|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ | | | | | | | | |
|  | |  |  | | |  |  | |
| Пермский государственный национальный  исследовательский университет | | | | | | | | |
|  |  | |  | | |  | |  |
|  | ОТЧЕТ  по лабораторной работе №1  «Создание клиентского приложения по выполнению CRUD операций на популярном информационном ресурсе»  по дисциплине  «Технологии разработки распределенных приложений» | | | | | | |  |
|  |  | | |  |  | | |  |
|  | Работу выполнил студент гр. КМБ-16  Тарарков Андрей Викторович  16.10.2020 | | |  | Проверил  доцент кафедры прикладной математики и информатики  Деменев Алексей Геннадьевич | | |  |
|  |  | | |  |  | | |  |
| Пермь, 2020 г. | | | | | | | | |

**Введение**

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

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

**Требования к выполнению работы**

| **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\***

**\*Проектирование\***

**Ход выполнения работы**

В качестве информационного ресурса для работы с его API был выбран Google Drive[2]

Для работы с API диска были использованы библиотеки google-api-python-client, google-auth-httplib2, google-auth-oauthlib

Приложение написано на языке Python 3.8[3]. Среда разработки приложения: PyCharm[1]

Исходный код приложения сохранён в GitHub репозиторий <https://github.com/Dasem/trrp1>.

**Сценарии использования приложения**

1. Авторизация
   1. Пользователь запускает приложение
   2. Пользователю предлагается авторизоваться с помощью OAuth Google аккаунта
   3. Пользователю выводится список возможных действий с приложением (стандартное меню)
2. Скачивание файла. Предусловия: приложение запущено в стартовом меню
   1. Пользователь вводит пункт меню скачивания файла
   2. Вводит необходимые данные, запрашиваемые приложением для инициации скачивания
   3. Пользователю загружается файл с уведомлением об окончании загрузки по завершении
3. Загрузка файла. Предусловия: приложение запущено в стартовом меню
   1. Пользователь вводит пункт меню загрузки файла
   2. Вводит необходимые данные, запрашиваемые приложением для инициации загрузки
   3. Приложение загружает файл с уведомлением об окончании загрузки по завершении
4. Переименование файла. Предусловия: приложение запущено в стартовом меню
   1. Пользователь вводит пункт меню изменения имени файла
   2. Вводит необходимые данные, запрашиваемые приложением для инициации переименования
   3. Приложение изменяет файла на сервере с уведомлением пользователя по завершении
5. Удаление файла. Предусловия: приложение запущено в стартовом меню
   1. Пользователь вводит пункт меню удаления файла
   2. Вводит необходимые данные, запрашиваемые приложением для инициации удаления
   3. Приложение удаляет файл с уведомлением об окончании удаления файла по завершении
6. Отсутствие повторной аутентификации при перезапуске программы
   1. Пользователь запускает приложение
   2. Пользователю предлагается авторизоваться с помощью OAuth Google аккаунта
   3. Пользователю выводится список возможных действий с приложением (стандартное меню)
   4. Пользователь закрывает приложение
   5. Пользователь повторно запускает приложение
   6. Выводится меню без предложения повторной авторизации

**Описание приложения**

С помощью приложения можно просматривать, создавать, редактировать и удалять файлы на личном Google Drive диске. Для работы в приложении необходимо аутентифицироваться с помощью аккаунта Google. При каждом новом запуске приложения производится попытка аутентификации под тем пользователем, который использовался в приложении в последний раз. Приложение не имеет графического интерфейса и запускается в консольном режиме. Если во время работы приложения возникают непредвиденные ошибки, то информация о них выводится в окно приожения, но это не мешает продолжению работы с приложением. Токены пользователей хранятся в файле , шифрование токенов производится посредством библиотеки Fernet[4].

Так же в целях проверки работоспособности приложения был разработан набор тестов, который можно выполнять также, как и само приложение.

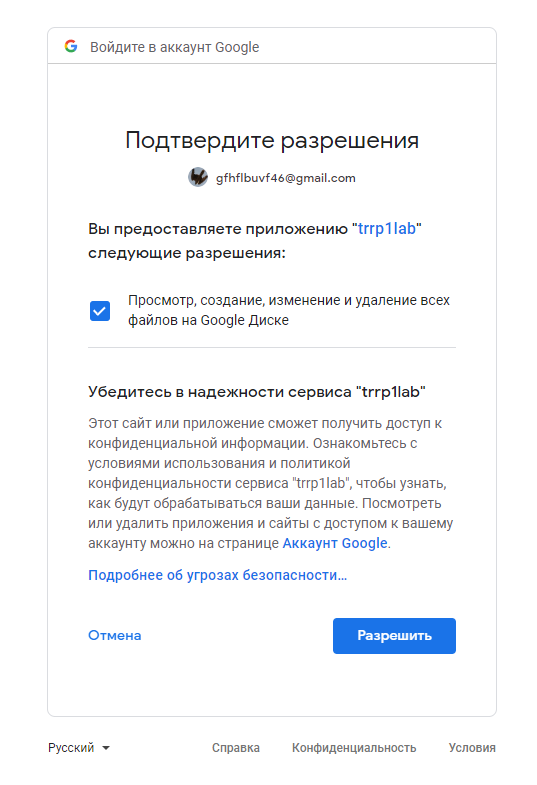
Первый запуск приложения инициирует OAuth авторизацию с помощью браузера, для создания уникального токена авторизированного пользователя: 

