Московский государственный технический университет им. Н.Э. Баумана

Факультет «Информатика и системы управления»

Кафедра «Автоматизированные системы обработки информации и управления»



**Отчет по лабораторной работе № 3**

**«Python-классы»**

**по курсу**

**“Разработка Интернет-приложений”**

**ИСПОЛНИТЕЛЬ:**

Березин И.С.

Группа ИУ5-53

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2016 г.

**ПРЕПОДАВАТЕЛЬ:**

Гапанюк Ю.Е.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"\_\_"\_\_\_\_\_\_\_\_\_\_\_2016 г.

Москва 2016

**Задание**

Вход:

username или vk\_id пользователя

Выход:

Гистограмма распределения возрастов друзей пользователя, поступившего на вход

Пример: Вход: reigning

Выход:

19 #

20 ##

21 ##

22 ####################

23 ################

24 ####

25 #

28 #

29 #

30 #

37 #

38 ##

45 #

lab3\_0.py

**class** BaseClient:  
 BASE\_URL = **None** method = **None** http\_method = **None  
  
 def** get\_params(self):  
 **pass  
  
 def** get\_json(self, response):  
 **return** response.json()  
  
 **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  
 )

lab3\_1.py

**from** lab3\_0 **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

lab3\_2.py

**from** lab3\_1 **import** MyClient  
**import** matplotlib.pyplot **as** plt  
  
user\_name = input(**'Введите id пользователя\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(**'The histogram of Ages'**)  
  
plt.subplots\_adjust()  
plt.show()

*Результаты выполнения:*

 