Государственное образовательное учреждение высшего профессионального образования

«Московский государственный технический университет имени Н.Э. Баумана» (МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ Информатика и системы управления КАФЕДРА Системы обработки информации и управления

Отчёт по лабораторной работе по курсу «Разработка интернет-приложений» Классы в Python

Исполнитель:
студентка группы **РТ5-51**
Карасева А. Д.
Преподаватель: **Гапанюк Ю.Е.**

Москва, 2017

Содержание папки Main

```
from user import User
from friends import Friends
import matplotlib.pyplot as plt
import random

Age = []
Val = []
def gen_random(begin, end, num_count):
    for i in range(num_count):
         Age.append(random.randint(begin, end))
         Val.append(random.randint(begin, end))



def draw_graph(ages):
    graph = {}
    for age in ages:
        graph[age] = graph.get(age, 0) + 1

    graph = sorted(graph.items())
    for age, val in graph:
        Age.append(age)
        Val.append(val)
        print(age, '#' * val)




def main():
    username = input()

    try:
        uid = User(username).execute()
    except User.UserNotFound as e:
        e.msg()
        return


    try:
        ages = Friends(uid).execute()
    except Friends.FriendsNotFound as e:
        e.msg()
        return

    draw_graph(ages)
    #gen_random(1, 50, 50)
    #gen_random(1, 60, 70)
    fig, ax = plt.subplots()
    # add a 'best fit' line

    plt.bar(Age, Val, align='center')
    ax.set_xlabel('Age')
    ax.set_ylabel('value')
    ax.set_title('Histogram of ages of friends')
    plt.show()


    return
```

```
if __name__ == '__main__':
    main()
```

Содержание папки User

```
from base_client import BaseClient


class User(BaseClient):
    class UserNotFound(Exception):
        @staticmethod
        def msg():
            print('User not found')

    method = 'users'
    http_method = 'get'

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

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

    def response_handler(self, response):
        user = response.json().get('response')
        if user:
            return user[0]['uid']
        else:
            raise self.UserNotFound
```

Содержание папки Base_client

```
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
```

```
    # 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 = 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(
            self.method,
            http_method=self.http_method
        )
```

Содержание папки Friends

```
import datetime
from base_client import BaseClient


class Friends(BaseClient):
    class FriendsNotFound(Exception):
        @staticmethod
        def msg():
            print('Friends not found')

    method = 'friends'
    http_method = 'get'

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

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

    def response_handler(self, response):
        friends = response.json().get('response')
        if not friends:
            raise self.FriendsNotFound

        ages = []
        today = datetime.datetime.today()
        c = []

        for friend in friends:
            date = friend.get('bdate')

            try:
                dt = datetime.datetime.strptime(date, '%d.%m.%Y')
            except TypeError:
                continue
```

```
            except ValueError:
                continue

            age = today.year - dt.year
            if today.month < dt.month:
                age -= 1
            elif today.month == dt.month and today.day < dt.day:
                age -= 1

            ages.append(age)



    return ages
```

Реализация программы: