МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА ШЕВЧЕНКА

кафедра інформаційних систем та технологій

ЗВІТ

із лабораторної роботи № **1**

з дисципліни «**Технології програмування об'єктів лінгвістичної предметної галузі**»

на тему: Базові завдання автоматичного опрацювання тексту

Варіант № **2-06**

Виконав:

Студент групи №2

Кличлієв К. С.

Перевірив:

доц. Костіков М. П.

Київ — 2022

Дані з індивідуального варіанту № 2-06

1) Замінити всі літери «g» на «h».
2) Розбити на слова та вивести кожне слово через «&».
3) Видалити з рядка всі літери, що позначають голосні звуки.
4) Транслітерація українська-шведська.

Мета роботи

1) Закріпити набуті знання з базового автоматичного опрацювання тексту шляхом реалізації наступних завдань: заміна символів у тексті, токенізація (розбиття тексту на слова) вбудованими методами мови програмування Python, видалення літер на позначення голосних звуків.
2) Розглянути поняття “транскрипція” та “транслітерація”, розробити власний метод транслітерації для пари мов українська-шведська.
3) З’ясувати, які існують методи заміни символів у рядках у мові програмування Python.

Середовище розробки

Мова програмування: Python

Назва та версія IDE: Jupyter Notebook 6.4.11

Назва і версія ОС: Linux Ubuntu 21.04

Мова ОС: англійська

Хід роботи

1) Створюємо консольний проект, який виводить на екран прізвище, ім’я, групу виконавця та номер ЛР, а також просить користувача ввести з клавіатури рядок тексту і зберігає його в змінну input_string:

2) Використовуючи вбудовані методи мови програмування Python для опрацювання тексту, було оброблено введений користувачем рядок за таким алгоритмом:
   - замінити всі літери “g” на “h”
   - розбити на слова та вивести кожне слово через “&”
   - видалити з рядка всі літери, що позначають голосні звуки

Заміна літер здійснена за допомогою методу replace(), для розбиття слів використано метод split(), а голосні звуки видалено шляхом перебирання всіх літер у вхідному тексті і відсіювання літер на позначення голосних звуків, що були попередньо збережені у змінній vowels.

Код програми:

```
print("Кличлієв Кирило\nГрупа №2\nЛабораторна робота №1\n")

input_string = input("Введіть рядок тексту: ")

new_string = input_string.replace("g", "h")
final_string = new_string.replace("G", "H")
print(f"\nЗамінити всі літери 'g' на 'h': {final_string}\n")

print("Розбити на слова та вивести кожне слово через символ «&»: ")
for i in input_string.split(" "):
    print(f'&{i}')

vowels = "aeiouAEIOU"
no_vowels = ''.join([l for l in list(input_string) if l not in vowels])
print(f"\nВидалити з рядка всі літери, що позначають голосні звуки: {no_vowels}")
```

Результат проведення вищеописаних операцій над вхідним рядком:

```
Кличлієв Кирило
Група №2
Лабораторна робота №1

Введіть рядок тексту: Swedish language is a North Germanic language

Замінити всі літери 'g' на 'h': Swedish lanhuahe is a North Hermanic lanhuahe

Розбити на слова та вивести кожне слово через символ «&»:
&Swedish
&language
&is
&a
&North
&Germanic
&language

Видалити з рядка всі літери, що позначають голосні звуки: Swdsh lngg s  Nrth Grmnc lngg
```

3) Створюємо власний метод для транслітерації тексту з української мови на шведську. З правилами, на основі яких побудовано транслітератор, можна ознайомитися за посиланням.

Код програми:

```
input_text = list(input('Введіть текст українською мовою: '))


def transliterate(input_string):

   uk_sw_dict = {

       'а':'a', 'б':'b', 'в':'v', 'г':'h', 'д':'d', 'е':'e', 'ж':'zj',
'з':'z', 'и':'y', 'й':'j', 'к':'k', 'л':'l', 'м':'m', 'н':'n',

       'о':'o', 'п':'p', 'р':'r', 'с':'s', 'т':'t', 'у':'u', 'ф':'f',
'х':'ch', 'ц':'ts', 'ч':'tj', 'ш':'sj', 'щ':'sjtj', 'ь':'',

       'ю':['ju', 'iu', 'u'], 'я':['ja', 'ia', 'a'], 'є':'je', 'ґ':'g',
'і':'i', 'ї':'ji', ' ': ' ', "'":"", '’':'', '’':'',

       'А':'A', 'Б':'B', 'В':'V', 'Г':'H', 'Д':'D', 'Е':'E', 'Ж':'ZJ',
'З':'Z', 'И':'Y', 'Й':'J', 'К':'K', 'Л':'L', 'М':'M', 'Н':'N',

       'О':'O', 'П':'P', 'Р':'R', 'С':'S', 'Т':'T', 'У':'U',
'Ф':'F','Х':'CH', 'Ц':'TS', 'Ч':'TJ', 'Ш':'SJ', 'Щ':'SJTJ', 'Ь':'',

       'Ю':['JU', 'IU', 'U'], 'Я':['JA', 'IA', 'A'], 'Є':'JE', 'Ґ':'G',
'І':'I', 'Ї':'JI'

   }
```

```
    transliterated = ''


    for count, letter in enumerate(input_text):
        if letter=='я':
            if input_text[count - 1] == 'ч' or input_text[count - 1] ==
'ш' or input_text[count - 1] == 'щ' or input_text[count - 1] == 'ж':
                transliterated += uk_sw_dict['я'][2]
            elif input_text[count - 1] == 'Ч' or input_text[count - 1]
== 'Ш' or input_text[count - 1] == 'Щ' or input_text[count - 1] == 'Ж':
                transliterated += uk_sw_dict['я'][2]
            elif input_text[count - 1] == 'з' or input_text[count - 1]
== 'с' or input_text[count - 1] == 'т' or input_text[count - 1] == 'ц':
                transliterated += uk_sw_dict['я'][1]
            elif input_text[count - 1] == 'З' or input_text[count - 1]
== 'С' or input_text[count - 1] == 'Т' or input_text[count - 1] == 'Ц':
                transliterated += uk_sw_dict['я'][1]
            else:
                transliterated += uk_sw_dict['я'][0]
        elif letter=='ю':
            if input_text[count - 1] == 'ч' or input_text[count - 1] ==
'ш' or input_text[count - 1] == 'щ' or input_text[count - 1] == 'ж':
                transliterated += uk_sw_dict['ю'][2]
            elif input_text[count - 1] == 'Ч' or input_text[count - 1]
== 'Ш' or input_text[count - 1] == 'Щ' or input_text[count - 1] == 'Ж':
                transliterated += uk_sw_dict['ю'][2]
            elif input_text[count - 1] == 'з' or input_text[count - 1]
== 'с' or input_text[count - 1] == 'т' or input_text[count - 1] == 'ц':
                transliterated += uk_sw_dict['ю'][1]
            elif input_text[count - 1] == 'З' or input_text[count - 1]
== 'С' or input_text[count - 1] == 'Т' or input_text[count - 1] == 'Ц':
                transliterated += uk_sw_dict['ю'][1]
```

```
            else:
                transliterated += uk_sw_dict['ю'][0]
        elif letter in uk_sw_dict:
            transliterated += uk_sw_dict[letter]
        else:
            transliterated += letter

    print(f"\nТранслітерація шведською: {transliterated}")

transliterate(input_text)
```

Суть роботи транслітератора: на вхід приймаємо рядок тексту українською мовою і передаємо його як аргумент у функцію **transliterate()**. Функція має такі складові:

- **словник uk_sw_dict**, де ключі - букви українського алфавіту, а значення - їхні шведські відповідники. Зазвичай співвідношення між ключами і значеннями 1:1, однак українські літери “ю” та “я” мають одразу по три шведські еквіваленти, які збережені в списку.
- **transliterated** - змінна, куди буде збережено транслітерований текст шведською.
- цикл **for** перебирає всі символи у вхідному тексті і перевіряє їх за 4 умовами. Метод **enumerate()** накладаємо на текст, щоб присвоїти кожному символу в нашому рядку номер (count), за яким можемо витягнути з нього, зокрема, й попередні елементи. Умови для транслітерації:
    1) Якщо символ є літерою “я”, то нам необхідно додатково перевірити попередню літеру у вхідному тексті (input_text[count - 1]), оскільки від неї залежить, який із трьох

шведських відповідників “ja”, “ia” чи “a” ми використовуємо для транслітерації. Наприклад, якщо в українському тексті літера “я” слідує за “ч”, то у шведській ми “я” передаємо як “a”, а якщо перед “я” стоїть “c”, то при транслітерації ми її замінюємо на “ia”. У більшості випадків “я” передаємо дифтонгом “ja”.

2) Другий випадок, коли нам треба перевірити попередній символ, - це літера “ю”, яка має наступні шведські відповідники: “ju”, “iu” та “u”. Наприклад, якщо в українському тексті літера “ю” слідує за “ш”, то у шведській “ю” передаємо як “u”, а якщо перед “ю” стоїть “з”, то при транслітерації ми її замінюємо на “iu”. У більшості випадків “ю” передаємо дифтонгом “ju”.
3) Третя умова для всіх інших літер і апострофа: якщо певний символ із вхідного рядка є в ключах словника, то додаємо в transliterated єдине значення цього ключа.
4) Останню умову else використовуємо для ідентифікації всіх інших символів (цифри, пробіли, пунктуація) і додаємо їх в transliterated без змін.

