



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, ...
    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) -> 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):
    self.user_id = 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)

# Обработка ответа от VK API
def response_handler(self, response):
    if response.status_code == requests.codes.ok:
        return response
    else:
        print("Error!")

def PrintAges(self, response):
    if response.json()['response'] is not None:    # Есть ли друзья
        age_list = [0 for i in range(120)]
        today = datetime.now()
        for f in response.json()['response']:
            bdate_str = f.get('bdate')
            try:
                bdate = datetime.strptime(bdate_str, '%d.%m.%Y')
                days = (today - bdate).days
                age = days // 365
                age_list[age] += 1
            except:
                pass
        # печать гистограммы
        for i in range(120):
            if age_list[i] > 0:
                print(i, ': ', '#' * age_list[i])

# Запуск клиента
def execute(self) -> object:
    res = self._get_data(self.method, http_method=self.http_method)
    if res == "Error!":
        print("Error!")
    return res
# Печать диаграммы
self.PrintAges (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 #