Рисунок 1

При успешной авторизации появляется меню, в котором доступны действия по работе с диском:

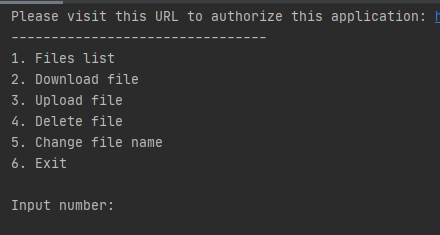


Рисунок 2

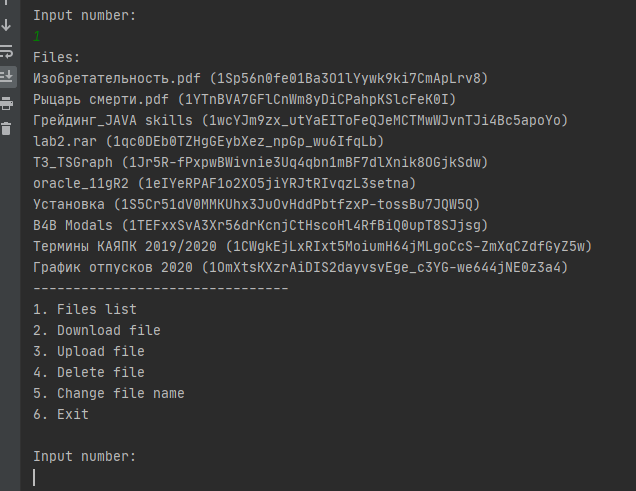
Для просмотра файлов на диске необходимо ввести «1»: 

Рисунок 3

Проверка работы функций в соответствии с требованиями к приложению:

1. Read (Скачивание файлов), производится путём копирования из списка файлов требуемого ID, а так же последующего ввода имени файла для сохранения

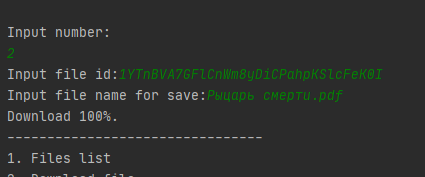


Рисунок 4

Проверяем наличие файла и целостность:

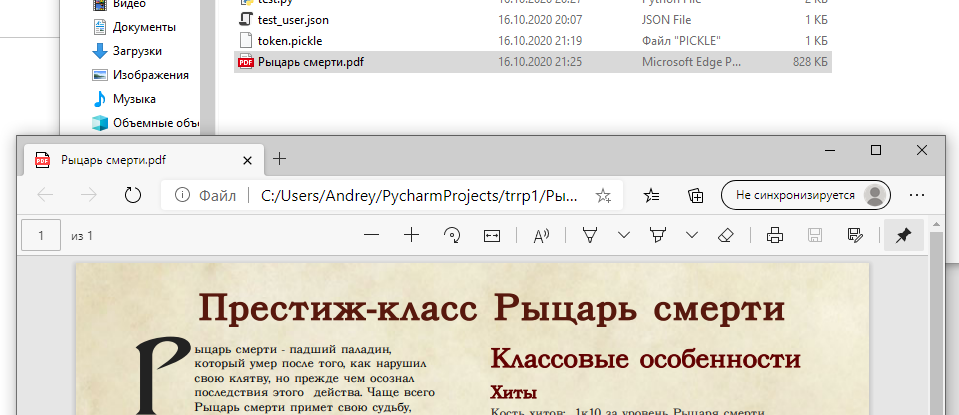


Рисунок 5

1. Create (Загрузка файла на диск), загружается файл из текущей директории с указанным именем

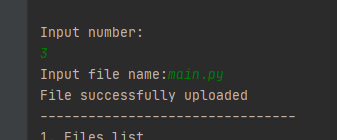


Рисунок 6

Проверим, выведя список файлов:

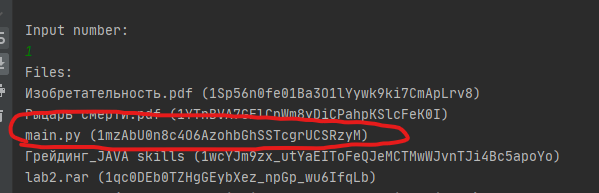


Рисунок 7

1. Update (Изменение имени файла) вводится ID файла и новое имя файла

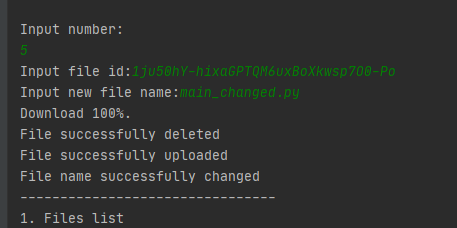


Рисунок 8

Проверяем:



