

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ  
И ПРОГРАММНОЙ ИНЖЕНЕРИИ (КАФЕДРА №43)

ОТЧЕТ ЗАЩИЩЕН С ОЦЕНКОЙ: \_\_\_\_\_

ПРЕПОДАВАТЕЛЬ:

Старший преподаватель

(должность, уч. степень, звание)

(подпись, дата)

Е. В. Павлов

(инициалы, фамилия)

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №1

«СОСТАВЛЕНИЕ ВАРИАНТОВ ТЕСТИРОВАНИЯ  
И МЕТРИКА ПОКРЫТИЯ ТРЕБОВАНИЙ»

ПО КУРСУ: «МЕТРОЛОГИЯ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

РАБОТУ ВЫПОЛНИЛ:

СТУДЕНТ ГР.

4831

(подпись)

(дата отчёта)

К.А Корнющенко

(инициалы, фамилия)

## 1. Цель работы

Целью данной работы является изучение способов тестирования функций приложения и оценка плотности покрытия тестами функциональных требований.

## 2. Задание на лабораторную работу

Разработать консольное приложение в соответствии с индивидуальным вариантом задания и представленной концепцией продукта.

Составить тест кейсы для тестирования функций данного приложения и построить матрицу соответствия требований для анализа тестового покрытия.

Выполнить тестирование приложения по составленным тест кейсам.

Реализовать не менее 7-ми профильных функций приложения и составить как минимум 10-ть тест кейсов к данным функциям (тест кейсы должны осуществлять проверку сформулированных функциональных требований приложения).

Вариант задания:

89. Приложение для обмена фотографиями

## 3. Спецификация требований к программному обеспечению

Данное приложение представляет собой адаптацию под сформулированную концепцию продукта и предназначено для того, чтобы делиться фотографиями с другими пользователями. Основные функции приложения заключаются в предоставлении пользователю возможности добавлять ссылку на фотографию и описание к ней.

Функциональные требования к приложению представлены в таблице 1 (каждому требованию соответствует уникальный идентификатор):

Таблица 1 — Функциональные требования

FR-01	Приложение должно предоставлять список ваших фотографий в формате: ссылка на изображение + описание (хранение списка данных осуществляется посредством внешнего файла).
FR-02	Приложение должно обеспечить добавления новых фотографий с описанием.
FR-03	Приложение должно выполнять удаление фотографий из списка.
FR-04	Приложение должно предоставлять возможность поиска пользователей по логину(хранение информации о пользователях осуществляется посредством внешнего файла).
FR-05	Приложение должно выполнять возможность подписаться на новых пользователей.
FR-06	Приложение должно выполнять возможность редактировать описание у фотографий.
FR-07	Приложение должно выполнять возможность изменять свой данные (логин/пароль).

FR-08	Приложение должно осуществлять вывод информации о программе (наименование приложения и год выпуска, короткое описание назначения, версия и автор программы).
FR-09	Приложение должно осуществлять планомерное завершение программы и сохранение данных в файл.
FR-10	Приложение должно осуществлять авторизацию/регистрацию пользователя.

#### 4. Тест дизайн и анализ тестового покрытия

ID и наименование: TC-1 Вывод списка фотографий пользователя	
Ссылка на требование: FR-01	
Дата создания / Автор: 25.02.2020 / Кирилл	
Предусловия: Необходимо, чтобы у пользователя было добавлено n фотографий в базу данных.	
Шаги теста:	Ожидаемый результат:
1. Авторизоваться или зарегистрироваться в приложении. 2. Выбрать в главном меню пункт: «1 - просмотр своих фотографий».	1. Приложение выводит список ссылок на фотографии и их описание.

