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

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

|  |  |
| --- | --- |
|  |  |
|  |  |

**Лабораторная работа №3 по дисциплине**

**“Разработка интернет приложений”**

**Вариант 5**

|  |  |
| --- | --- |
| ИСПОЛНИТЕЛЬ: |  |
| студент группы РТ5-51 |  |
| Гузилов.А.В | "\_\_"\_\_\_\_\_\_\_\_\_\_\_\_\_2016 г. |

Москва - 2016

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

Задание лабораторной работы

В этой ЛР Вы знакомитесь с модулями и ООП в Python, а также осваиваете работу с

сетью.

В лабораторной работе необходимо создать набор классов для реализации работы с

VK API

**Файл vkapi.ру**

**import** urllib.request  
**import** json  
  
**class** BaseClient:  
 BASE\_URL = **None** *# URL VK API* method = **None** *# Метод VK API* http\_method = **None  
  
 def** get\_params(self): *# Получение GET параметров запроса* **pass  
  
 def** get\_json(self): *# Получение данных POST запроса* **pass  
  
 def** get\_headers(self): *# Получение HTTP заголовков* **pass  
  
 def** generate\_url(self, method):  
 **return '{0}{1}'**.format(self.BASE\_URL, method)  
  
 **def** \_get\_data(self, method, http\_method): *# Отправка запроса к VK API* response = **None** *#* ***todo выполнить запрос* return** self.response\_handler(response)  
  
 **def** response\_handler(self, response): *# Обработка ответа от VK API* **return** response  
  
 **def** execute(self): *# Запуск клиента* **return** self.\_get\_data(  
 self.method,  
 self.http\_method  
 )  
  
  
**class** VkApi(BaseClient):  
 PARAMETERS = {}  
 response = **""  
  
 def** \_\_init\_\_(self):  
 self.http\_method = **"GET"** self.BASE\_URL = **"https://api.vk.com/method/"  
  
 def** set\_parameters(self, par):  
 self.PARAMETERS = par  
  
 **def** get\_json(self):  
 **return** json.loads(self.response)  
  
 **def** \_get\_data(self, method, http\_method):  
 **if** http\_method != **"GET"**:  
 **return 'unsupported'** params = self.generate\_url(method)  
 params += **"?"** params += **""**.join([**'%s=%s&'** % (key, value) **for** (key, value) **in** self.PARAMETERS.items()])  
 params += **'v=5.56'** response = urllib.request.urlopen(params)  
  
 **if** response **is not None**:  
 self.response = response.read().decode(**"utf-8"**)  
 **return** self.get\_json()  
 **else**:  
 **return False**

**Файл main.ру**

**from** vkapi **import** \*  
**from** datetime **import** date, datetime  
**import** collections  
**import** matplotlib.pyplot **as** plt  
  
  
**def** hash(x):  
 str = **""  
 for** i **in** range(x):  
 str += **"#"  
 return** str  
  
  
api = VkApi()  
  
params = {**'user\_id'**: **'152667880'**, **'fields'**: **'bdate'**}  
  
api.set\_parameters(params)  
api.method = **"friends.get"**response = api.execute()  
  
**if not** response **or** response == **"unsupported"**:  
 print(**"have some errors"**)  
response = response[**'response'**][**'items'**]  
  
ages = {}  
now = datetime.now().date()  
  
**for** user **in** response:  
 dat = user.setdefault(**'bdate'**, **''**).split(**"."**)  
 **if** len(dat) != 3:  
 **continue** d = date(int(dat[2]), int(dat[1]), int(dat[0]))  
 razn = now - d  
 years = int(razn.days / 365)  
 ages[years] = ages.setdefault(years, 0) + 1  
  
ages = collections.OrderedDict(sorted(ages.items()))  
  
hist = []  
bins = []  
flg, ax = plt.subplots()  
  
**for** k, v **in** ages.items():  
 print(**"{}: {}"**.format(k, hash(v)))  
 bins.append(k)  
 hist.append(v)  
 ax.bar(bins, hist, 1)  
  
*#вывод на гистограмму*plt.xlabel(**'Years'**)  
plt.ylabel(**'Number of friends'**)  
plt.hist(hist, bins=bins)  
plt.show()

Результат



