

Отчет по лабораторной работе №3
по курсу разработка интернет приложений
«Python-классы»

Вариант №8

Выполнила:

Студентка группы ИУ5-54
Журавлева Ульяна

Проверил:

Преподаватель кафедры ИУ5
Гапанюк Ю.У.

Москва, 2016год.

Содержание

1. Задание ЛР
2. Текст программы
3. Результаты выполнения программы

1. Задание ЛР

Цель данной лабораторной работы познакомиться с модулями и ООП в Python, а также освоить работу с сетью.

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

Пример результата, который будет выдавать программа:

Вход:

username или vk_id пользователя

Выход:

Гистограмма распределения возрастов друзей пользователя, поступившего на вход

Пример:

Вход:

reigning

Выход:

```
19 #
20 ##
21 ##
22 #####
23 #####
24 ####
25 #
28 #
29 #
30 #
37 #
38 ##
45 #
```

2. Текст программы

Модуль Main

```
import Child
from datetime import datetime

print("      Вход: ")
user = Child.Client("http://api.vk.com/method", 'users.get')
user.param = 'user_ids='+input("Введите username или id пользователя: ")
print(" ")
```

```

print("      Выход: ")
print("Пользователь: ")
print(user.execute())
resp = user.execute()
id = user.find_id(resp)
print(" ")
print("Гистограмма распределения возрастов друзей пользователя, поступившего
на вход: ")
friend = Child.Client('http://api.vk.com/method', 'friends.get')
friend.param = "user_id="+str(id)+"&fields=bdate"
friends = friend.execute()
friend_list = friends.get('response')
listf = friend.count_friend(friend_list)
friend.print_list(listf)

```

Модуль Child

```

import requests
import Base
from datetime import datetime

class Client(Base.BaseClient):
    param = None

    def __init__(self, url, methods):
        self.BASE_URL = url
        self.method = methods

    def generate_url(self, method):
        return '{0}/{1}'.format(self.BASE_URL, method)

    def _get_data(self, method):
        response = requests.get(self.generate_url(method), self.param)
        response.json()
        return response.json()

    def find_id(self, idf):
        try:
            cl_id1 = idf.get('response')
            cl_id = cl_id1[0].get('uid')
        except:
            print("Пользователь не найден")
            exit()
        return cl_id

    def find_age(self, date):
        try:
            bdate = datetime.strptime(date, '%d.%m.%Y')
        except: return None

        day_now = datetime.now().day
        month_now = datetime.now().month
        year_now = datetime.now().year

        age = year_now - bdate.year
        if (bdate.month == month_now and bdate.day > day_now) or bdate.month
> month_now:
            age -= 1
        return age

    def count_friend(self, friends_list):
        if (friends_list == None):
            print ('Страница пользователя была удалена или заблокирована')

```

```

        exit()
    if (len(friends_list) == 0):
        print ("У пользователя нет друзей, либо они скрыты")
        exit()
    friends_count = {'Возраст не указан у: ':0}
    for friend in friends_list:
        bdate = friend.get('bdate')
        age = self.find_age(bdate)
        if (age == None):
            friends_count['Возраст не указан у: '] += 1
            continue
        if (friends_count.get(age) == None):
            friends_count[age] = ''
        friends_count[age] += '#'
    return friends_count

def print_list(self, list):
    print('Возраст не указан у: ', list['Возраст не указан у: '])
    del list['Возраст не указан у: ']
    for key in sorted(list):
        print(key, ': ', list[key])

```

Модуль Base

```

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

    # Получение GET параметров запроса
    def get_params(self):
        pass

    # Получение данных POST запроса
    def get_json(self):
        pass

    # Получение HTTP заголовков
    def get_headers(self):
        pass

    # 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):
        response = None
        return self.response_handler(response)

    # Обработка ответа от VK API
    def response_handler(self, response):
        return response

    # Запуск клиента
    def execute(self):
        return self._get_data(
            self.method
        )

```

3. Результаты программы

```
Run Main
C:\Users\1\AppData\Local\Programs\Python\Python35\python.exe C:/Users/1/PycharmProjects/untitled5/Main.py
Вход:
Введите username или id пользователя: 2

Выход:
Пользователь:
{'response': [{'first_name': 'Александра', 'last_name': 'Владимирова', 'hidden': 1, 'uid': 2}]}

Гистограмма распределения возрастов друзей пользователя, поступившего на вход:
Возраст не указан у: 15
26 : #
28 : ##
29 : #
30 : #
31 : #
32 : #

Process finished with exit code 0

Run Main
C:\Users\1\AppData\Local\Programs\Python\Python35\python.exe C:/Users/1/PycharmProjects/untitled5/Main.py
Вход:
Введите username или id пользователя: u1ia960131

Выход:
Пользователь:
{'response': [{'uid': 231608653, 'hidden': 1, 'first_name': 'Уля', 'last_name': 'Журавлёва'}]}

Гистограмма распределения возрастов друзей пользователя, поступившего на вход:
Возраст не указан у: 32
19 : ###
20 : #####
21 : #
23 : #
26 : #
27 : #
48 : #
96 : #
115 : #

Process finished with exit code 0

Run Main
C:\Users\1\AppData\Local\Programs\Python\Python35\python.exe C:/Users/1/PycharmProjects/untitled5/Main.py
Вход:
Введите username или id пользователя: 3

Выход:
Пользователь:
{'response': [{'uid': 3, 'last_name': '', 'first_name': 'DELETED', 'deactivated': 'deleted'}]}

Гистограмма распределения возрастов друзей пользователя, поступившего на вход:
Страница пользователя была удалена или заблокирована

Process finished with exit code 0
```