Фактический результат:

```
Меню
1 - просмотр своих фотографий
2 - добавление фоточек
3 - удаление своих фоточек
4 - редактирование описания у фоточки
5 - редактирование своих данных login/password
6 - просмотр подписчиков
7 - поиск пользователей
8 - выход и сохранение данных
9 - информация о программе
1
dog->https://www.google.com/url?sa=i&url=https%3A%2F%2Fbigpicture
.ru%2F%3Fp%3D709152&psig=A0vVaw1v0-FBVow1f1CJTYXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRxq
FwoTCPCqyfju70cCFQAAAAAdAAAAABAJ
tramp->https://www.google.com/url?sa=i&url=https%3A%2F%2Ftvzvezda
.ru%2Fnews%2Fvstrane_i_mire%2Fcontent%2F201812121749-vhok
.htm&psig=A0vVaw1v0-FBVow1f1CJTYXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPCqyfju70c
CFQAAAAAdAAAAABA0
```

ID и наименование: TC-2 Добавление новых фотографий	
Ссылка на требование: FR-02	
Дата создания / Автор: 25.02.2020 / Кирилл	
Предусловия: необходимо авторизоваться.	
Шаги теста:	Ожидаемый результат:
1. Выбрать в главном меню пункт: «2 – добавление фотографий». 2. Приложение запрашивает ссылку на фотографию и просит ввести описание.	1. Приложение добавляет приложение в базу данных. 2. Приложение информирует об успешном добавлении фотографии.

Фактический результат:

Введите ссылку на фотографию  
[https://www.google.com/url?sa=i&url=https%3A%2F%2Frosphoto.com%2Fbest-of-the-best%2F55\\_luchshih\\_foto\\_online-galerei-4112&psig=A0vVaw1v0-FBVoWlf1CJTXXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPCqyfju70cCFQAAAAAdAAAAABAZ](https://www.google.com/url?sa=i&url=https%3A%2F%2Frosphoto.com%2Fbest-of-the-best%2F55_luchshih_foto_online-galerei-4112&psig=A0vVaw1v0-FBVoWlf1CJTXXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPCqyfju70cCFQAAAAAdAAAAABAZ)  
Введите описание  
Картинка  
**фото успешно добавлено**

ID и наименование: ТС-3 Удаление фотографий	
Ссылка на требование: FR-03	
Дата создания / Автор: 25.02.2020 / Кирилл	
Предусловия: необходимо авторизоваться и наличие хотя бы одной фотографии у пользователя.	
Шаги теста:	Ожидаемый результат:
1. Выбрать в главном меню пункт: «3 – удаление своих фотографий».	1. Сообщение об успешном удалении фотографии.
2. Выбрать номер изображения, которое хотим удалить	2. Приложение удаляет фотографию из файла.

Фактический результат:

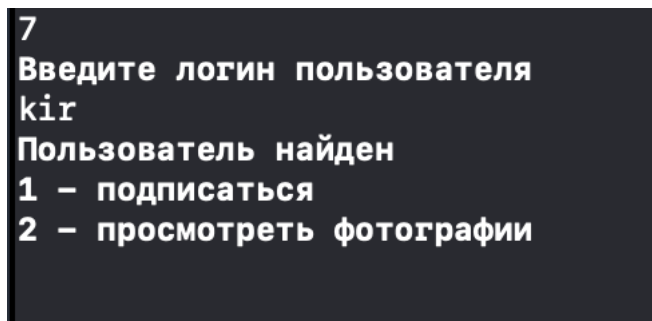
```

Выберите номер фотографии, которую хотите удалить
1:dog:https://www.google.com/url?sa=i&url=https%3A%2F%2Fbigpicture
.ru%2F%3Fp%3D709152&psig=A0vVaw1v0-FBVow1f1CJTYXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRxqFwoTC
PCqyfju70cCFQAAAAAdAAAAABAJ
2:tramp:https://www.google.com/url?sa=i&url=https%3A%2F%2Ftvzvezda
.ru%2Fnews%2Fvstrane_i_mire%2Fcontent%2F201812121749-vhok
.htm&psig=A0vVaw1v0-FBVow1f1CJTYXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPCqyfju70cCFQAA
AAAdAAAAABAO
1
Вы выбрали изображение :
https://www.google.com/url?sa=i&url=https%3A%2F%2Fbigpicture
.ru%2F%3Fp%3D709152&psig=A0vVaw1v0-FBVow1f1CJTYXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRxqFwoTC
PCqyfju70cCFQAAAAAdAAAAABAJ dog
Изображение успешно удалено

