

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

Факультет «Информатика и системы управления»

Кафедра ИУ5. Курс «Разработка интернет приложений»

**Отчет по лабораторной работе №2
«Python-классы»**

Выполнил:

студент группы ИУ5-52

Дудырев Е. О.

Подпись и дата:

Проверил:

доцент каф. ИУ5

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2017 г.

Оглавление

Исходный код.....	3
base_client.py	3
lab_classes.py	4
app.py.....	6
Результаты работы.....	7

Исходный код

base_client.py

```
class BaseClient:

    # URL vk api
    BASE_URL = None

    # 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 = None

        # 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,

        http_method=self.http_method

    )

```

lab_classes.py

```

from base_client import *

import requests

import json

import datetime


class VK_User(BaseClient):

    def __init__(self, username):

        self.name = username


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

method = 'users.get'

http_method = 'GET'


# Отправка запроса к VK API

def _get_data(self, method, http_method):

    url=self.generate_url(method)

    response = requests.get(url=url+'?user_ids='+self.name)

    return self.response_handler(response)


def response_handler(self, response):

    js = json.loads(response.text)

```

```

try:
    return js['response'][0]['uid']
except:
    raise Exception("There's no such user")

```

```
class VK_Friends(BaseClient):
```

```

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

```

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

```
method = 'friends.get'
```

```
http_method = 'GET'
```

```
# Отправка запроса к VK API
```

```

def _get_data(self, method, http_method):
    url=self.generate_url(method)
    response = requests.get(url=url+'?user_id='+str(self.id)+'&fields=bdate')
    return self.response_handler(response)

```

```
def response_handler(self, response):
```

```
    js = json.loads(response.text)
```

```
    try:
```

```
        dates = []
```

```
        for friend in js['response']:
```

```
            try:
```

```
                dates.append(datetime.datetime.strptime(friend['bdate'], "%d.%m.%Y").date())
```

```
            except:
```

```
                pass
```

```
    ages = [int((datetime.date.today()-date).days/365.2425) for date in dates]
```

```
    return ages
```

```
except KeyError:
```

```
return KeyError(js)
```

app.py

```
from lab_classes import *

from matplotlib import pylab

#user_id = VK_User('56337756').execute()

name = input('Type username or id: ')

print()

user_id = None

try:
    user_id = VK_User(name).execute()
except Exception as e:
    print(e)
    quit(1)

ages= None

try:
    ages = VK_Friends(user_id).execute()
except Exception as e:
    print(e)
    quit(1)

hist = [(x,ages.count(x)) for x in set(ages)]

hist.sort(key=lambda x: x[0])

for age, number in hist:
    print(age,'#'*number)

pylab.hist(ages,bins=((max(ages)-min(ages))))

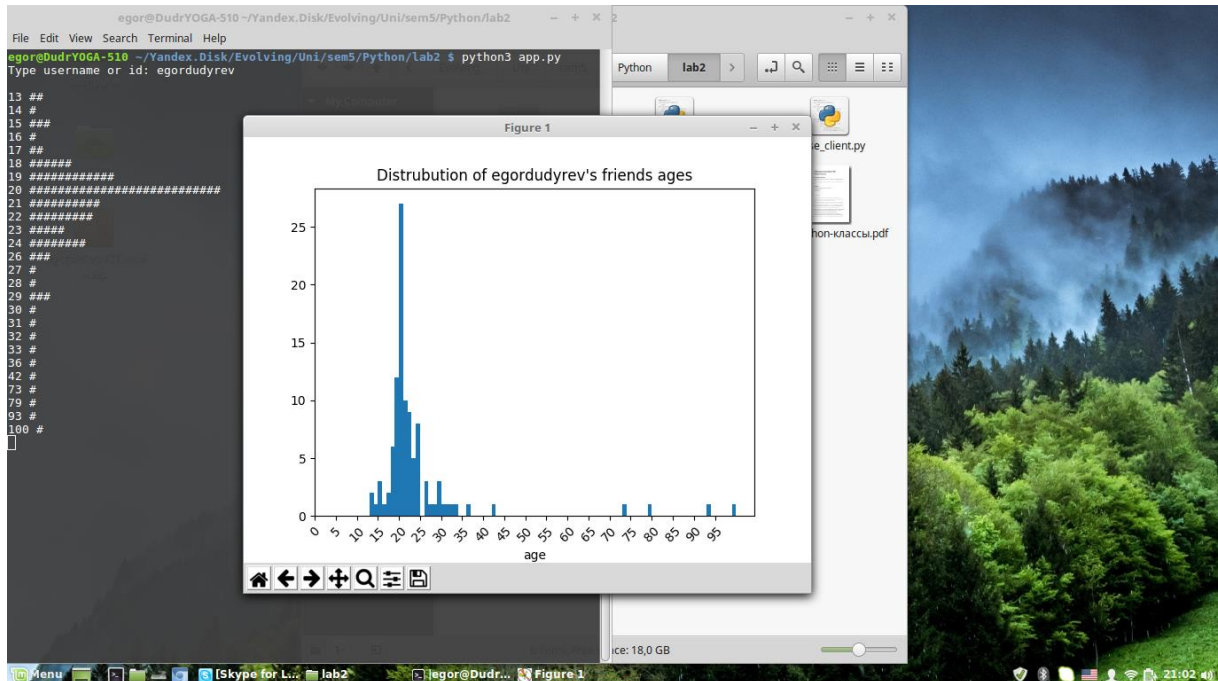
pylab.xlabel('age')

pylab.xticks(range(0,max(ages),5),rotation=45)

pylab.title('Distrubution of { }\s friends ages'.format(name))

pylab.show()
```

Результаты работы



Distrubution of egordudyrev's friends ages

