МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ

ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ

«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Факультет Информационной Безопасности

Пояснительная записка к курсовой работе по дисциплине

«Технологии, методы и языки программирования»:

«Разработка программы сканирования web-приложений»

Выполнил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ В.С. Ирицян  
 (подпись) ст. группы И-11

Проверил: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ Е.П. Тумоян  
 (подпись) доцент каф. БИТ

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc375751584)

[1 Теоретическая часть 4](#_Toc375751585)

[2 Разработка программы 6](#_Toc375751586)

[3 Примеры решения задачи 8](#_Toc375751587)

[ЗАКЛЮЧЕНИЕ 11](#_Toc375751588)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 12](#_Toc375751589)

[ПРИЛОЖЕНИЕ A 13](#_Toc375751590)

# 

# ВВЕДЕНИЕ

Сайт — совокупность электронных документов (файлов) частного лица или организации в компьютерной сети, объединённых под одним адресом (доменным именем или IP-адресом). Все сайты в совокупности составляют Всемирную паутину, где коммуникация (паутина) объединяет сегменты информации мирового сообщества в единое целое — базу данных и коммуникации планетарного масштаба. Для прямого доступа клиентов к сайтам на серверах был специально разработан протокол HTTP.

Задача данной курсовой работы разработка web-клиента, который может искать картинки и ссылки в сети vk.com с указанной глубиной прохождения. Результаты поиска можно выводить на экран либо в таблице. Так же, с помощью разработанной программы можно найти всех «друзей» в данной социальной сети с глубиной прохождения, выбранной пользователем.

# 1 Теоретическая часть

Всемирная паутина — распределённая система, предоставляющая доступ к связанным между собой документам, расположенным на различных компьютерах, подключенных к Интернету. Всемирную паутину образуют сотни миллионов веб-серверов. Веб-сервер - сервер, принимающий HTTP-запросы от клиентов, обычно веб-браузеров, и выдающий им HTTP-ответы, как правило, вместе с HTML-страницей, изображением, файлом, медиа-потоком или другими данными. Веб-сервером называют как программное обеспечение, выполняющее функции веб-сервера, так и непосредственно компьютер, на котором это программное обеспечение работает. Клиент, которым обычно является веб-браузер, передаёт веб-серверу запросы на получение ресурсов, обозначенных URL-адресами. Ресурсы — это HTML-страницы, изображения, файлы, медиа-потоки или другие данные, которые необходимы клиенту. В ответ веб-сервер передаёт клиенту запрошенные данные. Этот обмен происходит по протоколу HTTP.

Большинство ресурсов всемирной паутины основано на технологии гипертекста. Гипертекстовые документы, размещаемые во Всемирной паутине, называются веб-страницами. Несколько веб-страниц, объединённых общей темой, дизайном, а также связанных между собой ссылками и обычно находящихся на одном и том же веб-сервере, называются веб-сайтом. Для загрузки и просмотра веб-страниц используются специальные программы — браузеры.

В настоящее время наметились две тенденции в развитии Всемирной паутины: семантическая паутина и социальная паутина.

Семантическая паутина предполагает улучшение связности и релевантности информации во Всемирной паутине через введение новых форматов метаданных.

Социальная паутина полагается на работу по упорядочиванию имеющейся в Паутине информации, выполняемую самими пользователями Паутины. В рамках второго направления наработки, являющиеся частью семантической паутины, активно используются в качестве инструментов (RSS и другие форматы веб-каналов, OPML, микроформаты XHTML). Частично семантизированные участки Дерева категорий «Википедии» помогают пользователям осознанно перемещаться в информационном пространстве, однако, очень мягкие требования к подкатегориям не дают основания надеяться на расширение таких участков. В связи с этим интерес могут представлять попытки составления атласов Знания.