```

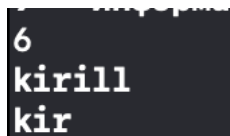
ID и наименование:      ТС-4 Поиск пользователей	
Ссылка на требование:    FR-04	
Дата создания / Автор:    25.02.2020 / Кирилл	
Предусловия: необходимо авторизоваться и наличие 2+ пользователей в базе данных.	
Шаги теста:	Ожидаемый результат:
1. Выбрать в главном меню пункт: «7 – Поиск пользователей»; 2. Ввести логин пользователя, которого хотим найти.	1. Приложение сообщает нам об успешном нахождении пользователя; 2. Приложение предлагает дальнейшие действия – подписаться или просмотреть фотографии;

Фактический результат:



ID и наименование: TC-5 Просмотр подписок	
Ссылка на требование: FR-05	
Дата создания / Автор: 25.02.2020 / Кирилл	
Предусловия: необходимо авторизоваться и наличие 2+ пользователей в базе данных.	
Шаги теста:	Ожидаемый результат:
1. Выбрать в главном меню пункт: «6 – просмотр подписок».	1. Приложение выводит список подписок;

Фактический результат:



```
6
kirill
kir
```



ID и наименование: TC-6 Редактирование описания у фотографий	
Ссылка на требование: FR-06	
Дата создания / Автор: 25.02.2020 / Кирилл	
Предусловия: необходимо авторизоваться и наличие хотя бы 1 фотографии	
Шаги теста:	Ожидаемый результат:
1. Выбрать в главном меню пункт: «4 - редактирование описания у фотографий». 2. Выбрать номер фотографии, описание которой хотим изменить. 3. Вести новое описание.	1. Приложение оповещает об успешном изменении описания. 2. Описание фотографии меняется в базе данных.

#### Фактический результат:

```

Выберите номер фотографии, описание которой хотите изменить
1:dog
https://www.google.com/url?sa=i&url=https%3A%2F%2Fbigpicture
.ru%2F%3Fp%3D709152&psig=A0vVaw1v0-FBVow1f1CJTYXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRx
PCqyfju70cCFQAAAAAdAAAAABAJ
2:tramp
https://www.google.com/url?sa=i&url=https%3A%2F%2Ftvzvezda
.ru%2Fnews%2Fvstrane_i_mire%2Fcontent%2F201812121749-vhok
.htm&psig=A0vVaw1v0-FBVow1f1CJTYXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRxqFwoTCPCqyfju70c
AAAdAAAAABA0
1
Вы выбрали изображение :
dog
https://www.google.com/url?sa=i&url=https%3A%2F%2Fbigpicture
.ru%2F%3Fp%3D709152&psig=A0vVaw1v0-FBVow1f1CJTYXbr7CK&ust=1582725726051000&source=images&cd=vfe&ved=0CAIQjRx
PCqyfju70cCFQAAAAAdAAAAABAJ
Введите новое описание
собака
Описание успешно изменено

```

ID и наименование: TC-7 Изменение логина и пароля	
Ссылка на требование: FR-07	
Дата создания / Автор: 25.02.2020 / Кирилл	
Предусловия: необходимо авторизоваться.	
Шаги теста:	Ожидаемый результат:
1. Выбрать в главном меню пункт: «5 - редактирование своих данных логин/пароль»; 2. Выбрать пункт: 1-логин 3. Ввести новый логин	1. Приложение выводит информацию о том, что логин успешно изменен. 2. Данные пользователя изменяются в базе данных.

Фактический результат:

```

Что вы хотите изменить ?
1 - логин
2 - пароль
1
Введите новый логин
qwerty
логин успешно изменен

```

ID и наименование: TC-8 Информация о программе	
Ссылка на требование: FR-08	
Дата создания / Автор: 25.02.2020 / Кирилл	
Предусловия: необходимо авторизоваться.	
Шаги теста:	Ожидаемый результат:
1. Выбрать в главном меню пункт: «9 - информация о программе»;	1. Приложение выводит информацию о программе.

