

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра САПР

ОТЧЕТ
по лабораторной работе №1
по дисциплине «ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ KOTLIN»
ТЕМА: Консольная программа для поиска в Википедии.

Студент гр. 0302

Кузнецов К.Е.

Студент гр. 0302

Головатюк К.А.

Преподаватель

Кулагин М.В

Санкт-Петербург

2022

Цель работы

Написать программу, которая с консоли считывает поисковый запрос пользователя, и выводит результат поиска по Википедии и открывает окно браузера с выбранной страницей.

Спецификация программы

WikiSearcher
<ul style="list-style-type: none">□ URL_TEMPLATE_SEARCH□ URL_TEMPLATE_OPEN□ search_page_request□ search_page_json
<ul style="list-style-type: none">● request(req)● show_result()● open_tab(page_num)

Описание интерфейса пользователя программы

Пользователю печатается информация о том, как выйти из программы и то, что нужно ввести слово для поиска. После ввода слова, пользователю печатается список найденных страниц. Пользователь вводит номер интересующей его страницы, после чего открывается окно браузера с ней.

Текст программы

```
import requests
import webbrowser
import json

class WikiSearcher:
    def __init__(self):
        self.__URL_TEMPLATE_SEARCH =
'https://ru.wikipedia.org/w/api.php?action=query&list=search&utf8=&format=json&srsearch='
        self.__URL_TEMPLATE_OPEN = 'https://ru.wikipedia.org/w/index.php?curid='
        self.__search_page_request = None
        self.__search_page_json = None
```

```

def request(self, req):
    self.__search_page_request = requests.get(self.__URL_TEMPLATE_SEARCH + req)
    if self.__search_page_request.status_code != 200:
        return False

    self.__search_page_json = json.loads(self.__search_page_request.text)
    return True

def show_result(self):
    if self.__search_page_json is None:
        return False

    if len(self.__search_page_json["query"]["search"]) == 0:
        return False
    else:
        counter = 1
        for i in self.__search_page_json["query"]["search"]:
            print(f'{counter}. {i["title"]}')
            counter += 1
        return True

def open_tab(self, page_num):
    if page_num.isdigit():
        if len(self.__search_page_json["query"]["search"]) >= int(page_num) > 0:
            page_id = str(self.__search_page_json["query"]["search"][int(page_num) -
1]["pageid"])
            webbrowser.open(self.__URL_TEMPLATE_OPEN + page_id, new=2)
        else:
            return False
    else:
        return False
    return True

```

```
def loop():
    searcher = WikiSearcher()

    while True:
        print("Введите 'exit' для выхода.\n"
              "Введите слово для поиска:")

        name = str(input())
        if name == "exit":
            return

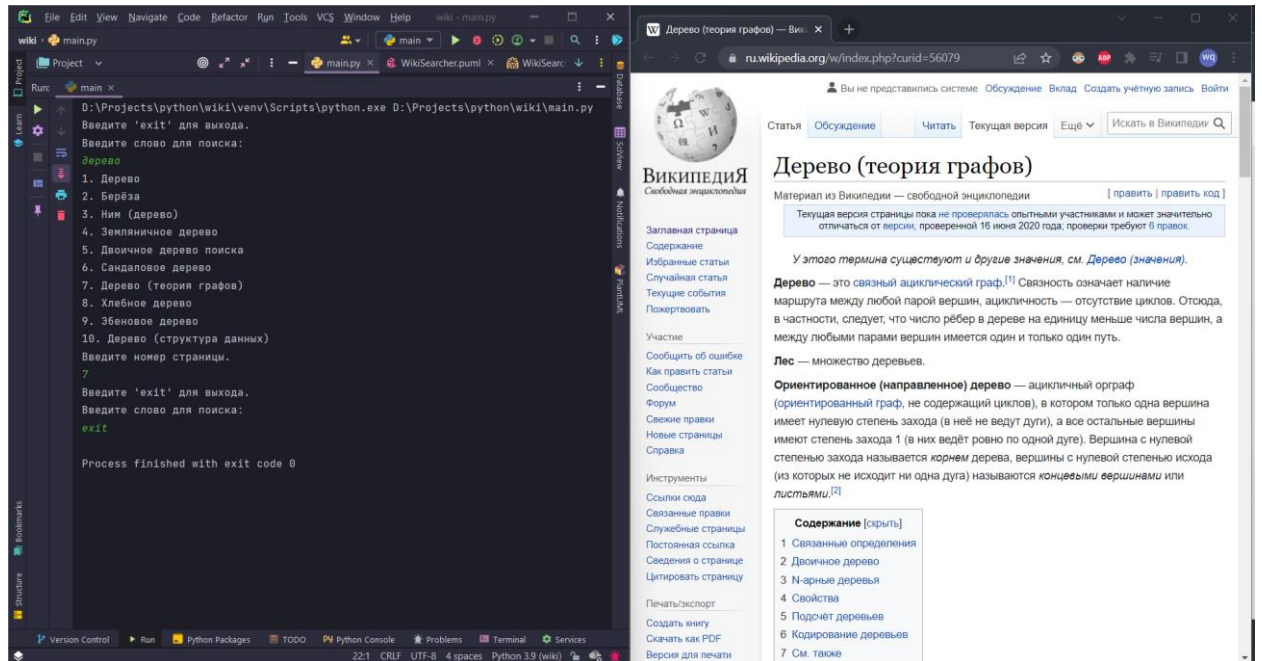
        if not searcher.request(name):
            print("Bad request!!!")
            continue

        if not searcher.show_result():
            print("bad request!!!")
            continue

        print("Введите номер страницы.")
        page = str(input())
        while not searcher.open_tab(page):
            print("Неправильный номер!!!\n"
                  "Введите еще раз.")
            page = str(input())

if __name__ == '__main__':
    loop()
```

Пример работы программы



Выводы

В ходе лабораторной работы были изучены запросы на сайты и работа с json файлами.