Московский Государственный Технический Университет Имени Н.Э. Баумана

Отчет по Лабораторной Работе №2

По Курсу «Разработка интернет приложений»

Выполнил:

# Студент группы ИУ5-52

# Шлыков Виктор

Москва 2017

Задание и порядок выполнения

Вход:

username или vk\_id пользователя Выход:

Гистограмма распределения возрастов друзей пользователя, поступившего на вход Пример:

Вход: reigning base.py

**class** BaseClient:

*# URL vk api*

BASE\_URL = **None**

*# метод vk api* method = **None** *# GET, POST, ...* http\_method = **None**

*# Получение GET параметров запроса* **def** get\_params(self):

**pass**

*# Получение данных POST запроса* **def** get\_json(self):

**pass**

*# Получение HTTP заголовков* **def** get\_headers(self):

**pass**

*# Склейка url* **def** generate\_url(self, method):

**return '{0}{1}'**.format(self.BASE\_URL, method)

*# Отправка запроса к VK API* **def** \_get\_data(self, method, http\_method):

response = **None**

*#* ***todo выполнить запрос***

**return** self.response\_handler(response)

*# Обработка ответа от VK API* **def** response\_handler(self, response):

**return** response

*# Запуск клиента* **def** execute(self): **return** self.\_get\_data( self.method, http\_method=self.http\_method

)

friends.py

**from** base **import** \* **import** requests **import** json

**from** datetime **import** datetime

**class** GetFriends(BaseClient):

BASE\_URL = **'https://api.vk.com/method/friends.get'** http\_method = **'GET'**

**def** \_\_init\_\_(self, uid):

self.uid = uid

**def** get\_params(self):

**return 'user\_id='** + str(self.uid) + **'&fields=bdate'**

**def** response\_handler(self, response): **try**:

obje = json.loads(response.text) friends = obje.get(**'response'**)

ages = []

**for** friend **in** friends:

b\_date = friend.get(**'bdate'**)

**if** b\_date **is None or** b\_date.count(**'.'**) < 2:

**continue**

b\_date = datetime.strptime(b\_date, **"%d.%m.%Y"**) n\_date = datetime.now()

ages.append(int((n\_date - b\_date).days) // 365.2425)

uniqages = list(set(ages))

**return** sorted([(x, ages.count(x)) **for** x **in** uniqages], key=**lambda** x: x[0]) **except**:

**raise** Exception(**"У пользователя нет друзей, либо они недоступны {}"**.format(self.uid))

**def** \_get\_data(self, method, http\_method): response = requests.get(self.BASE\_URL + **'?'** + self.get\_params()) **return** self.response\_handler(response) uid.py

**from** base **import** \* **import** requests **import** json

**class** GetID(BaseClient):

BASE\_URL = **'https://api.vk.com/method/users.get'** http\_method = **'GET'**

**def** \_\_init\_\_(self, name):

self.name = name

**def** get\_params(self):

**return 'user\_ids='** + self.name

**def** response\_handler(self, response): **try**:

obje = json.loads(response.text) **return** obje.get(**'response'**)[0].get(**'uid'**) **except**:

**raise** Exception(**"Данный пользователь не найден**

**{}"**.format(self.name))

**def** \_get\_data(self, method, http\_method):

response = **None**

response = requests.get(self.BASE\_URL + **'?'** + self.get\_params()) **return** self.response\_handler(response)

main.py

**import** matplotlib.pyplot **as** plt **from** uid **import** \* **from** friends **import** \* **import** random

input\_id = input(**'Введите id: '**) user = GetID(input\_id) user\_id = user.execute()

friends\_client = GetFriends(user\_id) friends = friends\_client.execute()

ages = [] counts = []

**for** (age, count) **in** friends: print(**'{} {}'**.format(int(age), int(count))) ages.append(int(age)) counts.append(int(count))

plt.grid() plt.axis([0, 60, 0, 50]) plt.figure(num=1, figsize=(8, 6)) plt.xlabel(**'age'**, size=14) plt.ylabel(**'count'**, size=14) plt.bar(ages, counts, width=0.5) plt.show()

