**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра CАПР**

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**

****

**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"

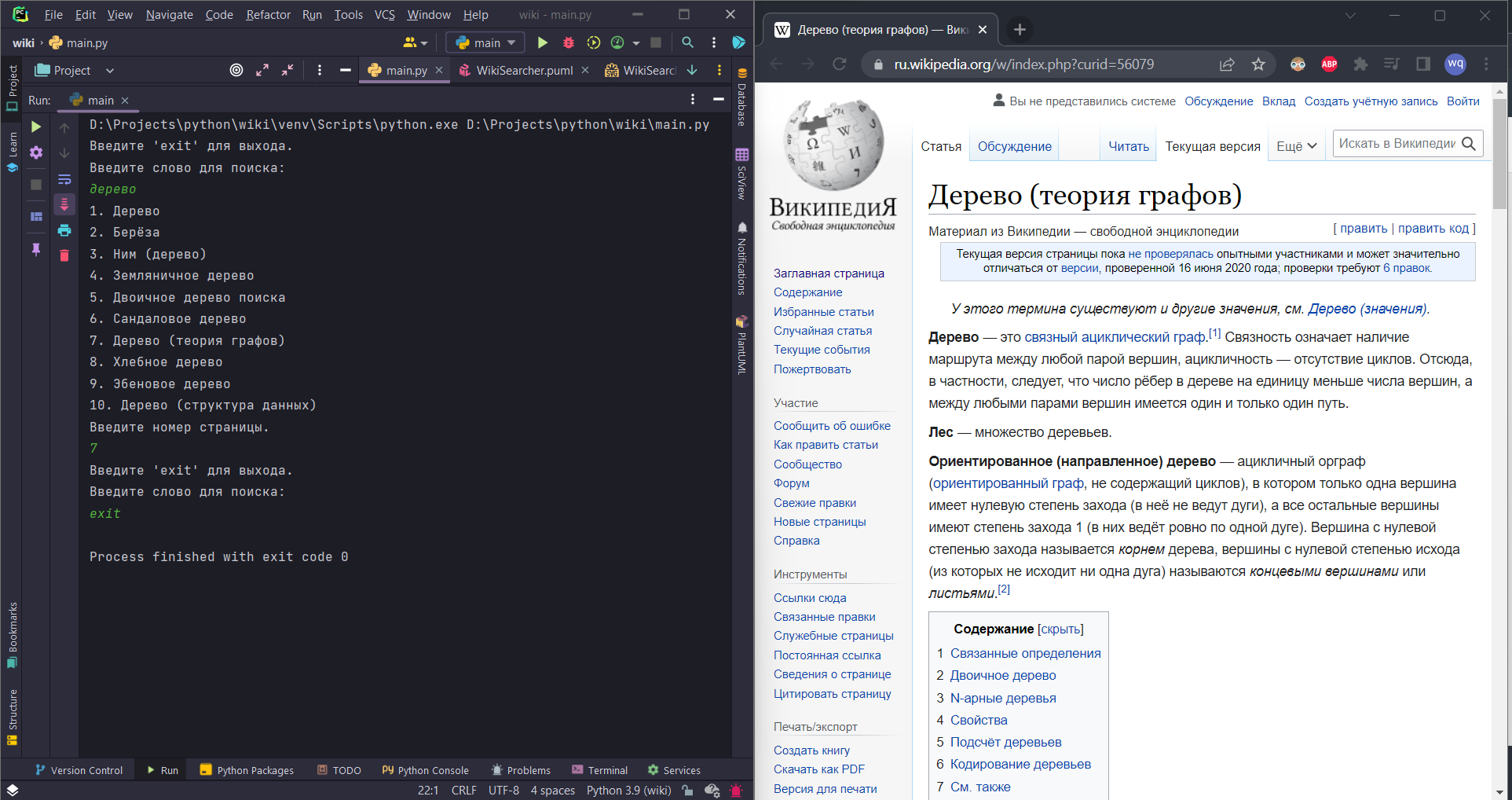
"Введите еще раз.")

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.