МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО

«Пермский государственный национальный исследовательский университет»

**Отчёт**

по лабораторной работе №1 «Создание клиентского приложения по выполнению CRUD операций на популярном информационном ресурсе»

по дисциплине «Технологии разработки распределенных приложений»

Работу выполнил Проверил

студент гр. ФИТ-8-2017 доцент кафедры

Габов Егор математического обеспечения

Владимирович вычислительных систем,

17.11.2020 к.ф.-м.н., доц.

Деменев Алексей

Геннадьевич

Пермь, 2020 г

**Содержание**

[**Введение** 3](#_Toc59445287)

[**Основная часть** 4](#_Toc59445288)

[**Постановка задачи** 4](#_Toc59445289)

[**Выбор предметной области и инструментов** 4](#_Toc59445290)

[**Сценарии использования** 4](#_Toc59445291)

[**Описание приложения** 7](#_Toc59445292)

[**Заключение** 11](#_Toc59445293)

[**Литература** 12](#_Toc59445294)

[**Код программы** 13](#_Toc59445295)

# **Введение**

***Цель:*** Изучение возможностей использования API современных популярных информационных ресурсов внешними приложениями.

***Формируемая компетенция:*** способность применять на практике теоретические основы и общие принципы разработки распределенных систем.

***Организация выполнения работы:*** Каждый студент выполняет индивидуальное задание.

***Требования к выполнению работы:***

1. Разработанное приложение должно предоставлять возможности создания, чтения, обновления и удаления некоторых объектов выбранного информационного ресурса (выполнения CRUD‑операций).
2. CRUD‑операции должны выполняться при помощи API выбранного ресурса, причём по крайней мере один тип операции должен требовать аутентификации на нём.
3. Должна быть возможность выполнения по крайней мере двух CRUD‑операций для одного и того же объекта выбранного информационного ресурса.

# **Основная часть**

## **Постановка задачи**

Спроектировать, реализовать, протестировать и оценить приложение в соответствии с таблицей №1.

***Таблица 1. Распределение баллов за выполнение работы (***Максимальное количество баллов — 10)***:***

| **Требование к заданию** | **Максимальное количество баллов** |
| --- | --- |
| Приложение позволяет выполнять CRUD‑операции над объектами выбранного ресурса | 1 (за каждый тип операции) |
| Приложение позволяет выполнять по крайней мере две CRUD‑операции для одного и того же объекта выбранного ресурса | 1 |
| Приложение выполняет по крайней мере одну из CRUD‑операций после успешной аутентификации | 2 |
| Приложение не требует повторной аутентификации при перезапуске программы | 1 |
| Параметры успешной аутентификации сохраняются при перезапуске программы в зашифрованном виде | 2 |

## **Выбор предметной области и инструментов**

В качестве ресурса была выбрана социальная сеть Вконтакте [1]. В качестве объекта, над которым будут выполняться CRUD-операции была выбрана «Стена» – это специальное пространство, где человек может размещать любую информацию.

Код программы написан на языке python версии 3.8 [2][3]. В качестве IDE использовался PyCharm Community Edition [4]. PyCharm Community Edition является бесплатной версией, обладающей усеченным набором возможностей [5].

Для работы с API Вконтакте [6] использовалась open-source библиотека VK [7].

Для шифрования данных для авторизации использовалась библиотека PyCryptodome [8].

Для того чтобы удостоверяться в правильности работы приложения, просматривалась стена «Вконтакте» с помощью браузера Google Chrome [9].

## **Сценарии использования**

Приложение может использоваться в следующих сценариях.

1. **Авторизация.**

**Шаги выполнения:**

1. Пользователь запускает приложение
2. Пользователь авторизуется в социальной сети Вконтакте, за счет ввода данных для авторизации (логин, пароль) и имени аккаунта.

**Ожидаемый результат:** Успешная авторизация пользователя в социальной сети Вконтакте. Получение пользователем вариантов дальнейшей работы.

**Способ тестирования:**

При успешной авторизации пользователю будет выведен список вариантов дальнейшей работы (рис 4).

Также факт авторизации через приложение будет отражён в социальной сети Вконтакте.

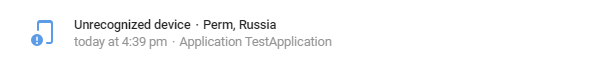


Рис. 1 Факт авторизации с помощью приложения

1. **Создание записи на стене**

**Шаги выполнения:**

1. Пользователь авторизуется в приложении
2. Пользователь выбирает пункт меню «Создать новую запись»
3. Вводит текст, который будет содержаться в создаваемой записи

**Ожидаемый результат:** Запись успешно создана. На экран выводится идентификатор новой записи (рис. 6).

**Способ тестирования:**

Факт успешного создания записи отражается на стене в социальной сети «Вконтакте» (рис. 8).

