

Защищено:
Гапанюк Ю.Е.

Демонстрация:
Гапанюк Ю.Е.

"__" _____ 2016 г.

"__" _____ 2016 г.

Отчет по лабораторной работе № 3 по курсу
«Разработка интернет-приложений»
«Python-классы»

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-53

Ореликов М.Г.

(подпись)

"__" _____ 2016 г.

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 #
```

base_client.py

```
class BaseClient:
    # URL vk api
    # id 191483545
    BASE_URL = None

    # 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
    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
    )

```

vk.py

```

from id import Id
from friends import Friends

name = input('Имя пользователя: ')
client = Id(name)
uid = client.execute()

friends_client = Friends(uid)
friends = friends_client.execute()

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

```

id.py

```

from base_client import BaseClient
import requests
import json

class Id(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

    # Отправка запроса к VK API
    def _get_data(self, method, http_method):
        response = None
        response = requests.get(self.BASE_URL + '?' + self.get_params())
        return self.response_handler(response)

```

```

# Обработка ответа от VK API
def response_handler(self, response):
    try:
        uobj = json.loads(response.text)
        print ('ID пользователя: ' + str((uobj.get('response')[0].get('uid'))))
        return uobj.get('response')[0].get('uid')
    except:
        raise Exception("Couldn't handle response for username
{}".format(self.name))

```

friends.py

```

from base_client import BaseClient
from urllib import parse
import requests
import json
from datetime import datetime

class Friends(BaseClient):
    BASE_URL = 'https://api.vk.com/method/friends.get'
    http_method = 'GET'

    def __init__(self, uid):
        self.uid = uid

    # Получение GET параметров запроса
    def get_params(self):
        return parse.urlencode({'user_id': self.uid, 'fields' : 'bdate'})

    # Отправка запроса к VK API
    def _get_data(self, method, http_method):
        response = requests.get(self.BASE_URL + '?' + self.get_params())
        return self.response_handler(response)

    # Обработка ответа от VK API
    def response_handler(self, response):
        try:
            age = []
            uobj = json.loads(response.text)
            friends = uobj.get('response')
            for friend in friends:
                bdate = friend.get('bdate')
                if bdate is None or bdate.count('.') < 2:
                    continue
                bdate = datetime.strptime(bdate, "%d.%m.%Y")
                ndate = datetime.now()
                age.append(int((ndate - bdate).days) // 365)

            unage = list(set(age))
            return sorted((x, age.count(x)) for x in unage)
        except:
            raise Exception("Couldn't handle response for friends of uid
{}".format(self.uid))

```

Результаты

Имя пользователя: `id191483545`

ID пользователя: 191483545

19 ####

20 #####

21 #

24 #

60 #

Process finished with exit code 0