Для того чтобы полноценно работать с некоторыми сайтами, необходимо пройти процедуру авторизации. Авторизация — предоставление определённому лицу или группе лиц прав на выполнение определённых действий; а также процесс проверки (подтверждения) данных прав при попытке выполнения этих действий. Часто можно услышать выражение, что какой-то человек «авторизован» для выполнения данной операции — это значит, что он имеет на неё право.

В ходе выполнения курсовой работы, мне понадобилось «доставать» из страниц некоторую информацию. Это можно было осуществить с помощью регулярных выражений. Регулярные выражения — система обработки текста, основанная на специальной системе записи образцов для поиска. Образец, задающий правило поиска, по-русски также иногда называют «шаблоном», «маской». Сейчас регулярные выражения используются многими текстовыми редакторами и утилитами для поиска и изменения текста на основе выбранных правил. Многие языки программирования уже поддерживают регулярные выражения для работы со строками. Например, [Python](http://ru.wikibooks.org/w/index.php?title=Tcl&action=edit&redlink=1) имеет синтаксис механизм обработки регулярных выражений.

# 2 Разработка программы

Программирование web-клиента - мощная техника для создания запросов в Web. Web-клиент - это любая программа, извлекающая данные с web -сервера при помощи протокола передачи гипертекста web-браузер является клиентом, так же, как и поисковики, то есть программы, автоматически перемещающиеся по Web для сбора информации. Программирование web-клиента входит в любой набор инструментов для разработчиков. В языке Python этот процесс достигает даже более высоких уровней удобства и гибкости. В работе был использован модуль requests.

Чтобы полноценно работать с некоторыми сайтами, необходимо пройти процедуру авторизации. Для того чтобы узнать какие запросы (get/ post) надо отослать, можно воспользоваться программой «Wireshark». Процедура авторизации в vk.com:

log = {'email': ваш email, ‘pass’: ваш пароль}

log\_url = "https://login.vk.com/?act=login"

session.post (log\_url, data=log)

В ходе выполнения курсовой работы, мне понадобилось искать на страницах картинки и ссылки на другие страницы. Это можно было осуществить с помощью регулярных выражений и XPath . Нахождение картинок: re.findall(ur'<img\s+.\*?src="(.\*?)".\*?', rt) и tree.xpath('////img[@src][1]/@src'). Нахождение ссылок: re.findall(ur'<a\s.\*?href="(.\*?)".\*?</a>', rt) и tree.xpath('////a[@href][1]/@href'). Для курсового проекта были выбраны регулярные выражения, т.к. им уделялось большее время в течение семестра.

Еще одной задачей была разработка программы поиска всех «друзей» в данной социальной сети с глубиной прохождения, выбранной пользователем. «Достать» всех друзей пользователя можно с помощью post-запроса следующего вида:

friends\_data = {"act": "load\_friends\_silent",

"al": "1",

"gid": "0",

"id": id пользователя}

friends\_url = http://vk.com/al\_friends.php

session.post(friends\_url, data=friends\_data)

После получения информации ее нужно отфильтровать в зависимости от того, друзей какого пользователя мы получаем: авторизованного или какого-либо другого. Пользователь может выбрать, куда выводить результат, и, в зависимости от выбора, результат выводится на экран консоли либо в виде таблицы (по моему мнению, более наглядно).

**3 Примеры решения задачи**

Пользователь ищет все картинки с глубиной прохождения равной 2.

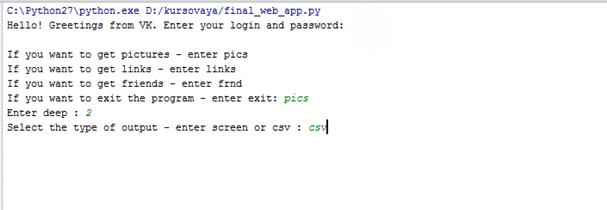


Рисунок 1- Запуск приложения

Результат поиска представлен на рисунке 2.

