МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЁТ

по учебной практике

Тема: Поиск максимального паросочетания

Студент гр. 8382 _____ Нечепуренко Н.А.

Студентка гр. 8382 _____ Терехов А.Е.

Студент гр. 8382 _____ Торосян Т.А.

Руководитель _____ Размочаева Н.В.

Санкт-Петербург
2020

ЗАДАНИЕ

НА УЧЕБНУЮ ПРАКТИКУ

Студент Нечепуренко Н.А. группы 8382
Студент Терехов А.Е. группы 8382
Студент Торосян Т.А. группы 8382

Тема практики: применение алгоритма Куна для поиска максимального паросочетания в двудольном графе на реальных данных из соц. сети Вконтакте.

Задание на практику:

Командная итеративная разработка визуализатора алгоритма(-ов) на Java с графическим интерфейсом.

Алгоритм: алгоритм поиска максимального паросочетания в двудольном графе (алгоритм Куна)

Сроки прохождения практики: 29.06.2020 – 12.07.2020

Дата сдачи отчета: 12.07.2020

Дата защиты отчета: 12.07.2020

Студент _____ Нечепуренко Н.А.

Студент _____ Терехов А.Е.

Студент _____ Торосян Т.А.

Руководитель _____ Размочаева Н.В.

АННОТАЦИЯ

Целью выполнения данной работы является практическое применения алгоритма поиска максимального паросочетания в двудольном графе. Данное решение позволяет визуализировать алгоритм поиска максимального паросочетания в двудольном графе. Доли графа представляют собой группу пользователей социальной сети ВКонтакте и группу сообществ, на которые они подписаны. Группа пользователей состоит из целевого пользователя и 5-15 его друзей. Группа сообществ – это множество состоящее из 5 первых сообществ каждого пользователя первой группы.

SUMMARY

The purpose of this work is a practical application of the max bipartite matching algorithm. This solution provides MBM algorithm visualisation. The shares of the graph represent a group of VKontakte social network users and a group of communities they are subscribed to. A user group consists of a target user and 5 to 15 of his friends. A community group is a set consisting of the first 5 communities of each user of the first group.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

На вход программе подаётся VK ID целевого пользователя. Целью является получение необходимой информации для построения двудольного графа и нахождение в нём максимального паросочетания.

Для решения задачи применяется алгоритм Куна. В начале работы программы и после получения VK ID программа делает несколько запросов через API соц. сети Вконтакте для получения списков друзей и сообществ. Затем из этих данных строится двудольный граф. На полученном графе запускается алгоритм Куна, который находит максимальное паросочетание между группой пользователей и группой сообществ.

1. ТРЕБОВАНИЯ К ПРОГРАММЕ

1.1. Исходные Требования к программе

1.1.1. Требования к входным данным

Для передачи программе ID пользователя используется стандартное текстовое поле. ID представляет собой целое число.

1.1.2. Требования к визуализации

Программа должна уметь выводить построенный граф так, чтобы можно было однозначно распознать пользователей и сообщества. Пользователь должен иметь возможность рассмотреть выполнение алгоритма пошагово.

1.1.3. Требования к выходным данным

Выходные данные (максимальное паросочетание) должны быть визуально представлены на данном графе.

1.2. Уточнение требований после консультации с преподавателем

По итогу консультации с преподавателем были отмечены следующие замечания, учтённые в последующих версиях программы:

- На малых разрешениях экрана часть данных не была видна. Был переработан GUI для поддержки на большинстве экранов.

2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ

2.1. План разработки

До 03.07.2020: разработать спецификацию, распределить роли и подготовить репозиторий.

До 06.07.2020: реализовать структуры данных для хранения графа, данных пользователя и групп; разработать базовую разметку пользовательского интерфейса.

До 06.07.2020: отладка и тестирование алгоритма поиска максимального паросочетания.

До 08.07.2020: визуализировать алгоритм на простейших тестовых данных.

До 10.07.2020: разработать парсер открытого API социальной сети ВКонтакте, интегрировать реальные данные в готовое решение.

До 12.07.2020: отладка программы, рефакторинг кода, написание комментариев.

