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

Кафедра «Системы обработки информации и управления»

---

# «Лабораторная работа №3 »

Лабораторная работа

(вид документа)

писчая бумага формата А4

(вид носителя)

(количество листов)

Исполнитель: студентка группы РТ5-51

_____ Коньшин К.И.

"_____"_____ 2016 г.

Москва – 2016

---

# Задание и порядок выполнения:

## Вход:

username или vk_id пользователя

## Выход:

Гистограмма распределения возрастов друзей пользователя, поступившего на вход

## Код класса наследника:

```
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
        )
```

## Код основной программы:

```
import requests
import json
import time
from collections import Counter

class vk_api(BaseClient):
    user_id,params,username = None,'',None
    age = []

    def __init__(self, username):
        self.BASE_URL = "http://api.vk.com/method/"
        self.http_method = "GET"
        self.username = username

    def get_data(self,method,params):
        self.method = method
        r = requests.get(self.generate_url(self.method),params)
        data = r.json()
        return data

    def get_params(self, **kwargs):
        for key in kwargs:
            self.params = self.params+"&"+key+"="+kwargs[key]
        return self.params

    def get_id(self):
        self.user_id =
self.get_data("users.get",self.get_params(uids=self.username,v='3.0'))["response"][0]["uid"]
        return self.user_id

    def get_friends(self):
        friends =
self.get_data("friends.get",self.get_params(user_id=str(self.user_id),fields="bdate",v='3.0'))["res
ponse"]
        for friend in friends:
            if friend.get("bdate"):
                date = friend["bdate"].split(".")
                if len(date)>2:
                    self.age.append(int((int(time.time())-int(time.mktime(time.strptime(friend["bdate"],
'%d.%m.%Y'))))/31536000))
        self.age = dict(Counter(self.age))
        return self.age

    def print_sharp(self,count):
        sharp = '';
        for i in range(count):
            sharp = sharp + "#"
        return sharp
```

```
    def print_age(self):
        for i in self.age:
            print("Age {} count {}".format(i,self.print_sharp(self.age[i])))

    def exe(self):
        self.get_id()
        self.get_friends()
        self.print_age()
```

## Вывод:

```
Age 14 count ###
Age 15 count #
Age 16 count #
Age 17 count ##
Age 18 count ########
Age 19 count ##################
Age 20 count ##########
Age 21 count ###
Age 22 count ###
Age 23 count #
Age 24 count #
Age 25 count ###
Age 26 count #
Age 29 count ##
Age 35 count #####
Age 47 count #
Age 114 count ###
```