

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

**Кафедра ИУ5 «Системы обработки информации и  
управления»**

**Отчет по лабораторной работе №3**

**«Python-классы»**

**по дисциплине «Разработка Интернет-приложений»**

**Выполнил: студент  
группы ИУ5-54  
Виноградов Андрей**

**Москва, 2016**

## 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 #
```

## baseclient.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
    )

```

## id\_from\_username.py

```

from baseclient import BaseClient
import requests
import json

```

```

class IdFromUsername(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:
            uobj = json.loads(response.text)
            return uobj.get('response')[0].get('uid')
        except:
            raise Exception("Couldn't handle response for username
{}".format(self.name))

    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 baseclient 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, uid):
        self.uid = uid

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

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

```

```

ages = []

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

    if bdate is None or bdate.count('.') < 2: continue

    bdate = datetime.strptime(bdate, "%d.%m.%Y") ndate =
    datetime.now()

    ages.append(int((ndate - bdate).days) // 365.2425)

uniqages = list(set(ages))
return sorted([(x, ages.count(x)) for x in uniqages], key=lambda
x: x[0])
except:
    raise Exception("Couldn't handle response for friends of uid
{}".format(self.uid))

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

```

vk.py

```

from id_from_username import IdFromUsername from
friends import Friends
import matplotlib.pyplot as plot

uname = input('Имя пользователя: ') uclient
= IdFromUsername(uname)
uid = uclient.execute()

friends_client = Friends(uid) friends =
friends_client.execute()

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

fig, ax = plot.subplots()
rects = ax.bar(list(map(lambda x: x[0], friends)), list(map(lambda x: x[1], friends)), 1)

plot.xlabel('Возраст')
plot.ylabel('Кол-во')
plot.title('Распределение друзей по возрастам')
plot.subplots_adjust()
plot.show()

```

Скриншот с результатом:

```
14 #
18 #####
19 #####
20 #####
21 #####
22 ###
25 ##
27 ##
29 ##
40 #
78 #
102 #
110 #
```