2.2. Распределение ролей в бригаде

- Нечепуренко Н. - лидер, алгоритмист.
- Терехов А. - фронтенд, тестирование.
- Торосян Т. - тестирование, документация.

3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ

3.1. Структуры данных

Таблица 1 – Основные структуры данных, необходимые для поиска максимального паросочетания.

Название	**Описание**
interface Visitor	Интерфейс, через который реализован паттерн «Посетитель» для обхода вершин графа.
public class NodeVisitor	Класс для обхода вершин графа.
public class SemiEdge	Класс полуребра, хранящий информацию о вершине и описание ребра.
public class GraphNode	Класс, описывающий вершину графа и хранящий её данные.
public class Edge	Класс, описывающий ребро графа.
public class Bipartite	Основной класс, реализующий поиск максимального паросочетания алгоритмом Куна.

3.2. Основные методы

Таблица 2 – Основные методы класса Bipartite

Название	**Описание**
Bipartite	Конструктор класса, инициализируемый входными данными.
GetMatchingList	Возвращает паросочетание в виде списка рёбер.
getResultMatching	Возвращает определённую вершину из текущего паросочетания, если таковая имеется.
setResultMatching	Добавляет грань в текущее решение.
getAdjacentList	Возвращает список примыкающих к данной вершине полурёбер.

toString	Возвращает результат выполнения алгоритма в текстовом виде. Отладочная функция.
getMaxMatching	Возвращает максимальное паросочетание в виде хеш-карты вершин.
getFirstSide	Возвращает первую вершину графа.

3.3. Особенности бэкенд, технологии и библиотеки

Для реализации алгоритма Куна в двудольном графе используется паттерн "Посетитель". Посетитель обходит вершины графа поиском в глубину, пытаясь найти дополняющий путь. В функции getMaxMatchingIteration вызывается очередная итерация алгоритма, номер которой получается из объекта mediator, обеспечивающего пошаговое выполнение. В алгоритме для каждой вершины меньшей доли (в данном случае для юзеров) выполняется обход графа в глубину с записью выбранных ребёр в HashMap текущего паросочетания.

Mediator поддерживает хранение состояния списка ребёр текущего паросочетания, что позволяет переключаться между итерациями вперед и назад.

Для отрисовки GUI было решено использовать Swing благодаря низкому порогу вхождения, легковесности и кроссплатформенности.

Для упрощения работы с зависимостями и автоматизации сборки проекта использован фреймворк Maven. Он был выбран в силу кроссплатформенности и временных рамок проекта, требующих от фреймворка простоты, быстродействия и поддержки различных операционных систем.

3.4. Особенности визуализации

Как было отмечено, для отрисовки интерфейса был использован Java Swing, на классах которого и основана графическая часть приложения. Основным Классом является MainWindow, подтягивающий остальные элементы. Класс текстового поля InputPanel используется для ввода пользовательского ID. После получения данных с сервера Вконтакте интерфейс отрисовывает аватары пользователей и сообществ. В дальнейшем, с помощью кнопок пользователь управляет ходом визуализации.

Помимо этого, через графический интерфейс можно получить контакты авторов, информацию о них и программе.

4. ТЕСТИРОВАНИЕ

4.1. Ручное тестирование программы

В силу компактности программы, малого количества входных данных и специфики работы с VK API выбор пал на ручное тестирование.

Были проведены тесты, покрывающие большинство тривиальных критических точек программы: ошибки пользовательского ввода, некорректные ID, ошибки внешнего сервиса (VK API).

Для воспроизведения «проблемных» комбинаций входных данных были созданы несколько тестовых аккаунтов в соцсети Вконтакте. Манипулируя списками друзей и сообществ этих аккаунтов, тестировщики проверили большинство уязвимых точек основного алгоритма.

ЗАКЛЮЧЕНИЕ

Проект был завершен успешно. Поставленные цели в техническом задании были выполнены. Был реализован интуитивно понятный пользовательский интерфейс и наглядная пошаговая визуализация выполнения алгоритма поиска максимального паросочетания в двудольном графе. Функционал приложения включает в себя: возможность многократного запуска визуализации на разных данных без необходимости закрывать программу, пошаговую визуализацию, возможности сразу получить финальный результат и начать сначала.

