



Лабораторные работы по курсу:
«Разработка Интернет Приложений»

Python-классы

Исполнитель:
Студент группы ИУ5-52

Миядин А.А.

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/"
    # 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):

        # todo выполнить запрос
        response = requests.get(self.BASE_URL + self.method + "." + self.http_method, params =
self.get_params())

        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,
            http_method=self.http_method
        )
```

ClientVK.py

```
import base_client
from datetime import datetime

class ClientGetID(base_client.BaseClient):
    username = None
    # метод vk api
    method = "users"
    # GET, POST, ...
    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 get_json(self):
        return self.json_data


class ClientGetFriendsAges(base_client.BaseClient):
    # метод vk api
    method = "friends"
    # id пользователя
    user_id = None
    # GET, POST, ...
    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 calculate_age(self, born, today):
        if today == None:
            today = datetime.utcnow()
        return today.year - born.year - ((today.month, today.day) < (born.month, born.day))

    def response_handler(self, response):
        self.json_data = response.json()
        ages = list()
        today = datetime.utcnow()
        date_tmp = None
```

```

for friend in self.json_data["response"]:
    date_tmp = friend.get("bdate")
    if date_tmp == None or len(date_tmp) < 6:
        continue
    ages.append( self.calculate_age( datetime.strptime( date_tmp, "%d.%m.%Y"), today))
    #if ages[-1] == 13:
    #    print(date_tmp)
return ages

def get_json(self):
    return self.json_data

```

gist.py

```

import numpy as np
import matplotlib.mlab as mlab
import matplotlib.pyplot as plt

class Gist ():
    # данные гистограммы
    _data = None
    _data_sorting = dict()

    def __init__(self, date):
        self._data = date
        for number in date:
            self._data_sorting.update(
                {number: self._data_sorting.get(number, 0) + 1})

    def get_data(self):
        return self._data

    def printHist(self):
        str_out = ""
        for age, stat in self._data_sorting.items():
            str_out += str(age).ljust(4) + str().ljust(stat, '#') + '\n'
        print(str_out)

    def showHist(self, title, title_x, title_y):

        fig, ax = plt.subplots()
        ax.hist(self._data, range(list(self._data_sorting.keys())[
            0], list(self._data_sorting.keys())[-1] + 1), normed=1)
        ax.set_title(title)
        ax.set_xlabel(title_x)
        ax.set_ylabel(title_y)
        plt.show()

    def showBar(self, title, title_x, title_y):
        plt.bar(list(self._data_sorting.keys()),
            self._data_sorting.values(), color='b')
        plt.show()

```

```

run.py
#!/usr/bin/python3

from ClientVK import *
from gist import Gist

debug = False
username = "alexdarkstalker98"

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

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

mygist = Gist(friends_ages)
mygist.printHist()
title = "Ages of Users "
title_x = "Ages"
title_y = "Users"
mygist.showHist(title,title_x,title_y)

```

Вывод:

```

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 #####
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 #
70 #
71 ###
72 #####
74 #####
75 #####
76 #####
77 #
78 ##
80 ##
82 ##
83 ##
84 #
85 ##
87 ##
88 ##
89 #
90 #
91 #
92 ##
93 ##
96 #####
97 #####
98 #
99 #####
100 ###
101 ###
102 ##
103 ##

104 #
105 #
106 #
107 ##
110 ##
112 ###
113 #
114 #
115 #####
116 #####