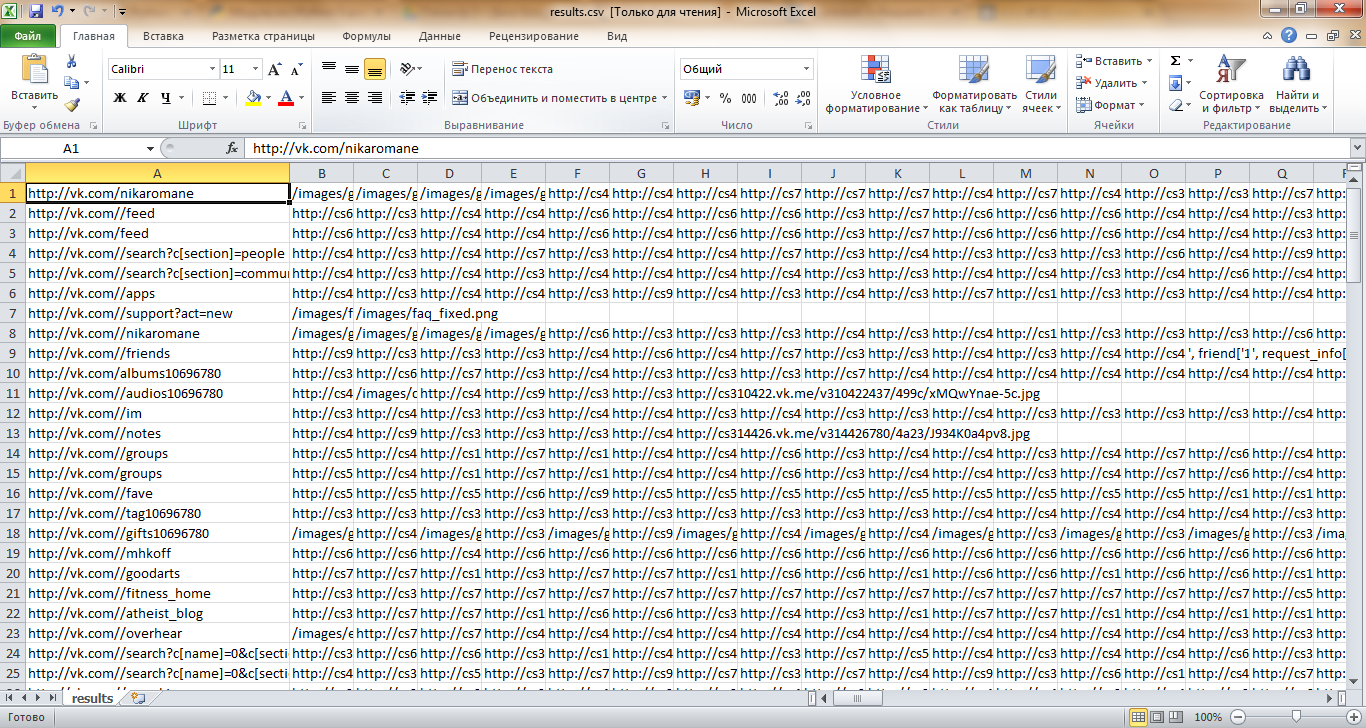


Рисунок 2 – Таблица результата поиска

Далее пользователь ищет ссылки, и результат выводится на экран.

