

Отчёт по лабораторной работе №3
по курсу «Разработка интернет-приложений»
Python-классы

Выполнил:

Волобуев В.Н., ИУ5-54

Преподаватель:

Гапанюк Ю.Е.

2016 г.

1) Задание лабораторной работы

Основная задача работы – ознакомление с модулями и ООП в Python, а также осваивание работы с сетью.

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

Вход:

username или vk_id пользователя

Выход:

Гистограмма распределения возрастов друзей пользователя, поступившего на вход

Доп. требования:

Построить гистограмму с использованием *matplotlib*

2) 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):
    return self._get_data(
        self.method,
        http_method=self.http_method
    )
```

VK_get_id.py

```
from base_client import BaseClient
import requests
```

```

class GetIDFromUsername(BaseClient):
    # users.get
    BASE_URL = 'https://api.vk.com/method/users.get'
    http_method = 'GET'

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

    def get_params(self):
        return 'user_ids='+self.username

    def response_handler(self, response):
        return response.json()['response'][0]['uid']

    def _get_data(self, method, http_method):
        response = requests.get(\
            self.BASE_URL+'?' + self.get_params())

        return self.response_handler(response)

if __name__ == '__main__':
    test = GetIDFromUsername('wilhelmjayther')
    print(test.execute())

```

VK_parser.py

```

from base_client import BaseClient
from VK_get_id import GetIDFromUsername
import requests
from datetime import datetime

```

```

class ParseFriends(BaseClient):
    # friends.get(fields=bdate)
    BASE_URL = 'https://api.vk.com/method/friends.get'
    http_method = 'GET'

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

    def get_params(self):
        return 'user_id='+str(self.uid)+'&fields=bdate'

    def count_age(self, bdate, curdate):
        years = curdate.year-bdate.year
        if curdate.month < bdate.month or\
            curdate.month == bdate.month and\
            curdate.day < bdate.day:
            years = years-1
        return years

    def response_handler(self, response):
        friends = response.json()['response']
        hist = dict()
        for friend in friends:
            if 'bdate' in friend and\
                friend['bdate'].count('.')==2:
                bdate = datetime.strptime(friend['bdate'],\
                    "%d.%m.%Y")
                curdate = datetime.now()
                years = self.count_age(bdate, curdate)
                if years not in hist:
                    hist[years] = 0
                hist[years] = hist[years]+1
        result_string = str()
        for age in sorted(hist):

```

```

        result_string = result_string + str(age) + ':'
        for i in range(hist[age]):
            result_string = result_string + '#'
        result_string = result_string + '\n'
    return result_string

def _get_data(self, method, http_method):
    response =\
        requests.get(self.BASE_URL+'?' + self.get_params())

    return self.response_handler(response)

if __name__ == '__main__':
    test = GetIDFromUsername('wilhelmjayther')
    uid = test.execute()
    test = ParseFriends(uid)
    print(test.execute())

```

my_vk_client.py

```

import VK_get_id
import VK_parser
from datetime import datetime
import matplotlib.pyplot as plt

class MatPlotParser(VK_parser.ParseFriends):
    #returns list of ages for matplotlib histogram
    def response_handler(self, response):
        friends = response.json()['response']
        hist_list = list()

```

```

        for friend in friends:
            if 'bdate' in friend and\
                friend['bdate'].count('.')==2:
                bdate = datetime.strptime(friend['bdate'],\
                    "%d.%m.%Y")
                curdate = datetime.now()
                years = self.count_age(bdate, curdate)
                hist_list.append(years)
        return hist_list

if __name__ == '__main__':
    str_id = input('Type some ID: ')
    test = VK_get_id.GetIDFromUsername(str_id)
    uid = test.execute()
    test = VK_parser.ParseFriends(uid)
    print(test.execute())
    test = MatPlotParser(uid)
    ages = test.execute()
    #time to get rid of seniles
    ages = list(filter(lambda x: x > 0 and x < 100, ages))
    bins = max(ages)-min(ages)+1

    fig, ax0 = plt.subplots(ncols=1, figsize=(8, 4))

    ax0.hist(ages, bins, facecolor='#ff8000', alpha=0.75)
    ax0.set_title(str_id+'\''s friends ages')

    plt.tight_layout()
    plt.show()

```

3) Результаты работы

```
Терминал
Файл  Правка  Вид  Поиск  Терминал  Справка
Type some ID: wilhelmjayther
21:#
22:####
23:#####
24:#####
25:###
26:#
27:####
28:#
29:##
31:#
32:#
35:#
115:#

>>> █
```

