

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

(количество листов)

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-52

Матвеева П.Р.

(подпись)

"__" _____ 2016 г.

Москва, МГТУ - 2016

Оглавление

1. Описание задания лабораторной работы.	3
2. Модули.	3
3. Результаты работы.	5

1. Описание задания лабораторной работы.

Знакомство с модулями и ООП в Python, а также освоение работы с сетью.

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

2. Модули.

Base.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

```
from Base import BaseClient
import requests
import json

class Client(BaseClient):
    BASE_URL = 'https://api.vk.com/method/users.get'
    http_method = 'GET'

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

    def get_params(self):
        return 'user_ids=' + self.name

    def response_handler(self, response):
        try:
            return json.loads(response.text).get('response')[0].get('uid')
        except:
```

```

        raise Exception("Запрос не может быть обработан")

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

        response = requests.get(self.BASE_URL + '?' + self.get_params())
        return self.response_handler(response)

```

Friends.py

```

from Base import BaseClient
import requests
import json
from datetime import datetime

class Friends(BaseClient):
    BASE_URL = 'https://api.vk.com/method/friends.get'
    http_method = 'GET'

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

    def get_params(self):
        return 'user_id=' + str(self.user_id) + '&fields=bdate'

    def response_handler(self, response):
        try:
            friends = json.loads(response.text).get('response')

            ages = []

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

                if birth_date is None or len(birth_date) < 8:
                    continue

                birth_date = datetime.strptime(birth_date, "%d.%m.%Y")
                cur_date = datetime.now()

                ages.append(int((cur_date - birth_date).days) // 365.2425)

            ages_ = list(set(ages))
            return sorted([(x, ages.count(x)) for x in ages_])
        except:
            raise Exception("Запрос не может быть обработан")

    def _get_data(self, method, http_method):
        response = requests.get(self.BASE_URL + '?' + self.get_params())
        return self.response_handler(response)

```

Request.py

```

from Client import Client
from Friends import Friends

u_id = Client('id6684003').execute()
friends = Friends(u_id).execute()

for (age, count) in friends:
    print('{} {}'.format(int(age), '#' * count))

```

3. Результаты работы.

```
18 ###
19 #####
20 #####
21 #####
22 #####
23 ###
24 #####
25 #####
26 #####
27 #####
28 ##
29 #
30 #####
31 ##
32 ##
33 #
36 #
37 #
46 #
67 #
115 #
```