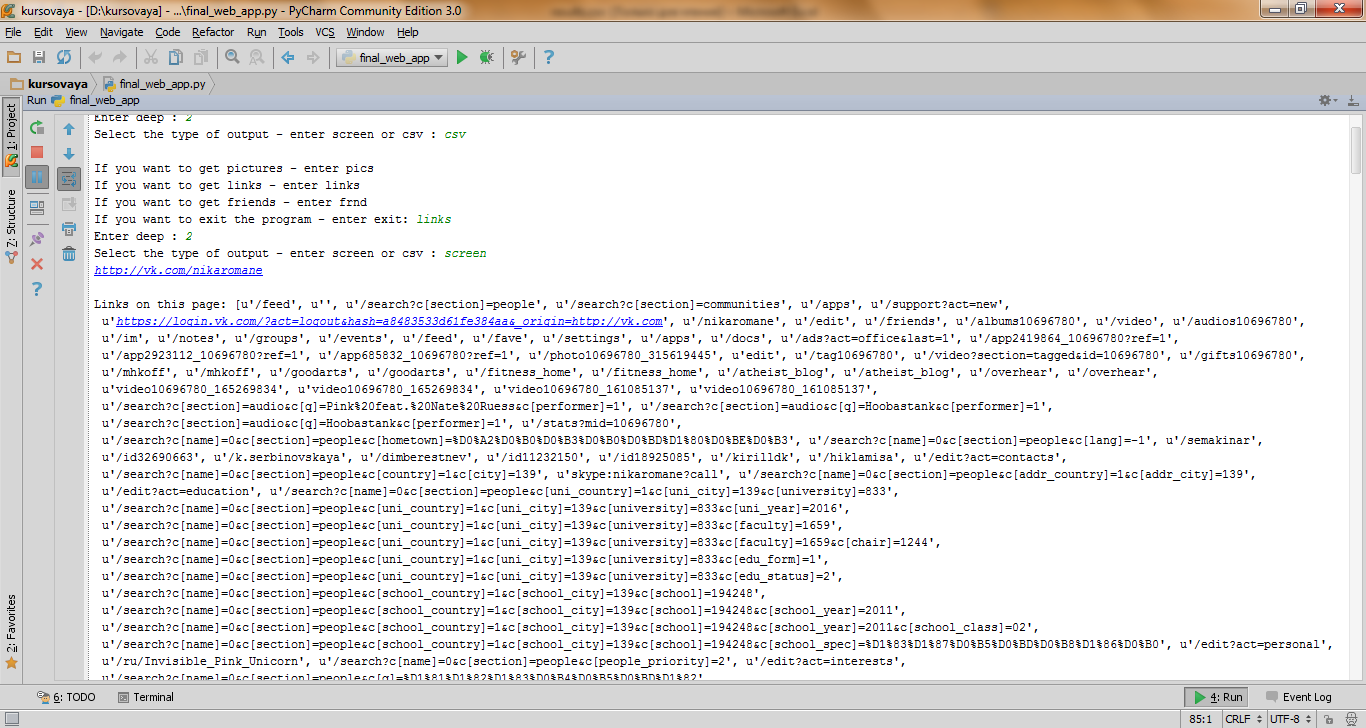


Рисунок 3 – Результат поиска ссылок

Поиск друзей с глубиной равной 2, вывод результата в таблице.