Фактический результат:

**Приложение для обмена фотографиями**  
**Разработчик – Кирилл Корнющенко**  
**Год выпуска – 2020**

ID и наименование:	ТС-9 Завершение работы приложения и сохранение данных		
Ссылка на требование:	FR-09		
Дата создания / Автор:	25.02.2020 / Кирилл		
Предусловия: необходимо авторизоваться.			
Шаги теста:		Ожидаемый результат:	
1. Выбрать в главном меню пункт: «8 - выход и сохранение данных».		1. Приложение завершает свою работу и сохраняет данные в базе данных.	

Фактический результат:

**Program ended with exit code: 0**

ID и наименование: TC-10 Вывод информации о программе	
Ссылка на требование: FR-10	
Дата создания / Автор: 25.02.2020 / Кирилл	
Предусловия: предварительная настройка среды или системы не требуется.	
Шаги теста:	Ожидаемый результат:
1. Выбрать в пункт: «2 - регистрация». 2. Ввести логин 3. Ввести пароль 4. Ввести повторно пароль	1. Приложение сохраняет нового пользователя. 2. Приложение выводит сообщение об успешной регистрации.

Фактический результат:

```

Выберите пункт меню
1 – авторизация
2 – регистрация
2
Введите логин
QWERTY
Введите пароль
QWERTY
Введите пароль еще раз
QWERTY
Вы успешно зарегистрировались

```

Выполним анализ тестового покрытия при помощи построения матрицы соответствия требований:

Требование	FR-01	FR-02	FR-03	FR-04	FR-05	FR-06	FR-07	FR-08	FR-09	FR-10
Тест кейсы	1	1	1	1	1	1	1	1	1	1
ТС-1	✓									
ТС-2		✓								
ТС-3			✓							
ТС-4				✓						
ТС-5					✓					
ТС-6						✓				
ТС-7							✓			
ТС-8								✓		
ТС-9									✓	
ТС-10										✓

Таким образом, из данной матрицы следует, что для всех функциональных требований приложения есть хотя бы по одному тесту, соответственно тестовое покрытие функциональных требований в данном случае составляет 100%.

## Выводы по работе

В результате выполнения данной работы было разработано консольное приложение для обмена фотографиями. Составлено необходимое количество тест кейсов, в соответствии с которыми выполнено тестирование приложения, и построена матрица соответствия требований для анализа тестового покрытия.

Анализ покрытия показывает, что составленные тесты покрывают 100% функциональных требований, однако нельзя говорить о том, что само приложение было полностью протестировано, так как выбранная метрика тестового покрытия работает только с требованиями и не учитывает конечную реализацию, соответственно вопрос о дефектах в коде остается открытым. В тоже время по результатам тестирования серьезных расхождений со спецификацией выявлено не было.

В соответствии с предложенной концепцией продукта был сформулирован ряд нефункциональных требований, одним из которых является защита от очевидно ошибочных/неверных действий пользователя. Данное требование было реализовано лишь частично посредством следующих проверок:

- если встречается неизвестный символ, то приложение вылетает;
- описание и ссылка фотографии ограничены длинной string;
- приложение проверяет наличие используемых файлов.

Очевидно, что представленный контроль данных для ввода не учитывает изобретательность пользователя и оставляет некоторые критические моменты без внимания, в частности можно отметить следующее:

- нет контроля дублирующей информации (можно добавить два или более одинаковых пользователя/фотографии/подписчика);
- нет защиты от редактирования данных в файлах вручную (приложение не проверяет целостность и корректность данных в файлах, соответственно при выводе информации может возникнуть ситуация, которую возможно интерпретировать как дефект);

В фактических результатах тестирования приведены только скриншоты реакции приложения через интерфейс пользователя, скриншоты изменения данных в файлах не приводятся в виду их малой информативности.

