



Лабораторные работы по курсу:
«Разработка Интернет Приложений»

Python-классы

Исполнитель:
Студент группы РТ5-51
Макаров А.В.
Преподаватель:
Гапанюк Ю.Е.,
« _ » _____

Цель работы:

Познакомиться с модулями и ООП в Python, а также освоить работу с сетью.

Листинг

base_client.py:

```
import requests

class BaseClient:
    # URL vk api
    BASE_URL = "https://api.vk.com/method/"

    # Отправка запроса к VK API
    def _get_data(self, method, http_method):
        response = requests.get(self.BASE_URL + self.method + "." + self.http_method,
                                params=self.get_params())

        return self.response_handler(response)

    # Запуск клиента
    def execute(self):
        return self._get_data(
            self.method,
            http_method=self.http_method
        )
```

ClientVK.py

```
import base_client
from datetime import datetime

debug = True


class ClientGetID(base_client.BaseClient):
    method = "users"
    http_method = "get"
    json_data = None

    def __init__(self, username):
        self.username = username

    def get_params(self):
        return {
            "user_ids": self.username
        }

    def response_handler(self, response):
        self.json_data = response.json()
        return self.json_data["response"][0]["uid"]


def calculate_age(born):
    today = datetime.utcnow()
    return today.year - born.year


class ClientGetFriendsAges(base_client.BaseClient):
    method = "friends"
    http_method = "get"
    json_data = None

    def __init__(self, user_id):
        self.user_id = user_id

    def get_params(self):
        return {
            "user_id": self.user_id,
            "fields": "bdate"
        }

    def response_handler(self, response):
        self.json_data = response.json()
        ages = list()

        for friend in self.json_data["response"]:
            date_tmp = friend.get("bdate")
            # print (type(date_tmp), date_tmp)

            if date_tmp is None:
                date_tmp = "01.01.2016"
                # print (date_tmp)
                # continue

            try:
                ages.append(calculate_age(datetime.strptime(date_tmp, "%d.%m.%Y")))
            except ValueError:
                ages.append(0)
        ages = sorted(ages)
        return ages
```

gist.py

```
class Gist:
    # данные гистограммы

    def __init__(self, ages):

        self._data_sorting = dict()
        for number in ages:
            self._data_sorting.update(
                {number: self._data_sorting.get(number, 0) + 1})

    def print_hist(self):

        str_out = ""
        for age, stat in self._data_sorting.items():
            str_out += str(age).ljust(3) + str().ljust(stat, '#') + '\n'
        print(str_out)
```

Main.py

```
from ClientVK import ClientGetID
from ClientVK import ClientGetFriendsAges
from gists import Gist

debug = True
username = "Laptogol"

get_id = ClientGetID(username).execute()
friends_ages = ClientGetFriendsAges(get_id).execute()

if debug:
    print("ID: ", get_id)
    print("Ages: ", friends_ages)

title = "Ages of Users "
title_x = "Ages"
title_y = "Users"

mygist = Gist(friends_ages)
mygist.print_hist()

Вывод:
0 #####
16 ##
17 ##
18 ##
19 ###
20 #####
21 ###
22 #
23 #
24 #
25 #
26 #
36 #
49 #
69 #
```


