Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет среднего профессионального образования

Санкт-Петербург 2019

Специальность: 09.02.07 Информационные системы и программирование

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №3

по теме: Работа с внешними приложениями

**Цель работы:** Написать программу, которая по REST протоколу получает публичные данные с сайта <http://vk.com/>.

Выполнил:

Студент группы Y2433

Тертень Д. В.

Проверил:

\_\_\_\_\_\_\_\_ Антонов М. Б.

Дата: “\_\_” \_\_\_\_\_\_ 2020г.

Оценка:\_\_\_\_\_\_\_\_\_\_\_

Этапы создания приложения:

1. создать новое приложение в настройках Вконтакте;
2. получить сервисный токен для отправки запросов на API Вконтакте;
3. изучить API Вконтакте ([https://vk.com/dev/methods).](https://vk.com/dev/openapi)

Само приложение должно предоставлять функцию ввода уникального ID профиля и кнопку его получения. При нажатии на кнопку данные должны показаться в заранее подготовленной форме. Данные которые необходимо вывести (если у профиля они не заданы или скрыты оставить пустую строку):

- фамилия;

- имя;

- ник-нейм;

- день рождения;

- пол;

- фотография профиля;

- список друзей (минимум ID каждого друга).

**Код программы:**

Файл mainwindow.h:

#ifndef MAINWINDOW\_H

#define MAINWINDOW\_H

#include <QMainWindow>

#include <QString>

#include <QtNetwork/QNetworkAccessManager>

#include <QtNetwork/QNetworkReply>

#include <QtNetwork/QNetworkRequest>

#include <QTimer>

#include <QEventLoop>

#include <QUrl>

#include <QObject>

#include <QJsonObject>

#include <QJsonDocument>

#include <QJsonArray>

#include <QFile>

#include <ui\_mainwindow.h>

QT\_BEGIN\_NAMESPACE

namespace Ui { class MainWindow; }

QT\_END\_NAMESPACE

class MainWindow : public QMainWindow

{

Q\_OBJECT

private:

Ui::MainWindow \*ui;

QString token = "9f3ecd5d9f3ecd5d9f3ecd5d349f4e444e99f3e9f3ecd5dc1455d52725a8d57db27f2f3";

QNetworkAccessManager networkManager;

public:

MainWindow(QWidget \*parent = nullptr);

~MainWindow();

void getInfo()

{

/\*\*

\* @brief Получение информации и пользователе

\*/

QNetworkRequest requestUser(QUrl("https://api.vk.com/method/users.get?user\_ids=" + ui->idLineEdit->text() + "&fields=bdate,nickname,sex,photo\_100&lang=3&access\_token=" + token + "&v=5.103"));

QNetworkReply \*response = networkManager.get(requestUser);

QEventLoop event;

connect(response,SIGNAL(finished()),&event,SLOT(quit()));

event.exec();

disconnect(response,SIGNAL(finished()),&event,SLOT(quit()));

QJsonArray itemArray = QJsonDocument::fromJson(response->readAll()).object()["response"].toArray();

QJsonObject obj = itemArray.at(0).toObject();

getPhoto(obj.value("photo\_100").toString());

ui->surnameLabel->setText("Фамилия: " + obj.value("last\_name").toString());

ui->nameLabel->setText("Имя: " + obj.value("first\_name").toString());

ui->nicknameLabel->setText("Никнейм: " + obj.value("nickname").toString());

ui->birthdayLabel->setText("День рождения: " + obj.value("bdate").toString());

int sex = obj.value("sex").toInt();

if(sex==1)

ui->sexLabel->setText("Пол: женский");

else if(sex==2)

ui->sexLabel->setText("Пол: мужской");

else ui->sexLabel->setText("Пол: не указан");

delete response;

}

void getFrinds(){

/\*\*

\* @brief Получение списка друзей пользователей

\*/

QNetworkRequest requestFriends(QUrl("https://api.vk.com/method/friends.get?user\_id=" + ui->idLineEdit->text() + "&fields=&lang=3&access\_token=" + token + "&v=5.103"));

QNetworkReply \*response = networkManager.get(requestFriends);

QEventLoop event;

connect(response,SIGNAL(finished()),&event,SLOT(quit()));

event.exec();

disconnect(response,SIGNAL(finished()),&event,SLOT(quit()));

QJsonArray itemArray = QJsonDocument::fromJson(response->readAll())["response"].toObject().value("items").toArray();

QJsonDocument doc;

doc.setArray(itemArray);

QString friends(doc.toJson());

qDebug() << friends;

friends.remove('\n').remove('[').remove(']');

if(friends == "")

ui->friendsLabel->setText("Друзей нет или они скрыты настройками приватности");

else ui->friendsLabel->setText("Друзья: " + friends);

delete response;

}

void getPhoto(QString photoURL){

/\*\*

\* @brief Загрузка изображения профиля пользователя

\* @param Ссылка на изображение профиля

\*/

QUrl getURL = QUrl(photoURL);

QNetworkRequest requestPhoto(getURL);

QNetworkReply \*response = networkManager.get(requestPhoto);

QEventLoop event;

connect(response,SIGNAL(finished()),&event,SLOT(quit()));

event.exec();\

disconnect(response,SIGNAL(finished()),&event,SLOT(quit()));

QFile picture("picture.jpg");

picture.open(QIODevice::WriteOnly);

picture.write(response->readAll());

picture.close();

QPixmap pixmap("picture.jpg");

ui->pictureLabel->setPixmap(pixmap);

delete response;

}

};

#endif // MAINWINDOW\_H

Файл mainwindow.cpp:

#include "mainwindow.h"

#include "ui\_mainwindow.h"

MainWindow::MainWindow(QWidget \*parent): QMainWindow(parent), ui(new Ui::MainWindow)

{

ui->setupUi(this);

connect( ui->searchButton, &QPushButton::clicked, this, &MainWindow::getInfo );

connect( ui->searchButton, &QPushButton::clicked, this, &MainWindow::getFrinds );

}

MainWindow::~MainWindow()

{

delete ui;

}

Файл main.cpp:

#include "mainwindow.h"

#include <QApplication>

int main(int argc, char \*argv[])

{

QApplication a(argc, argv);

MainWindow w;

w.show();

return a.exec();

}

**Вывод:** в ходе лабораторной работы были изучены основы работы с REST-запросами, создано приложение, использующее API стороннего сайта.