Московский государственный
технический университет
им. Н.Э. Баумана

Разработка интернет-приложений
Лабораторная работа № 3
“Python – Классы”
С доп. заданием

Выполнил:
студент группы ИУ5-53
Степанов Д. Г
Подпись:
Дата:

Задание

Москва 2017г.

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/024c1dcaf92f1ff733c63a07e447ab51

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

Описание методов можно найти тут:
https://vk.com/dev/methods

Разнесите базовый класс, классы наследники и основную программу в разные модули.
Про модули можно прочитать тут:
https://docs.python.org/3/tutorial/modules.html
https://habrahabr.ru/post/166463/

Для выполнения запросов нужно использовать библиотеку *requests*
http://docs.python-requests.org/en/master/

Для обработки дат (дней рождения) используйте встроенную библиотеку *datetime*
https://docs.python.org/3/library/datetime.html

Чтобы установить библиотеку используйте пакетным менеджером *pip*
https://pip.pypa.io/en/stable/quickstart/

Подсказки:

1. Метод get библиотеки requests принимает вторым аргументом словарь get-параметров.
2. Не забывайте, что в классах-наследниках можно перегружать статические поля наследуемого класса.

Дополнительное задание

Постройте гистограмму с использованием *matplotlib*
http://matplotlib.org/examples/statistics/histogram_demo_features.html

Ф а й л base_client.py

```python
class BaseClient:

    # URL vk api
    BASE_URL = "https://api.vk.com/"
    # 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):
        response = None
        # todo выполнить запрос
        return self.response_handler(response)
```

```python
    # 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
        )
```

Ф а й л vk_get_userid.py

```python
import requests
import datetime
import json
from base_client import BaseClient


class VkUserId(BaseClient):
    BASE_URL = "https://api.vk.com/"
    http_method = "?"

    def __init__(self, user_domain):
        self.params = {"user_ids": user_domain, "fields": "bdate", "access_token": "", "v": "5.68" }
        self.user_domain = user_domain
        self.method = "method/users.get"
        self.response = requests.Response()
        self.response_content = ""
        self.user_id = ""

    def get_params_string(self):
        params_string = ""
        for key, value in self.params.items():
            params_string = params_string + key + "=" + value + "&"
        return params_string

    def _get_data(self, method, http_method):
        response = requests.Response()
        print(self.BASE_URL + self.method + self.http_method + self.get_params_string())
        # Argument y in requests.get(x, y) is a dictionary of parameters
        try:
            response = requests.get(self.BASE_URL + self.method + self.http_method,  self.params)
        except ConnectionError:
            print("Ошибка соединения с сервером")
            exit(1)
        return self.response_handler(response)

    def response_handler(self, response):
        if response.status_code != 200:
            print("response :", response.status_code)
```

VkUserId

```
    def response_handler(self, response):
        if response.status_code != 200:
            print("response :", response.status_code)
            exit(1)
        self.response = response
        self.response_content = response.content.decode("utf-8")
        data = json.loads(self.response_content)
        try:
            self.user_id = data["response"][0]["id"]
        except KeyError:
            print("Invalid user domain")
            exit(1)
        return None

    def get_headers(self):
        return self.response.headers()

    def execute(self):
        return self._get_data(
            self.method,
            http_method=self.http_method
        )
```

Ф а й л vk_get_userfriends.py

```
import requests
import datetime
import json
from base_client import BaseClient


class VkUserFriends(BaseClient):
    BASE_URL = "https://api.vk.com/"
    http_method = "?"

    def __init__(self, user_id):
        self.params = {"user_id": str(user_id), "fields": "bdate", "access_token": "", "v": "5.68"}
        self.user_id = user_id
        self.method = "method/friends.get"
        self.response_content = ""
        self.dates_list = []
        self.ages_list = []
        self.response = requests.Response()
```

```
    def get_params_string(self):
        params_string = ""
        for key, value in self.params.items():
            params_string = params_string + key + "=" + value + "&"
        return params_string

    def _get_data(self, method, http_method):
        response = requests.Response()
        print(self.BASE_URL + self.method + self.http_method + self.get_params_string())
        try:
            response = requests.get(self.BASE_URL + self.method + self.http_method,  self.params)
        except ConnectionError:
            print("Ошибка соединения с сервером")
            exit(1)
        return self.response_handler(response)
```

```
    def response_handler(self, response):
        if response.status_code != 200:
            print("response :", response.status_code)
            exit(1)
        self.response = response
        self.response_content = response.content.decode("utf-8")
        data = json.loads(self.response_content)

        try:
            for el in data["response"]["items"]:
                if "bdate" in el:
                    self.dates_list.append(el.get("bdate", "None"))
        except KeyError:
            print("Invalid user domain")
            exit(1)
```

```
        date_obj_list = []
        for dates in self.dates_list:
            try:
                date_obj_list.append(datetime.datetime.strptime(dates, "%d.%m.%Y"))
            except ValueError:
                pass

        for obj in date_obj_list:
            age = int((datetime.datetime.now()-obj).days // 365.25)
            self.ages_list.append(age)
        return self.ages_list

    def get_headers(self):
        return self.response.headers()

    def execute(self):
        return self._get_data(
            self.method,
            http_method=self.http_method
        )
```

Ф а й л main_prog.py

```
import vk_get_userfriends
import vk_get_userid
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt


user_domain = input("Введите имя пользователя \n ")
client_object1 = vk_get_userid.VkUserId(user_domain)
client_object1.execute()

print(client_object1.response_content)
print("user_id = ", client_object1.user_id)


client_object2 = vk_get_userfriends.VkUserFriends(client_object1.user_id)
client_object2.execute()

unique_ages_list = sorted(set(client_object2.ages_list))
for el in unique_ages_list:
    print(el, ":", "#"*client_object2.ages_list.count(el))

fig, ax = plt.subplots()
plt.hist(client_object2.ages_list, max(client_object2.ages_list),color="gray", rwidth=1)
ax.set_xlabel('Age')
ax.set_ylabel('count')
ax.set_title("Histogram of Ages")
fig.tight_layout()
plt.show()


print('Script Ends Here')
```

Результат выполнения программы

```
C:\Users\0\AppData\Local\Programs\Python\Python36-32\python.exe C:/GIT/python_labs/lab3/main_p
Введите имя пользователя
 taron997
https://api.vk.com/method/users.get?user_ids=taron997&fields=bdate&access_token=&v=5.68&
{"response":[{"id":14377480,"first_name":"Тарон","last_name":"Степанян","bdate":"4.10.1997"}]}
user_id =  14377480

https://api.vk.com/method/friends.get?user_id=14377480&fields=bdate&access_token=&v=5.68&
14 : ###
15 : ##
16 : ######
17 : ##################
18 : ########################
19 : ##############################################
20 : ####################################################
21 : #############
22 : ################
23 : ########
24 : ##
25 : ##
26 : ###
27 : ###
28 : #
29 : #####
31 : #
32 : #
34 : #
35 : #
45 : #
48 : #
72 : ##
82 : #
90 : #
95 : ##
97 : ##
102 : ##
107 : #
113 : #
115 : ##
```

Гистограмма распределения возрастов друзей пользователя