**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**

**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

«КАЛУЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ им. К.Э. ЦИОЛКОВСКОГО»

Физико–технологический институт

Кафедра информатики и информационных технологий

Направление подготовки 09.03.02 Информационные системы и технологии

*ДИПЛОМНАЯ РАБОТА*

«Создание бота для «Автопостинга» в социальной сети ВКонтакте»

студента 4 курса очной формы обучения

**Критенко Валентина Григорьевича**

**Работа выполнена:**

**научный руководитель**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*Старший преподаватель*

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

*Виноградский Вадим Генадьевич*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*/подпись/*

**г. Калуга, 2019 г.**

Содержание:

[Введение 3](#_Toc8032766)

[Глава 1. Теоретические основы разработки веб приложения 5](#_Toc8032767)

[1.1 Назначение и область применения разработки 5](#_Toc8032768)

[1.2 Установление требований к разрабатываемой системе 6](#_Toc8032769)

[1.3 Анализ существующих разработок 8](#_Toc8032770)

[Глава 2. Разработка бота веб приложения  десктопной версии 14](#_Toc8032771)

[для социальной сети "ВКонтакте 14](#_Toc8032772)

[2.1. Описание и обоснование выбора состава программных средств разработки 14](#_Toc8032773)

[2.2 Алгоритм программы 17](#_Toc8032774)

[2.3 Руководство разработчика 18](#_Toc8032775)

[Заключение 25](#_Toc8032776)

[Список литературы 26](#_Toc8032777)

[Приложения 27](#_Toc8032778)

[Приложение 1. Код файла Resizer 27](#_Toc8032779)

[Приложение 2. Код файла VK API 30](#_Toc8032780)

[Приложение 3. Код внутри кнопок 35](#_Toc8032781)

# Введение

Интернет и автоматизация всё больше и больше проникают в жизнь людей. Он доказал свою способность объединять людей, промышленность, предприятия в общем информационном пространстве. Автоматизация помогает высвобождать человеческие ресурсы, (которые в дальнейшем повышают свою квалификацию или переходят в другие сферы), понижает цену продукта (учет разных дефектов, логистики и тп).  
 Самым первым социально-сетевым ресурсом следует считать американский портал Classmates.com. Именно по его образу и подобию создавались в свое время «Одноклассники». Проект был запущен в 1995 году Рэнди Конрадсом и предоставлял всем желающим возможность найти своих одноклассников, однокурсников и сослуживцев.

Первая популярная социальная сеть Это - Facebook, запущенный ещё в 2004 году. Соц сети, заслужившие популярность во всём мире, открывает именно этот ресурс. Однако, первые два года он был корпоративным и только в 2006 году (в сентябре), каждый получил возможность на нем зарегистрироваться. Именно с 2006 года принято считать зарождение мира социальных сетей, так как именно с него началась активная популяризация таких популярных проектов как MySpace и LinkedIn.

В данной работе речь будет вестись о боте для социальной сети ВКонтакте и автоматизации некоторых процессов при взаимодействии с нею. Бот – специальная программа, выполняющая автоматически и/или по заданному расписанию какие–либо действия через интерфейсы, предназначенные для людей. Актуальность данной работы заключается в расширении сферы использования ботов, распространения информационных технологий среди пользователей, создании дополнительного инструментария для работы с социальной сетью. Веб приложения стали появляться в последние несколько лет, в связи с развитием рынка мобильных платформ. Программное обеспечение, которое позволяет автоматизировать процессы пользователя (бот), распространено крайне мало, в большинстве случаев является платным и не даёт всех возможностей для пользователя.

Боты призваны помочь человеку с поиском чего–либо в сети интернет: услуга, информация или какая–либо вещь. Потребность людей в общении, знакомствах породила социальные сети. ВКонтакте – большая социальная сеть, нацеленная на молодежь в первую очередь и лиц до 30 лет. Пользователям в таком возрастном диапазоне проще адаптироваться к современным технологиям. Бот даст дополнительный инструмент для взаимодействия с социальной сетью, в более удобном и простом формате.

Цель дипломной работы: создание приложения, которое позволит взаимодействовать пользователю с социальной сетью ВКонтакте, получать и отправлять какую–либо информацию, вносить изменения, в дальнейшем производить автоматизацию выкладки постов.

Соответственно были выбраны задачи дипломной работы:

1. История развития и возникновения ботов в современных интернет приложениях.
2. Виды авторизации и взаимодействия с социальной сетью
3. Боты в социальных сетях
4. Обзор аналогов
5. Выбор программных средств для разработки бота

Работа состоит из введения, двух глав, заключения, списка использованной литературы и приложения.

# Глава 1. Теоретические основы разработки веб приложения

1. **История развития и возникновения ботов в современных интернет приложениях.**

Боты для интернет приложений появились относительно недавно. Их потенциал пока не исчерпан. Первыми появились чат-боты. Чат-бот — компьютерная программа для общения с человеком через понятные для последнего «интерфейсы», текст или голос.

Первым чат ботом была Элиза (1966г) создал Джозеф Вейценбаум. Она была программой для психологической помощи людям. Могла выделять ключевые слова и подставлять под шаблон, если нужной вариативности фраз не было, то ответ звучал общими фразами («понятно» и др) и переводила тему разговора. Программа не имитировала специалиста – психотерапевта, а только пародировала поведение с пациентом.

Второй чат бот эмитировал поведение пациента страдающего параноидальной шизофренией Пэри. Создал его психиатр Кеннет Колби. В ходе тестирования 48% врачей психиатров не понимали что общаются с машиной.  
В 1972 году на ICCC (международной конференции компьютерных коммуникаций) Пэри и Элиза были соединены по сети ARPANET и пытались общаться между собой.  
 A.L.I.C.E. (Элис) — это аббревиатура от Artificial Linguistic Internet Computer Entity (Искусственное лингвистическое интернетное компьютерное существо).Был создан в 1995 году. Первый кто смог вести разговор на естественном для человека языке, к примеру русском и английском. Она с помощью эвристического анализа фраз людей с образцами в базе данных могла вести разговор близкий к реальному. Печалило только-то, что бот отвечал общими диалогами, но на это время был прорыв. Получила 3 премии Лёбнера, как человечный бот, но тест Тьюринга так и не смогла пройти.

В 1988 году началась первая разработка в области искусственного интеллекта для чат-ботов. Запуск произошел спустя 9 лет в 1997 году. Назвали его Jabberwacky. Он имитировал развлекательный разговор. Бот запоминает информацию из диалога с человеком, затем при следующих диалогах обращался к ней с помощью контекстных шаблонов. Он запоминает сленг, формирует отдельные характеры для разной вариативности диалогов.

На его основе написан чат-бот веб-приложение Cleverbot. В начале оно поддерживало беседу только на английском, позже перечень поддерживаемых языков расширили. Бот стал поддерживать поэтов.

В 2001 году был создан SmarterСhild в компании ActiveBuddy. Она представляла словесную приключенческую игру. В дальнейшем его функционал был расширен: информация о погоде, быстрый доступ к новостям, калькулятор, переводчик.

Чат-бот Ватсон (2006г). Стал первым в эпоху больших данных. Система может обрабатывать естественные языки, анализировать запросы, искать информацию и предоставлять её пользователю. Он способен обучаться в процессе диалога и может моментально отвечать на вопросы пользователя.

Чат-бот Тау. Был создан в 2016 году компанией Microsoft.За сутки благодаря пользователям Twitter научился расизму и жестокости. В дальнейшем в связи со скандалом по его поведению был отключен. Компания принесла свои извинения.

## 1.2 Область применения ботов

Ботов можно встретить на любой платформе, будь это мессенджер, веб-портал или смартфон. Они применяются в рамках определенных сценариев и под конкретные нужды пользователя.

Выделяют 4 вида ботов по видам использования:  
1 Техническая поддержка

2 Поиска по заданным параметрам

3 Потребление контента и развлечения

4 Покупки

Разрабатываемый программный продукт предназначен для обычных людей. Первое и основное требование к приложению – применение авторизации с социальной сетью Вконтакте с помощью методики OAuth 2.0– требование является обязательным для всех приложений социальной сети ВКонтакте, служит стандартом для защиты информации и аккаунта пользователя. Её реализовала команда разработчиков ВКонтакте – vk api. Метод OAuth 2.0 – протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе. Протокол избавляет от необходимости доверять приложению логин и пароль, а также позволяет выдавать ограниченный набор прав, а не все сразу. Второе – взаимодействие пользователя с социальной сетью оптимизированными способами – уменьшено количество переходов между элементами, соответственно тратится меньше времени на то, чтобы пользователь добрался до нужной операции. Для нормальной работы программы необходим IBM совместимый компьютер с тактовой частотой процессора не ниже 1.8ГГц и выше, оперативной памятью 4 ГБ, жесткий диск объемом не менее 30 ГБ, Windows 8(x64) и выше.

Требования к дизайну разрабатываемого программного продукта:

1. Структурированная информация
2. Эргономика
3. Приятная визуализация.

Программные требования:

1. Работа с vk api не ниже 5 версии
2. Написание приложения на версии netframework 4 и выше
3. Возможность модернизации
4. Возможность работы с БД

## Анализ существующих разработок

В ходе исследования рынка программных средств, которые занимаются созданием ботов и сбором статистики, были найдены следующие:

*Kuku*

Kuku – многофункциональная платформа, возможности которой составляют автопостинг, создание контент–плана и аналитику социальных сетей. Кроме того, сервис позволяет объединять различные аккаунты в группы или проекты, адаптировать контент под разные сети, а так же автоматически ставить лайки на новые публикации друзей. Является официальным партнёром 10 соц. сетей. Отчёты выгружаются в PDF формате.

Рис 1. Главное окно «Kuku».

Плюсы: Есть бесплатный 14–дневный пробный период.

Минусы:

Далее минимальный тариф стоит 399 р (не более 5 аккаунтов и не более 500 постов).

*TargetHunter*

Програмный продукт – TargetHunter – предоставляет современные инструменты для анализа и подбора активной целевой аудитории ВКонтакте. Сервис позволяет значительно снизить рекламные расходы, за счет проработанных алгоритмов.

Рис 2. Окно запроса «TargetHunter»

Плюсы:

Приложение предоставляет 20 функций совершенно бесплатно.

Минусы:

Стандартный тариф включает 90 функций для сбора ЦА и стоит 799 рублей на 30 дней.

Не обеспечивает возможности постить записи больше недели

Для наших целей является избыточным функционал, платным.

*YouScan*

YouScan мониторинг социальных сетей, сайтов отзывов и СМИ. Отслеживает упоминания бренда или любых друг слов.

Рис 3. Окно отчета «YouScan»

Плюсы:  
Профессиональный программный продукт, содержит нужные инструменты.

Профессиональная поддержка

Удобный интерфейс

Информативные отчёты

Поддержка 24/7.

Пробный период

Минусы:

Пакет, с минимальными возможностями стоит 500$ в месяц. Максимальный пакет стоит 5000$.

Демо доступно только по запросу.

*Sobot*

Sobot – сервис автоматизации действий в социальных сетях, запущенный в 2015 году. На сайте можно скачать бот для соц. сетей: Мой Мир, ВКонтакте, Одноклассники. Есть партнерская программа – 45% от заработка системы. На сайте можно скачать бот–программу, ознакомится с инструкцией по управлению ботом. Программа имеет массу полезных функций, например, может произвести выборку пользователей из указанной группы. Бот может вступить во множество сообществ из заданного списка, публиковать новости в разные группы в автоматическом режиме. Делать репосты, ставить лайки, публиковать фото и видео. Приглашать друзей, принимать приглашения. Через бот можно редактировать данные анкеты. Может осуществить массовую регистрацию пользователей из списка.

Рис 4. Окно запроса «Sobot»

Возможности Sobot

Делает публикацию постов, комментариев, ищет группы;

Занимается рассылкой личных сообщений;

Делает приглашения в группу и друзья;

Публикует комментарии в новостной ленте;

Создает черный список групп и пользователей;

Пакетное добавление аккаунтов;

Бот лайкает посты и фотографиям, делает репосты;

Автоматически подтверждает запрос на дружбу.