Работа программы основывается на алгоритме Куна, осуществляющем поиск максимального паросочетания в двудольном графе.

Работоспособность алгоритма была проверена тестированием на реальных данных, тривиальными и синтетическими тестами.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Шилдт Герберт. Swing. Руководство для начинающих.: Вильямс, 2007
2. Шилдт Герберт. Java. Полное руководство.: Диалектика, 2018
3. Описание и теория алгоритма Куна. – https://e-maxx.ru/algo/kuhn_matching
4. Описание паттерна «Посетитель». – https://refactoring.guru/ru/design-patterns/visitor
5. Описание паттерна «Одиночка». – https://refactoring.guru/ru/design-patterns/singleton
6. Учебный курс «Java для начинающих». – https://stepik.org/course/187/syllabus
7. Репозиторий проекта. – http://github.com/nechepurenkoN/spbetu2020_summer_practice

ПРИЛОЖЕНИЕ А
UML-ДИАГРАММЫ

Рисунок А.1 – UML для графического интерфейса.

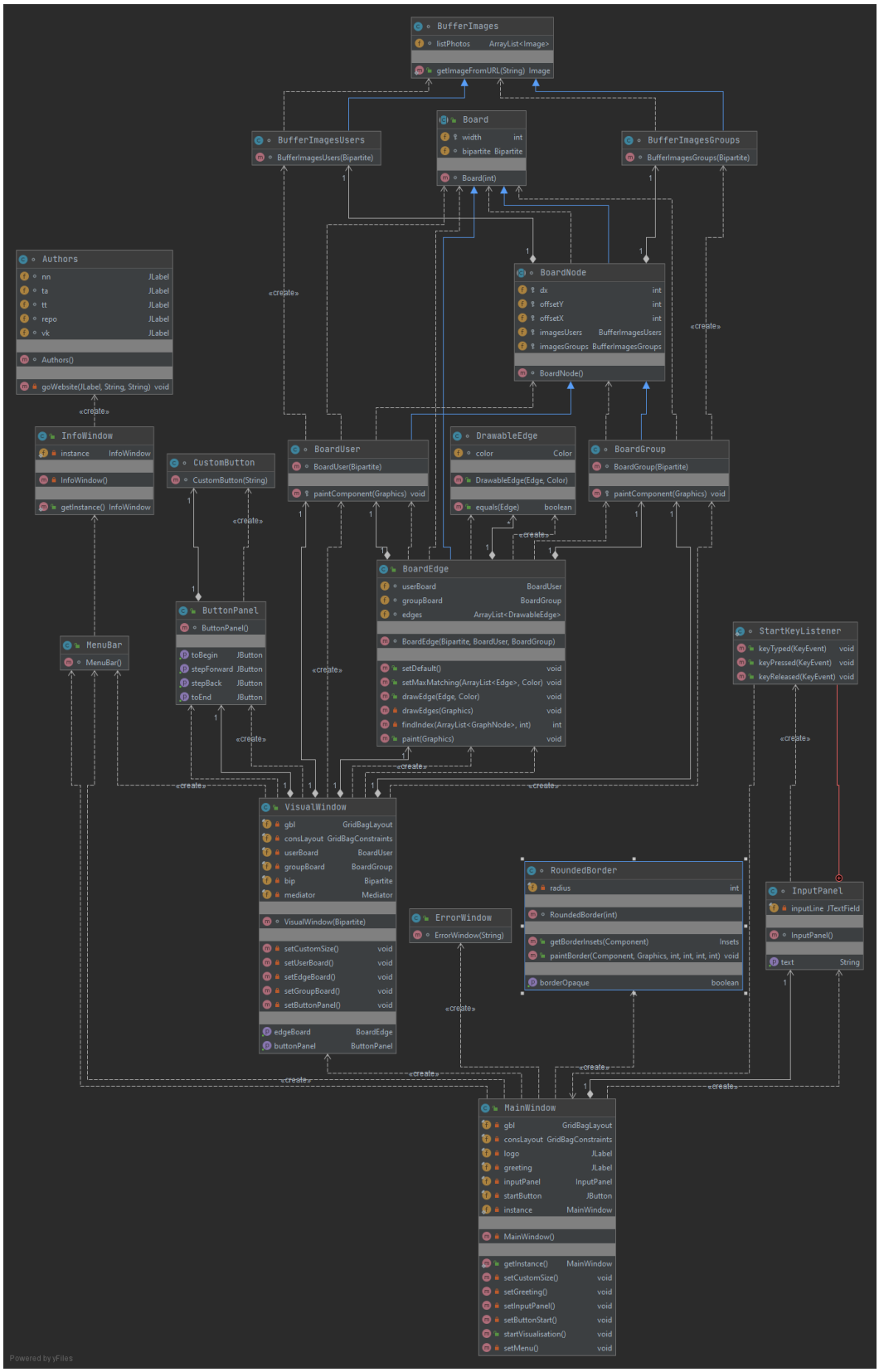

Рисунок А.2 – UML для парсера.

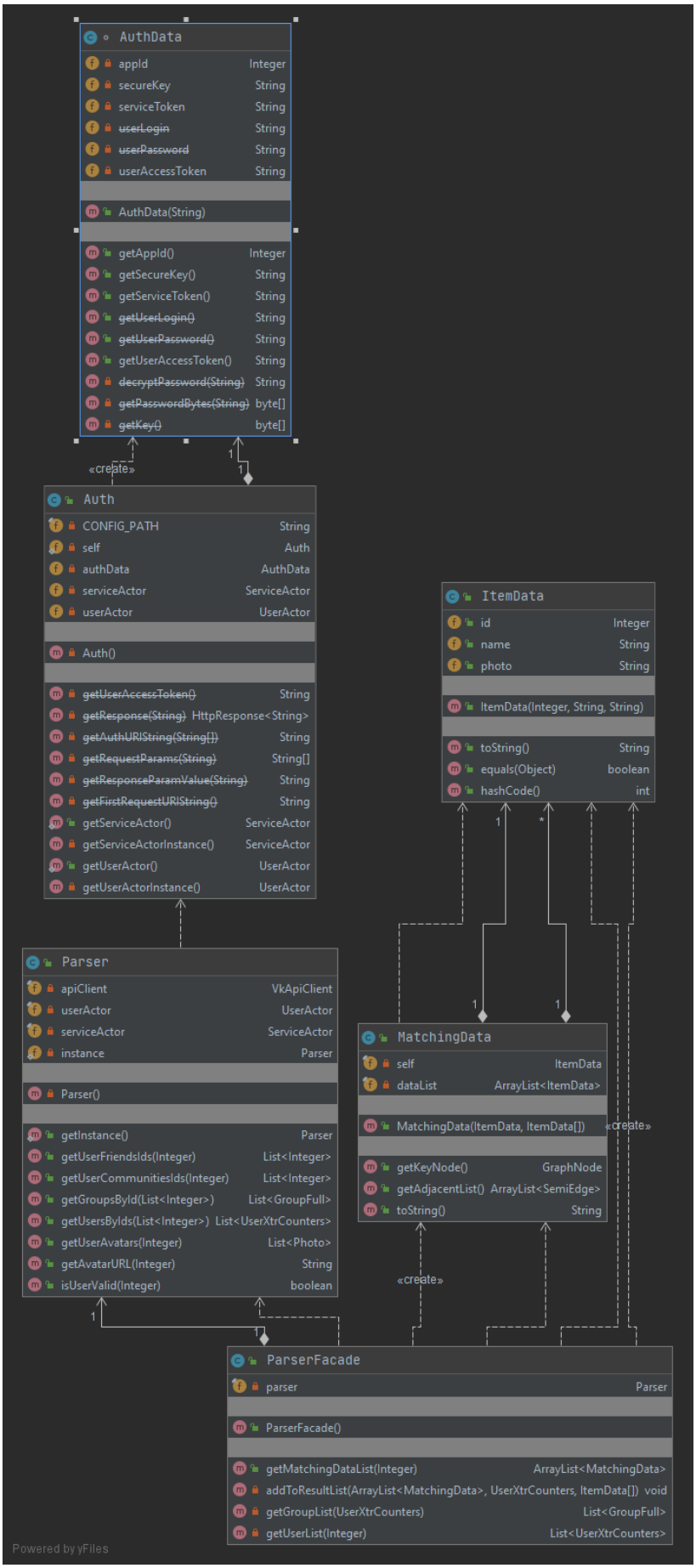

Рисунок А.3 – UML для основного алгоритма.