Рисунок 9

1. Delete (Удаление файла) вводится ID удаляемого файла

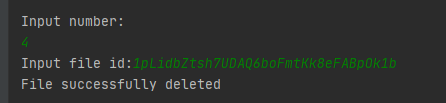


Рисунок 10

Проверяем:



Рисунок 11

При необходимости возможно провести все те же проверки путём автоматического тестирования, с помощью запуска скрипта test.py:

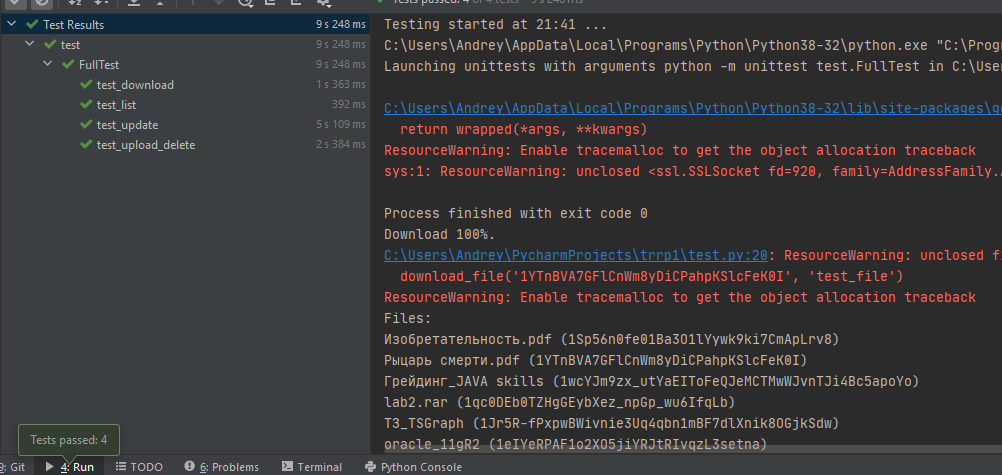


Рисунок 12

При повторном входе в приложение не требуется повторная аутентификация:

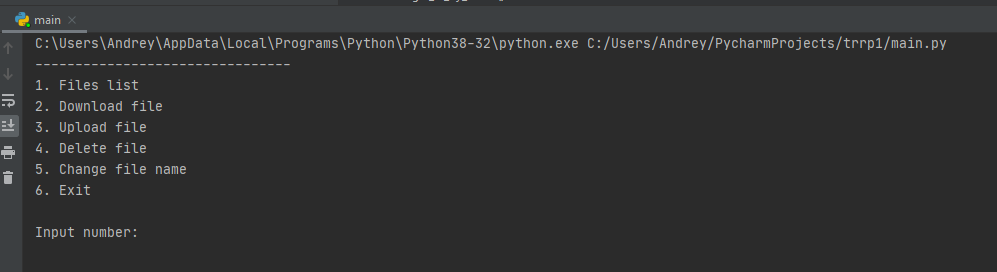
****

Рисунок 13

Токен хранится в зашифрованном виде:

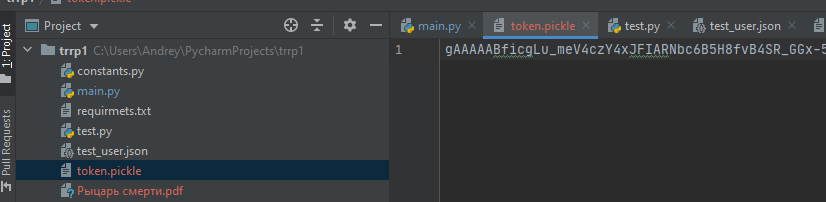
****

Рисунок 14

**Заключение**

В работы достигнута цель и выполнены следующие требования:

Приложение позволяет выполнять CRUD операции выбранного ресурса - выполнено корректность операций доказана путём проверки изменения состояния диска. Оценка 4б.

Приложение позволяет выполнять по крайней мере 2 CRUD операции для одного и того же объекта выбранного информационного ресурса выполняется. Создание (create), переименование (update) и удаление (delete) происходило с одним и тем же файлом в примере. Оценка 1б.

В приложении работает аутентификация с помощью аккаунта Google. По крайней мере одна операция выполняется после успешной авторизации. Оценка 2б.

Приложение не требует повторной аутентификации при перезапуске программы. Оценка 1б

Параметры успешной аутентификации сохраняются при перезапуске программы в зашифрованном виде - выполняется. Оценка 2б.

Требования выполнены полностью, корректность операций доказана приёмочными тестами.

**Программное обеспечение, использовавшееся при выполнении работы, а также лицензии**

1. PyCharm, ссылка для получения бесплатной лицензии студентами и преподавателями: <https://www.jetbrains.com/shop/eform/students>, лицензия: <https://www.jetbrains.com/company/privacy.html>
2. Google Drive API, апи: <https://developers.google.com/drive?hl=ru>, лицензия: <https://developers.google.com/terms>
3. Python 3.8, скачать: <https://www.python.org/downloads/>, лицензия: <https://docs.python.org/3/license.html>
4. Fernet (python библиотека шифрования) <https://github.com/oz123/python-fernet/blob/master/LICENSE>