Вывод: в своем приложении будем объединять возможности других приложений (Сбор и предоставление аналитики в форме графиков, информации о пользователях, ведения нескольких аккаунтов)

# Глава 2. Разработка бота веб приложения  десктопной версии

# для социальной сети "ВКонтакте

## 2.1. Описание и обоснование выбора состава программных средств разработки

При начале разработки был выбор из нескольких программных сред разработки : VisualStudio2017, DevelStudio, Delphi 7. Delphi 7 – уже не отвечает современным требованиям, он поддерживает ограниченно разработку с веб формами и запросами. VisualStudio2017 – позволяет полноценно сделать клиент для декстопной версии приложения на языке C#. DevelStudio – среда программирования на php, можно полностью реализовать приложение. В итоге выбор пал на DevelStudio, т.к дальше при интеграции приложения с сервером не надо будет создавать с нуля или переписывать.

PHP является наиболее распространенным языком веб–программирования. Подавляющее большинство сайтов и веб–сервисов в интернете написано с помощью PHP. По некоторым оценкам PHP применяется более чем на 80% сайтов, среди которых такие сервисы, как facebook.com, vk.com, baidu.com и другие. И такая популярность неудивительна. Простота языка позволяет быстро и легко создавать сайты и порталы различной сложности. PHP был создан в 1994 году датским программистом Расмусом Лердорфом и изначально представлял собой набор скриптов на другом языке Perl. Позже этот набор скриптов был переписан в интерпретатор на языке Си. И с самого возникновения PHP (сокращение от PHP: Hypertext Preprocessor – PHP: Препроцессор гипертекста) представлял удобный набор инструментов для упрощенного создания веб–сайтов и веб–приложений. Для всех наиболее распространенных операционных системам (Windows, MacOS, Linux) есть свои версии пакетов разработки на PHP, а это значит, что можно создавать веб–сайты на любой из этих операционных систем. PHP может работать в связке с различными веб–серверами: Apache, Nginx, IISПростота и легкость освоения. Как правило, уже имея небольшой опыт в программировании на PHP, можно создавать простенькие веб–сайты. PHP похож на язык Си, поэтому, зная Си или один из языков с сиподобным синтаксисом, будет проще овладеть PHP. PHP поддерживает работу с множеством систем баз данных (MySQL, MSSQL, Oracle, Postgre, MongoDB и другие).Распространенность хостинговых услуг и их дешевизна. Так как, как правило, хостинговые компании размещают веб–сайты на PHP на веб–серверах Apache или Nginx, которые работают на одной из операционных систем семейства Linux. И веб–серверы, и операционные системы на базе Linux бесплатны, что снижает общую стоимость использования хостинга. Постоянное развитие. PHP продолжает развиваться, выходят все новые версии, которые несут новые функции, адаптируя язык программирования к новым вызовам. И, как правило, перейти на новую версию не составляет труда.

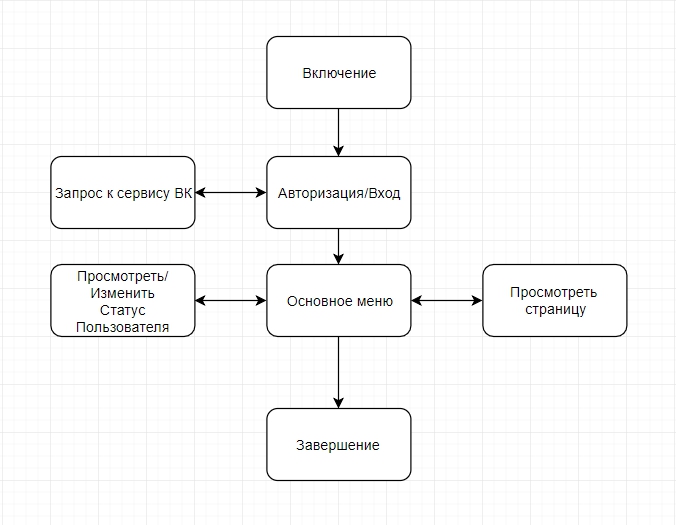
DevelStudio – среда программирования, ориентированная на создание программ с помощью языка PHP под Windows. Позволяет создавать приложения с помощью языка PHP. Для работы с вк api используется открытая архитектура авторизации OAuth 2.0 и язык PHP.

OAuth 2.0 – протокол авторизации, позволяющий выдать одному сервису (приложению) права на доступ к ресурсам пользователя на другом сервисе. Протокол избавляет от необходимости доверять приложению логин и пароль, а также позволяет выдавать ограниченный набор прав, а не все сразу.Результатом авторизации является access token – некий ключ (обычно просто набор символов), предъявление которого является пропуском к защищенным ресурсам. Обращение к ним в самом простом случае происходит по HTTPS с указанием в заголовках или в качестве одного из параметров полученного access token'а. Обычно, access token имеет ограниченный срок годности. Это может быть полезно, например, если он передается по открытым каналам. Чтобы не заставлять пользователя проходить авторизацию после истечения срока действия access token'а, во всех перечисленных выше вариантах, в дополнение к access token'у может возвращаться еще refresh token. По нему можно получить access token с помощью HTTP–запроса, аналогично авторизации по логину и паролю.

Приложение использует авторизацию – Implicit Flow для вызова методов API ВКонтакте непосредственно с устройства пользователя. Ключ доступа, полученный таким способом, не может быть использован для запросов с сервера. После успешного входа пользователю будет предложено авторизовать приложение, разрешив доступ к необходимым настройкам, запрошенным при помощи параметра scope.

