

Защищено:
Гапанюк Ю.Е.

"__" _____ 2016 г.

Демонстрация ЛР:
Гапанюк Ю.Е.

"__" _____ 2016 г.

**Отчет по лабораторной работе № 3 по курсу
Разработка интернет приложений**

**"Лабораторная работа №3.
Python-классы."**

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-52

Смирнов А. И.

(подпись)

"__" _____ 2016 г.

Краткое описание:

В этой ЛР Вы знакомитесь с модулями и ООП в Python, а также осваиваете работу с сетью.

В лабораторной работе необходимо создать набор классов для реализации работы с VK API

Задание:

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 #
```

Текст программы:

Base_client.py:

```
class BaseClient:
    BASE_URL = None

    method = None
    http_method = None

    def get_params(self):
        pass

    def get_json(self):
        pass

    def get_headers(self):
        pass

    def generate_url(self, method):
        return '{0}{1}'.format(self.BASE_URL, method)
```

```

def _get_data(self, method, http_method):
    response = None

    # todo выполнить запрос

    return self.response_handler(response)

def response_handler(self, response):
    return response

def execute(self):
    return self._get_data(
        self.method,
        http_method=self.http_method
    )

```

Client.py:

```

import requests, json

from base_client import BaseClient

class Client(BaseClient):
    BASE_URL = 'https://api.vk.com/method/'
    user_name = ''
    user_id = ''
    payload = {}

    def __init__(self):
        self.user_name = input('Введите id пользователя: ')
        self.payload = {'v': '5.57', 'user_ids': self.user_name}
        self.user_id = self.get_id()

    def get_id(self):
        resp = self._get_data('users.get', self.payload)
        return resp['response'][0]['id']

    def _get_data(self, method, param):
        response = requests.get(self.generate_url(method), params=param)
        response = response.json()
        return self.response_handler(response)

    def response_handler(self, response):
        if 'error' not in response:
            return response
        else:
            raise NameError

    def execute(self):
        return self.user_id

```

Client_friends.py:

```

from client import *
import datetime

class ClientFriends(Client):

```

```

today = datetime.datetime.toordinal(datetime.datetime.now())
age_friends = {}

def __init__(self, user_id):
    self.user_id = user_id
    self.payload = {'v': '5.57', 'fields': 'bdate', 'order': 'random',
'user_id': self.user_id}

def get_friends(self):
    resp = self._get_data('friends.get', self.payload)
    for i in range(len(resp['response']['items'])):
        if 'bdate' in resp['response']['items'][i]:
            if len(resp['response']['items'][i]['bdate']) >= 8:
                bdate =
datetime.datetime.strptime(resp['response']['items'][i]['bdate'], '%d.%m.%Y')
                bdate = datetime.datetime.toordinal(bdate)
                age = int((self.today - bdate)//365.25)
                if self.age_friends.get(age) == None:
                    self.age_friends[age] = '#'
                else:
                    self.age_friends[age] += '#'
    return self.age_friends

def print_friends(self):
    ages = self.get_friends()
    for i in range(max(ages.keys())+1):
        if ages.get(i) == None:
            continue
        else:
            print(i, ': ', ages.get(i))

```

Main.py:

```

from client import Client
from client_friends import ClientFriends

user = Client()
user_friends = ClientFriends(user.execute())
user_friends.print_friends()

```

Результаты работы программы:

```
Run main
E:\Python\python.exe E:/Проекты/Python/Лаб3/main.py
Введите id пользователя: durov
20 : #####
21 : #####
22 : #####
23 : #####
24 : #####
25 : #####
26 : #####
27 : #####
28 : #####
29 : #####
30 : #####
31 : #####
32 : #####
33 : #####
34 : #####
35 : #####
36 : #####
37 : ##
38 : #####
39 : ###
40 : ###
41 : ###
42 : ###
43 : ##
44 : #
45 : ##
46 : ##
50 : #
51 : ###
52 : #
53 : #
55 : #
62 : #

Process finished with exit code 0
```