

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



Отчет по Лабораторной работе 3
По курсу
“Разработка интернет-приложений”

Преподаватель_____Гапанюк Ю.Е.

Студентка группы ИУ5-52 Гергиева Д.Р.

Москва, 2017г.

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 #
```

Исходный код

base_client.py

```
import requests
#https://api.vk.com/method/

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 printf("%s %d", str, num)
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):
    resp = requests.get(url=self.generate_url(self.method),
params=self.get_params())
    return self.response_handler(resp.json())

# Обработка ответа от VK API
def response_handler(self, response):
    return response

# Запуск клиента
def execute(self):
    return self._get_data(
        method=self.method,
        http_method=self.http_method
    )

```

client_get_friends_ages.py

```

from base_client import BaseClient
import datetime

class ClientGetFriendsAges(BaseClient):
    user_id = None
    method = "friends.get"
    http_method = "GET"

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

    def get_params(self):
        params = {"user_id": self.user_id, "fields": "bdate"}
        return params

    def response_handler(self, response):
        friends = response["response"]
        ages = []
        for f in friends:
            if "bdate" in f:
                date = f["bdate"] # %d.12.2012
                if len(date)>5:
                    now = datetime.datetime.now()
                    date_time = datetime.datetime.strptime(date, "%d.%m.%Y")
                    age = now - date_time
                    ages.append(age.days//365)

        return ages

```

get_user_id.py

```
from base_client import BaseClient

class GetUserId(BaseClient):
    user_ids = None
    method = 'users.get'

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

    def get_params(self):
        params = {"user_ids": self.user_ids}
        return params

    def response_handler(self, response):
        return response["response"][0]["uid"]
```

histogram.py

```
import matplotlib.pyplot as plt

from client_get_friends_ages import ClientGetFriendsAges

from get_user_id import GetUserId

id = GetUserId("salastorus").execute()

arr = ClientGetFriendsAges(id)

list = arr.execute()

plt.hist(list, range(min(list), max(list)))

plt.show()
```

Скриншот работы программы