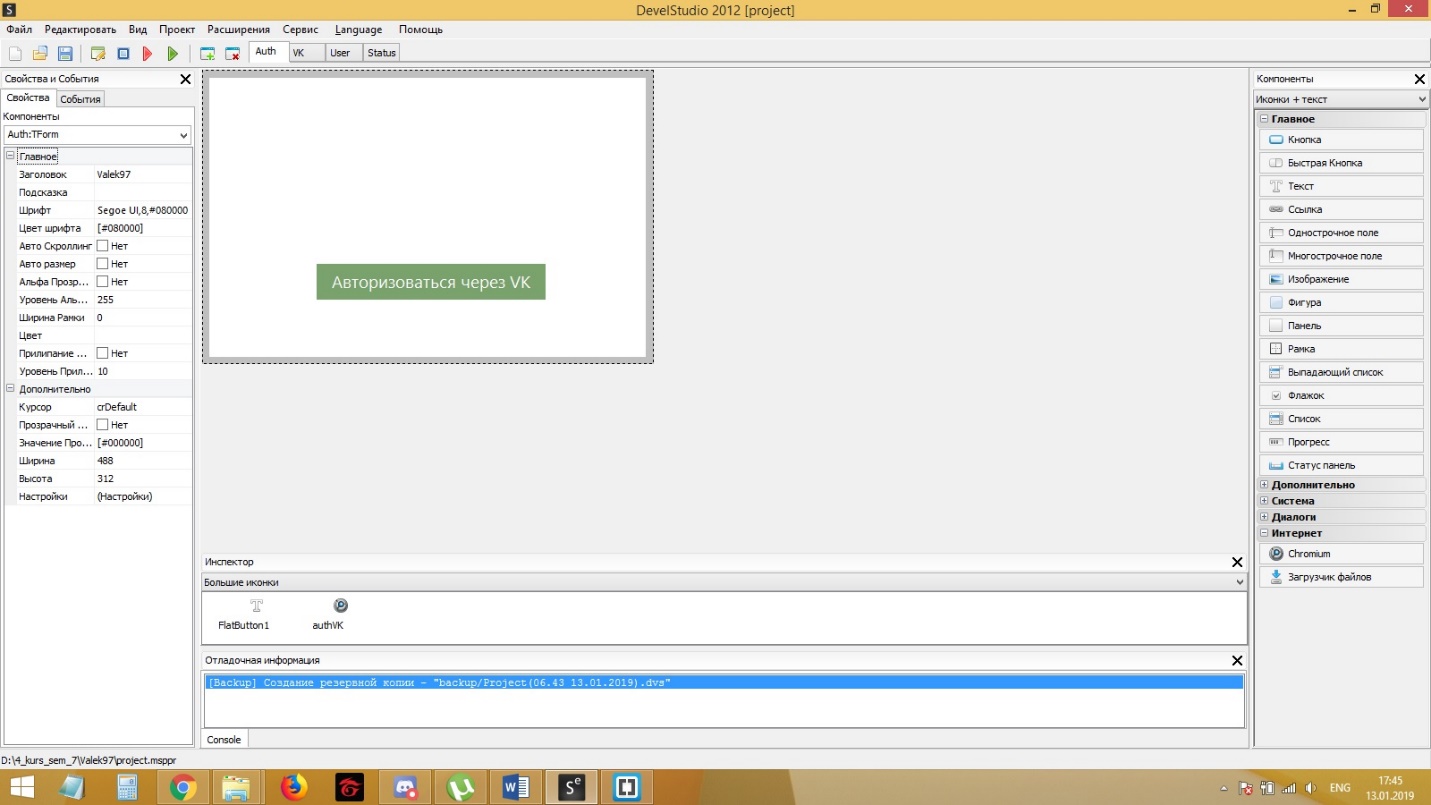
## 2.2 Алгоритм программы

Приложение для пользователей дает возможность пройти полностью безопасно авторизацию социальной сети ВКонтакте методом OAuth 2.0.

Рис 5. Алгоритм программы

## 2.3 Руководство разработчика

Приложение создается в IDE DevelStudio. Это окно авторизации. Здесь окно сделано широким, т.к при авторизации разворачивается небольшое окно браузера, которое может некорректно отобразиться при низком разрешении.

Рис 6. Окно авторизации

На форме размещаем кнопку и окно браузера, которое будет всплывать при запросе авторизации.

**Код внутри кнопки «авторизации»:**

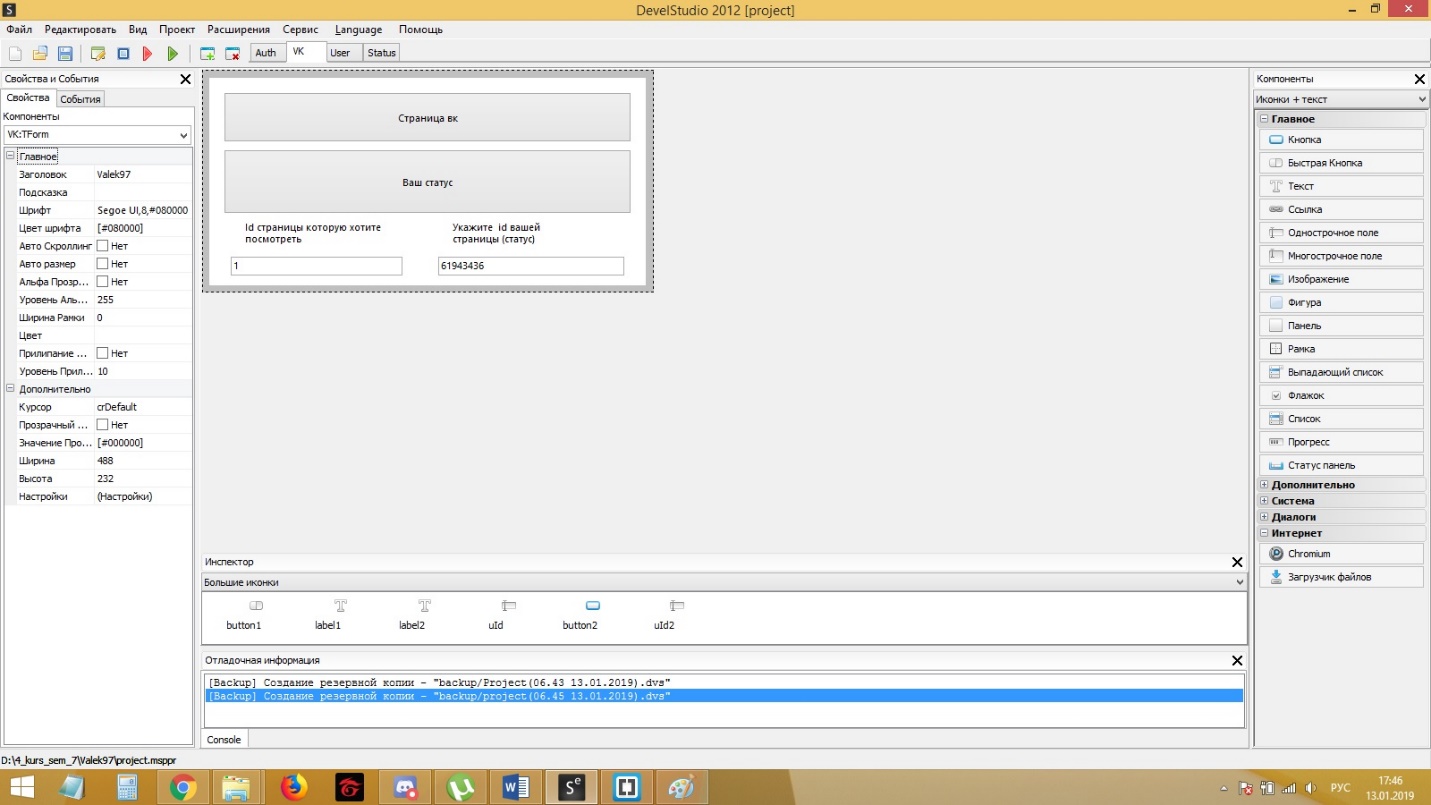
resize::resize\_object(c('authVk'), array("y" => 0));

