

Защищено:  
Гапанюк Ю. Е.

Демонстрация ЛР:  
Гапанюк Ю. Е.

"\_\_" \_\_\_\_\_ 2016 г.

"\_\_" \_\_\_\_\_ 2016 г.

## **Отчет по лабораторной работе №3 «Python-классы»**

ИСПОЛНИТЕЛЬ:  
студент группы ИУ5-52

\_\_\_\_\_  
(подпись)

Лунев М. А.

"\_\_" \_\_\_\_\_ 2016 г.

# 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 #
```

## Листинг

base\_client.py

```
import requests
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):
        payload = self.get_params()
        response = requests.get(self.BASE_URL, params=payload)
        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
        )
```

## friend\_age.py

```
from base_client import BaseClient
from datetime import datetime
import requests, json
class FriendsAge(BaseClient):
    BASE_URL = 'https://api.vk.com/method/friends.get'
    http_method = 'GET'
    def __init__(self, userId):
        self.userId = userId
    def get_params(self):
        return {
            'user_id': self.userId ,
            'fields': 'bdate'
        }
    def response_handler(self, response):
        try:
            parsed = json.loads(response.text)
            list = parsed.get('response')
            ages = []
            for person in list:
                bdate = person.get('bdate')
                if bdate is not None:
                    try:
                        date = datetime.strptime(bdate, "%d.%m.%Y")
                        now = datetime.now()
                        ages.append(int((now - date).days / 365))
                    except ValueError:
                        continue
            return ages
        except:
            return None
```

## id\_from\_username.py

```
from base_client import BaseClient
import requests, json
class IDFromUsername(BaseClient):
    BASE_URL = 'https://api.vk.com/method/users.get'
    http_method = 'GET'
    def get_params(self):
        return {
            'user_ids': self.name
        }
    def __init__(self, name):
        self.name = name
    def response_handler(self, response):
        parsed = json.loads(response.text)
        try:
            return parsed.get('response')[0].get('uid')
        except:
            return -1
```

## iterators.py

```
# Итератор для удаления дубликатов
class Unique(object):
    def __init__(self, items, **kwargs):
        self.items = list(items)
        self.max_size = len(self.items)
        self.index = 0
        self.set = set()
```

```

try:
    self.ignore_case = kwargs['ignore_case']
except:
    self.ignore_case = False
def __next__(self):
    if self.index == self.max_size:
        raise StopIteration
    item = self.items[self.index]
    if self.ignore_case is True:
        item = item.lower()
    while item in self.set:
        self.index = self.index + 1
        if self.index == self.max_size:
            raise StopIteration
        item = self.items[self.index]
        if self.ignore_case is True:
            item = item.lower()
    self.set.add(item)
    return item
def __iter__(self):
    return self

```

main.py

```

from friends_age import FriendsAge
from id_from_username import IDFromUsername
import matplotlib.pyplot as plt
input = 1 #input()
getID = IDFromUsername(input)
ID = getID.execute()
getAges = FriendsAge(ID)
ages = getAges.execute()
max_age = max(ages)
osY = [0] * (max_age + 1)
for age in ages:
    osY[age] += 1
osX = range(max_age + 1)
plt.axis([0, max_age, 0, max(osY)])
plt.xlabel('Age')
plt.ylabel('Friends')
plt.plot(osX, osY)
plt.show()

```

Проверка

