

Московский Государственный Технический Университет
Имени Н.Э. Баумана

Отчет по Лабораторной Работе №2
По Курсу «Разработка интернет приложений»

Выполнил:
Студент группы ИУ5-52
Поляков Дмитрий

Москва 2017

Задание и порядок выполнения Вход:

username или vk_id пользователя

Выход:

Гистограмма распределения возрастов друзей пользователя, поступившего на вход Пример:

Вход: reigning

base.py

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

    # Получение GET параметров запроса
    def get_params(self):
        pass

    # Получение данных POST запроса
    def get_json(self):
        pass

    # Получение HTTP заголовков
    def get_headers(self):
        pass

    # 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)

    # 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
    )
```

friends.py

```
from base import *
import requests import
json
from datetime import datetime
class
GetFriends(BaseClient):
    BASE_URL = 'https://api.vk.com/method/friends.get'
    http_method = 'GET'
    def __init__(self,
uid):
        self.uid = uid
    def
get_params(self):
        return 'user_id=' + str(self.uid) + '&fields=bdate'
    def response_handler(self,
response):
        try:
            obje = json.loads(response.text)
            friends = obje.get('response')

            ages = []
            for friend in
friends:
                b_date = friend.get('bdate')
                if b_date is None or b_date.count('.') <
2:
                    continue

                b_date = datetime.strptime(b_date, "%d.%m.%Y")
            n_date = datetime.now()

            ages.append(int((n_date - b_date).days) // 365.2425)

            uniqages = list(set(ages))
            return sorted([(x, ages.count(x)) for x in uniqages], key=lambda
x: x[0])
        except:
            raise Exception("У пользователя нет друзей, либо они недоступны
{}".format(self.uid))
    def _get_data(self, method, http_method):
        response =
requests.get(self.BASE_URL + '?' + self.get_params())
        return
self.response_handler(response) uid.py

from base import *
import requests import
json
class
GetID(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
    def response_handler(self,
response):
        try:
            obje = json.loads(response.text)
            return obje.get('response')[0].get('uid')
        except:
```

```

        raise Exception("Данный пользователь не найден
{}".format(self.name))
    def _get_data(self, method,
http_method):
        response = None

        response = requests.get(self.BASE_URL + '?' + self.get_params())
return self.response_handler(response)

```

main.py

```

import matplotlib.pyplot as plt
from uid import * from friends
import * import random

input_id = input('Введите id: ')
user = GetID(input_id) user_id =
user.execute()
    friends_client =
GetFriends(user_id) friends =
friends_client.execute()
    ages = []
counts = []
    for (age, count) in friends:    print('{}
{}'.format(int(age), int(count)))
ages.append(int(age))
counts.append(int(count))
    plt.grid() plt.axis([0, 60, 0,
50]) plt.figure(num=1,
figsize=(8, 6)) plt.xlabel('age',
size=14) plt.ylabel('count',
size=14) plt.bar(ages, counts,
width=0.5) plt.show()

```

Введите id: 82858857

14 2
17 1
18 1
19 6
20 20
21 10
22 3
26 3
27 2
28 1
29 1
34 1
76 1
116 1

