

Министерство образования Республики Беларусь  
Учреждение образования  
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерного проектирования

Кафедра инженерной психологии и эргономики

Дисциплина: Современные языки программирования

ЛАБОРАТОРНАЯ РАБОТА № 3

Выполнил:

Атаев И.М. гр. 910101

Проверила:

Василькова А.Н.

Минск 2022

## Задание:

### 3.1 Исследование и парсинг

Цель работы – приобрести навыки работы с исходными данными, получаемыми из открытых источников сети Интернет

**Задание:** Получить на вход программы URL страницы интернет-ресурса, а также параметр depth - уровень глубины исследования ссылок, присутствующих на заданной странице.

С помощью пакетов scrapy (уже присутствует в Anaconda), либо BeautifulSoup, lxml плюс модуля для оформления запросов - Requests осуществить просмотр/парсинг заданной веб-страницы и собрать статистику следующего вида:

Вариант 2. Количество слов на странице анализируемого уровня

## Исходный код:

```
import requests
import re
from bs4 import BeautifulSoup
from collections import Counter
from string import punctuation

url = "https://www.geeksforgeeks.org/fundamentals-of-algorithms/?ref=shm"

headers = {
    "Accept": "*//*",
    "User-Agent": "Mozilla/5.0 (iPad; CPU OS 11_0 like Mac OS X)
AppleWebKit/604.1.34 (KHTML, like Gecko) Version/11.0 Mobile/15A5341f
Safari/604.1"
}

req = requests.get(url, headers=headers)
src = req.text
print(src)
with open("index.html", "w") as file:
    file.write(src)
with open("index.html") as file:
    src = file.read()

soup = BeautifulSoup(src, "lxml")
# ПОИСК ВСЕХ ССЫЛОК
```

```

for link in soup.findAll('a'):
    print(link.get('href'))
print("Количество всех ссылок", len(soup.findAll('a')))
#поиск слов
find_all_clothes = soup.find_all(text=re.compile("([Aa]lgorithm)"))
print("Количество слова 1", len(find_all_clothes))
find_all_clothes = soup.find_all(text=re.compile("([Qq]uiz)"))
print("Количество слова 2", len(find_all_clothes))
find_all_clothes = soup.find_all(text=re.compile("([Ll]oops)"))
print("Количество слова 3", len(find_all_clothes))

def loopit():
    NUM = 0
    for TAG in soup.find_all():
        if TAG.string is not None:
            NUM = NUM + len(TAG.string.split())

            # print(TAG.string.split())
    print("Общее количество слов в параграфах", NUM)

loopit()
# #####
stop_list = [ "|", ":", "-", "/", ".", ", ", "\"", "▲", "“", "+", "Б", "]", "!", "—", "<", "&"]
word_count = Counter()
all_words = soup.get_text(" ", strip=True).lower().split()

#count words
for i in range(1,8):
    word_count.clear()
    for word in all_words:
        cln_word = word.strip('.,?')

        if len(cln_word) == i:

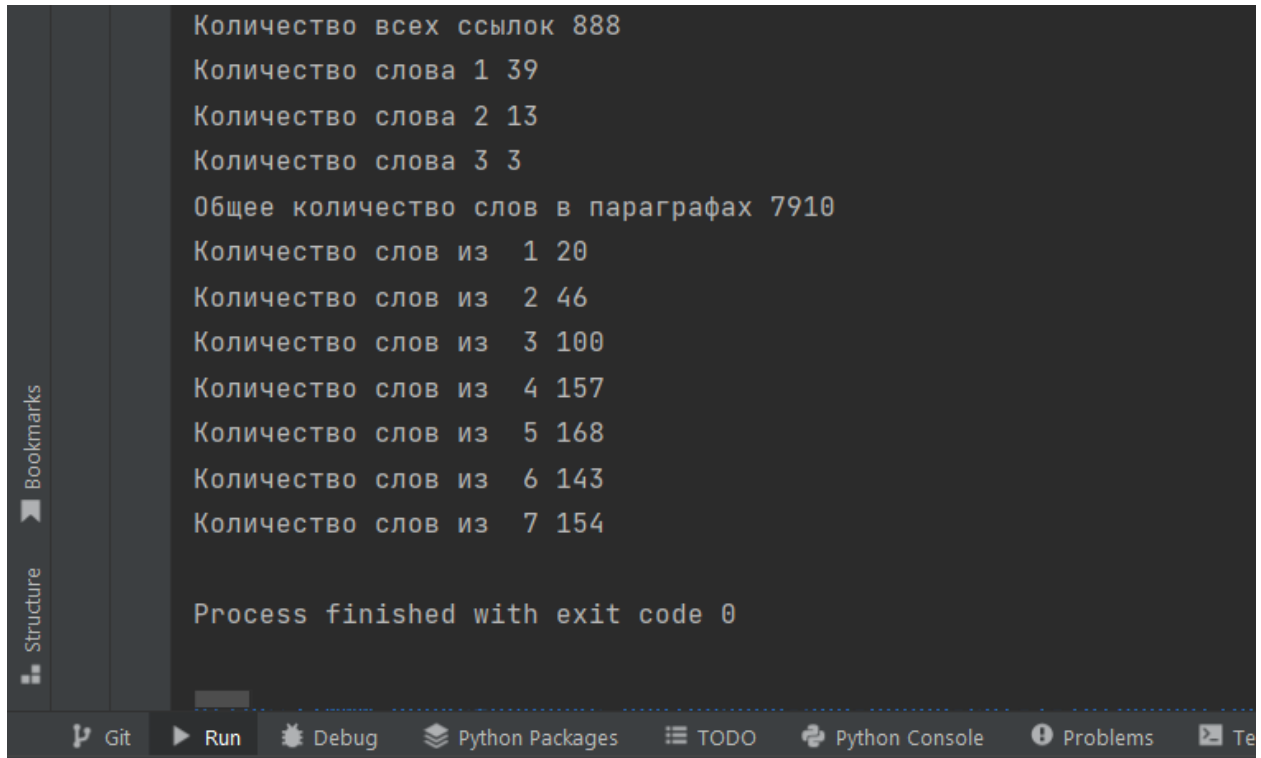
            if cln_word in stop_list:
                continue
            word_count[cln_word] += 1

    print("Количество слов из ", i, len(word_count.keys()))

#
# list=[1,2,3,4,5,6,7]
# list2=[20,44,98,152,167,144,151]

```

## Результаты работы программы:



```
Количество всех ссылок 888
Количество слова 1 39
Количество слова 2 13
Количество слова 3 3
Общее количество слов в параграфах 7910
Количество слов из 1 20
Количество слов из 2 46
Количество слов из 3 100
Количество слов из 4 157
Количество слов из 5 168
Количество слов из 6 143
Количество слов из 7 154

Process finished with exit code 0
```

Рис. 1 – Количество слов на странице анализируемого уровня.

### 3.2 Получение и подготовка исходных данных для анализа из закрытых источников,

(продолжительность 4 часа)

Получение и подготовка исходных данных для анализа из закрытых источников, требующих авторизации, но имеющих API.

Цель работы – приобрести навыки работы с исходными данными, получаемыми из закрытых источников сети Интернет, на примере социальной сети «ВК»; получить навыки работы с алгоритмом кластеризации k-means.

#### Исходный код:

```
import vk_api
session = vk_api.VkApi(token="***тут должен быть твой токен")
vk = session.get_api()
#находит айди всех друзей и друзей друзей
list=[]
def get_user_status(user_id):
    friends = session.method("friends.get", {"user_id": user_id})
    for friend in friends["items"]:
        user = session.method("users.get", {"user_ids":
friend,"fields":"bdate,counters"} )
        print(f"{user[0]['first_name']} ")
        list.append(user[0]['id'])
    # print(list)
    for i in range(0,2):
```

```

        user = session.method("friends.get", {"user_id": list[i]})#плучили
друзей
    # list.append(user["items"])
    for friend in user["items"]:
        user = session.method("users.get", {"user_ids": friend,
"fields": "bdate,counters"})
        print( f"{user[0]['first_name']} ")
        list.append(user[0]['id'])
    print(list)
    print(len(list))
all=[]#<---скопируй сюда список list из консоли чтобы не гонять функцию
get_user_status(user_id)
ints=list(set(all))
# print(ints)
# print(len(ints))
# counters_photo=[6, 24, 316, 1, 60, 2, 2, 0, 135, 502, 36, 3, 21, 646, 7,
831, 19, 40, 175, 155, 14, 39, 15, 1333, 36, 95, 0, 33, 62, 29, 1, 9, 226,
10, 3, 1, 1, 224, 8, 0, 12, 293, 6, 269, 17, 196, 370, 131, 1084, 817, 188,
46, 3, 305, 18, 30, 2, 60, 920, 70, 74, 194, 135, 169, 83, 11, 16, 1, 108,
11, 1, 3, 1564, 48, 407, 5, 198, 0, 62, 100, 4, 9, 11, 115, 471, 14, 4, 12,
7, 7, 7, 276, 20, 2, 2, 37, 6, 59, 10241, 31, 49, 0, 17, 34, 16, 1411, 33,
46, 15, 30, 7, 46, 61, 5, 248, 1, 908, 7, 2, 113, 5, 1, 12, 237, 0, 4, 1, 56,
5, 84, 416, 29, 8, 310, 0, 12, 28, 3, 631, 1, 190, 567, 740, 15, 746, 399]
# y=range(0,2000)
# counters_video=[649, 289, 0, 4, 13, 70, 3, 47, 5, 0, 85, 0, 3, 0, 9, 0,
1251, 0, 0, 145, 101, 10, 9, 6, 101, 0, 41, 0, 0, 2, 53, 99, 5, 0, 123, 5, 0,
