

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



Гапанюк Ю.А.

Отчет по Лабораторной работе 2  
По курсу  
“Разработка интернет-приложений”

Выполнил:  
Кемерова Д.М.  
Студент группы ИУ5-54

Москва 2017

## Задание

Вход:

username или vk\_id пользователя

Выход:

Гистограмма распределения возрастов друзей пользователя, поступившего на вход

## ЛИСТИНГ

vk\_clients.py

```
from base_class import *
import requests
from datetime import datetime
class GetID(BaseClient):
    v = 5.58
    user_id = None

    def __init__(self, username):
        super().__init__('https://api.vk.com/method/', 'users.get', 'GET')
        self.user_id = username

    def get_params(self):
        return {'user_ids': self.user_id, 'v': self.v}

    def _get_data(self, method, http_method):
        r = requests.get(self.generate_url(method), self.get_params())
        return self.response_handler(r)

    def response_handler(self, response):
        res_dic = response.json()
        if not res_dic.get('error') is None:
            err = res_dic.get('error')
            err_msg = err.get('error_msg')
            raise Exception(err_msg)
        else:
            return res_dic.get('response')[0]

class GetFriends(BaseClient):
    user_id = None
    fields = 'bdate'
    v = 5.68

    def __init__(self, user_id):
        super().__init__('https://api.vk.com/method/', 'friends.get', 'GET')
        self.user_id = user_id

    def get_params(self):
        return {'user_id': self.user_id, 'v': self.v, 'fields': self.fields}

    def _get_data(self, method, http_method):
        r = requests.get(self.generate_url(method), self.get_params())
        return self.response_handler(r)

    def response_handler(self, response):
        res_dic = response.json()
        if not res_dic.get('error') is None:
            err = res_dic.get('error')
            err_msg = err.get('error_msg')
            raise API_Exception(err_msg)
        else:
            return res_dic.get('response').get('items')
```

## base\_class

```
#import requests
class BaseClient:
    # URL vk api
    BASE_URL = None
    # метод vk api
    method = None
    # GET, POST, ...
    http_method = None

    def __init__(self, url, m, h):
        self.BASE_URL = url
        self.method = m
        self.http_method = h

    # 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
        print('base')

        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,
            self.http_method
        )
```

## diagram

```
from vk_clients import *
username=input()
get_user = GetID(username)
user = get_user.execute()
user_id = user.get('id')

get_friends = GetFriends(user_id)
friends = get_friends.execute()

age_list = [0 for i in range(120)]
```

```

today = datetime.now()

print(user.get('last_name'), user.get('first_name'), sep=' ')

for f in friends:
    bdate_str = f.get('bdate')
    try:
        bdate = datetime.strptime(bdate_str, '%d.%m.%Y')
        days = (today-bdate).days
        age = days // 365
        age_list[age] += 1
    except:
        pass
for i in range(120):
    if age_list[i]>0:
        print(i, ': ', '@'*age_list[i])

import matplotlib.pyplot as plt
import numpy as np

y = np.arange(len(age_list))
plt.bar(y, age_list, align='center')
plt.show()

```

## Результат

```

Дудырев Егор
13 : #
14 : ##
15 : ###
16 : #
17 : #
18 : #####
19 : #####
20 : #####
21 : #####
22 : #####
23 : #####
24 : #####
26 : ###
27 : #
28 : #
29 : ###
30 : #
31 : #
32 : #
33 : #
36 : #
42 : #
73 : #
79 : #
93 : #
100 : #

```

