

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

**ОТЧЕТ**  
**по учебной практике**  
**Тема: VKFriends**

Студент гр. 7382	_____	Гаврилов А.В.
Студент гр. 7382	_____	Гиззатов А.С.
Студент гр. 7382	_____	Государкин Я.С.
Руководитель	_____	Фирсов М.А.

Санкт-Петербург  
2019

## ЗАДАНИЕ НА УЧЕБНУЮ ПРАКТИКУ

Студент Гаврилов А.В. группы 7382

Студент Гиззатов А.С. группы 7382

Студент Государкин Я.С. группы 7382

Тема практики: VKFriends

Задание на практику:

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

Алгоритм: Поиск общих друзей двух пользователей во вконтакте.

Сроки прохождения практики: 01.07.2019 – 14.07.2019

Дата сдачи отчета: 00.07.2019

Дата защиты отчета: 00.07.2019

Студент	_____	Гаврилов А.В.
Студент	_____	Гиззатов А.С.
Студент	_____	Государкин Я.С.
Руководитель	_____	Фирсов М.А.

## **Аннотация**

Темой данной учебной практики является командная итеративная разработка визуализатора алгоритма на языке программирования Java.

Цель работы научиться писать на языке программирования java, также обучиться налаживанию коммуникации между участниками практики, разобраться с декомпозицией обязанностей. Визуализировать любой алгоритм для работы с графами. В работе представлена визуализация поиска общих друзей в ВК.

# Содержание

<b>ВВЕДЕНИЕ .....</b>	<b>5</b>
<b>1. ТРЕБОВАНИЯ К ПРОГРАММЕ .....</b>	<b>6</b>
1.1. Исходные Требования к программе .....	6
1.1.1. Примеры работы программы. ....	6
1.1.2. Примеры тестирования программы. ....	6
<b>2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ .....</b>	<b>10</b>
2.1. План разработки.....	10
2.1.1. Создать базовый gui с кнопками добавления/удаления вершины, удаления всего графа, графическое поле для графа и поле/окно для вывода информации об аккаунте. ....	10
2.1.2. Добавить возможность строить граф, где вершина - аккаунт, а ребро - кол-во общ. Друзей. ....	10
2.1.3. Добавить возможность добавлять/удалять новые вершины. ....	10
2.1.4. Реализовать базовый набор unit тестов. ....	10
2.1.5. Свести все наработки в один проект (реализовать взаимодействие между частями проекта).....	10
2.1.6. Выводить некоторую информацию об аккаунте по нажатию на вершину. ....	10
2.1.7. Строить минимальное/максимальное остовное дерево на основе базового графа. ....	10
<b>3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ .....</b>	<b>11</b>
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>15</b>
<b>СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ .....</b>	<b>16</b>
<b>ПРИЛОЖЕНИЕ А.....</b>	<b>17</b>

## **ВВЕДЕНИЕ**

Научиться программировать на языке java, а конкретно:

- 1) Разобраться с созданием графического интерфейса при помощи встроенных библиотек.
- 2) Построить граф друзей, ребра которого показывает количество общих друзей между двумя людьми связанными этим ребром, а вершина содержит краткую информацию о пользователе.
- 3) Написать алгоритм, который ищет всех общих друзей между двумя выбранными пользователями.
- 4) Написать юнит тесты для проверки корректности работы программы.

## **1. ТРЕБОВАНИЯ К ПРОГРАММЕ**

### **1.1. Исходные Требования к программе**

При запуске проекта будет создаваться отдельное окно. Данное окно будет иметь 3 поля:

1. Поле графического представления графа и визуализации алгоритма. Граф строится по данным, введенным пользователем с помощью графического интерфейса.

2. Поле с элементами управления:

- Добавление вершины в граф.
- Перестройка графа.
- Удаление вершины.
- Удаление графа.

3. Диалог создания графа:

- Ввод URL/ID пользователя.
- Ввод URL/ID двух или более пользователей для нахождения общих друзей.

