

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

Факультет «Информатика и управление»

Отчёт по лабораторной работе №3
по курсу «Разработка интернет-приложений»

«Python - классы»

Выполнила:

студентка группы ИУ5-52

Горбовцова К.М.

Подпись и дата:

Проверил:

преподаватель каф. ИУ5

Подпись и дата:

Москва, 2017 г.

Цель работы: Изучение модулей и ООП в Python, работы с сетью. Необходимо создать набор классов для реализации работы с VK API.

Код программы:

BaseClient.py – базовый класс

```
class BaseClient:

    # URL vk api
    BASE_URL = "https://api.vk.com/method/"
    # 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
        )
```

NewMyClass.py – классы для получения ID пользователя и формирования списка друзей

```
from BaseClient import BaseClient
import requests
```

```
class GetId(BaseClient):
    BASE_URL = "https://api.vk.com/method/"
    method = 'users.get'
    http_method = '?user_ids='
    username = None
    """vk_id = None"""

    def __init__(self):
        self.username = raw_input("Input user's name ")

    def generate_url(self):
        return '{0}{1}{2}{3}'.format(self.BASE_URL, self.method, self.http_method,
self.username)
```

```
    def get_data(self):

        r = requests.get(self.generate_url()).json()
        try:
            if r["response"][0]["deactivated"]:
                print("this page was banned or not exist")
                exit()
        except Exception:
            return r["response"][0]["uid"]
```

```
class GetFriend(BaseClient):
    def __init__(self, vk_id):
        self.vk_id = vk_id
        self.get_data()

    BASE_URL = "https://api.vk.com/method/"
    method = 'friends.get'
    http_method = '?uid='
    fields = "&fields=bdate"

    def generate_url(self):
        return '{0}{1}{2}{3}{4}'.format(self.BASE_URL, self.method, self.http_method,
self.vk_id, self.fields)

    def get_data(self):

        r = requests.get(self.generate_url()).json()
        return r["response"]

    def bdayArr(self):
```

```

new_arr = []
for friend in self.get_data():
    try:
        if len(friend["bdate"]) >= 8:
            new_arr.append(friend["bdate"])

    except Exception:
        continue
return new_arr

```

func.py – функции для вычисления возраста и построения гистограммы

```

import datetime
from datetime import timedelta
import matplotlib.pyplot as plt
import time

```

```

def count_bdate(bdate_list):
    ages = []
    curr_date = datetime.date.today()
    for i in bdate_list:
        tempd = i.split(".")
        my_date = datetime.date(int(tempd[2]), int(tempd[1]), int(tempd[0]))
        delta = ((curr_date - my_date).total_seconds()) / 365 / 24 / 3600

        ages.append(int(delta))

    return ages

```

```

def histog(array):
    plt.hist(array)
    plt.show()

```

main.py

```

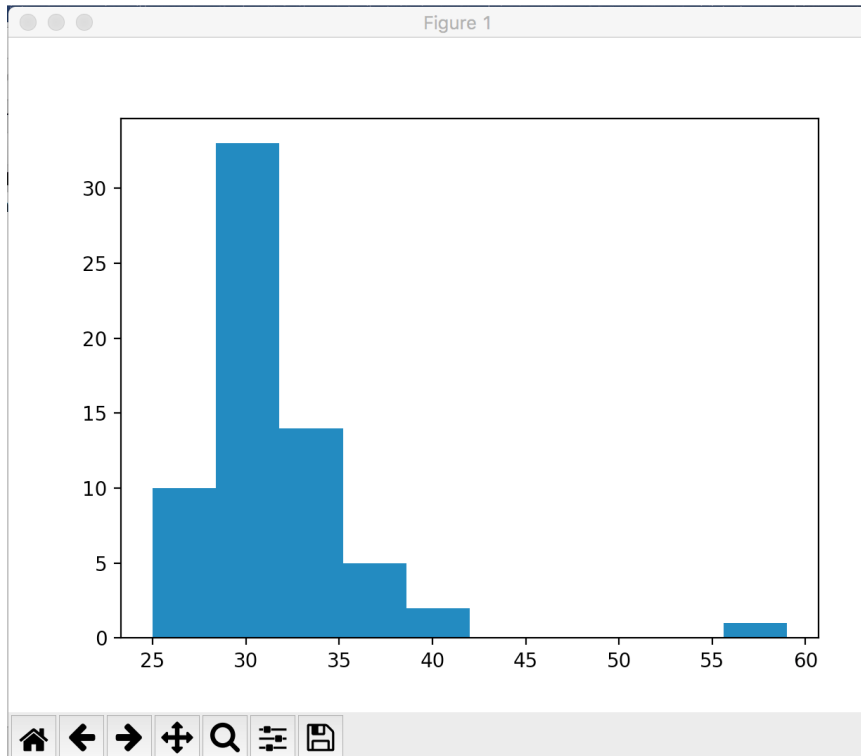
from func import *
from NewMyClass import *
histog(
    count_bdate(
        GetFriend(
            GetId().get_data()
        ).bdayArr()
    )
)

```

Входные данные:

```
/Users/GorbovtsovaKsenya/anaconda/bin/python /Users/GorbovtsovaKsenya/Desktop/lab3.1/main.py
Input user's name ksenya21
```

Результат работы программы:



Входные данные:

```
/Users/GorbovtsovaKsenya/anaconda/bin/python /Users/GorbovtsovaKsenya/Desktop/lab3.1/main.py
Input user's name 1231234
```

Результат работы программы:

```
this page was banned or not exist
```

```
Process finished with exit code 0
```