1. **Удаление записи**

**Шаги выполнения:**

1. Пользователь авторизуется в приложении
2. Пользователь выбирает пункт меню «Удалить запись»
3. Пользователь вводит идентификатор записи, которую хочет удалить

**Ожидаемый результат:** Успешное удаление записи (рис. 9).

**Способ тестирования:**

Факт успешного удаления записи отражается на стене в социальной сети «Вконтакте» (рис. 10)

1. **Просмотр последних записей**

**Шаги выполнения:**

1. Пользователь авторизуется в приложении
2. Пользователь выбирает пункт меню «Просмотреть последние n записей на странице»
3. Пользователь вводит количество записей, которое он хочет просмотреть.

**Ожидаемый результат:** на экран выводится список последних записей со страницы пользователя (рис. 5).

**Способ тестирования:**

Проверить правильность выведенных записей можно, просмотрев записи на стене в социальной сети «Вконтакте» (рис. 8).

1. **Редактирование записи**

**Шаги выполнения:**

1. Пользователь авторизуется в приложении
2. Пользователь выбирает пункт меню «Изменить запись»
3. Пользователь вводит идентификатор записи, которую хочет изменить
4. Пользователь вводит текст, на который будет изменён текст выбранной записи.

**Ожидаемый результат:** Успешное изменение записи (рис. 7)

**Способ тестирования:**

Факт изменения записи можно проверить, просмотрев запись на стене в социальной сети «Вконтакте» (рис. 8)

1. **Повторный запуск программы, без повторной авторизации**

**Шаги выполнения:**

1. Пользователь запускает приложение
2. Приложение просматривает файл AuthData.txt и использует данные для аутентификации в социальной сети «Вконтакте»

**Ожидаемый результат:** Успешная авторизация пользователя в социальной сети Вконтакте. Получение пользователем вариантов дальнейшей работы.

**Способ тестирования:**

При успешной авторизации пользователю будет выведен список вариантов дальнейшей работы (рис 4).

Также факт авторизации через приложение будет отражён в социальной сети Вконтакте (рис 1).

## **Описание приложения**

Приложение позволяет авторизированному пользователю выполнять следующие операции:

* Создание записи на стене;
* Просмотр списка записей со стены;
* Редактирование записи со стены;
* Удаление записи со стены.

При запуске приложение просматривает файл authData.txt, если такого файла нет, то приложение запрашивает у пользователя данные для аутентификации. При успешной авторизации данные сохраняются в файле authData. Данный файл содержит информацию лишь об одном аккаунте. Данные для авторизации шифруются алгоритмом симметричного шифрования DES.

Содержимое файла authData.txt:

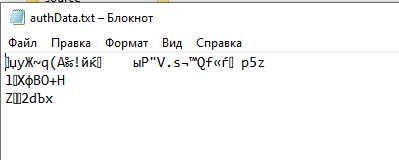


Рис. 1 Файл AuthData.txt

Если файл authData отсутствует, приложение предлагает авторизоваться.

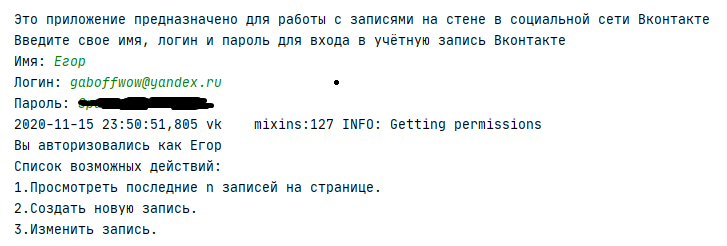


Рис. 2 Процесс авторизации

При успешной авторизации пользователю представляются следующие варианты дальнейшей работы:

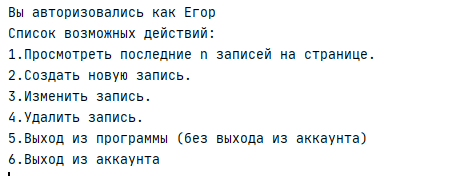


Рис. 3 Список функций

* 1. Просмотр последних записей.

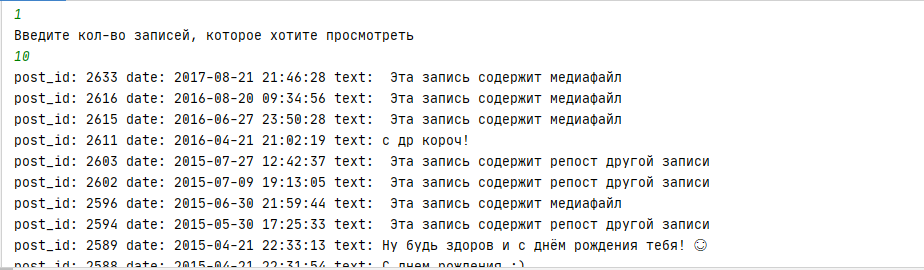


Рис. 4 Результат функции getLastPosts()

* 1. Создание новой записи.