**1.1.1. Примеры работы программы.**

**1.1.2. Примеры тестирования программы.**

```
Abduali Ondasyn 222956247 - Abduali Ondasyn 222956247 - equals
Adlet Assanov 88417687 - Adlet Assanov 88417687 - equals
Aldiyar Bekturganov 120190064 - Aldiyar Bekturganov 120190064 - equals
Alexander Chigaleychik 23251408 - Alexander Chigaleychik 23251408 - equals
Alexander Drozd 534202467 - Alexander Drozd 534202467 - equals
Alexander Zheleznyak 28078 - Alexander Zheleznyak 28078 - equals
Alexey Egorov 183622198 - Alexey Egorov 183622198 - equals
Aliakbar Sagidulla 92211088 - Aliakbar Sagidulla 92211088 - equals
Alibi Rakhanov 124712186 - Alibi Rakhanov 124712186 - equals
Alikhan Musabekov 132664181 - Alikhan Musabekov 132664181 - equals
Altynai Akhmettayeva 185526800 - Altynai Akhmettayeva 185526800 - equals
Alua Shyntay 172077069 - Alua Shyntay 172077069 - equals
Amina Askarbek 45342782 - Amina Askarbek 45342782 - equals
Amina Bimuratova 147902153 - Amina Bimuratova 147902153 - equals
Anastasia Lyashevskaya 58887369 - Anastasia Lyashevskaya 58887369 - equals
Andrey Gavrilov 110883708 - Andrey Gavrilov 110883708 - equals
Andrey Serov 212690498 - Andrey Serov 212690498 - equals
Andrey Tokarev 131900209 - Andrey Tokarev 131900209 - equals
Anuar Orazbekov 273101127 - Anuar Orazbekov 273101127 - equals
Anna Eryomenko 265211250 - Anya Eryomenko 265211250 - not equals

Process finished with exit code 0
```

Рис.1 – Пример тестирования метода parseFriendsJson.

```
Alexander Chigaleyichik 23251408
Alexander Drozd 534202467
Alexandra Ratushnaya 62777107
Alexey Egorov 183622198
Alexey Pashko 1566618
Alina Ereemeeva 3012466
Alyona Shadrina 138167857
Amir Gizzatov 206043986
Anastasia Lyashevskaya 58887369
Anastasia Tsvetkova 218284976
Andrey Bykov 87210355
Andrey Tokarev 131900209
Anton Kondratyev 198691145
Anya Eryomenko 265211250
Anya Petrova 141845542
Daniil Bakherov 140858924
Dilya Shekhmametyeva 84779358
Dmitry Shadrin 5331843
Elena Ereemeeva 97012700
Felix Sokolov 145821876
Igor Bledny 478285968
Kostya Zyuzin 88639708
Margarita Sorokina 83762485
Marlen Bergaliev 175436356
Maxim Kabanchenko 250210234
Maxim Maximov 141038889
Mikhail Petrov 23335051
Nastasya Zhukova 4446814
Natasha Dorogan 37780718
Nikita Balagura 42004237
Nikita Gavrilov 110872794
Polina Zelenkova 104365046
Roma Ivanov 138618783
Samuil Gusselnikov 183940633
Sasha Drozd 148917226
Sergey Glazunov 179878269
Sergey Tropashko 7928845
Tanya Shadrina 12436859
Valentina Davkaeva 83241708
Valeria Chayka 104583503
Vlad Zhdanov 251128179
Yaroslav Gosudarkin 111008298
Yaroslav Piskunov 223745613

Process finished with exit code 0
```

Рис.2 – Пример тестирования метода getUser.



```
Andrey Gavrilov 110883708 - Amir Gizzatov 206043986 in common: 14
Yaroslav Gosudarkin 111008298 - Amir Gizzatov 206043986 in common: 20
Anya Petrova 141845542 - Amir Gizzatov 206043986 in common: 26
Ilyas Kasymbekov 79537315 - Amir Gizzatov 206043986 in common: 10
Polina Deryabina 60050060 - Amir Gizzatov 206043986 in common: 12
Evgenia Karibdzhanova 92122073 - Amir Gizzatov 206043986 in common: 4

Process finished with exit code 0
```

Рис.3 – Пример тестирования метода getCommonFriends.

## **2. ПЛАН РАЗРАБОТКИ И РАСПРЕДЕЛЕНИЕ РОЛЕЙ В БРИГАДЕ**

### **2.1. План разработки**

**2.1.1. Создать базовый gui с кнопками добавления/удаления вершины, удаления всего графа, графическое поле для графа и поле/окно для вывода информации об аккаунте.**

**2.1.2. Добавить возможность строить граф, где вершина - аккаунт, а ребро - кол-во общ. Друзей.**

**2.1.3. Добавить возможность добавлять/удалять новые вершины.**

**2.1.4. Реализовать базовый набор unit тестов.**

**2.1.5. Свести все наработки в один проект (реализовать взаимодействие между частями проекта).**

**2.1.6. Выводить некоторую информацию об аккаунте по нажатию на вершину.**

**2.1.7. Строить минимальное/максимальное остовное дерево на основе базового графа.**

### **2.2. Распределение ролей в бригаде:**

- **Государкин Ярослав - ответственный за алгоритмы.**
- **Гаврилов Андрей - ответственный за графический интерфейс.**
- **Гиззатов Амир - ответственный за юнит тестирование и отчёт.**

### 3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ

#### 3.1. Используемые структуры данных.

Граф для реализации алгоритма представляется

#### 3.2. Описание классов.

Программа состоит из классов: ApplicationMain, VKClient, VKUser, MainWindow.

