МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра "Системи автоматизованого проектування"

Звіт

до лабораторної роботи №6

на тему: ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ВИКОРИСТАННЯ РЕГУЛЯРНИХ ВИРАЗІВ ДЛЯ ОБРОБКИ ТЕКСТУ. ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.

з дисципліни "Комп'ютерна лінгвістика"

Виконала:
студентка групи ПРЛм-11
Гарбуз Л.В.
Прийняв:
викладач
Дупак Б.П.

Львів 2015

Мета роботи: вивчення основ програмування на мові Python. Використання регулярних виразів для обробки текстів.

Тексти програм на мові *Python*.

Варіант – 3

1. Описати, які класи стрічок відповідають наступному регулярному виразу. [a-zA-Z]+. Результати перевірити використовуючи nltk.re_show().

Шукаємо всі слова, які складаються з малих та великих літер англійського алфавіту:

```
from __future__ import division
import nltk, re, pprint
f = open('D:/corpus.txt')
raw = f.read()
nltk.re_show('[a-zA-Z]+', raw)
```

```
>>>
{Many} {websites} {will} {try} {to} {tell} {you} {which} {careers} {offer} {the}
 {best} {prospects} {for} {the} {future}. {Their} {choices} {are} {usually} {bas
ed} {on} {percentage} {growth} {statistics} {for} {recent} {years}, {which} {are
} {a} {clear} {indicator} {of} {where} {the} {numbers} {of} {jobs} {are} {increa
sing}.
>>>
```

2. Описати, які класи стрічок відповідають наступному регулярному виразу. [A-Z][a-z]*. Результати перевірити використовуючи nltk.re_show().

Шукаємо послідовності символів, що починаються з великої літери англійського алфавіту:

```
>>> from __future__ import division
>>> import nltk, re, pprint
>>> line='Many websites will try to tell you which careers offer the best prospects for the future. Their choi
for recent years, which are a clear indicator of where the numbers of jobs are increasing.'
>>> nltk.re_show('[A-Z][a-z]*',line)
{Many} websites will try to tell you which careers offer the best prospects for the future. {Their} choices ar
ecent years, which are a clear indicator of where the numbers of jobs are increasing.
>>>
```

3. Описати, які класи стрічок відповідають наступному регулярному виразу. \d+(\.\d+)?. Результати перевірити використовуючи nltk.re_show().

Шукаємо послідовності символів, що складаються з цифр, які пвторюються з 1 і більше разів; (\.\d+)? – послідовність цифр після крапки не є обов’язковою:

```
from __future__ import division
import nltk, re, pprint
f = open('D:/corpus.txt')
raw = f.read()
nltk.re_show('\d+(\.\d+)?', raw)
```

```
>>>
Many websites will try to tell you which careers offer the best prospects for th
e future. Their choices are usually based on  {4.5} percentage growth statistics
 for recent years, which are a clear indicator of where the numbers of jobs are
increasing.
>>>
```

4. Описати, які класи стрічок відповідають наступному регулярному виразу. ([^aeiou][aeiou][^aeiou])*. Результати перевірити використовуючи nltk.re_show().

Шукаємо послідовності символів, що складаються з трьох символів, перший і третій з яких не є голосною, а другий – будь-яка голосна з набору [aeiou] і зустрічаються 0 і більше разів. Якщо послідовність не знайдено – виводиться {}:

```
from __future__ import division
import nltk, re, pprint
f = open('D:/corpus.txt')
raw = f.read()
nltk.re_show('([^aeiou][aeiou][^aeiou])*', raw)
```

```
>>>
{Man}y{} {websit}e{}s{} {wil}l{} {}t{}r{}y{} {to tel}l{} {}y{}o{}u{} {}w{hic}h{}
 {car}e{}e{}r{}s{ offer} {}t{he bes}t{} {}p{rospec}t{}s{} {for} {}t{he fut}u{re.
} {}T{}h{}e{}i{}r{} {}c{}h{}o{}i{ces ar}e{ us}u{}a{}l{}l{}y{} {bas}e{}d{ on} {pe
rcentag}e{} {}g{row}t{}h{} {}s{tat}i{}s{tic}s{} {for} {rec}e{}n{}t{} {}y{}e{}a{}
r{}s{},{} {}w{hic}h{ ar}e{ a }c{}l{}e{}a{}r{ indic}a{tor of} {}w{her}e{} {}t{he
number}s{ of} {job}s{ ar}e{ in}c{}r{}e{}a{sin}g{}.{}
>>>
```

5. Описати, які класи стрічок відповідають наступному регулярному виразу. \w+|[^\w\s]+.. Результати перевірити використовуючи nltk.re_show().

Шукаємо послідовності, що складаються з літер, цифр чи символів, які повторюються 1 і більше разів:

```
from __future__ import division
import nltk, re, pprint
f = open('D:/corpus.txt')
raw = f.read()
nltk.re_show('\w+|[^\w\s]+.', raw)
```

```
>>>
{Many} {websites} {will} {try} {to} {tell} {you} {which} {careers} {offer} {the}
 {best} {prospects} {for} {the} {future}{. }{Their} {choices} {are} {usually} {b
ased} {on}  {4}{.5} {percentage} {growth} {statistics} {for} {recent} {years}{,
}{which} {are} {a} {clear} {indicator} {of} {where} {the} {numbers} {of} {jobs}
{are} {increasing}.
>>>
```

6. Описати, які класи стрічок відповідають наступному регулярному виразу. p[aeiou]{,2}t Результати перевірити використовуючи nltk.re_show().

Шукаємо стрічки, які складаються з букви “p”, жодної, одної або двох голосних з набору і літери “t”:

```
from __future__ import division
import nltk, re, pprint
f = open('D:/corpus.txt')
raw = f.read()
nltk.re_show('p[aeiou]{,2}t', raw)
```

```
>>>
Many websites will try to tell you which careers offer the best prospects for th
e future. Their choices are usually based on  4.5 percentage growth statistics f
or recent years, which are a clear indicator of where the numbers of jobs are in
creasing.{pt} {pat} {pet} {pit}
>>>
```

8. Написати регулярний вираз, який встановлює відповідність наступному классу стрічок:арифметичний вираз з цілими значеннями і, який містить операції множення та додавання (2*3+8).

```
from __future__ import division
import nltk, re, pprint
f = open('D:/corpus.txt')
raw = f.read()
print 'TEXT:'
print
print raw
print 'RESULT:'
print re.findall(r"\d+[+|*]\d+[+|*]\d+", raw)
```

```
>>>
TEXT:

Many websites will try to tell you which careers offer the best prospects for th
e future. Their choices are usually based on  4.5 percentage growth statistics f
or recent years, which are a clear indicator of where the numbers of jobs are in
creasing.2*3+8
RESULT:
['2*3+8']
>>>
```

12. Написати регулярний вираз для токенізації такого тексту, як don't до do та n't? Пояснити чому цей регулярний вираз не працює: «n't|\w+».

```
import nltk
s="don't"
nltk.re_show("do|\w+",s)


>>>
{do}{n}'{t}
>>>
```

```
import nltk
s="don't"
nltk.re_show("n't|\w+",s)

>>>
{don}'{t}
>>>
```

14. Прочитати файл допомоги про функцію re.sub() використовуючи help(re.sub) . Використовуючи re.sub напишіть програму видалення HTML розмітки замінивши її на пробіли.

```
>>> from __future__ import division
>>> import nltk, re, pprint
>>> help(re.sub)
Help on function sub in module re:

sub(pattern, repl, string, count=0, flags=0)
    Return the string obtained by replacing the leftmost
    non-overlapping occurrences of the pattern in string by the
    replacement repl.  repl can be either a string or a callable;
    if a string, backslash escapes in it are processed.  If it is
    a callable, it's passed the match object and must return
    a replacement string to be used.

>>> from urllib import urlopen
>>> url="http://www.bbc.com/news/world-europe-34442127.stn"
>>> html=urlopen(url).read()
>>> html[:350]
' <!DOCTYPE html>\n<html lang="en" id="responsive-news" prefix="og: http://ogp.m
e/ns#">\n<head >\n    <meta charset="utf-8">\n    <meta http-equiv="X-UA-Compati
ble" content="IE=edge,chrome=1">\n    <title>Ukraine crisis: OSCE encouraged as
rivals confirm pullback - BBC News</title>\n    <meta name="description" content
="Officials confirm that Ukrainian go'
>>> s=html[:350]
>>> re.sub(r"<.*>",'',s)
' \n\n\n    \n    \n    \n    <meta name="description" content="Officials confir
m that Ukrainian go'
>>>
```

15. Прочитати Додаток А. Дослідити явища описані у Додатку А використовуючи корпуси текстів та метод findall()для пошуку в токенізованому тексті.

```
import nltk
from nltk.corpus import gutenberg, nps_chat, brown
moby = nltk.Text(gutenberg.words('melville-moby_dick.txt'))
print moby.findall(r"<the> <best> <.*> <can>")
chat=nltk.Text (nps_chat.words())
print chat.findall(r"<the> <best> <.*> <can>")


>>>
the best we can
None

None
>>>
```

Висновок: на цій лабораторній роботі я у своїх програмах використовувала регулярні вирази для обробки текстів.