2, 0, 11, 0, 0, 286, 0, 0, 0, 0, 2, 0, 0, 77, 10, 8, 0, 75, 365, 11, 10, 10,
461, 0, 2, 1, 0, 24, 6, 0, 1, 38, 0, 249, 0, 22, 30, 40, 223, 4, 3, 81, 0,
204, 90, 0, 4, 41, 1, 16, 0, 21, 0, 1, 12, 0, 8, 40, 157, 170, 0, 132, 302,
49, 0, 0, 4, 102, 13, 0, 21, 5, 171, 0, 7, 106, 193, 124, 0, 2, 0, 8, 3, 0,
4, 223, 18, 0, 2, 4, 64, 74, 0, 21, 0, 131, 0, 7, 0, 10, 0, 17, 2, 16, 1, 0,
0, 0, 0, 2, 14, 176, 2, 168, 0, 7, 0, 13, 0, 14, 0, 0, 0, 44, 97, 119, 91,
206, 116, 0, 2, 0, 19]
# counters_notes=[]
# counters_groups=[28, 353, 190, 34, 77, 58, 36, 100, 201, 7, 102, 12, 0, 7,
130, 2, 8, 239, 48, 421, 38, 9, 6, 10, 150, 42, 650, 33, 20, 169, 32, 472, 9,
43, 203, 373, 47, 74, 81, 54, 3, 7, 42, 16, 0, 173, 35, 26, 54, 180]
# age=['4.2', '25.12.1990', '29.4', '9.10', '30.4.1917', '5.11', '14.6.1991',
'24.9', '12.1', '20.12', '4.10', '21.12.1991', '11.8', '14.10', '22.2.1998',
'10.8.1987', '12.8.1989', '21.1.1987', '17.3', '9.2.2001', '2.5.1988',
'10.5', '28.5', '20.7', '5.5', '4.12.1986', '27.4.1999', '7.4', '7.10.2007',
'16.3.1985', '7.3.1998', '18.3.2001', '10.5', '20.1.2003', '14.3', '15.2',
'29.6.1974', '1.4.2008', '30.7', '5.3', '29.9', '4.10', '18.12',
'19.12.1987', '28.5', '9.9', '13.12', '22.10.1987', '14.6', '6.2', '22.2',
'21.12.1999', '11.2', '25.4', '29.5', '30.4', '15.11', '6.4.1994', '9.10',
'7.9.1991', '3.3', '7.5', '17.11', '24.7.1998', '2.8', '14.5', '16.1',
'24.7.2001', '6.10.1987', '7.7.1994', '14.7', '19.1', '16.2', '23.12',
'2.11.2000', '11.10.1987', '8.9', '23.12.2001', '27.8.2001', '24.6.1987',
'1.4.2002', '9.3', '31.3', '8.3.1987', '6.7', '3.5', '13.2', '28.5',
'27.6.1991', '10.8.2000', '13.11', '21.5', '29.11', '8.9', '11.4',
'23.5.2000', '5.11', '6.10', '28.1', '4.9.1999', '7.8', '6.4', '14.12.1988',
'16.12', '13.10.1987', '6.4.1994', '21.5', '26.2.1987', '14.11.1984',
'2.8.1987', '7.1', '7.12', '23.5', '5.4.1985', '14.3', '11.11.1987', '21.7',
'24.3', '5.1', '28.9', '13.9', '9.3.2002', '14.6', '8.4', '9.8.1989',
'22.11', '7.1.1987']
#тут можно парсить любые данные по типу даты рождения, города и тд только
надо вписать этот вид в после fields
for i in range(0,50):
    user = session.method("users.get", {"user_ids": ints[i], "fields":
"bdate,counters,status,city,country"})
    try:
        print(f"{user[0]['first_name']} {user[0]['last_name']}
{user[0]['city']} {user[0]['country']} ")
        # counters_photo.append(user[0]['counters']['photos'])
        # counters_video.append(user[0]['counters']['videos'])
        # counters notes.append(user[0]['counters']['notes'])