- Класс ApplicationMain отвечает за выполнение программы в целом. Он создает граф и вызывает методы для нахождения общих друзей.
- Класс VKClient отвечает за получение списка друзей, списка общих друзей, информации о пользователе, отправка запроса в vk api.

Поля класса VKClient предоставлены в таблице 1, а методы перечислены в таблице 2.

Таблица 1-Поля класса VKClient

Поле	Значение поля
public static final String[] basicArgs	Информация о пользователе
private static final String accessVkApiToken	Ключ для отправки запроса к vk api
private static final String versionVkApi	Версия api vk
private static final String beginVkApi	Начало запроса к api vk
private static final String endVkApi	Конец запроса к api vk

МЕТОДЫ
<code>public ArrayList&lt;VKUser&gt; parseFriendsJson(String str)</code>
<code>public ArrayList&lt;int[]&gt; getCommonFriends(int srcId, int[] targetIds)</code>
<code>public VKUser getUser(int userId, String[] args)</code>
<code>public String getUserFriends(int userId, String order, String[] args)</code>
<code>public ArrayList&lt;VKUser&gt; getFriends( int userId, String order, String[] args )</code>
<code>private String createGetRequest(String method, int id, String order, String[] args)</code>
<code>private String getRequest(String url)</code>

• Класс VKUser отвечает за информацию о пользователе во вконтакте. Имеет методы для получения строки из полей класса(userId, firstName, lastName), получение строки из списка друзей, поиска и удаления пользователя в списке. Поля класса представлены в таблице 3.

Таблица 3 – Поля класса VKUser.

Поле	Значение поля
<code>public int userId</code>	Уникальный номер пользователя.
<code>public String firstName</code>	Имя пользователя.
<code>public String lastName</code>	Фамилия пользователя.

Методы
<code>public String toString()</code>
<code>public static String[] arrayToStrings(ArrayList&lt;VKUser&gt; list)</code>
<code>public static void findAndRemove(ArrayList&lt;VKUser&gt; list, int id)</code>

• Класс `MainWindow` отвечает за графический интерфейс, строит граф друзей и граф общих друзей пользователя. Поля и методы класса представлены в таблицах 4 и 5.

Таблица 4 – Поля класса `MainWindow`.

Поле	Значение поля
<code>private final int leftPanelWidth</code>	Высота левой панели
<code>private final int colNum</code>	Количество столбцов в <code>textfiled</code>
<code>private JPanel toolBar</code>	Панель инструментов
<code>private JPanel buttonBar</code>	Панель кнопок
<code>private JPanel LeftPanel</code>	Левая панель
<code>private final Color colorForTools</code>	Цвет инструментов
<code>private final Color colorForOther</code>	Цвет графа и текстового поля
<code>private JList&lt;String&gt; mList</code>	Динамический список пользователей
<code>private JTextField InputField</code>	Поле ввода <code>id</code> или <code>URL</code> пользователя
<code>private mxGraph graph</code>	Граф
<code>private VKClient mClient</code>	Информация о пользователе
<code>private ArrayList&lt;VKUser&gt; mUsers</code>	Список пользователей типа <code>VKUser</code>

Методы
protected mxGraphComponent makeGraph()
protected void makeToolBar()
public void actionPerformed(ActionEvent actionEvent)
protected void makeLeftPane(mxGraphComponent graphComponent)
private void setUpVKClient()