c("authVK")–>url = 'https://oauth.vk.com/authorize?client\_id=6769897&display=mobile&scope=friends,status,offline&response\_type=token&v=5.44';

client\_id – клиент приложение ВК к которому мы делаем запрос

v=5.44 – версия авторизации, по которой мы делаем запрос к серверу авторизации ВК.

Здесь у нас идет обращение к файлу resize, который отвечает за авторизацию в ВК.

Рис 7. Основное меню

На форме размещаем несколько кнопок, label, edit полей.

**Код кнопки «Страница ВК»:**

Profile::User('');

**Код внутри кнопки «Ваш статус»:**

$get = Status::statusGet();

if(!isset($get)){

messageDlg("Не удалось получить статус!", mtConfirmation, MB\_OK);

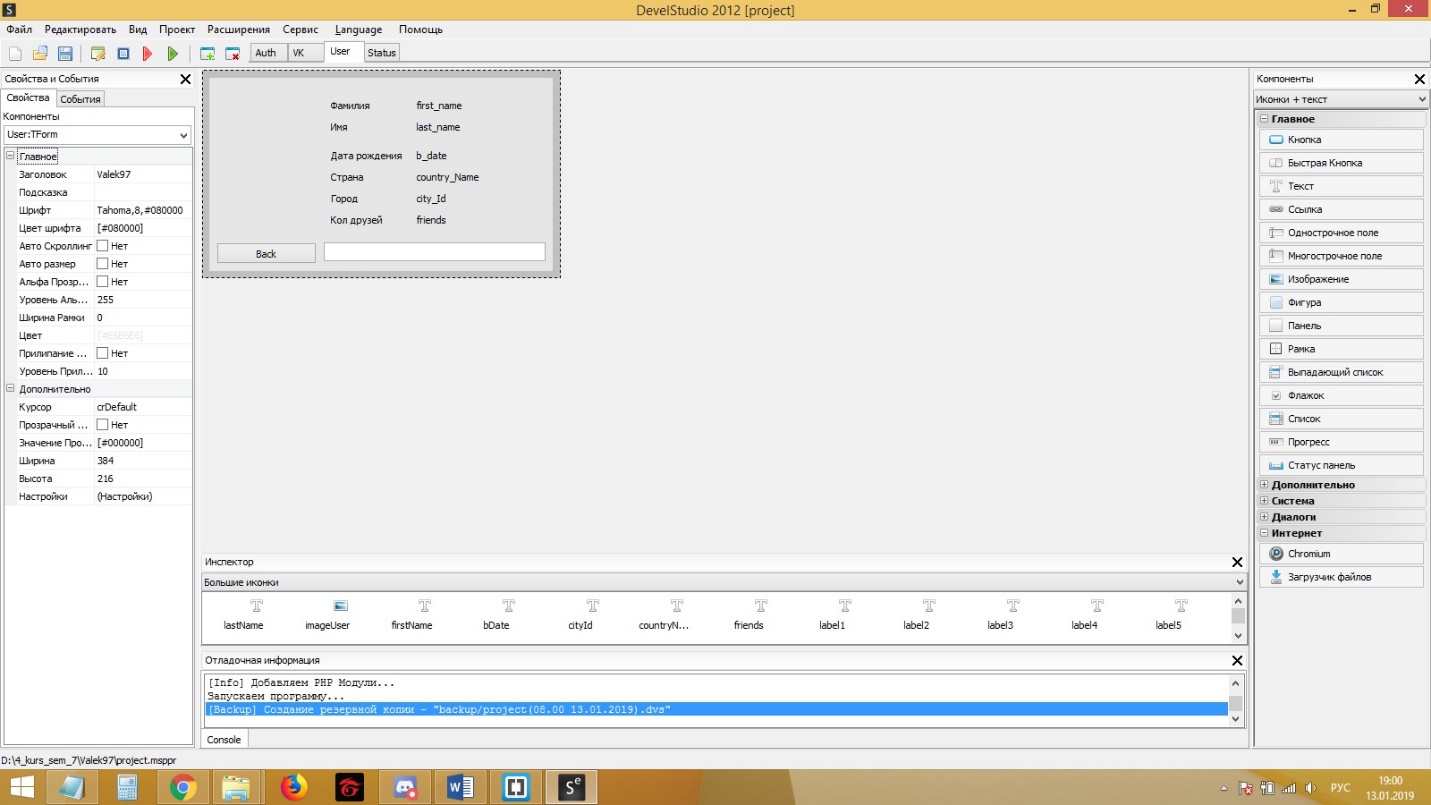
}else{

c("Status–>statusEdit")–>text = $get;

LoadForm(c('Status'), LD\_NONE);

}

Здесь уже обработка происходит в файле VK API. Идет обращение к классу User в котором строится основной запрос к ВК.

Рис 8. Окно User

В этом окне появляется информация о нужной странице ВКонтакте.  
Размещаем на форму: label, edit, кнопки Выводится такая информация как: Фамилия, Имя, Дата рождения, Страна, Город, Количество друзей. В поле Edite выводится статус, если он есть.

**Код кнопки «Back»:**

LoadForm(c('VK'), LD\_NONE);

**Код из файла VK API**, который обрабатывает данную форму:

function User($uid2){

$uid2= c("uId2")–>text;

//$uid2=61943436;

$user = VK::request('users.get', 'user\_id=' . $uid2.'&fields=photo\_100,bdate,city,country,followers\_count,online,online\_mobile,contacts,connections,status,last\_seen,counters,sex&v=5.92');

if(!$user){

messageDlg("Не удалось получить информацию о пользователе!", mtConfirmation, MB\_OK);

}else{

//pre($user);

//c('User–>imageUser')–>loadFromUrl ('https://vk.com/id61943436?z=photo61943436\_456241353%2Falbum61943436\_0%2Frev');

c('User–>firstName')–>caption = iconv('UTF–8', 'cp1251', $user['response']['0']['first\_name']);

c('User–>lastName')–>caption = iconv('UTF–8', 'cp1251', $user['response']['0']['last\_name']);

c('User–>bDate')–>caption = iconv('UTF–8', 'cp1251', $user['response']['0']['bdate']);

c('User–>status')–>text = iconv('UTF–8', 'cp1251', $user['response']['0']['status']);

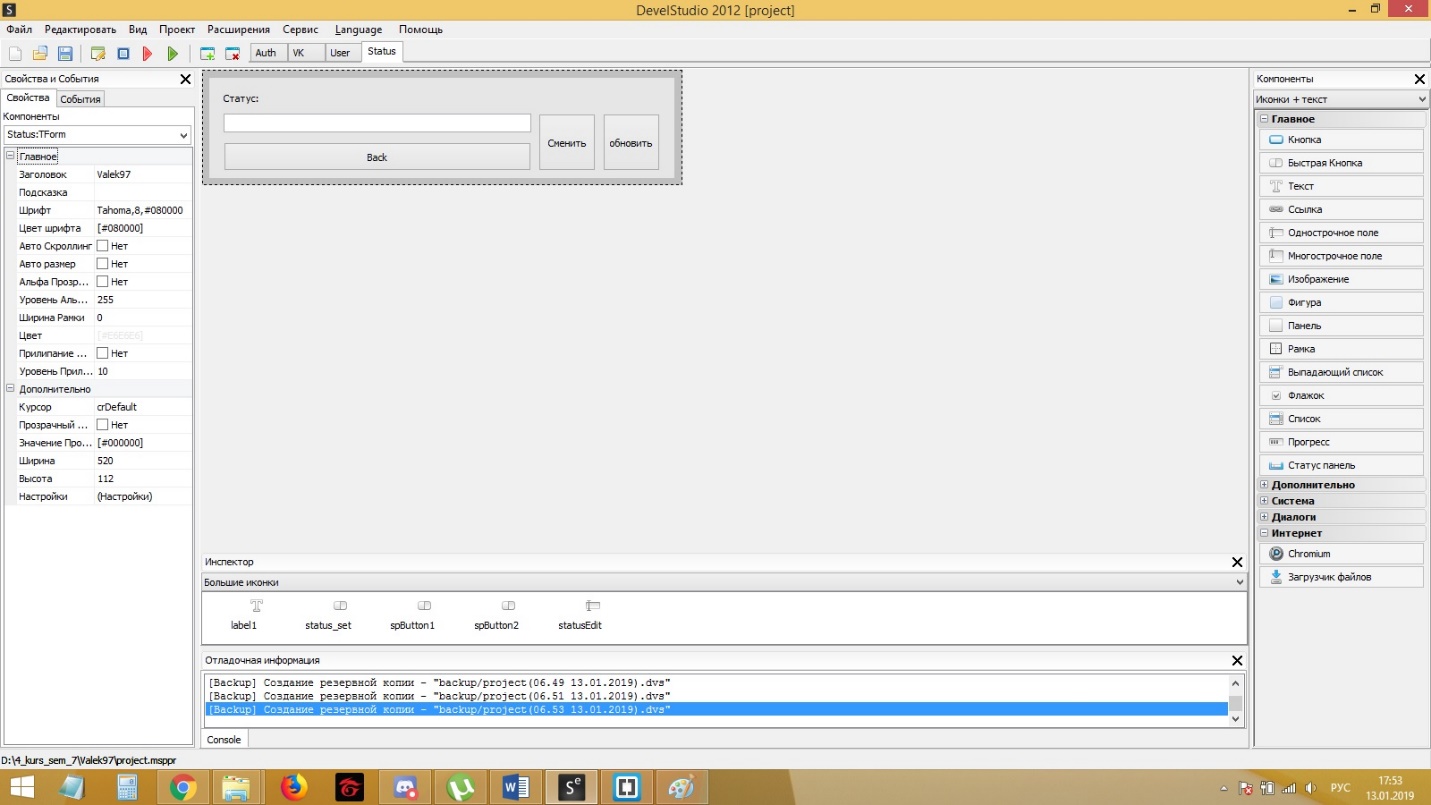
c('User–>cityId')–>text = iconv('UTF–8', 'cp1251', $user['response']['0']['city']['title']);

c('User–>countryName')–>text = iconv('UTF–8', 'cp1251', $user['response']['0']['country']['title']);

c('User–>friends')–>text = iconv('UTF–8', 'cp1251', $user['response']['0']['counters']['friends']);

LoadForm(c("User"), LD\_NONE);

}

Рис 9. Окно статуса

В окне статуса размещены: кнопки, поле edite, label. В поле edite выводим данные о статусе. Можем отредактировать статус как хочет пользователь. Для закрепления статуса жмем кнопку «Сменить». Если пользователю нужно обновить строку состояния статуса, то нужно нажать кнопку «Обновить»

**Код кнопки «Сменить»:**

if(c("statusEdit")–>text == null){

messageDlg("Поле статуса пустое!", mtConfirmation, MB\_OK);

}else{

$set = Status::statusSet(c("statusEdit")–>text);

if($set == 1){

messageDlg("Статус успешно изменен!", mtConfirmation, MB\_OK);

}else{

messageDlg("Не удалось сменить статус!", mtConfirmation, MB\_OK);

