Московский Государственный Технический Университет им. Н.Э. Баумана

Разработка интернет-приложений

Отчёт по лабораторной работе №3

«Python классы»

Выполнил:

студент группы ИУ5-54

Ваняшкин Юрий

1. Цель работы

В лабораторной работе необходимо создать набор классов для реализации работы с VK API.

2. Листинг программы

Модуль `base_client.py`

```
import requests
class BaseClient:
    # URL vk api
    BASE_URL = None
    # method vk api
    method = None
    # GET, POST, ...
    http_method = None
    # GET
    def get_params(self):
        return None
    # POST
    def get_json(self):
        # _get_data
        #response.json()
        return None
    # HTTP
    def get_headers(self):
        # _get_data
        #response.headers['content-type']
        return None
    # url
    def generate_url(self, method):
        return '{0}{1}'.format(self.BASE_URL, method)
    # VK API
    def _get_data(self, parameters):
        response = None
        return self.response_handler(response)
    # VK API
    def response_handler(self, response):
        return response
    # Start client
    def execute(self, parameters):
        return self._get_data(
            parameters
        )
```

Модуль `catch_id.py`

```
from base_client import BaseClient
import requests
class CatchID(BaseClient):
    # ID user vk.com
    ID = None
    def __init__(self, username):
        self.BASE_URL = "https://api.vk.com/method/"
        self.method = "users.get"
        self.http_method = "GET"
        params = {"user_ids": username, "fields": "", "name_case": "nom"}
        self.ID = self.execute(params)
    # Response handler
    def response_handler(self, response):
        try:
            response.raise_for_status()
            if len(response.json()["response"]) > 0:
                return response.json()["response"][0]["uid"]
            else:
                return None
        except requests.exceptions as err:
            print('Oops. Something occured')
            print(err)
    # VK API
    def _get_data(self, parameters):
        response = None
        if self.http_method == "GET":
            response = requests.get(self.generate_url(self.method),
params=parameters)
        return self.response_handler(response)
```

Модуль `catch_friends.py`

```
from base_client import BaseClient
from datetime import datetime, timedelta
import requests
class CatchFriends(BaseClient):
    # friends for our user
    friends = None
    def __init__(self, ID):
        self.BASE_URL = "https://api.vk.com/method/"
        self.method = "friends.get"
        self.http_method = "GET"
        params = {"user_id": ID, "order": "random", "list_id": None,
                  "count": None, "offset": None, "fields": "bdate",
                  "name_case": "nom"}
        self.friends = self.execute(params)
    # Response handler
    def response_handler(self, response):
            response.raise_for_status()
```

```
            dateOfB = []
            for x in response.json()["response"]:
                if x.get("bdate") != None:
                    #arr = datetime.strptime(x.get("bdate"), "%d.%m.%Y")
                    #dateOfB.append( datetime.now().year - arr.year )
                    arr = (x.get("bdate")).split('.')
                    if len(arr) == 3:
                        dateOfB.append(datetime.now().year - int(arr[2]))
            dateOfB.sort()
            return dateOfB
    # VK API
    def _get_data(self, parameters):
        response = None
        if self.http_method == "GET":
            response = requests.get(self.generate_url(self.method),
params=parameters)
        return self.response_handler(response)
```

Модуль `main.py`

```
from itertools import groupby
from catch_id import CatchID
from catch_friends import CatchFriends
import numpy as np
import matplotlib.pyplot as plt
def PrintSimpleGraph(listOfDates):
    newListOfDates = [elem for elem, _ in groupby(listOfDates)]
    for x in newListOfDates:
        num = listOfDates.count(x)
        a = ''.join("#" for x in range(0, num))
        print(str(x) + a)
def PrintGraph(listOfDates):
    # hist()
    fig = plt.figure()
    plt.hist(listOfDates)
    plt.title('Result histogramm')
    plt.grid(True)
    # Tweak spacing to prevent clipping of ylabel
    fig.tight_layout()
    plt.show()
username = input()
user = CatchID(str(username))
#print(user.ID)
listOfFriends = CatchFriends(user.ID)
#print(listOfFriends.friends)
print('\n')
PrintSimpleGraph(listOfFriends.friends)
print('\n')
PrintGraph(listOfFriends.friends)
```

3. Результат работы

Никнейм пользователя: oblomoff (3000+ друзей)
В режиме вывода в консоль:

```
13 #
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 ###
```

В режиме построения графика: