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

Кафедра “Системи автоматизованого проектування”

Звіт
до лабораторної роботи №1
на тему: “ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.
ОСНОВИ ПРОГРАМУВАННЯ НА МОВІ PYTHON(частина 1) ”
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*.

Теоретичні відомості.

Python - це проста і потужна об'єктно-орієнтована мова програмування високого рівня з чудовими можливостями для обробки лінгвістичних даних.

Natural Language Toolk (NLTK) – набір *Python* бібліотек, які призначені для аналізу текстів природною мовою. *NLTK* дозволяє здійснювати символьний та статистичний аналіз текстів, створювати графічні звіти та містить детальну документацію і використовується в проектах з лінгвістики, штучного інтелекту, машинного навчання, автоматизації документообігу. Його можна застосовувати як начальний комплекс, готовий аналітичний інструмент або платформу для створення прикладних систем опрацювання текстів. *NLTK* вільно розповсюджується (http://www.nltk.org) і всі бажаючі можуть його встановити згідно інструкції розробників.

Python – інтерпретаційна мова, яка дозволяє зекономити час, що витрачається на компіляцію. Інтерпретатор можна використовувати інтерактивно, що дозволяє експериментувати з можливостями мови і створювати фрагменти програм або тестувати окремі функції. Інтерпретатор – це програма яка виконує Python програми.

```
Python 2.4.3 (#1, Mar 30 2006, 11:02:16)
[GCC 4.0.1 (Apple Computer, Inc. build 5250)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

При запуску інтерпретатора ми бачимо інформацію про його версію, додаткову інформацію і запрошення >>> вводити оператори Python. У випадку використання Interactive DeveLopment Environment (IDLE) нам доступні додаткові зручності, зокрема у відображенні тексту програми на екрані.

Текст або частини тексту в програмах на Python представляються за допомогою *стрічок* (*string*) і повинен відділятися від решти програми лапками (одинарними, подвійними або потрійними).

Якщо стрічка містить одинарні лапки необхідно використовувати лівий слеш перед апострофом для того, щоб символ апострофа не розглядався як символ завершення стрічки або використовувати подвійні лапки. Якщо цього не зробити то отримаємо помилку.

Деколи стрічки можуть складатися з декількох рядків. Python забезпечує декілька способів роботи з ними. В наступному прикладі послідовність з двох стрічок об'єднується в одну. Потрібно використовувати лівий слеш або круглі дужки для того щоб інтерпретатор знав що ввід стрічки ще не завершився після введення першого рядка.

Lists(списки) – тип даних для опису послідовності значень. В *Python* списки представляються як послідовність записана через кому і у квадратних дужках.

Tuples(кортежі) – подібний до списків тип даних але подібно до стрічок кортежі також не можемо змінювати.

Тексти програм на мові *Python*.

Варіант - 3

3.3 Здійснити арифметичні операції зі стрічкою msg.

```
>>> msg = 'Lidia Harbuz'
>>> msg
'Lidia Harbuz'
>>> 'Lidia Harbuz'.count ('i')
2
>>> 'Lidia Harbuz' .count ('r')
1
```

3.7 Використовуючи зрізи видаліть афікси у наступних словоформах: dish-es, run-ning, nation-ality, un-do, pre-heat.

```
>>> str = 'dishes'
>>> str [0:-2]
'dish'
>>> stp = 'running'
>>> stp [0: -4]
'run'
>>> stf = 'nationality'
>>> stf [0: -5]
'nation'

>>> stm = 'undo'
>>> stm [2:]
'do'

>>> stl = 'preheat'
>>> stl [3:]
'heat'
```

3.10 Поясніть результат виконання msg[::-1].

```
>>> msg [::-1]
'zubraH aidiL'
```

Зріз [::-1] міняє всі символи у стрічці місцями з кінця на початок.

3.13 Представити прізвище, ім’я та по батькові як список стрічок. Використовуючи метод .reverse() та зріз [::-1] змінити стрічку. Результати пояснити.

```
>>> a = ['Harbuz', 'Lidia', 'Viktorivna']
>>> a
['Harbuz', 'Lidia', 'Viktorivna']
>>> b = a.reverse()
>>> a
['Viktorivna', 'Lidia', 'Harbuz']
>>> b
>>> ['Viktorivna', 'Lidia', 'Harbuz'][::-1]
['Harbuz', 'Lidia', 'Viktorivna']
```

Метод .reverse() міняє місцями списки стрічок без зміни у них символів з кінця на початок, а зріз [::-1] повертає списки у початкове положення.

3.16 Створити змінну words яка містить список слів. Дослідіть операції words.sort() і sorted(words).

```
>>> words = ['fjgjgjg', 'ffoof', 'sjshd']
>>> words.sort()
>>> words
['ffoof', 'fjgjgjg', 'sjshd']
>>> lida = sorted(words)
>>> lida
['ffoof', 'fjgjgjg', 'sjshd']
```

3.21 Напишіть програму, яка створить стрічку в якій будуть записані другі символи всіх слів з стрічки silly.

```
>>> silly = 'sdshebadslkpfmekad'
>>> len(silly)
18
```

```
>>> silly [1:18:2]
'dhbdlpmkd'
```

3.24 Використайте функцію index() наступним чином ’inexpressible’.index(’e’). Що станеться якщо виконати ’inexpressible’.index(’re’)

```
>>> 'inexpressible'.index('e')
2
>>> 'inexpressible'.index('re')
5
```

Висновок: у цій лабораторній роботі я вивчила основи програмування на мові *Python,* дізналася, що є такі типи даних: стрічка, список, кортеж, а також дізналася, які є методи роботи зі стрічкою та зі списком.