}

}

**Код кноки «Обновить»:**

$get = Status::statusGet();

if(!isset($get)){

messageDlg("Не удалось получить статус!", mtConfirmation, MB\_OK);

}else{

c("statusEdit")–>text = $get;

}

**Код кнопки «Back»:**

LoadForm(c('VK'), LD\_NONE);

**Код из файла VK API** обрабатывающего данную форму:  
function statusGet(){

$uid= c("VK–>uId")–>text;

//$uid=61943436;

$get = VK::request('status.get','user\_id='.$uid.'&v=5.92');

$get = iconv('UTF–8', 'cp1251', $get['response']['text']);

return $get;

}

function statusSet($set){

$set = iconv('cp1251', 'UTF–8', $set);

$set = str\_replace(' ', '+', $set);

$set = VK::request('status.set', 'text=' . $set.'&v=5.92');

return $set['response'];

}

# Заключение

В ходе работы над курсовой работой было прочитано и проанализированное много ресурсов посвященных языку программирования PHP, среде разработки – DevelStudio, написанию скриптов для взаимодействия с социальной сетью ВКонтакте, особенностям разных приложений, их ограничений и возможностей. Полностью с чистого листа шло освоение PHP и DevelStudio.

Решена проблема с авторизацией в социальной сети ВКонтакте. Изучены команды запросов к методам VK API. Разобраны разные методы авторизации, их плюсы и минусы. Разобраны базовые основы PHP. Не решена проблема с модулями DevelStudio. обратился за решением проблемы к разработчику и сообществу, пока решение не найдено и приходится вручную вводить данные в поля ID вашей страницы и которой хотите просмотреть. По итогу работы создано приложение, которое в дальнейшем после модернизации и более углубленного изучения сможет отвечать всем запросам, а именно: получение данных от сервера и обработка в приложении, авторизация разных пользователей социальной сети ВКонтакте, автоматическое получение ид пользователей, без ручного ввода.

# Список литературы

1. Мейер Б. Объектно-ориентированное программирование и программная инженерия / Б. Мейер. Интернет-Университет Информационных Технологий (ИНТУИТ), 2016. — 285 c. — 2227-8397
2. Зандстра М. PHP. Объекты, шаблоны и методики программирования/ М. Зандстра, Вильямс, 2016. 576 с.
3. Котеров Д. В. PHP 7/ Д. В Котеров, И. В. Симдянов, БХВ–Петербург, 2018. 1088 с.
4. Кузнецов А. С. Многоэтапный анализ архитектурной надежности и синтез отказоустойчивого программного обеспечения сложных систем/ А. С. Кузнецов, С.В.  Ченцов, Р. Ю. Царев, Инфра–М, 2018. 144 с.
5. Кузнецов М. Самоучитель PHP 7/ М Кузнецов, И. В. Симдянов, БХВ–Петербург, 2018. 448 с.
6. Локхарт Д. Современный PHP. Новые возможности и передовой опыт/ Д. Локхарт, ДМК Пресс, 2016. 304 с.
7. МакГрат М. PHP7 для начинающих с пошаговыми инструкциями/ М. МакГрат, Эксмо–Пресс, 2018. 256 с.
8. Мартин Р. Чистая архитектура. Искусство разработки программного обеспечения/ Р. Мартин, Питер, 2018. 352 с.
9. Назаров, С.В. Архитектура и проектирование программных систем/ С.В. Назаров, Инфра–М, 2016. 376 с.
10. Поляков Е. В PHP на примерах/ Е. В. Поляков, Наука и техника, 2017. 256 с.

# Приложения

# Приложение 1. Код файла Resizer

<?

class resize{

static $objects = array();

static $speed = 15;

static function resize\_object($obj, $params=false)

{

if(!$params) $params = array();

$pars = array('x', 'y', 'w', 'h', 'func', 'speed', 'time');

$s = 7;

for($i=0;$i<$s;++$i)

{

if(!isset($params[$pars[$i]])) $params[$pars[$i]] = false;

}

if($params['time']!==false) $params['time'] += microtime(1);

$sides = array($params['x'], $params['y'], $params['w'], $params['h']);

resize::$objects[$obj->self] = array($sides, $params['func'], $params['speed'], $params['time']);

}

static function set\_speed($i)

{

resize::$speed = $i;

}

static function get\_speed()

{

return resize::$speed;

}

static function tick()

{

$objs = &resize::$objects;

$size = count($objs);

if($size>0)

{

$m = microtime(1);

foreach($objs as $self=>$data)

{

list($sides, $func, $speed, $time) = $data;

$sname = array('Left', 'Top', 'Width', 'Height');

$s = 4;

if( $time !== false )if( $time > $m ) continue;

for($i=0;$i<$s;++$i)

{

$sidetmp[$i] = gui\_propGet($self, $sname[$i]);

$remp[$i] = resize::eval\_formule($sidetmp[$i], $sides[$i], $speed);

}

if($remp[0]===false && $remp[1]===false && $remp[2]===false && $remp[3]===false)

{

unset($objs[$self]);

for($i=0;$i<$s;++$i)

{

if($sides[$i]!==false) gui\_propSet($self, $sides[$i], $sides[$i]);

}

if($func!==false && is\_callable($func) ) $func($self);

}

else

{

for($i=0;$i<$s;++$i)

{

if($remp[$i]!==false) gui\_propSet($self,$sname[$i],$sidetmp[$i]+$remp[$i]);

}

}

}

}

}

static function syncfunc()

{

}

private static function eval\_formule($x, $y, $speed=false)

{

if($x===false || $y===false) return false;

if($speed===false) $speed = resize::$speed;

$remp = ($y-$x)/100\*resize::$speed;

if($remp==0) return false;

return $remp>0?ceil($remp):floor($remp);

}

} setTimer(5, 'resize::tick();');

?>

# Приложение 2. Код файла VK API

<?php

class VK{

function auth($login, $pass){

if(empty($login) and empty($pass)) return false;

$vk = curl\_init('http://m.vk.com/');

curl\_setopt\_array($vk, array(

CURLOPT\_USERAGENT => 'Mozilla/5.0 (Windows NT 6.3; rv:38.0) Gecko/20100101 Firefox/38.0',

