

**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
Заровная Н.А.**

Проверил:

**преподаватель каф. ИУ5
Гапанюк Ю.Е.**

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, method, http_method):
    self.BASE_URL = BASE_URL
    self.method = method
    self.http_method = http_method
```

```
    # 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

# Склеивка 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')
    # 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,
        self.http_method
    )

```

vk_client.py

```

import requests
from base_client import *
from exception import *
from datetime import datetime

class GetVkID(BaseClient):
    v = 5.68

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

# Получение GET параметров запроса
def get_params(self):
    return {'user_ids': self.user_id, 'v': self.v}

# Отправка запроса к VK API
def _get_data(self, method, http_method):
    response = requests.get(self.generate_url(method), self.get_params())
    return self.response_handler(response)

# Обработка ответа от VK API
def response_handler(self, response):
    try:
        response_dict = response.json()
        if response_dict.get('error') is not None:

```

```

        error = response_dict.get('error')
        error_msg = error.get("error_msg")
        raise APIException(error_msg)
    else:
        return response_dict.get('response')[0]
except APIException as ex:
    print(ex.err)

class GetVkFriends(BaseClient):
    v = 5.68
    fields = 'bdate'

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

    # Получение GET параметров запроса
    def get_params(self):
        return {'user_id': self.user_id, 'v': self.v, 'fields': self.fields}

    # Отправка запроса к VK API
    def _get_data(self, method, http_method):
        response = requests.get(self.generate_url(method), self.get_params())
        return self.response_handler(response)

    # Обработка ответа от VK API
    def response_handler(self, response):
        try:
            response_dict = response.json()
            if response_dict.get('error') is not None:
                error = response_dict.get('error')
                error_msg = error.get("error_msg")
                raise APIException(error_msg)
            else:
                return response_dict.get('response').get('items')
        except APIException as ex:
            print(ex.err)

```

exception.py

```

class APIException(Exception):
    def __init__(self, err):
        Exception.__init__(self)
        self.err = err

```

get_vkfriends.py

```

from vk_client import *
get_user = GetVkID('g_s_f_o')
user = get_user.execute()
user_id = user.get('id')
get_friends = GetVkFriends(user_id)
friends = get_friends.execute()
now = datetime.now()
ages = [0] * 1000

for fr in friends:
    try:
        date_str = fr.get('bdate')
        date = datetime.strptime(date_str, '%d.%m.%Y')
        days = (now - date).days
        age = days // 365
    except:
        pass

```

```

        ages[age] += 1
    except:
        pass

print(user.get("first_name"), user.get("last_name"))
for age in range(1000):
    if ages[age] != 0:
        print(age, " ", "#" * ages[age])

```

Результат

C:\Python37\python.exe C:/Users/АЛЬБЕРТ/PycharmProjects/lab3/get_vkfriends.py

Анастасия Соломатина

```

13  #
15  #
17  ###
18  #####
19  #####
20  #####
21  #####
22  #####
23  #####
24  #####
25  #####
26  ###
27  ####
28  ##
29  #
30  #
31  ##
32  ##
55  #
70  #
72  #
93  #
97  #
102 ##

```