Таким образом, можно заключить, что выполненная работа соответствует поставленной задаче и отвечает всем сформулированным в методических указаниях требованиям.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Виггерс, Карл. Разработка требований к программному обеспечению = Software Requirements: пер. с англ.; 3-е издание, дополненное / Карл Виггерс, Джой Битти — М.: Издательство «Русская редакция», 2014. — 736 с.: ил.
2. Куликов, С. С. Тестирование программного обеспечения. Базовый курс / С. С. Куликов — Минск: Четыре четверти, 2017 — 312 с.
3. Traceability matrix [Электронный ресурс]: From Wikipedia, the free encyclopedia. — Wikipedia, 2020 — URL: [https://en.wikipedia.org/wiki/Traceability\\_matrix](https://en.wikipedia.org/wiki/Traceability_matrix) (дата обращения: 18.02.2020)
4. Работа с файлами в C++. Библиотека fstream [Электронный ресурс]. — PureCodeCpp, 2014-2020 — URL: <https://purecodecpp.com/archives/2751> (дата обращения: 18.02.2020)



## ПРИЛОЖЕНИЕ А

### Исходный код программы

```
// main.cpp
// tinder
//
// Created by Кирилл on 22.02.2020.
// Copyright © 2020 Кирилл. All rights reserved.
//

#include <iostream>
#include <fstream>
#include <vector>
#include <string>

using namespace std;

//MARK: Структуры
struct Photo{
    string href;
    string description;
};

struct User{
    string login;
    string password;
    vector<string> followers;
    vector<Photo> photoVector;
};

//MARK: Class
class Instagramm{
private:
    const string path = "/Users/kirill/Desktop/tinder/tinder/file.txt";
    vector<User> dataAllUser;
    User activeUser;
    //обработка строки из файла
    void readDataSupport(string data);
    // регистрация
    void registration();
    // вход
    void entrance();
    //menu
    void menu();
    //add image
    void addImage();
    //finc people
    void findPeople();
    //просмотр фото
    void lookPhotoActiveUser();
    //просмотр подписчиков
    void lookFollowers();
    //редактирование данных своих
    void editData();
    //редактирование описания у фотографий
    void editDescriptionPhoto();
    //удаление фотографий
    void removePhoto();
    //save
    void saveAndClose();
    //старт
    void start();
    //info
```

```

    void info();
public:
    //считывание данных
    void readData();
};

int main() {
    Instagramm insta;
    insta.readData();
    return 0;
}

//MARK: Считывание данных из файла
void Instagramm::readDataSupport(string data){
    User user;
    string line;
    int count = 0;
    vector<string> followers;
    vector<Photo> photoVector;
    Photo photo;
    for (int i=0;i<data.length();i++){
        if (data[i]!='#' && count==0){
            line+=data[i];
        }else if (data[i]=='#' && count==0){
            user.login = line;
            line = "";
            count++;
        }else if (data[i]!='#' && count==1){
            line+=data[i];
        }else if (data[i]=='#' && count==1){
            user.password = line;
            count++;
            line = "";
        }else if (data[i]!='*' && count==2){
            if (data[i]!='|'){
                line+=data[i];
            }else{
                followers.push_back(line);
                line = "";
            }
        }else if (data[i]=='*' && count==2){
            count++;
            user.followers = followers;
            line = "";
        }else{
            if (data[i]!=' ' && data[i]!='|'){
                line+=data[i];
            }else if (data[i]==' '){
                photo.description = line;
                line = "";
            }else if (data[i]=='|'){
                photo.href = line;
                line = "";
                photoVector.push_back(photo);
            }
        }
    }
    user.photoVector = photoVector;
    dataAllUser.push_back(user);
}

void Instagramm::readData(){
    ifstream file(path);
    if (!file) {
        cout << "файл не найден" << endl;
    }
}

