

**Лабораторная работа № 3**  
**по курсу "Разработка интернет приложений"**  
**Основы python**

**Выполнил:**  
**Студент III курса**  
**группы ИУ5-53**  
**Иванников Александр**

# 1. Задание

## Вход:

username или vk\_id пользователя

## Выход:

Гистограмма распределения возрастов друзей пользователя, поступившего на вход

## Условия:

Для реализации необходимо наследоваться от базового класса base\_client.py

## Итоговый результат:

Согласно указаниям, исходный результат состоит из трёх файлов inheritor\_class.py — созданный класс, main.py , base\_client.py — базовый класс.

## Листинг файла inheritor\_class.py:

```
from base_client import BaseClient
from collections import Counter
from datetime import date, datetime
import requests

class MyClient(BaseClient):
    BASE_URL = 'https://api.vk.com/method/'
    user_id = None
    user_friends = None

    def _get_data(self, method, http_method):
        response = requests.get(self.generate_url(method))
        return self.response_handler(response)

    def __init__(self, user_name):
        self.user_id = self._get_id(user_name)
        self.user_friends = self._get_friends(self.user_id)

    def _get_id(self, user_name):
        self.method = 'users.get?user_ids=' + str(user_name) + '&v=5.57'
        try:
            user_id = self.get_json(self.execute())['response'][0].get('id')
            return user_id
        except Exception as ex:
            raise Exception('User not found')
```

```

def _get_friends(self, user_id):
    self.method = 'friends.get?user_id=' + str(user_id) + '&fields=bdate&v=5.57'
    try:
        user_friends = self.get_json(self.execute())['response']['items']
        return user_friends
    except Exception as ex:
        raise Exception('Friends not found')

def _get_age(self, friend_bdate):
    try:
        day = int(friend_bdate[0])
        month = int(friend_bdate[1])
        year = int(friend_bdate[2])
    except KeyError:
        raise Exception('Incorrect bdate format')
    bdate = date(year, month, day)
    today = date.today()
    age = today.year - bdate.year
    if today.month < bdate.month:
        age -= 1
    elif today.month == bdate.month and today.day < bdate.day:
        age -= 1
    return age

def get_age_list(self):
    user_age_list = []
    for friend in self.user_friends:
        if friend.get('bdate'):
            bdate = friend['bdate'].split('.')
            if len(bdate) == 3:
                age = self._get_age(bdate)
                user_age_list.append(age)
    user_age_list = Counter(user_age_list)
    return user_age_list

```

## Листинг файла main.py:

```

from inheritor_class import MyClient
import matplotlib.pyplot as plt

user_name = input('insert username or id \n')
obj = MyClient(user_name)
age_list = obj.get_age_list()
keys = list(age_list.keys())
keys.sort()
for i in keys:
    print(i, ': ', '#' * age_list[i])

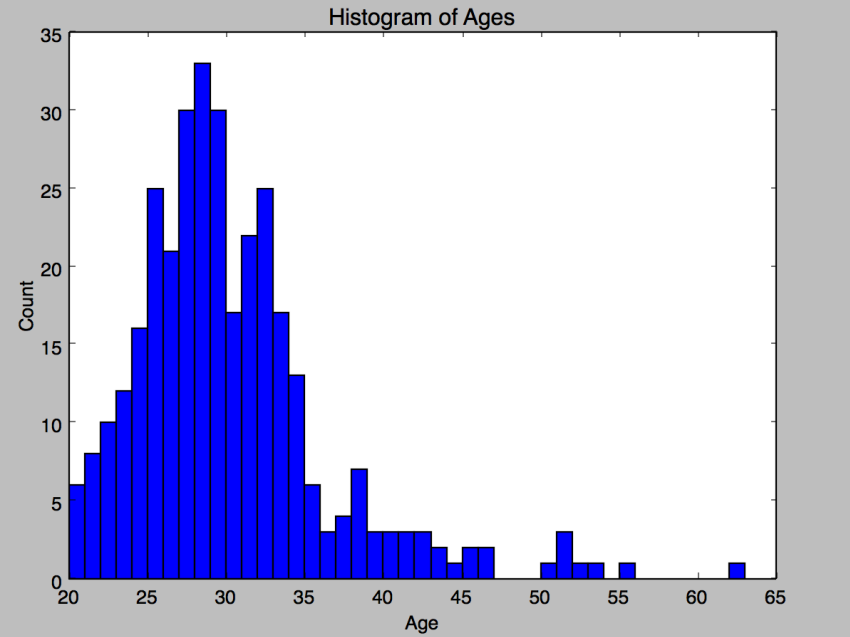
fig, ax = plt.subplots()
rects1 = ax.bar(age_list.keys(), age_list.values(), 1)

plt.xlabel('Age')
plt.ylabel('Count')
plt.title('Histogram of Ages')

plt.subplots_adjust()
plt.show()

```

Вывод:



```
1
(20, ':', '#####')
(21, ':', '#####')
(22, ':', '#####')
(23, ':', '#####')
(24, ':', '#####')
(25, ':', '#####')
(26, ':', '#####')
(27, ':', '#####')
(28, ':', '#####')
(29, ':', '#####')
(30, ':', '#####')
(31, ':', '#####')
(32, ':', '#####')
(33, ':', '#####')
(34, ':', '#####')
(35, ':', '#####')
(36, ':', '###')
(37, ':', '###')
(38, ':', '#####')
(39, ':', '###')
(40, ':', '###')
(41, ':', '###')
(42, ':', '###')
(43, ':', '##')
(44, ':', '#')
(45, ':', '##')
(46, ':', '##')
(50, ':', '#')
(51, ':', '###')
(52, ':', '#')
(53, ':', '#')
(55, ':', '#')
(62, ':', '#')
```