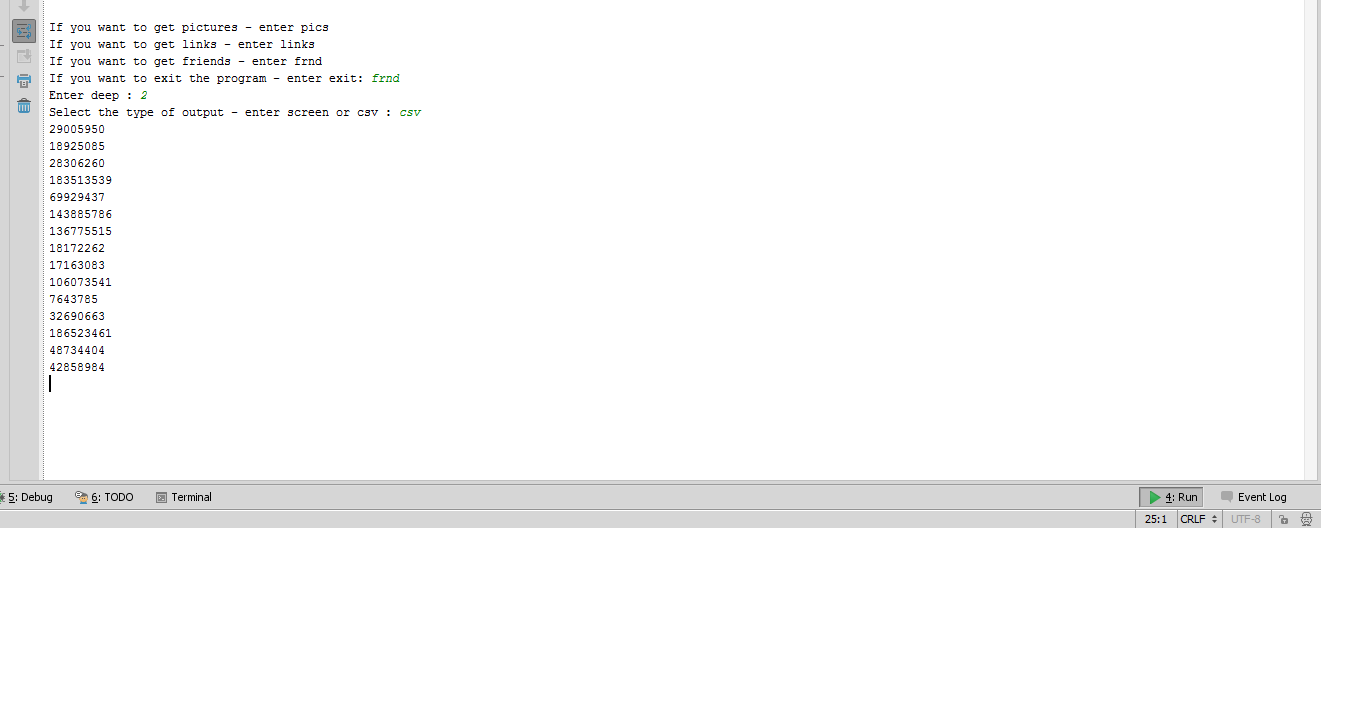


Рисунок 4 – Поиск друзей

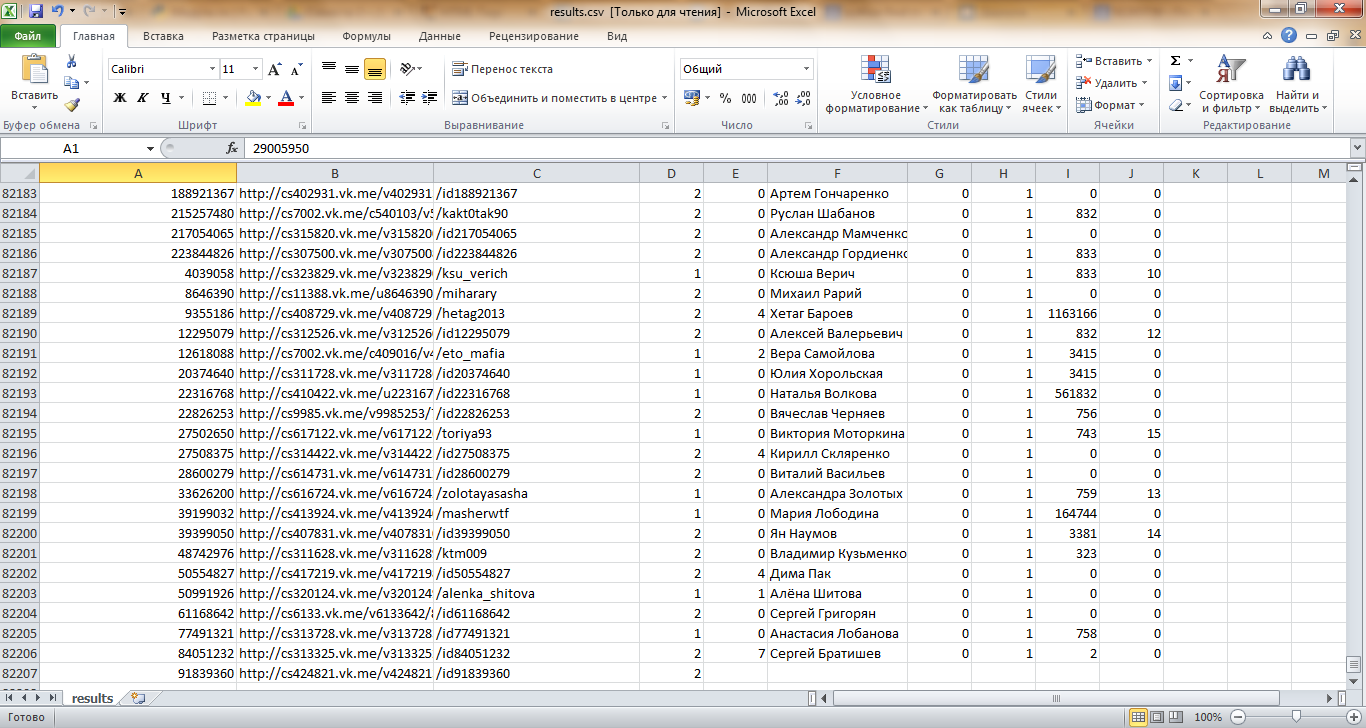


Рисунок 5 – Результат поиска друзей

# ЗАКЛЮЧЕНИЕ

Целью настоящей работы была разработка проекта сканирования web-приложений. Основной задачей являлась задача нахождения картинок, ссылок или друзей на выбор пользователя с указанной глубиной прохождения в социальной сети ‘vk.com’, проведение тестирования разработанного проекта и анализ полученных результатов.

Таким образом, можно сделать вывод, что с помощью данной программы можно найти все ссылки и картинки на странице с глубиной, выбранной пользователем, осуществить поиск всех друзей, и, в свою очередь, друзей их друзей. По моему мнению, мой проект не достаточно полезен, т.к. поиск сущностей занимает большое количество времени. В будущем постараюсь решить эту проблему. В рамках данной курсовой работы цель и все поставленные задачи можно считать выполненными.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Аутентификация и авторизация: новый взгляд. Журнал Connect! Мир связи.
2. Авторизация. [Электронный ресурс]. − URL: http://ru.wikipedia.org/wiki/Авторизация (дата обращения: 26.11.13)
3. Регулярные выражения. [Электронный ресурс]. − URL: http://ru.wikipedia.org/wiki/Регулярные\_выражения (дата обращения: 20.11.13)
4. Методы моделирования и анализа вычислительных систем. Учебное пособие. 1996г.В.Л. Бурковский, Л.В. Холопкина, Н.Л. Райхель, О.Я. Кравец.
5. Html tags. [Электронный ресурс]. − URL: http://htmldog.com/reference/htmltags/ (дата обращения: 26.11.13)
6. Requests: HTTP for Humans. [Электронный ресурс]. − URL: http://www.python-requests.org/en/latest/ (дата обращения: 6.12.13)
7. Quickstart. [Электронный ресурс]. − URL: http://www.python-requests.org/en/latest/user/quickstart/#make-a-request (дата обращения: 6.12.13)

# ПРИЛОЖЕНИЕ A

**А Исходный текст программы**

Листинг А - final\_web\_app.py

import requests

import re

import json

import csv

import os

input\_log\_pass = raw\_input("Hello! Greetings from VK. Enter your login and password: ")

session = requests.session()

log = {'email': input\_log\_pass.split()[0], 'pass': input\_log\_pass.split()[1]}

log\_url = "https://login.vk.com/?act=login"

res = session.post(log\_url, data=log)

def encode\_for\_excel(values):

return [value.encode("utf8") for value in values]

f\_out = open('D:\\results.csv', 'wb')

f\_out.write(u'\ufeff'.encode('utf8'))

writer = csv.writer(f\_out, delimiter=';', lineterminator='\n',

quoting=csv.QUOTE\_ALL, dialect='excel')

def search\_pics\_and\_links(parameter, output\_tp, inp\_deep, completed\_links, link='http://vk.com/nikaromane', count=0):

count = +1

res = session.get(link)

if output\_tp == 'screen':

print res.url + '\n'

rt = res.text

if parameter == 'pics':

picture\_urls = re.findall(ur'<img\s+.\*?src="(.\*?)".\*?', rt)

if picture\_urls:

if output\_tp == 'screen':

print 'Pictures on this page: ' + str(picture\_urls) + '\n'

elif output\_tp == 'csv':

picture\_urls[0] = res.url

writer.writerow(encode\_for\_excel(picture\_urls))

else:

if output\_tp == 'screen':

print 'Pictures not found \n'

if parameter == 'links' or parameter == 'pics':

urls = re.findall(ur'<a\s.\*?href="(.\*?)".\*?</a>', rt)

if urls:

if parameter == 'links':

if output\_tp == 'screen':

print 'Links on this page: ' + str(urls) + '\n'

elif output\_tp == 'csv':

urls[0] = res.url

writer.writerow(encode\_for\_excel(urls))

for url in urls:

status = False

if not url[:7] == 'http://':

url = 'http://vk.com/'+url

for c\_link in completed\_links:

if url == c\_link:

status = True

if status is False and count < int(inp\_deep):

completed\_links.append(url)

search\_pics\_and\_links(parameter, output\_tp, str(int(inp\_deep)-1), completed\_links, url, count)

def search\_friends(how\_deep, output\_tp, friends\_data\_id='nikaromane', count=0):

friends\_data = {"act": "load\_friends\_silent",

"al": "1",

"gid": "0",

"id": friends\_data\_id}

friends\_url = "http://vk.com/al\_friends.php"

login\_html = session.post(friends\_url, data=friends\_data)

start\_ind = login\_html.text.find('{')

if friends\_data\_id == 'nikaromane':

end\_ind = login\_html.text.find(',"all\_requests"')

else:

end\_ind = login\_html.text.find('}<!><!json>')

json\_data = login\_html.text[start\_ind + 7: end\_ind]

if json\_data:

json\_data = json\_data.replace("'", '"')

if output\_tp == 'screen':

print friends\_data\_id + '\n' + json\_data + '\n'

js = json.loads(json\_data)

if output\_tp == 'csv':

for ljs in js:

writer.writerow(encode\_for\_excel(ljs))

count = +1

if int(how\_deep) > count:

for i in range(0, len(js)):

id = js[i][0]

search\_friends(str(int(how\_deep)-1), output\_tp, id, count)

else:

if output\_tp == 'screen':

print '\n---------------------------\n'

if output\_tp == 'csv':

writer.writerow(encode\_for\_excel(['---------------------------']))

while True:

param = raw\_input("\nIf you want to get pictures - enter pics \nIf you want to get links - enter links \n"

"If you want to get friends - enter frnd\n"

"If you want to exit the program - enter exit: ")

if param != 'exit':

input\_deep = raw\_input('Enter deep : ')

if input\_deep:

output\_type = raw\_input('Select the type of output - enter screen or csv : ')

if param == 'frnd':

search\_friends(input\_deep, output\_type)

writer.writerow(encode\_for\_excel(['\n']))

if param == 'links' or param == 'pics':

compl\_links = []

search\_pics\_and\_links(param, output\_type, input\_deep, compl\_links)

for i in range(0,5):

writer.writerow(encode\_for\_excel(['\n']))

os.system("start excel.exe D:\\results.csv")

else:

print 'Wrong deep'

if param == 'exit':

print 'See you later, bye!'

break

f\_out.close()