CURLOPT\_FOLLOWLOCATION => true,

CURLOPT\_RETURNTRANSFER => true,

CURLOPT\_SSL\_VERIFYPEER => false,

CURLOPT\_COOKIEFILE => '')

);

preg\_match('/<form method="post" action="([\w\W]+)" novalidate>/U', curl\_exec($vk), $url);

if(empty($url[1])) {

curl\_close($vk);

return false;

}

curl\_setopt($vk, CURLOPT\_URL, $url[1]);

curl\_setopt($vk, CURLOPT\_CUSTOMREQUEST, 'POST');

curl\_setopt($vk, CURLOPT\_HTTPHEADER, array('Host' => 'login.vk.com'));

curl\_setopt($vk, CURLOPT\_POST, true);

curl\_setopt($vk, CURLOPT\_POSTFIELDS, http\_build\_query(array('email' => (string)$login, 'pass' => (string)$pass)));

curl\_exec($vk);

curl\_setopt($vk, CURLOPT\_URL, curl\_getinfo($vk, CURLINFO\_EFFECTIVE\_URL));

curl\_setopt($vk, CURLOPT\_CUSTOMREQUEST, 'GET');

curl\_setopt($vk, CURLOPT\_HTTPHEADER, array('Host' => 'm.vk.com'));

curl\_setopt($vk, CURLOPT\_POST, false);

curl\_exec($vk);

curl\_setopt($vk, CURLOPT\_URL, 'http://oauth.vk.com/authorize?client\_id=5077418&scope=2079999&redirect\_uri=https://oauth.vk.com/blank.html&display=wap&response\_type=token&v=5.44');

preg\_match('/<form method="post" action="([\w\W]+)">/U', curl\_exec($vk), $url);

if(empty($url[1])) {

curl\_close($vk);

return false;

}

curl\_setopt($vk, CURLOPT\_URL, $url[1]);

curl\_setopt($vk, CURLOPT\_CUSTOMREQUEST, 'POST');

curl\_setopt($vk, CURLOPT\_HTTPHEADER, array('Host' => 'login.vk.com'));

curl\_setopt($vk, CURLOPT\_POST, true);

curl\_exec($vk);

$url = parse\_url(curl\_getinfo($vk, CURLINFO\_EFFECTIVE\_URL));

parse\_str($url['fragment'], $url);

curl\_close($vk);

if(array\_key\_exists('access\_token', $url)) {

return $url;

}

return false;

}

function request($method, $pars){

global $token;

$curl = curl\_init();

curl\_setopt($curl, CURLOPT\_RETURNTRANSFER, true);

curl\_setopt($curl, CURLOPT\_SSL\_VERIFYPEER, false);

curl\_setopt($curl, CURLOPT\_URL, 'https://api.vk.com/method/' . $method . '?' . $pars . '&access\_token=' . $token);

$response = curl\_exec($curl);

curl\_close($curl);

$response = json\_decode($response, true);

return $response;

}

}

class Profile{

function User($uid2){

$uid2= c("uId2")->text;

//$uid2=61943436;

$user = VK::request('users.get', 'user\_id=' . $uid2.'&fields=photo\_100,bdate,city,country,followers\_count,online,online\_mobile,contacts,connections,status,last\_seen,counters,sex&v=5.92');

if(!$user){

messageDlg("Не удалось получить информацию о пользователе!", mtConfirmation, MB\_OK);

}else{

//pre($user);

//c('User->imageUser')->loadFromUrl ('https://vk.com/id61943436?z=photo61943436\_456241353%2Falbum61943436\_0%2Frev');

c('User->firstName')->caption = iconv('UTF-8', 'cp1251', $user['response']['0']['first\_name']);

c('User->lastName')->caption = iconv('UTF-8', 'cp1251', $user['response']['0']['last\_name']);

c('User->bDate')->caption = iconv('UTF-8', 'cp1251', $user['response']['0']['bdate']);

c('User->status')->text = iconv('UTF-8', 'cp1251', $user['response']['0']['status']);

c('User->cityId')->text = iconv('UTF-8', 'cp1251', $user['response']['0']['city']['title']);

c('User->countryName')->text = iconv('UTF-8', 'cp1251', $user['response']['0']['country']['title']);

c('User->friends')->text = iconv('UTF-8', 'cp1251', $user['response']['0']['counters']['friends']);

LoadForm(c("User"), LD\_NONE);

}

}

}

class Status{

function statusGet(){

$uid= c("VK->uId")->text;

//$uid=61943436;

$get = VK::request('status.get','user\_id='.$uid.'&v=5.92');

$get = iconv('UTF-8', 'cp1251', $get['response']['text']);

return $get;

}

function statusSet($set){

$set = iconv('cp1251', 'UTF-8', $set);

$set = str\_replace(' ', '+', $set);

$set = VK::request('status.set', 'text=' . $set.'&v=5.92');

return $set['response'];

}

}

?>

# Приложение 3. Код внутри кнопок

**Кнопка Авторизации**

resize::resize\_object(c('authVk'), array("y" => 0));

c("authVK")–>url = 'https://oauth.vk.com/authorize?client\_id=6769897&display=mobile&scope=friends,status,offline&response\_type=token&v=5.44';

**Кнопка Страница ВК**

Profile::User('');

**Кнопка Ваш статус**

$get = Status::statusGet();

if(!isset($get)){

messageDlg("Íå óäàëîñü ïîëó÷èòü ñòàòóñ!", mtConfirmation, MB\_OK);

}else{

c("Status–>statusEdit")–>text = $get;

LoadForm(c('Status'), LD\_NONE);

}

**Кнопка Сменить**

if(c("statusEdit")–>text == null){

messageDlg("Поле статуса пустое!", mtConfirmation, MB\_OK);

}else{

$set = Status::statusSet(c("statusEdit")–>text);

if($set == 1){

messageDlg("Статус успешно изменен!", mtConfirmation, MB\_OK);

}else{

messageDlg("Не удалось сменить статус!", mtConfirmation, MB\_OK);

}

}

**Кнопка Обновить**

$get = Status::statusGet();

if(!isset($get)){

messageDlg("Не удалось получить статус!", mtConfirmation, MB\_OK);

}else{

c("statusEdit")–>text = $get;

}

**Кнопка Back**

LoadForm(c('VK'), LD\_NONE);