```

```

        # counters_groups.append(user[0]['counters']['groups'])
        # counters_groups.append(user[0]['counters']['groups'])
        # age.append(user[0]['bdate'])
    except KeyError:
        print("no data")

#
# print(counters_photo)
# print(counters_video)
# print(counters_notes)
# print(counters_groups)
# print(age)
# print(user)
#     try:
#         print(f"{user[0]['first_name']} {user[0]['last_name']}
{user[0]['bdate']} {user[0]['counters']['photos']}")
#         list.append(user[0]['counters']['photos'])
#     except KeyError:
#         print("no data")
# print(list)
get_user_status(тут твоё айди цифрами)

```

## Результаты работы программы:

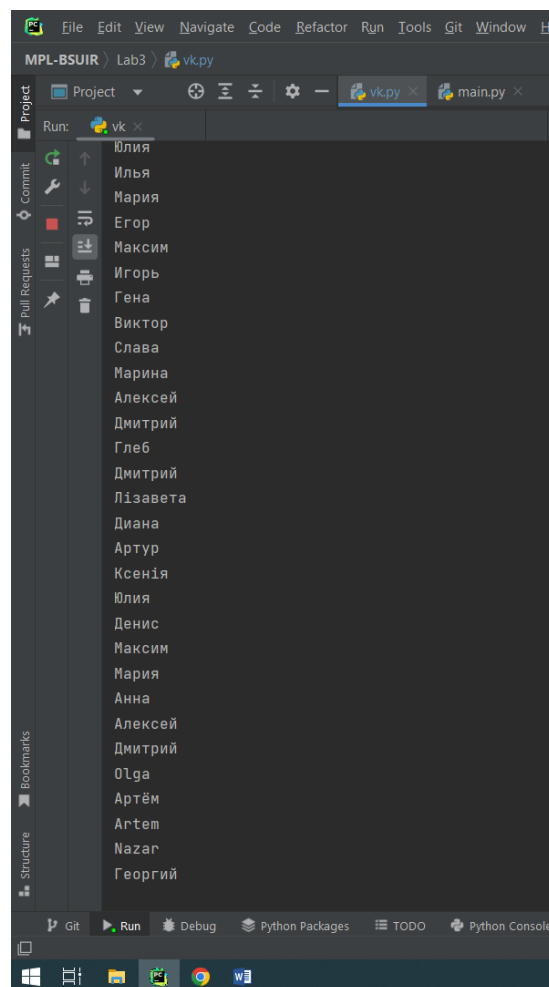


Рис. 1 – Получение и подготовка исходных данных для.

**Вывод:** В ходе выполнения лабораторной работы ознакомились с приобретением навыка работы с исходными данными, получаемыми из открытых источников сети Интернет