false в противном случае*

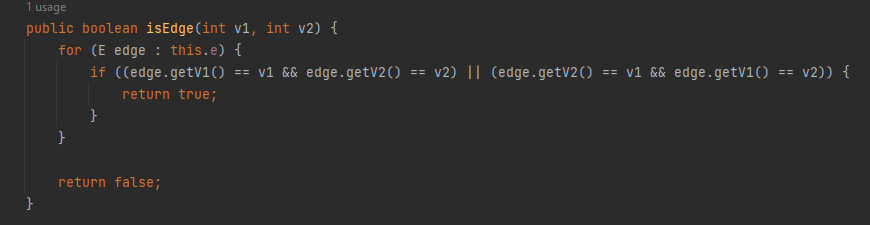


Рисунок 11 - Проверка смежности вершин

1. *Функция (float), возвращающая вес ребра для заданных двух вершин*

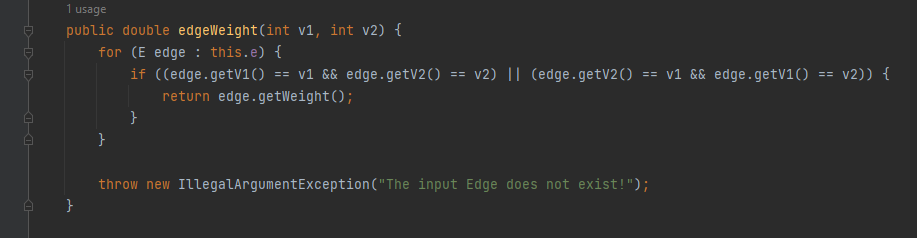


Рисунок 12 - Вес ребра

**Дополнительное задание**

Для проверки связности графа использован обход в глубину:

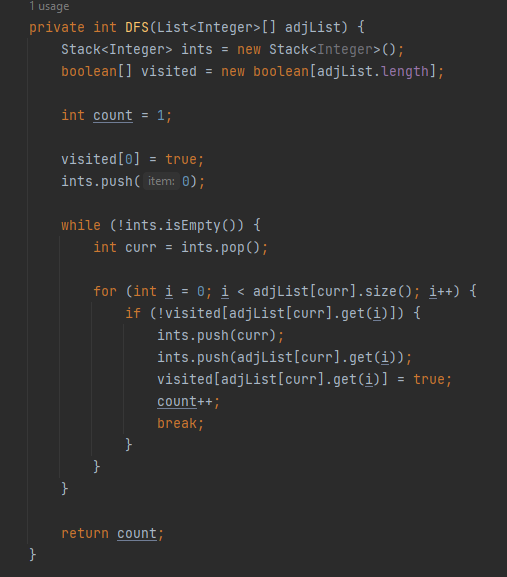


Рисунок 13 - Обход в глубину со счетчиком вершин

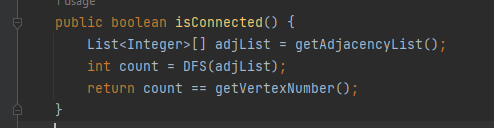


Рисунок 14 - Проверка на связность графа