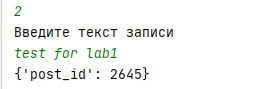


Рис. 5 Результат функции createPost()

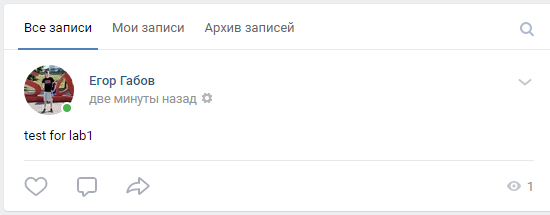


Рис. 6 Результат создания записи

* 1. Изменение записи.

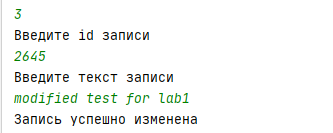


Рис. 7 Результат функции editPost()

Все записи можно просмотреть на «Стене» в социальной сети Вконтакте.

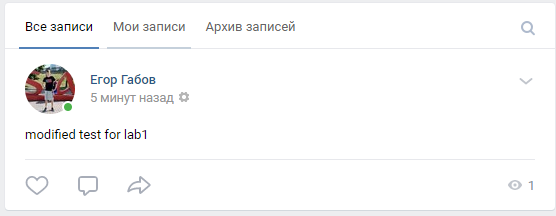


Рис. 8 Результат изменения записи

* 1. Удаление записи.

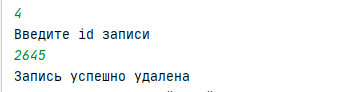


Рис. 9 Результат функции deletePost()

Запись удалена.

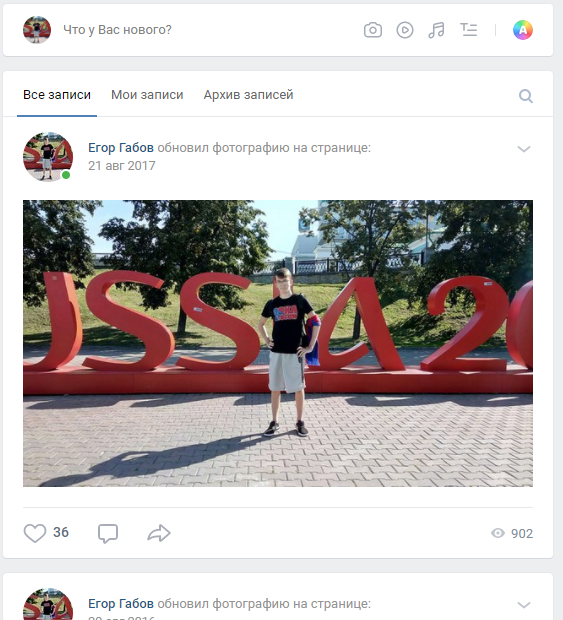


Рис. 10 Результат удаления записи на стене

* 1. Выход из программы.

При выборе данной опции пользователь выходит из приложения, а его данные сохраняются в authData.txt, при следующем запуске приложения он автоматически авторизуется.

* 1. Выход из аккаунта.

При выборе данной опции пользователь выходит из приложения, а файл с его данными удаляется.

# **Заключение**

В ходе работы достигнута цель и выполнены следующие требования:

Приложение позволяет выполнять CRUD операции над объектами выбранного ресурса. Приложение позволяет выполнять 4 операции (создание, чтение, редактирование, удаление) над записями на стене Вконтакте. (4 балла)

Приложение позволяет выполнять по крайней мере две CRUD операции для одного и того же объекта выбранного ресурса. К одной и той же записи в примерах применялись все 4 операции. (1 балл)

Приложение выполняет по крайней мере одну из CRUD операций после успешной аутентификации. Операции можно выполнять лишь после авторизации. (2 балла)

Приложение не требует повторной аутентификации при перезапуске программы. Данные для аутентификации сохраняются в отдельном файле. (1 балл)

Параметры успешной аутентификации сохраняются при перезапуске программы в зашифрованном виде. Данные для аутентификации шифруются DES и сохраняются в отдельном файле. (2 балла)

