Московский Государственный Технический Университет им. Н. Э. Баумана

Факультет «Информатика и системы управления»

**Отчет по лабораторной работе № 3**

**Курс «Разработка интернет-приложений»**

|  |  |
| --- | --- |
| Выполнил: |  |
| студент группы ИУ5-54 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  | (подпись) |
| Харлашкин А. И. | "\_\_"\_\_\_\_\_\_\_\_\_\_2016 г. |

Проверил:

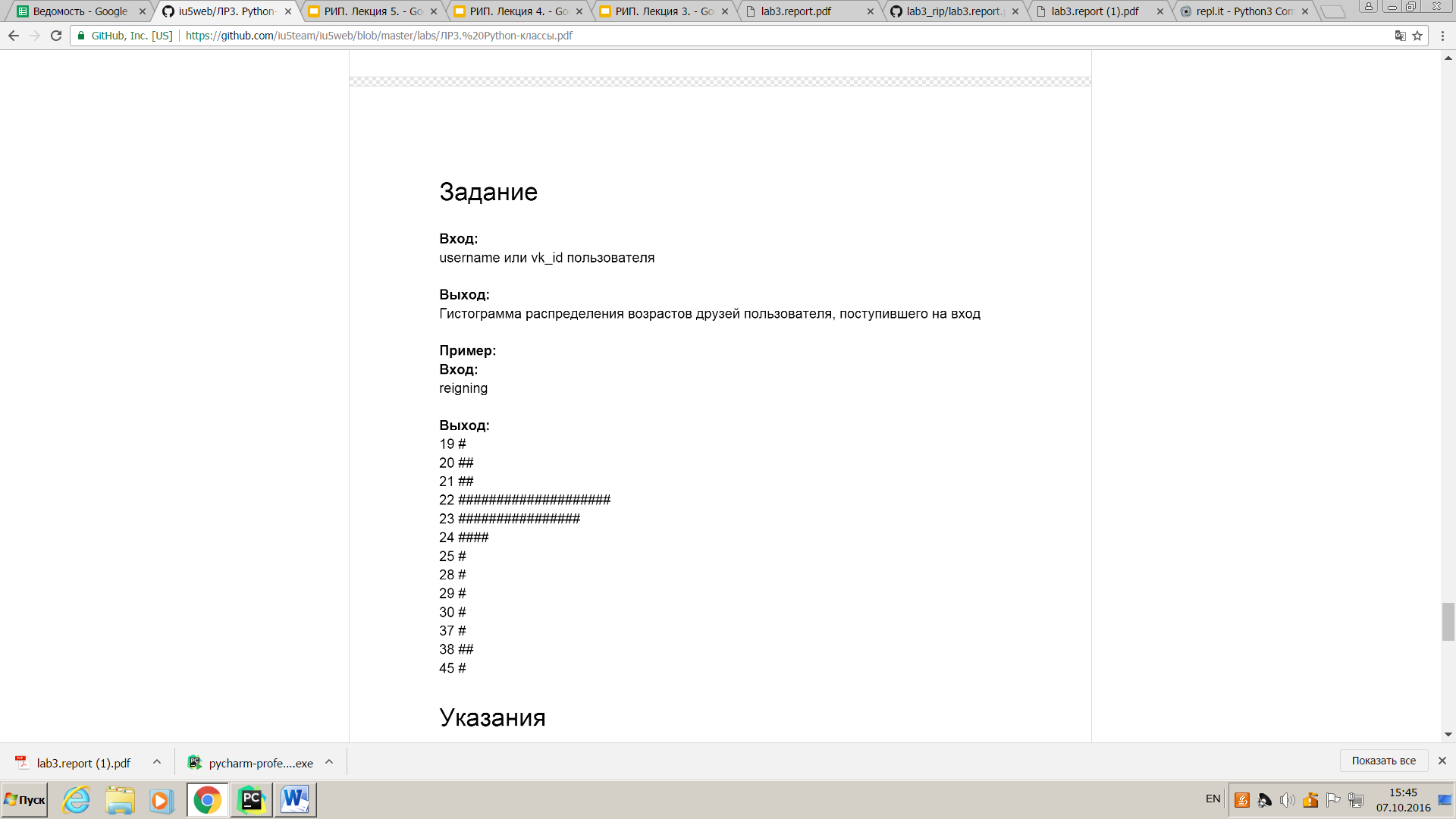
Преподаватель каф. ИУ5

|  |
| --- |
| \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
| (подпись) |

Гапанюк Ю. Е.

"\_\_"\_\_\_\_\_\_\_\_\_2016 г.

Москва, МГТУ – 2016 г.



**BaseClient.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** *#* ***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  
 )

**id\_from\_username.py**

**from** baseclient **import** BaseClient  
**import** requests *#библиотека запросы***import** json  
  
  
**class** IdFromUsername(BaseClient):  
 BASE\_URL = **'https://api.vk.com/method/users.get'** http\_method = **'GET'  
  
 def** \_\_init\_\_(self, name):  
 self.name = name  
  
 **def** get\_params(self):  
 **return 'user\_ids='** + self.name  
  
 **def** response\_handler(self, response): *#обработчик ответа* **try**:  
 uobj = json.loads(response.text)  
 **return** uobj.get(**'response'**)[0].get(**'uid'**)  
 **except**:  
 **raise** Exception(**"Couldn't handle response for username {}"**.format(self.name))  
  
 **def** \_get\_data(self, method, http\_method):  
 response = **None** response = requests.get(self.BASE\_URL + **'?'** + self.get\_params())  
 **return** self.response\_handler(response)

**vk.py**

**from** id\_from\_username **import** IdFromUsername  
**from** friends **import** Friends  
  
uclient = IdFromUsername(**'1'**)  
uid = uclient.execute()  
  
friends\_client = Friends(uid)  
friends = friends\_client.execute()  
  
**for** (age, count) **in** friends:  
 print(**'{} {}'**.format(int(age), **'#'** \* count))

**friends.py**

**from** baseclient **import** BaseClient  
**import** requests  
**import** json  
**from** datetime **import** datetime  
  
  
**class** Friends(BaseClient):  
 BASE\_URL = **'https://api.vk.com/method/friends.get'** http\_method = **'GET'  
  
 def** \_\_init\_\_(self, uid):  
 self.uid = uid  
  
 **def** get\_params(self):  
 **return 'user\_id='** + str(self.uid) + **'&fields=bdate'  
  
 def** response\_handler(self, response): *#обработчик ответа* **try**:  
 uobj = json.loads(response.text)  
 friends = uobj.get(**'response'**)  
  
 ages = []  
  
 **for** friend **in** friends:  
 bdate = friend.get(**'bdate'**)  
  
 **if** bdate **is None or** bdate.count(**'.'**) < 2:  
 **continue** bdate = datetime.strptime(bdate, **"%d.%m.%Y"**)  
 ndate = datetime.now()  
  
 ages.append(int((ndate - bdate).days) // 365.2425)  
  
 uniqages = list(set(ages)) *#устранение дубликатов в массиве возрастов* **return** sorted([(x, ages.count(x)) **for** x **in** uniqages], key=**lambda** x: x[0])  
 **except**:  
 **raise** Exception(**"Couldn't handle response for friends of uid {}"**.format(self.uid))  
  
 **def** \_get\_data(self, method, http\_method):  
 response = requests.get(self.BASE\_URL + **'?'** + self.get\_params())  
 **return** self.response\_handler(response)

**Скриншот результата:**