```

```

    } else {
        string data;
        while (getline(file, data)) {
            readDataSupport(data);
        }
        file.close();
    }
    start();
}

//MARK: Вход в приложение
void Instagramm::start(){
    int choose;
    cout << "Выберите пункт меню " << endl;
    cout << "1 - авторизация " << endl;
    cout << "2 - регистрация " << endl;
    cin >> choose;
    switch (choose) {
        case 1:
            entrance();
            break;
        case 2:
            registration();
            break;
        default:
            cout << "error" << endl;
            break;
    }
}

void Instagramm::registration(){
    string login, password, password2;
    cout << "Введите логин " << endl;
    cin >> login;
    cout << "Введите пароль " << endl;
    cin >> password;
    cout << "Введите пароль еще раз " << endl;
    cin >> password2;
    if (password == password2){
        cout << "Вы успешно зарегистрировались " << endl;
        User user;
        user.login = login;
        user.password = password;
        dataAllUser.push_back(user);
        activeUser = user;
        menu();
    }else{
        cout << "Пароли не совпадают " << endl;
    }
}

void Instagramm::entrance(){
    string login, password;
    cout << "Введите логин " << endl;
    cin >> login;
    cout << "Введите пароль " << endl;
    cin >> password;

    for(int i=0;i<dataAllUser.size();i++){
        if (dataAllUser[i].login == login){
            if(dataAllUser[i].password == password){
                activeUser = dataAllUser[i];
                cout << "Вы успешно авторизировались " << endl;
                menu();
            }else {

```

```

        cout << "Пароль введен неверно " << endl;
    }
    break;
}
}
}

//MARK: меню
void Instagramm::menu(){
    int choose = -1;
    while (choose!=8) {
        system("cls");
        cout << "Меню" << endl;
        cout << "1 - просмотр своих фотографий" << endl;
        cout << "2 - добавление фотографий " << endl;
        cout << "3 - удаление своих фотографий " << endl;
        cout << "4 - редактирование описания у фотографий " << endl;
        cout << "5 - редактирование своих данных login/password " << endl;
        cout << "6 - просмотр подписок " << endl;
        cout << "7 - поиск пользователей " << endl;
        cout << "8 - выход и сохранение данных " << endl;
        cout << "9 - информация о программе " << endl;
        cin >> choose;
        switch (choose) {
            case 1:
                lookPhotoActiveUser();
                break;
            case 2:
                addImage();
                break;
            case 3:
                removePhoto();
                break;
            case 4:
                editDescriptionPhoto();
                break;
            case 5:
                editData();
                break;
            case 6:
                lookFollowers();
                break;
            case 7:
                findPeople();
                break;
            case 8:
                saveAndClose();
                break;
            case 9:
                info();
                break;
            default:
                break;
        }
    }
}

void Instagramm::addImage(){
    Photo photo;
    cout << "Введите ссылку на фотографию " << endl;
    cin >> photo.href;
    cout << "Введите описание " << endl;
    cin >> photo.description;
    activeUser.photoVector.push_back(photo);
    cout << "фото успешно добавлено " << endl;
}

```

```

}

void Instagramm::findPeople(){
    string login;
    bool find = false;
    cout << "Введите логин пользователя " << endl;
    cin >> login;
    for (int i=0;i<dataAllUser.size();i++){
        if(dataAllUser[i].login == login){
            find = true;
            cout << "Пользователь найден " << endl;
            cout << "1 - подписаться" << endl;
            cout << "2 - просмотреть фотографии " << endl;
            int choose;
            cin >> choose;
            switch (choose) {
                case 1:
                    activeUser.followers.push_back(dataAllUser[i].login);
                    cout << "Вы успешно подписались" << endl;
                    break;
                case 2:
                    for (int j=0;j<dataAllUser[i].photoVector.size();j++){
                        cout << dataAllUser[i].photoVector[j].href << " " << dataAllUser[i].photoVector[j].description <<
endl;
                    }
                    if (dataAllUser[i].photoVector.size() == 0){
                        cout << "у пользователя нету фото" << endl;
                    }
                    break;
                default:
                    break;
            }
            break;
        }
    }
    if (!find){
        cout << "Пользователь не найден " << endl;
    }
}

void Instagramm::lookPhotoActiveUser(){
    for (int i=0;i<activeUser.photoVector.size();i++){
        cout << activeUser.photoVector[i].description << "->" << activeUser.photoVector[i].href << endl;
    }
    if (activeUser.photoVector.size() == 0){
        cout << "у вас нету фото" << endl;
    }
}

void Instagramm::lookFollowers(){
    for (int i=0;i<activeUser.followers.size();i++){
        cout << activeUser.followers[i] << endl;
    }
    if (activeUser.followers.size() == 0){
        cout << "у вас нету подписчиков" << endl;
    }
}

void Instagramm::editData(){
    cout << "Что вы хотите изменить?" << endl;
    cout << "1 - логин" << endl;
    cout << "2 - пароль" << endl;
    int choose;
    cin >> choose;
    switch (choose) {