Итого: 10 баллов

# **Литература**

* + 1. Социальная сеть Вконтакте, URL: <https://vk.com> (Дата обращения 10.11.2020)
    2. Лицензия Python 3.8, [https://docs.python.org/3.8/license.html](https://docs.python.org/3.7/license.html) (Дата обращения 10.11.2020)
    3. Python 3.8.0. Ссылка наскачивание URL: <https://www.python.org/downloads/release/python-380/> (дата обращения: 10.11.2020)
    4. PyCharm Community Edition. Распространяется под лицензией Apache 2, URL: <https://www.jetbrains.com/pycharm/features/editions_comparison_matrix.html> (Дата обращения 10.11.2020)
    5. PyCharm Community Edition. Ссылка на скачивание. URL: <https://www.jetbrains.com/pycharm/download/#section=windows> (Дата обращения 10.11.2020)
    6. Документация VK API, URL: <https://vk.com/dev/manuals> (Дата обращения 10.11.2020)
    7. VK. Open-source библиотека, разработанная Дмитрием Ворониным, URL: <https://github.com/voronind/vk> (Дата обращения 10.11.2020)
    8. Библиотека PyCryptodome, URL: <https://pycryptodome.readthedocs.io/en/latest/src/license.html> (Дата обращения 10.11.2020)
    9. Google Chrome, браузер распространяется на условиях собственнической лицензии, URL: <https://www.google.ru/intl/ru/chrome/terms/> (Дата обращения 10.11.2020)

# **Код программы**

|  |
| --- |
| import os import vk from Cryptodome.Cipher import DES import datetime  key = b'abcdefgh' des = DES.new(key, DES.MODE\_ECB) filePath = **"D:**\\**authData.txt"** value = 1   def pad(text):  while len(text) % 8 != 0:  text += b' '  return text   def createPost():  message = input(**"Введите текст записи**\n**"**)  response = vkApi.wall.post(message=message)  print(response)  return response   def deletePost():  postId = input(**"Введите id записи**\n**"**)  response = vkApi.wall.delete(post\_id=postId)  if response == 1:  print(**'Запись успешно удалена'**)  else:  print(**'Произошла ошибка при удалении сообщения'**)  return response   def getLastPosts():  posts = input(**"Введите кол-во записей, которое хотите просмотреть**\n**"**)  response = vkApi.wall.get(count=posts)  items = response[**'items'**]  text1 =**''** text2 =**''** for item in items:  date = datetime.datetime.fromtimestamp(item[**'date'**]).strftime(**'%Y-%m-%d %H:%M:%S'**)  try:  if item[**'copy\_history'**]:  text1 = **'Эта запись содержит репост другой записи'** except Exception:  text1 = **''** try:  if item[**'attachments'**]:  text2 = **'Эта запись содержит медиафайл'** except Exception:  text2 = **''** print(**'post\_id: '**+str(item[**'id'**])+**' date: '**+date+**' text: '**+item[**'text'**]+**' '** + text1 + text2)  return response   def editPost():  postId = input(**"Введите id записи**\n**"**)  message = input(**"Введите текст записи**\n**"**)  response = vkApi.wall.edit(post\_id=postId, message=message)  if response == 1:  print(**'Запись успешно изменена'**)  else:  print(**'Произошла ошибка при изменении записи'**)  return response   def exitProgram():  print(**'Завершение работы'**)   def exitAcc():  os.remove(filePath)  print(**'Завершение работы'**)   print(**'Это приложение предназначено для работы с записями на стене в социальной сети Вконтакте'**) if(os.path.isfile(filePath)):  with open(filePath, **'rb'**) as f:  encryptedData = f.read()  decryptedData = des.decrypt(encryptedData)  data = decryptedData.decode(**'utf-8'**).split(**'!'**)  name = data[0]  login = data[1]  password = data[2].replace(**' '**, **''**) else:  print(**'Введите свое имя, логин и пароль для входа в учётную запись Вконтакте'**)  name = input(**"Имя: "**)  login = input(**"Логин: "**)  password = input(**"Пароль: "**)  data = name+**'!'**+login+**'!'**+password  encryptedData = des.encrypt(pad(bytes(data, **'utf-8'**)))  with open(filePath, **'wb'**) as f:  f.write(encryptedData) try:  session = vk.AuthSession(7664147, login, password, scope=**'wall'**)  vkApi = vk.API(session, v=**'5.35'**, lang=**'ru'**)  print(**"Вы авторизовались как "** + name) except Exception:  print(**'Произошла ошибка при авторизации, проверьте введённые данные'**)  os.remove(filePath)  value = -1  while value > 0 and value < 5:  value = int(input(**"Список возможных действий:**\n**1.Просмотреть последние n записей на странице.**\n**2.Создать новую запись.**\n**3.Изменить запись.**\n**4.Удалить запись.**\n**5.Выход из программы (без выхода из аккаунта)**\n**6.Выход из аккаунта**\n**"**))  result = {  1: getLastPosts,  2: createPost,  3: editPost,  4: deletePost,  5: exitProgram,  6: exitAcc,  }.get(value, exitProgram)() |