

Московский государственный технический университет им. Н.Э. Баумана

Факультет «Радиотехнический»

Кафедра «Автоматизированные системы обработки информации и управления»



**Отчёт по лабораторной работе №2
по дисциплине Разработка Интернет приложений**

Выполнил:

Студент группы РТ5-51

Богомолов Д.Н.

Москва 2017

Задание

Вход:

username или vk_id пользователя

Выход:

Гистограмма распределения возрастов друзей пользователя, поступившего на вход

base_client.py:

```
import requests

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

    # Отправка запроса к VK API
    def _get_data(self):
        response = requests.get(self.BASE_URL + self.method + "." +
self.http_method, params = self.get_params())
        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()
```

ClientVK.py

```
import base_client
from datetime import datetime
debug = True

class ClientGetID(base_client.BaseClient):
    method = "users"
    # GET, POST, ...
    http_method = "get"
```

```

def __init__(self, username):
    self.username = username
    self.json_data = None

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

def response_handler(self, response):
    self.json_data = response.json()
    return self.json_data["response"][0]["uid"]

def get_json(self):
    return self.json_data

class ClientGetFriendsAges(base_client.BaseClient):
    # метод vk api
    method = "friends"
    # GET, POST, ...
    http_method = "get"

    def __init__(self, user_id):
        self.user_id = user_id
        self.json_data = None

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

    def calculate_age(self, born, today):
        return today.year - born.year - ((today.month, today.day) <
(born.month, born.day))

    def response_handler(self, response):
        self.json_data = response.json()
        ages = list()
        today = datetime.utcnow()
        print("0-дата рождения не указана или она неполная")
        for friend in self.json_data["response"]:
            date_tmp = friend.get("bdate")
            if date_tmp is None or len(date_tmp) < 6:
                ages.append(self.calculate_age( datetime.strptime(
"12.10.2017", "%d.%m.%Y"), today))
                continue
            ages.append( self.calculate_age( datetime.strptime( date_tmp,
"%d.%m.%Y"), today))
        return ages

    def get_json(self):
        return self.json_data

gist.py
class Gist ():
    # данные гистаграммы
    _data = None
    _data_sorting = dict()

    def __init__(self, date):
        self._data = date

```

```

        for number in date:
            self._data_sorting.update({number: self._data_sorting.get(number,
0) + 1})

    def print_hist(self):
        str_out = ""
        for age, stat in self._data_sorting.items():
            str_out += str(age).ljust(4) + str().ljust(stat, '#') + '\n'
        print(str_out)

```

run.py

```

from ClientVK import ClientGetID
from ClientVK import ClientGetFriendsAges
from gist import Gist

debug = True
username = "demonBogomolov"

get_id = ClientGetID(username).execute()
friends_ages = ClientGetFriendsAges(get_id).execute()

if debug:
    print("ID: ", get_id)
    print("Ages: ", friends_ages)

mygist = Gist(friends_ages)
mygist.print_hist()

```

Вывод

```

0-дата рождения не указана или она неполная
ID: 17862090
Ages: [0, 34, 0, 0, 29, 0, 0, 0, 0, 0, 0, 0, 18, 0, 21, 20, 0, 0, 0, 0, 20, 20, 0, 0, 0, 97, 0, 0, 23, 105,
0 #####
34 #
29 ##
18 #####
21 #####
20 #####
97 #
23 ##
105 #
17 ##
19 #####
33 #
53 #
99 #
24 #
14 #
27 ##
15 ####
32 #
115 #
112 #
87 #

```