### 3.3 UML – диаграмма классов.

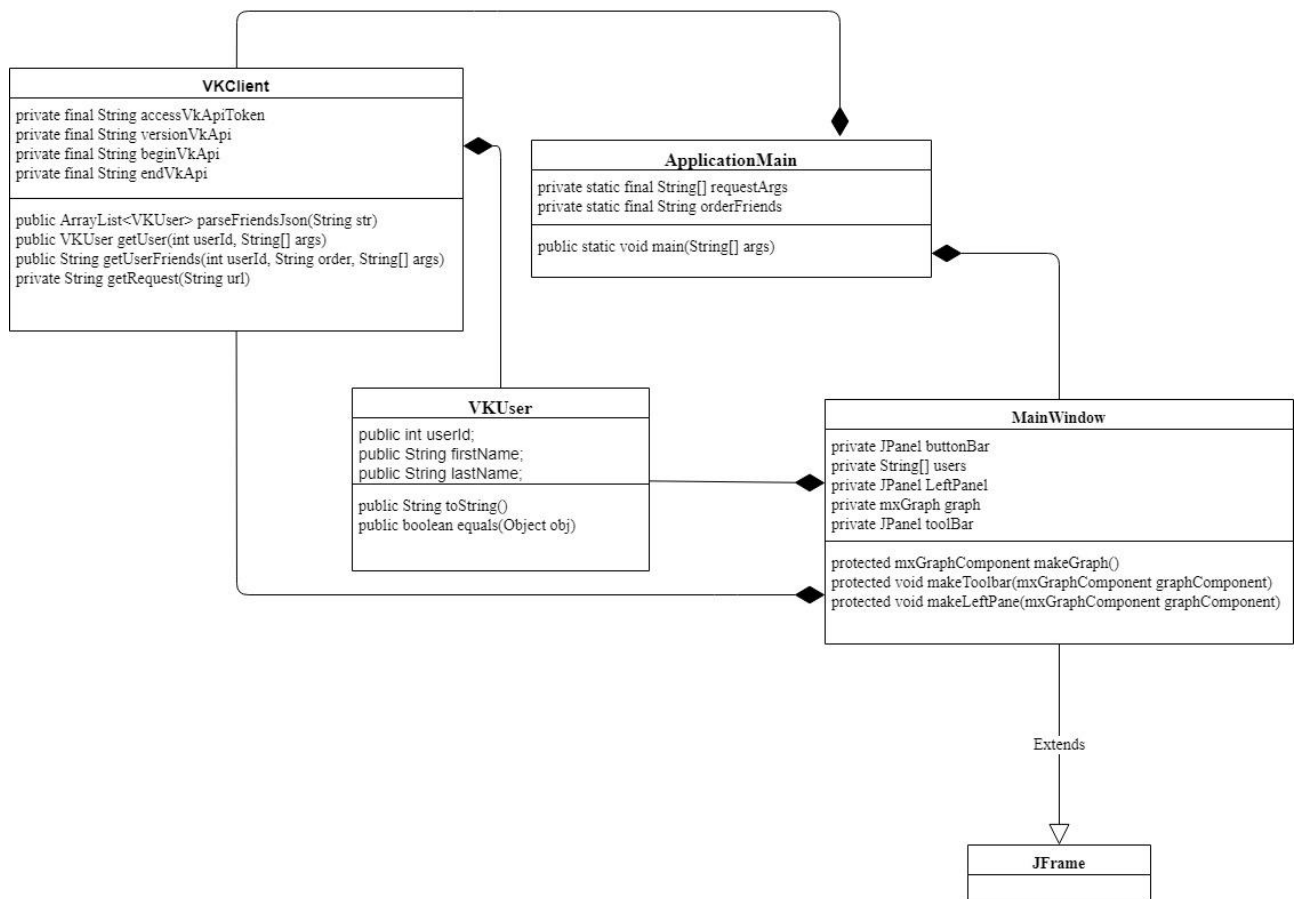


Рис.3 - диаграмма классов.

## **ЗАКЛЮЧЕНИЕ**

Таким образом, был реализован алгоритм поиска общих друзей во вконтакте с визуализацией. Был разработан интерфейс удобный и понятный для пользователя. Так же в ходе учебной практики были получены знания по языку программирования Java, получены навыки в визуализации алгоритмов и работе в команде.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1) VK API [Электронный ресурс] URL:  
<https://vk.com/dev/manuals>
- 2) HABR[Электронный ресурс] URL:  
<https://habr.com/>
- 3) ORACLE JFRAME[Электронный ресурс] URL:  
<https://docs.oracle.com/javase/7/docs/api/javax/swing/JFrame.html>
- 4) JUNIT ON JAVARUSH[Электронный ресурс] URL:  
<https://javarush.ru/groups/posts/605-junit>



## **ПРИЛОЖЕНИЕ А**

### **НАЗВАНИЕ ПРИЛОЖЕНИЯ**

полный код программы должен быть в приложении, печатать его не надо