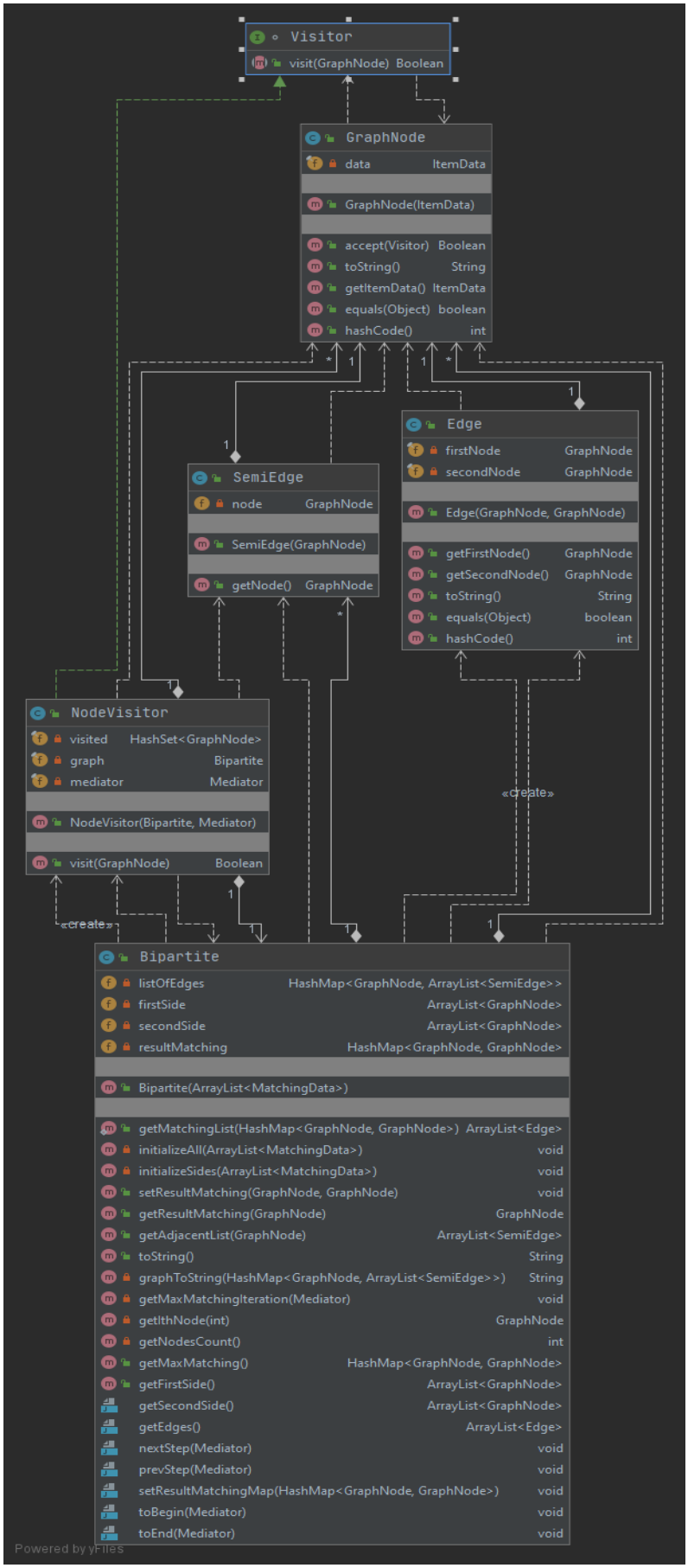

Рисунок А.4 – UML для связей в проекте.