- **print(f"Транслітерація шведською: {transliterated}")** - виводить на екран транслітерований текст.

transliterate(input_text) - пропускаємо наш вхідний рядок input_text через функцію transliterate() - отримуємо транслітерацію шведською мовою. На ілюстрації нижче подано результати транслітерації фрагмента з книги Ліни Костенко “Записки українського самасшедшого”:

```
Введіть текст українською мовою: Мужчини зникають як явище. їхнє місце посіли круті — ерзац, замінник, гібрид гамадрила й Шва
рценеґера. Словом, еволюція вспак. Час хисткий і непевний. У нас тепер «перезмінка». Думали, що нове століття почнеться з кру
глої дати, з 2000 року, а виявляється, що з наступного, коли крізь нуль проклюнеться одиничка. Але це арифметика. Все одно вж
е цей чорний лебідь з трьома нулями завершує свій царственний круг. Хоча, може, це й не лебідь, а Вселенський Змій з хвостом,
звинутим у три нулі — символ страшний, космогонічний, не треба б людству накликати його на себе.

Транслітерація шведською: Muzjtjyny znykajut jak javysjtje. jichnje mistse posily kruti — erzats, zaminnyk, hibryd hamadryla
j SJvartsenegera. Slovom, evoljutsija vspak. TJas chystkyj i nepevnyj. U nas teper «perezminka». Dumaly, sjtjo nove stolittia
potjnetsia z kruhloji daty, z 2000 roku, a vyjavljajetsia, sjtjo z nastupnoho, koly kriz nul prokljunetsia odynytjka. Ale tse
aryfmetyka. Vse odno vzje tsej tjornyj lebid z troma nuljamy zaversjuje svij tsarstvennyj kruh. CHotja, mozje, tse j ne lebi
d, a Vselenskyj Zmij z chvostom, zvynutym u try nuli — symvol strasjnyj, kosmohonitjnyj, ne treba b ljudstvu naklykaty joho n
a sebe.
```

Відповідь на контрольне питання № 6

У мові програмування Python існує декілька методів для заміни символів у рядках:

1) **string.replace(old, new, count)** - вбудований метод, що бере на вхід рядок (string) і замінює певний підрядок (old) на новий (new). Параметр count - кількість разів заміни старого підрядка новим; необов’язковий:

```
myStr = "This is my first program"
newStr = myStr.replace("first", "second")
print(newStr)
```

```
This is my second program
```

```
myStr = "Thos os my forst program"
newStr = myStr.replace("o", "i", 3)
print(newStr)
```

```
This is my first program
```

2) **re.sub(old, new, string)** - шаблон регулярного виразу бібліотеки re для пошуку в рядку (string) певного підрядка (old) і заміни його новим підрядком (new)

```
import re

myStr = "German lanhuahe"
newStr = re.sub("h", "g", myStr)
print(newStr)
```

```
German language
```

3) Конвертація рядка в список і заміна елемента за його індексом в списку:

```
oldStr = "aba"

strList = list(oldStr)    # конвертація рядка в список
strList[2] = "c"    # заміна третього елемента в списку на "c"

newStr = "".join(strList)
print(newStr)
```

```
abc
```

4) **string.translate()** - метод, що повертає рядок де певні символи замінені іншими символами, вказаними в словнику:

```
# використовуємо словник із ascii з кодами щоб замінити 83 (S) на 80 (P)
myDict = {83: 80}
txt = "Hello Sam!"
print(txt.translate(myDict))
```

```
Hello Pam!
```

чи таблиці відповідності (mapping table), створеної за допомогою метода maketrans():

```
txt = "Hello Sam!"
myTable = txt.maketrans("S", "P")
print(txt.translate(myTable))
```

```
Hello Pam!
```

Висновок

В ході виконання лабораторної роботи №1 досягнуто всіх цілей, поставлених у меті роботи. Зокрема, на практиці закріплено знання з базового опрацювання тексту шляхом написання коду для вирішення наступних проблем: заміна великих та малих літер “g” на “h” у відповідному регістрі, розбиття тексту на слова та видалення з тексту літер на позначення голосних звуків.

Було реалізовано власний метод транслітерації українського тексту шведською мовою з урахуванням позиції літер відносно інших літер у слові:

```
Введіть текст українською мовою: Запоріжжя, Сян, яблуко

Транслітерація шведською: Zaporizjzja, Sian, jabluko
```

транслітерація літери “я” шведськими відповідниками “a”, “ia” та “ja” залежно від попередньої літери в українських словах

Також з’ясовано, які є методи для заміни символів у рядках у мові програмування Python. Існує 4 основні способи:

- метод replace()
- метод sub() бібліотеки re для роботи з регулярними виразами
- за допомогою конвертації рядка в список і заміни його елементів за індексом
- метод translate()

Додаток

Посилання на всі коди з ЛР №1:

https://github.com/klychliiev/lab-1.git