

Московский государственный технический
университет им. Н.Э. Баумана
Факультет «Информатика и системы управления» Кафедра
«Системы обработки информации и управления»



“Разработка интернет- приложений”

«Python-классы»

Лабораторная работа № 3

Студент группы ИУ5-53

_____ Атаманов В. В.

Преподователь

_____ Гапанюк Е. Ю.

Москва 2017

Задание

Вход: username или vk_id пользователя

Выход: Гистограмма распределения возрастов друзей пользователя, поступившего на вход

Пример:

Вход:

reigning

Выход:

```
19 #
20 ##
21 ##
22 #####
23 #####
24 ####
25 #
28 #
29 #
30 #
37 #
38 ##
45 #
```

Указания За основу возьмите базовый класс:

<https://gist.github.com/Abashinos/024c1dc9f92f1ff733c63a07e447ab51>

Для реализации методов ВК наследуйтесь от этого базового класса. Создайте один класс для получения id пользователя из username и один для получения и обработки списка друзей. В классах-наследниках необходимо реализовать методы: ● get_params - если есть get параметры (необязательно). ● get_json - если нужно передать post данные (необязательно). ● get_headers - если нужно передать

дополнительные заголовки (необязательно). • `response_handler` - обработчик ответа. В случае успешного ответа необходим, чтобы преобразовать результат запроса. В случае ошибочного ответа необходим, чтобы сформировать исключение. • `_get_data` - внутренний метод для отправки http запросов к VK API.

Для решения задачи нужно обратиться к двум методам VK API

- 1) `users.get` - для получения vk id по username
- 2) `friends.get` - для получения друзей пользователя. В этом методе нужно передать в get параметрах `fields=bdate` для получения возраста. Нужно принять во внимание, что не у всех указана дата рождения

Код программы: `base_client.py`

```
import requests
class BaseClient:
    # URL vk api
    BASE_URL = "https://api.vk.com/method/"
    # метод vk api
    method = None
    # GET, POST, ...
    http_method = None
    # Получение GET параметров запроса
    def get_params(self):
        return None
    # Получение данных POST запроса
    def get_json(self):
        return None
    # Получение HTTP заголовков
    def get_headers(self):
        return None
    # 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):
        if http_method == 'GET':
            response = requests.get('{0}?{1}'.format(self.generate_url(method), self.get_params()))
        elif http_method == 'POST':
            response = requests.post(self.generate_url(method), data=self.get_json())
        else:
            return None
        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
    )
```

Код программы: friends.py

```
from base_client import BaseClient
from json import loads
class Friends(BaseClient):
    # метод vk api
    method = "friends.get"
    # GET, POST, ...
    http_method = "GET"
    def __init__(self, user_id):
        self._user_id = user_id
    def get_params(self):
        return 'user_id={0}&fields=bdate'.format(self._user_id)
    def response_handler(self, response):
        return loads(response.text)['response']
```

Код программы: user_id.py

```
from base_client import BaseClient
from json import loads
class UserId(BaseClient):
    # метод vk api
    method = "users.get"
    # GET, POST, ...
    http_method = "GET"
    def __init__(self, username):
        self._username = username
    def get_params(self):
        return 'uids=' + self._username
    def response_handler(self, response):
        data = loads(response.text)['response']
        return data[0].get('uid')
```

Код программы: main.py

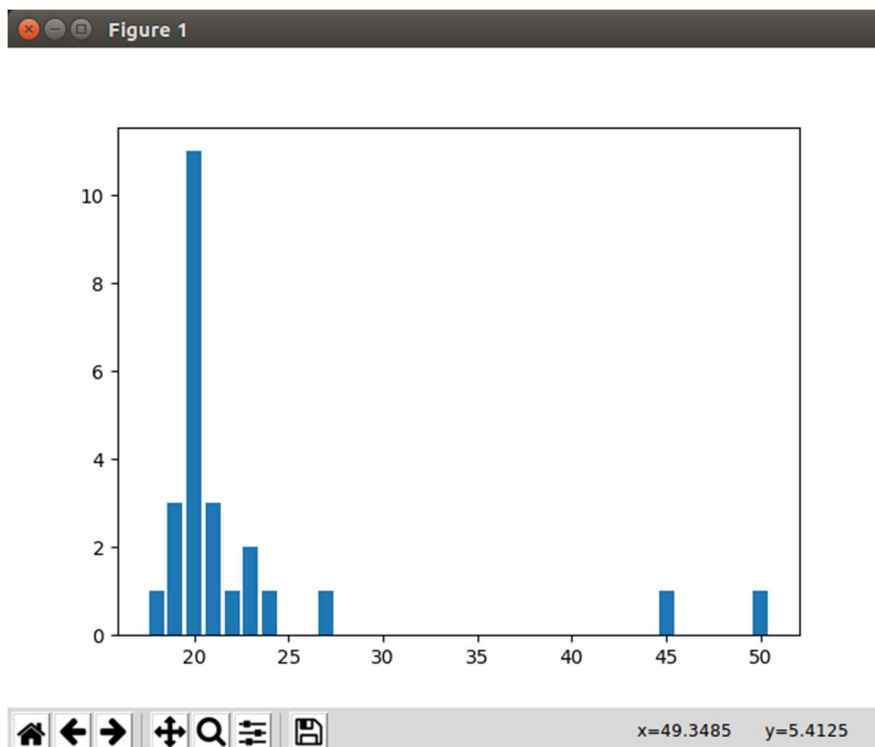
```
from datetime import datetime, date
from user_id import UserId
from friends import Friends
from random import randint, normalvariate
import matplotlib.pyplot as plt
if __name__ == "__main__":
    ages_dct = dict()
    get_user_id = UserId(input())
    get_friends = Friends(get_user_id.execute())
    it_dct = get_friends.execute()
    for f in range(100000):
        if f.get('bdate') is None:
            continue
```

```

bdate = f['bdate'].split('.')
if len(bdate) != 3:
    continue
age = datetime.now(tz=None).date() - date(int(bdate[2]), int(bdate[1]), int(bdate[0]))
age = int(age.days / 365)
if ages_dct.get(age) is None:
    ages_dct[age] = 1
else:
    ages_dct[age] += 1
x_axis = []
y_axis = []
items = sorted(ages_dct.items(), key=lambda item: item[0])
for key, value in items:
    x_axis.append(key)
    y_axis.append(value)
    print(key, end=' ')
    for i in range(value):
        print('#', end='')
    print()
plt.bar(x_axis, y_axis, align='center')
plt.show()

```

Результаты работы:



lab_3 - [~/GitHub/BMSTU_Python/lab_3] - .../main.py - PyCharm 2017.2.4

File Edit View Navigate Code Refactor Run Tools VCS Window Help

lab_3 > main.py >

Terminal

```
+ viewsharp@viewsharp-DELL:~/GitHub/BMSTU_Python/lab_3$ python3
main.py
viewsharp
18 #
19 ###
20 #####
21 ###
22 #
23 ##
24 #
27 #
45 #
50 #
```

1: Project
2: Structure
Favorites

Data View
Database

6: TODO Python Console Terminal Event Log

Unregistered VCS root detected: The direc... (5 minutes ago) 13:21 LF: UTF-8