Согласовано:
Гапанюк Ю.Е.

"__"_____2016 г.

Утверждаю:
Гапанюк Ю.Е.

«__»__

Лабораторная работа №3 по курсу

Разработка интернет приложений

ИСПОЛНИТЕЛЬ:

студент группы ИУ5-53
Семенова Е.В.

_____

"07" октября 2016 г.

Москва - 2016

Base_Client_Mod.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

        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
        )
```

Get_User_Id_Mod.py

```
import requests
from Base_Client_Mod import BaseClient
import sys

class GetUserId:

        BASE_URL = 'https://api.vk.com/method/'
        method = None

        def __init__(self, identifier):
                self.method = 'users.get?user_ids={0}&v=V'.format(identifier)


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

```
        def _get_data(self, method):
                response = None

                response = requests.get(self.generate_url(method))

                return self.response_handler(response)

        def response_handler(self, response):

                if response.json().get("response") is None or len(response.json().get("response")) ==
0:
                        return []
                return response.json().get("response")[0].get("uid")

        def execute(self):
                return self._get_data(self.method)
```

Get_Friends_Bdate_Mod.py

```
import requests
from Base_Client_Mod import BaseClient
import datetime

class GetFriendsBdate(BaseClient):

        BASE_URL = 'https://api.vk.com/method/'
        method = None

        def __init__(self, ID):
                self.method = 'friends.get?user_id={0}&fields=bdate&v=V'.format(ID)

        def generate_url(self, method):
                return '{0}{1}'.format(self.BASE_URL, method)

        def _get_data(self, method):
                response = None

                response = requests.get(self.generate_url(method))

                return self.response_handler(response)

        def response_handler(self, response):
                bDateList = []

                now = datetime.datetime.now()

                if response.json().get("response") is None:
                        return []

                else:
                        for x in response.json().get("response"):
                                if(x.get("bdate") != None):
```

```
                                        if(len(x.get("bdate").split(".")) == 3):
                                                bDateList.append(now.year -
int(x.get("bdate").split(".")[2], 10))
                        return bDateList

        def execute(self):
                return self._get_data(self.method)
```

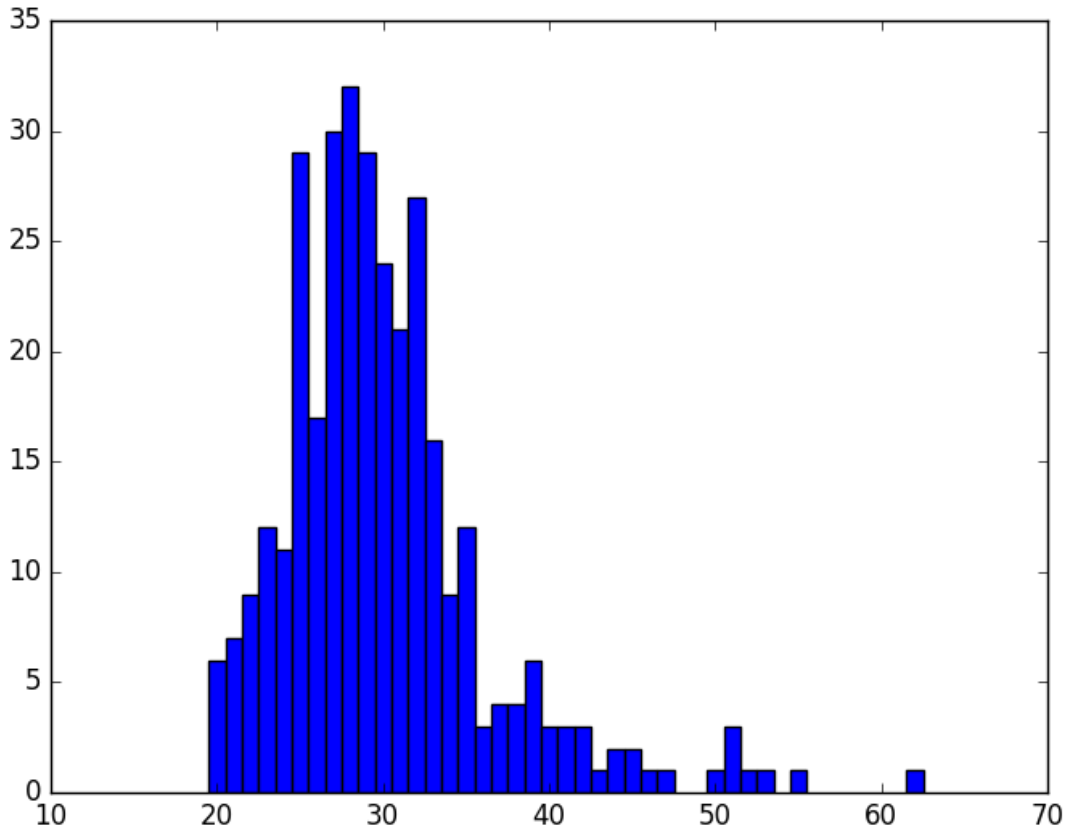
main.py

```
import matplotlib.pyplot as plt
import numpy as np
import sys

from Get_Friends_BDate_Mod import GetFriendsBdate
from Get_User_Id_Mod import GetUserId

x = input()

getId = GetUserId(x)
ID = getId.execute()

getBDate = GetFriendsBdate(ID)
bDateList = getBDate.execute()

List = {}
for x in bDateList:
        if x not in List:
                List[x] = 1
        else:
                List[x] += 1

List2 = List.copy()

print()
while(len(List2) != 0):
        min = 100
        for x in List2.keys():
                if x < min:
                        min = x
        print(min, end="")
        for i in range(List2[min]):
                print("#", end="")
        print()
        List2.pop(min)

x= []
y = []
for i in List.keys():
        x.append(i - 0.5)
        y.append(List[i])
width = 1;
```

```
locs = x
width = 1
plt.bar(locs, y, width=width)
plt.show()
```

Результаты работы программы:

```
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#
47#
50#
51###
52#
53#
55#
62#
```


35
30
25
20
15
10
5
0
10
20
30
40
50
60
70