



Лабораторные работы по курсу:
«Разработка Интернет Приложений»

Лабораторная работа №3

Исполнитель:
Студентка группы ИУ5-54

Ворошилова М.А.

Преподаватель:

Гапанюк Ю.Е,



Москва 2017г.

Задание

Вход: username или vk_id пользователя

Выход: Гистограмма распределения возрастов друзей пользователя, поступившего на вход

Листинг

base_client.py:

```
class BaseClient:
    # URL vk api
    BASE_URL = None
    # метод vk api
    method = None
    # GET, POST, ...

    # Получение GET параметров запроса
    def get_params(self):
        return None

    # Получение данных POST запроса

    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) -> object:
        return self._get_data(
            self.method,
            http_method=self.http_method
        )
```

classes.py:

```
import requests
import base_client
import json
import datetime

class GetId(base_client.BaseClient):
    BASE_URL = "https://api.vk.com/method/"
    method = 'users.get'
    http_method = 'get'

    # Отправка запроса к VK API
    def _get_data(self, method, http_method):
        screen_name = input()
        response = requests.get(base_client.BaseClient.generate_url(self, GetId.method), params={'user_ids': screen_name})
        return self.response_handler(response)

    # Обработка ответа от VK API
    def response_handler(self, response):
        if response.status_code == requests.codes.ok:
            res = str(response.json()[0][0]['uid'])
            return res
        else:
            print("Error!")

class FriendsAnalytics(base_client.BaseClient):
    BASE_URL = "https://api.vk.com/method/"
    method = 'friends.get'
    http_method = 'post'
    user_id = None
```

```

def __init__(self, vk_id):

    # Отправка запроса к VK API
def _get_data(self, method, http_method):
    data = {'user_id': self.user_id, 'count': '5000', 'fields': 'bdate'}
    response = requests.post(base_client.BaseClient.generate_url(self, self.method), data=data)
    return self.response_handler(response)

s # Обработка ответа от VK API
e def response_handler(self, response):
l     if response.status_code == requests.codes.ok:
f         return response
u     else:
s         print("Error!")
e
r def Diagram(self, response):
i     if response.json()['response'] is not None:     # Есть ли друзья
d         blist = list()
v         for person in response.json()['response']:
k
continue
        else:
            try:
                blist.append(datetime.datetime.strptime(person['bdate'], '%d.%m.%Y'))
            except:
                continue
        today = datetime.datetime.today()
        agelist = list()
        for b in blist:
            age = today.year - b.year - 1
            if today.month <= b.month:
                if today.day <= b.day:
                    age += 1
            agelist.append(age)
        agelist.sort()
i     temp = agelist[0]
f     print(agelist[0], end='\t')
p     for a in agelist:
e         if a != temp:
r             print()
s             print(a, end='\t')
o             print('#', end='')
n             temp = a
g         else:
e             print('#', end='')
t
b
d # Запуск клиента
a def execute(self) -> object:
t     res = self._get_data(self.method, http_method=self.http_method)
e     if res == "Error!":
i         print("Error!")
s     return res
None # Печать информации
# Печать информации
self.Diagram(res)

```

main.py:

```

import requests
import datetime
import json
import base_client

```

```

import classes

```

```

a = classes.GetId().execute()
print(a)

```

```

b = classes.FriendsAnalytics(a)
b.execute()

```

Вывод:

```
youra_avdeev
154129192
14      ###
15      #####
16      #####
17      #####
18      #####
19      #####
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      #####
48      #####
49      #####
50      #####
51      #####
52      #####
53      #####
54      #####
55      #####
56      #####
57      #####
58      #####
59      #####
60      #####
61      ###
62      #####
63      ###
64      #####
65      ###
66      #####
67      #
68      ####
69      ##
70      ####
71      ####
72      ##
73      #
74      #
76      #
77      #
83      ##
90      #
93      #
94      #
95      ##
96      ##
98      #
99      #####
100     #
104     #
105     #
106     #
107     ##
108     #
111     #####
112     #
113     #
114     ##
115     #####
116     #
```