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

**Факультет «Информатика и управление»**

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

«Python - классы»

Выполнил: Проверил:

студент группы ИУ5-52 преподаватель каф. ИУ5

Сергеев В.Р.

Подпись и дата: 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

)

getID.py – классы для получения ID пользователя

**from BaseClient import BaseClient**

**import requests**

**import sys**

**class GetUserID(BaseClient):**

**def \_\_init\_\_(self, username):**

**self.username = username**

**method = "users.get"**

**http\_method = "?user\_ids="**

**def generate\_url(self):**

**return '{0}{1}{2}'.format(BaseClient.generate\_url(self),self.http\_method,self.username)**

**def get\_data(self):**

**r = requests.get(self.generate\_url()).json()**

**try:**

**if r["response"][0]["deactivated"]:**

**print("Такого ID не существует")**

**sys.exit()**

**except Exception:**

**return r["response"][0]["uid"]**

getFriends.py- классы для получения ID пользователя и формирования списка друзей

**from BaseClient import BaseClient**

**import requests**

**class Friends(BaseClient):**

**def \_\_init\_\_(self, vk\_id):**

**self.vk\_id = vk\_id**

**method = "friends.get"**

**http\_method = "?uid="**

**fields = "&fields=bdate"**

**def generate\_url(self):**

**return '{0}{1}{2}{3}'.format(BaseClient.generate\_url(self), self.http\_method, self.vk\_id, self.fields)**

**def get\_data(self):**

**try:**

**response = requests.get(self.generate\_url()).json()**

**return response["response"]**

**except Exception:**

**print ('Такого ID не существует')**

**exit()**

**def make\_list(self):**

**t\_list = []**

**for i in self.get\_data():**

**try:**

**if len(i["bdate"]) >= 8:**

**t\_list.append(i["bdate"])**

**except Exception:**

**continue**

**return t\_list**

Make\_gist.py – функции для вычисления возраста и построения гистограммы

**import datetime**

**import matplotlib.pyplot as plt**

**def count\_bdate(bdate\_list):**

**age = []**

**curr\_date = datetime.date.today()**

**for i in bdate\_list:**

**t\_date = i.split(sep='.')**

**if int(curr\_date.month) > int(t\_date[1]):**

**age.append(int(curr\_date.year) - int(t\_date[2]))**

**elif int(curr\_date.month) == int(t\_date[1]):**

**if int(curr\_date.day) > int(t\_date[2]):**

**age.append(int(curr\_date.year) - int(t\_date[2]))**

**else:**

**age.append(int(curr\_date.year) - int(t\_date[2]) - 1)**

**elif int(curr\_date.month) < int(t\_date[1]):**

**age.append(int(curr\_date.year) - int(t\_date[2]) - 1)**

**return age**

**def gist(ages\_list):**

**plt.hist(ages\_list)**

**plt.show()# -\*- coding: utf-8 -\*-**

main\_file.py

**from getID import \***

**from getFriends import \***

**from Gist import \***

**ClientID = input("Введите ID: \n")**

**user = GetUserID(ClientID)**

**user\_id = user.get\_data()**

**friend = Friends(user\_id)**

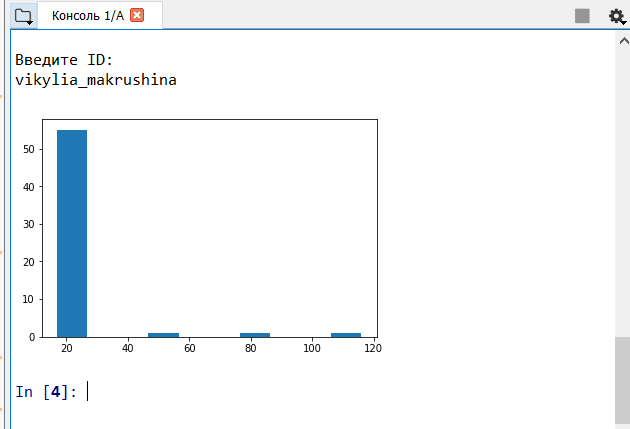
**friend\_list = friend.make\_list()**

**ages = count\_bdate(friend\_list)**

**fr\_gist = gist(ages)**

Входные данные: vikylia\_makrushina

Результат работы программы:



Входные данные: 1212121

Результат работы программы:

