

**Московский государственный технический
университет им. Н.Э. Баумана.**

Факультет «Информатика и управление»

Кафедра ИУ5. Курс «Разработка интернет приложений»

Отчет по лабораторной работе №2

«Python-классы»

Выполнил:

студент группы ИУ5-52

Злобина С.В.

Проверил:

преподаватель каф.

ИУ5

Гапанюк Ю.Е.

Москва, 2017 г.

1 Задание

1. Знакомство с модулями и ООП в Python, работа с сетью.
2. Создать набор классов для реализации работы с VK API:

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 #
```

2 Листинг

base_client.py

class BaseClient:

```
# URL vk api
BASE_URL = None
# метод vk api
method = None
# GET, POST, ...
http_method = None
def __init__(self, base_url):
    self.BASE_URL = base_url
# 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
    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
    )

```

client_friends.py

from base_client **import** BaseClient

import requests

from datetime **import** datetime

from exception **import** HandlerException

class ClientFriends(BaseClient):

BASE_URL = 'https://api.vk.com/method/'

method = 'friends.get'

http_method = 'GET'

def __init__(self,
 user_id,
 order='random',
 list_id=None,
 count=None,
 offset=None,
 fields='bdate',
 name_case=None):

super().__init__(self.BASE_URL)

self.user_id = user_id

self.order = order

self.list_id = list_id

self.count = count

self.offset = offset

self.fields = fields

self.name_case = name_case

self.params = {}

self.ages_list = None

def _get_data(self, method, http_method):

response = requests.get(self.generate_url(method), self.get_params())

return self.response_handler(response)

def get_params(self):

self.params['user_id'] = self.user_id

```

self.params['order'] = self.order
if self.list_id is not None:
    self.params['list_id'] = self.list_id
if self.count is not None:
    self.params['count'] = self.count
if self.offset is not None:
    self.params['offset'] = self.offset
self.params['fields'] = self.fields
if self.name_case is not None:
    self.params['name_case'] = self.name_case
return self.params
def response_handler(self, response):
    if response.status_code != 200:
        raise HandlerException('response in ClientFriends with not 200 status')
    else:
        response_dict = response.json()
        self.ages_list = []
        for elem in response_dict.get('response'):
            try:
                bdate = datetime.strptime(elem['bdate'], '%d.%m.%Y')
                self.ages_list.append((datetime.today() - bdate).days // 365)
            except ValueError:
                continue
            except KeyError:
                continue
            except Exception:
                raise HandlerException('unknown exception while processing bdates')

```

exception.py

class HandlerException(Exception):

```

def __init__(self, value):
    self.value = value
def __str__(self):
    print(self.value)

```

client_id_from_username.py

from base_client **import** BaseClient

import requests

from exception **import** HandlerException

class ClientIdFromUsername(BaseClient):

BASE_URL = 'https://api.vk.com/method/'

method = 'users.get'

http_method = 'GET'

def __init__(self, user_id, fields=None, name_case=None):

super().__init__(self.BASE_URL)

self.user_id = user_id

self.fields = fields

self.name_case = name_case

self.params = {}

self.real_id = None

def get_params(self):

```

self.params['user_ids'] = self.user_id
if self.fields is not None:
    self.params['fields'] = self.fields
if self.name_case is not None:
    self.params['name_case'] = self.name_case
return self.params
def _get_data(self, method, http_method):
    response = requests.get(self.generate_url(method), self.get_params())
    print(response.url)
    return self.response_handler(response)
def response_handler(self, response):
    if response.status_code != 200:
        raise HandlerException('response in ClientIdFromUsername with not 200 status')
    else:
        self.real_id = response.json()['response'][0]['uid']

```

friends_ages.py

from client_friends import ClientFriends

from client_id_from_username **import** ClientIdFromUsername

#import matplotlib.pyplot as pyplot

client_by_username = ClientIdFromUsername('pakahontas')

client_by_username.execute()

print(client_by_username.real_id)

client = ClientFriends(client_by_username.real_id)

client.execute()

print(client.ages_list)

ages_dict = [0] * 120

for age **in** client.ages_list:

ages_dict[age] += 1

for number **in** range(110):

if ages_dict[number] > 0:

print(number, ': ', '#'*ages_dict[number])

pyplot.hist(client.ages_list, client.ages_list.len)

Результат

```
https://api.vk.com/method/users.get?user_ids=pakahontas
5249605
[24, 29, 24, 31, 29, 27, 24, 22, 26, 24, 20, 21, 22, 25, 22, 27, 23, 23, 26, 21, 24, 21, 31, 30, 30, 24, 18, 27, 21, 27, 23, 20, 27, 16,
15 : ###
16 : #####
17 : #####
18 : #####
19 : #####
20 : #####
21 : #####
22 : #####
23 : #####
24 : #####
25 : #####
26 : #####
27 : #####
28 : #####
29 : #####
30 : #####
31 : #####
32 : #####
33 : ##
34 : #
35 : #
36 : ##
37 : #
38 : #####
39 : #
41 : ##
42 : #
```

```
42 : #
44 : #
49 : ##
52 : #
62 : #
76 : #
```

Process finished with exit code 0