МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра “Системи автоматизованого проектування”

Звіт

до лабораторної роботи №2

на тему: “ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ. ОСНОВИ ПРОГРАМУВАННЯ НА МОВІ PYTHON(частина 2) ”

з дисципліни “Комп’ютерна лінгвістика”

Виконала:

студентка групи ПРЛм-11

Липак О.В.

Прийняв:

викладач

Дупак Б.П.

Львів-2015

Мета роботи: вивчити основи програмування на мові *Python*, ознайомитись з контрольними структурами та класом FreqDist.

Теоретичні відомості.

Python підтримує широкий набір операторів для встановлення взаємозв'язків між змінними (значеннями). Повний набір цих операторів наведений у таблиці 1.

Таблиця 1.

Operator	Relationship
<	less than
<=	less than or equal to
==	equal to (note this is two not one = sign)
!=	not equal to
>	greater than
>=	greater than or equal to

Загальна схема роботи цих прикладів ([w for w in text if *condition*]), де *condition* умова, яка справджується або ні (приймає значення True або False).

Звичайно ми використовуємо умовні оператори, як частину *If* операторів. Для перевірки властивостей окремих слів існує набір наступних функцій (Таблиця2.).

Функція	Пояснення
s.startswith(t)	чи починається s з t
s.endswith(t)	чи закінчується s на t
t in s	Чи t міститься в s
s.islower()	Чи всі символи в s є малі
s.isupper()	Чи всі символи в s є великі
s.isalpha()	Чи всі символи в s є букви
s.isalnum()	Чи всі символи в s є букви і цифри
s.isdigit()	Чи всі символи в s є цифри

Функція	Пояснення
s.istitle()	Чи всі слова в s є з великої літери

В даних прикладах наступні вирази: [f(w) for ...] or [w.f() for ...], де f це функція, яка або визначає довжину слова або перетворює малі літери на великі. В кожному з цих прикладів здійснюється обробка кожного елемента списку. Змінній W послідовно присвоюються значення слів з текста і над цією змінною виконуються передбачені програмою дії. Такий запис [f(w) for ...] називається "list comprehension." (включення списків або спискові висловлювання) і є важливим для написання та розуміння програм на Python.

Для автоматичного визначення слів, які є найбільш інформативними для текстів певного жанру або певної тематики спочатку інтуітивно виникає думка побудувати частотний список або частотний розподіл. Частотний розподіл вказує на частоту з якою в тексті зустрічається кожне зі слів. Такий частотний список називають розподілом тому, що він вказує яким чином загальна кількість слів розподіляється між словниковими статями (оригінальні слова) в тексті. Враховуючи що побудова частотних розподілів часто необхідна при обробці природної мови в NLTK реалізовано окремий клас FreqDist в модулі nltk.probability . Застосуємо цей клас для знаходження 50 найчастотніших слів в тексті *Moby Dick*.

Тексти програм на мові ***Python.***

Варіант №7

1. Створіть змінну *sentence* і присвойте їй значення ‘she sells sea shells by the sea shore’ та напишіть фрагмент програми для виведення на екран всіх слів які починаються з ‘sh’.

```
>>> sentence = 'she sells sea shells by the sea shore'
>>> words = sentence.split ()
>>> for word in words:
        if word.startswith ('sh'):
                print word


she
shells
shore
```

4. Напишіть програму, яка видаляє всі голосні зі стрічки, яка відповідає імені, по батькові та прізвищу студента. Програма повинна здійснювати наступну послідовність дій: створення початкової стрічки; створення стрічки, у якій буде зберігатися результат; for цикл для обробки стрічки символ за символом і запису неголосних символів в результуючу стрічку.

```
>>> msg = 'Olena Volodymyrivna Lypak'
>>> vowels = ['a', 'i', 'y', 'o', 'e']
>>> result = ("".join ([w for w in msg.lower() if w not in vowels]))
>>> result
'ln vldmrvn lpk'
```

8. Виконати наступні приклади і пояснити різницю між ними

w.isupper()

not w.islower()

```
>>> W =('OLENA')
>>> W.isupper()
True
>>> not W.islower()
True
```

Функція ***w.isupper()*** перевіряє чи дані літери належать до великого регістру, а функція ***not w.islower()*** перевіряє чи дані літери не належать до малого регістру.

9. Знайдіть в тексті № 5 всі слова довжина яких дорівнює 4 і побудуйте для них частотний розподіл.

```
>>> fdist5 = FreqDist (sorted ([w for w in set (text5) if len(w) == 4]))
>>> fdist5
<FreqDist with 1181 samples and 1181 outcomes>
>>> vocabulary5 = fdist5.keys()
>>> vocabulary5[:50]
[u'!!!!', u'!!!.', u'!...', u'!???', u'"...', u'####', u'((((', u'))))', u
u'1900', u'1930', u'1980', u'1985', u'1996', u'1cos', u'2006', u'2:55', u'
'6:53', u'7:45', u'9.53', u'98.5', u'98.6', u'9:10', u':o *', u'; ..', u'<
```

11. Напишіть вираз для знаходження в тексті №6 всіх слів які відповідають наступним вимогам: закінчуються на ize; містять літеру z; містять послідовність літер pt; написані з великої літери. Результат представити, як список слів.

```
>>> fdist6 = FreqDist (text6)
>>> sorted ([w for w in set (text6) if w.endswith('ize') or 'z' in w or 'pt' in w or w.istitle()])
[u'A', u'Aaaaaaaaah', u'Aaaaaaaah', u'Aaaaaah', u'Aaaah', u'Aaaaugh', u'Aaagh', u'Aaah', u'Aaauggh', u
uuuuugh', u'Aauuuves', u'Action', u'Actually', u'African', u'Ages', u'Aggh', u'Agh', u'Ah', u'Ahh', u'
', u'Anarcho', u'And', u'Angnor', u'Anthrax', u'Antioch', u'Anybody', u'Anyway', u'Apples', u'Aramaic'
```

20. Побудуйте колокації для текстів №2 та №4. Результати порівняйте.

```
>>> import nltk
>>> from nltk.book import*
*** Introductory Examples for the NLTK Book ***
Loading text1, ..., text9 and sent1, ..., sent9
Type the name of the text or sentence to view it.
Type: 'texts()' or 'sents()' to list the materials.
text1: Moby Dick by Herman Melville 1851
text2: Sense and Sensibility by Jane Austen 1811
text3: The Book of Genesis
text4: Inaugural Address Corpus
text5: Chat Corpus
text6: Monty Python and the Holy Grail
text7: Wall Street Journal
text8: Personals Corpus
text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

```
>>> text2.collocations()
Building collocations list
Colonel Brandon; Sir John; Lady Middleton; Miss Dashwood; every thing;
thousand pounds; dare say; Miss Steeles; said Elinor; Miss Steele;
every body; John Dashwood; great deal; Harley Street; Berkeley Street;
Miss Dashwoods; young man; Combe Magna; every day; next morning
>>> text4.collocations()
Building collocations list
United States; fellow citizens; four years; years ago; Federal
Government; General Government; American people; Vice President; Old
World; Almighty God; Fellow citizens; Chief Magistrate; Chief Justice;
God bless; every citizen; Indian tribes; public debt; one another;
foreign nations; political parties
>>> a=[]
>>> a.append(text2.collocations)
>>> b=[]
>>> b.append(text4.collocations)
>>> if (b==a or a==b):
        print 'Collocations are identical'
else:
        print 'Collocations are not identical'


Collocations are not identical
```

Висновок: у цій лабораторній роботі я вивчила основи програмування на мові *Python*, ознайомилась з контрольними структурами та класом FreqDist.