```

```

    case 1:{
        string login;
        int index = 0;
        cout << "Введите новый логин" << endl;
        cin >> login;
        for (int i=0;i<dataAllUser.size();i++){
            if (dataAllUser[i].login == activeUser.login){
                index = i;
                break;
            }
        }
        activeUser.login = login;
        dataAllUser[index] = activeUser;
        cout << "логин успешно изменен" << endl;
        break;
    }
    case 2:{
        string password,password2;
        cout << "Введите новый пароль " << endl;
        cin >> password;
        cout << "Повторите новый пароль" << endl;
        cin >> password2;
        if (password == password2){
            activeUser.password = password;
            cout << "Пароль успешно изменен" << endl;
        }else{
            cout << "Пароли разные " << endl;
        }
        break;
    }
    default:
        cout << "error" << endl;
        break;
}
}

void Instagramm::editDescriptionPhoto(){
    cout << "Выберите номер фотографии, описание которой хотите изменить" << endl;
    for (int i=0;i<activeUser.photoVector.size();i++){
        cout << i+1 << ":" << activeUser.photoVector[i].description << " " << activeUser.photoVector[i].href << endl;
    }
    int index;
    cin >> index;
    cout << "Вы выбрали изображение :" << endl;
    cout << activeUser.photoVector[index-1].description << " " << activeUser.photoVector[index-1].href << endl;
    string description;
    cout << "Введите новое описание" << endl;
    cin >> description;
    activeUser.photoVector[index-1].description = description;
    cout << "Описание успешно изменено" << endl;
}

void Instagramm::removePhoto(){
    cout << "Выберите номер фотографии, которую хотите удалить" << endl;
    for (int i=0;i<activeUser.photoVector.size();i++){
        cout << i+1 << ":" << activeUser.photoVector[i].description << " " << activeUser.photoVector[i].href << endl;
    }
    int index;
    cin >> index;
    cout << "Вы выбрали изображение :" << endl;
    cout << activeUser.photoVector[index-1].href << " " << activeUser.photoVector[index-1].description << endl;
    activeUser.photoVector.erase(activeUser.photoVector.begin()+index-1);
    cout << "Изображение успешно удалено" << endl;
}

```

```

void Instagramm::saveAndClose(){
    ofstream file(path);
    if (file.is_open()){
        for (int i=0;i<dataAllUser.size();i++){
            if (dataAllUser[i].login == activeUser.login){
                dataAllUser[i] = activeUser;
                break;
            }
        }
        for (int i=0;i<dataAllUser.size();i++){
            file << dataAllUser[i].login << "#" << dataAllUser[i].password << "#";
            for (int j=0;j<dataAllUser[i].followers.size();j++){
                file << dataAllUser[i].followers[j] << "!";
            }
            file << "*";
            for (int k=0;k<dataAllUser[i].photoVector.size();k++){
                file << dataAllUser[i].photoVector[k].description << " " << dataAllUser[i].photoVector[k].href << "|";
            }
            file << endl;
        }
    }
    else{
        cout << "файл не найден " << endl;
    }
}

void Instagramm::info(){
    cout << "Приложение для обмена фотографиями" << endl;
    cout << "Разработчик - Кирилл Корнющенко" << endl;
    cout << "Год выпуска - 2020" << endl;
}

```