Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

Кафедра кібербезпеки та програмного забезпечення

Звіт

з лабораторної роботи № 4

з дисципліни “Скриптові мови програмування”

на тему

“ Робота з рядками у Python ”

Виконав студент

академічної групи КІ-20

Прокопенко Є.С.

Перевірив викладач

Савеленко О. К.

Кропивницький-2021

**Мета:** набути навичок роботи з вбудованими функціями для роботи з рядками у Python

**Хід виконання лабораторної роботи**

1. Ознайомитися з наступним лекційним матеріалом:

**Тема 4.** Функції для роботи з рядками та словниками у мові Python

4.1. Базові операції роботи з рядками

4.2. Функції та методи для роботи з рядками

4.3. Зрізи рядків

4.4. Словники. Базові операції роботи зі словниками

4.5. Методи словників

2. Розробити алгоритми для виконання завдання до даної лабораторної роботи.

3. Розробити програму на мові програмування Python версії 3.х для рішення поставленого завдання на основі розробленого алгоритму.

4. Здійснити тестування та відлагодження розробленої програми.

5. Результати виконання лабораторної роботи повинні бути відображені у звіті. Звіт виконання лабораторної роботи повинен містити:

1) Титульний лист.

2) Тема та мета лабораторної роботи.

3) Номер варіанту.

4) Завдання до лабораторної роботи.

5) Опис принципу роботи та проектних рішень, а також схеми (структурні, блок-схеми тощо) розробленої програми **(за бажанням, на додаткові бали)**.

6) Лістинг розробленої програми.

7) Скриншоти з результатами роботи розробленої програми.

8) Відповіді на контрольні питання.

6. Захистити лабораторну роботу:

1) Продемонструвати викладачу процес і результати роботи програми.

2) Відповісти на питання викладача по роботі програми, лістингу, теорії.

3)  Здати оформлений звіт з лабораторної роботи.

**Завдання 1:**

Створіть програму, яка буде складати випадкові фрази на основі трьох списків зі словами. З кожного списку вона повинна брати випадковим чином слова і поєднувати їх в одну фразу.

**Завдання 2:**

Візьміть текстовий файл, що містить Вашу улюблену художню книгу.

1. Визначте загальну кількість символів у тексті з пробілами та без пробілів.

2. Визначте загальну кількість слів у тексті, загальну кількість різних слів (без повторів) та кількість унікальних слів, що зустрічаються тільки один раз.

**Завдання 3:**

**Варіант 4.** Визначте частоту повторів 3 слів у тексті.

Лістинг модулю та програмного застосунку:

from lab03 import seach\_file

import random

def random\_sentence(list1,list2,list3):

return random.choice(list1) + " " + random.choice(list2) + " " + random.choice(list3)

def how\_word\_in\_file(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

words = 0

for string in file:

for letter in string:

if( letter.isalpha() == False and letter != ' ' ):

string = string.replace(letter,'')

words += len(list(filter(None, str(string).split(' '))))

return words

return 0

def symbols\_in\_text(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

symbols = 0

for string in file:

symbols += len(str(string))

return symbols

return 0

def symbols\_in\_text\_no\_gap(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

symbols = 0

for string in file:

symbols += len(str(string).replace(' ',''))

return symbols

return 0

def how\_word\_in\_file\_no\_replay(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

words = 0

no\_replay\_words = []

for string in file:

string = str(str(string).upper())

for letter in string:

if( letter.isalpha() == False and letter != ' ' ):

string = string.replace(letter,'')

string = list(filter(None, str(string).split(' ')))

words += len(string)

for word in string:

unical = True

if(len(no\_replay\_words) == 0):

no\_replay\_words.append(word)

else:

for reply in no\_replay\_words:

if word == reply or word == '':

unical = False

words -= 1

break

if(unical):

no\_replay\_words.append(word)

return words

return 0

def how\_word\_in\_file\_unical(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

words = 0

no\_replay\_words = []

frequency\_words = []

for string in file:

string = str(str(string).upper())

for letter in string:

if( letter.isalpha() == False and letter != ' ' ):

string = string.replace(letter,'')

string = list(filter(None, str(string).split(' ')))

words += len(string)

for word in string:

unical = True

if(len(no\_replay\_words) == 0):

no\_replay\_words.append(word)

frequency\_words.append(1)

else:

for reply in no\_replay\_words:

if word == reply or word == '':

frequency\_words[no\_replay\_words.index(reply)] += 1

unical = False

words -= 1

break

if(unical):

no\_replay\_words.append(word)

frequency\_words.append(1)

words = 0

for cout in frequency\_words:

if cout == 1:

words += 1

return words

return 0

def how\_sentences\_in\_file(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

sentense = 0

for string in file:

string = str(str(string).upper())

sentense += len(list(find\_all(str(string),'!')))

sentense += len(list(find\_all(str(string),'.')))

sentense += len(list(find\_all(str(string),'?')))

sentense += len(list(find\_all(str(string),'...')))-len(list(find\_all(str(string),'...')))\*3

return sentense

return 0

def how\_sentences\_in\_file\_question(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

sentense = 0

for string in file:

string = str(str(string).upper())

sentense += len(list(find\_all(str(string),'?')))

return sentense

return 0

def how\_sentences\_in\_file\_exclamatory(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

sentense = 0

for string in file:

string = str(str(string).upper())

sentense += len(list(find\_all(str(string),'!')))

return sentense

return 0

def how\_sentences\_in\_file\_normal(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

sentense = 0

for string in file:

string = str(str(string).upper())

sentense += len(list(find\_all(str(string),'.'))) - len(list(find\_all(str(string),'...')))\*3

return sentense

return 0

def how\_sentences\_in\_file\_three\_dots(file\_name,path\_of\_file = "./",encoding\_file = "utf-8"):

if (seach\_file(file\_name,path\_of\_file) != None):

with open(file\_name, mode="r",encoding=encoding\_file) as file:

sentense = 0

for string in file:

string = str(str(string).upper())

sentense += len(list(find\_all(str(string),'...')))

return sentense

return 0

def find\_all(a\_str, sub):

start = 0

while True:

start = a\_str.find(sub, start)

if start == -1: return

yield start

start += len(sub)

if \_\_name\_\_ == '\_\_main\_\_':

list1 = ["Наруто","Оксана","Влад","Андрій","Котик","Артем","Тетяна","Дарина","Єлизавета","Егор","Жабка"]

list2 = ["лопає","сідає","спить","наївся","страждає","відпочиває","лежить","стоїть","снідає","живе","нюхає"]

list3 = ["у ЦНТУ","вдома","на заводі","на сесії","на пуфику","на кладовищі","на стілець","на вулиці","у літаку"]

print("Випадкове речення : " + str(random\_sentence(list1, list2, list3)))

print("Кількість унікальних слів у тексті: "+ str(how\_word\_in\_file\_unical("бернар вербер древо возможностей сборник.txt")))

print("Кількість речень у тексті: "+str(how\_sentences\_in\_file("бернар вербер древо возможностей сборник.txt")))

**Тестування**

Для тестування було взято книгу Бернара Вербера «Дерево можливостей» сбірка розповідей та результати порівнювались із даними взятими із ПЗ Microsoft Word.

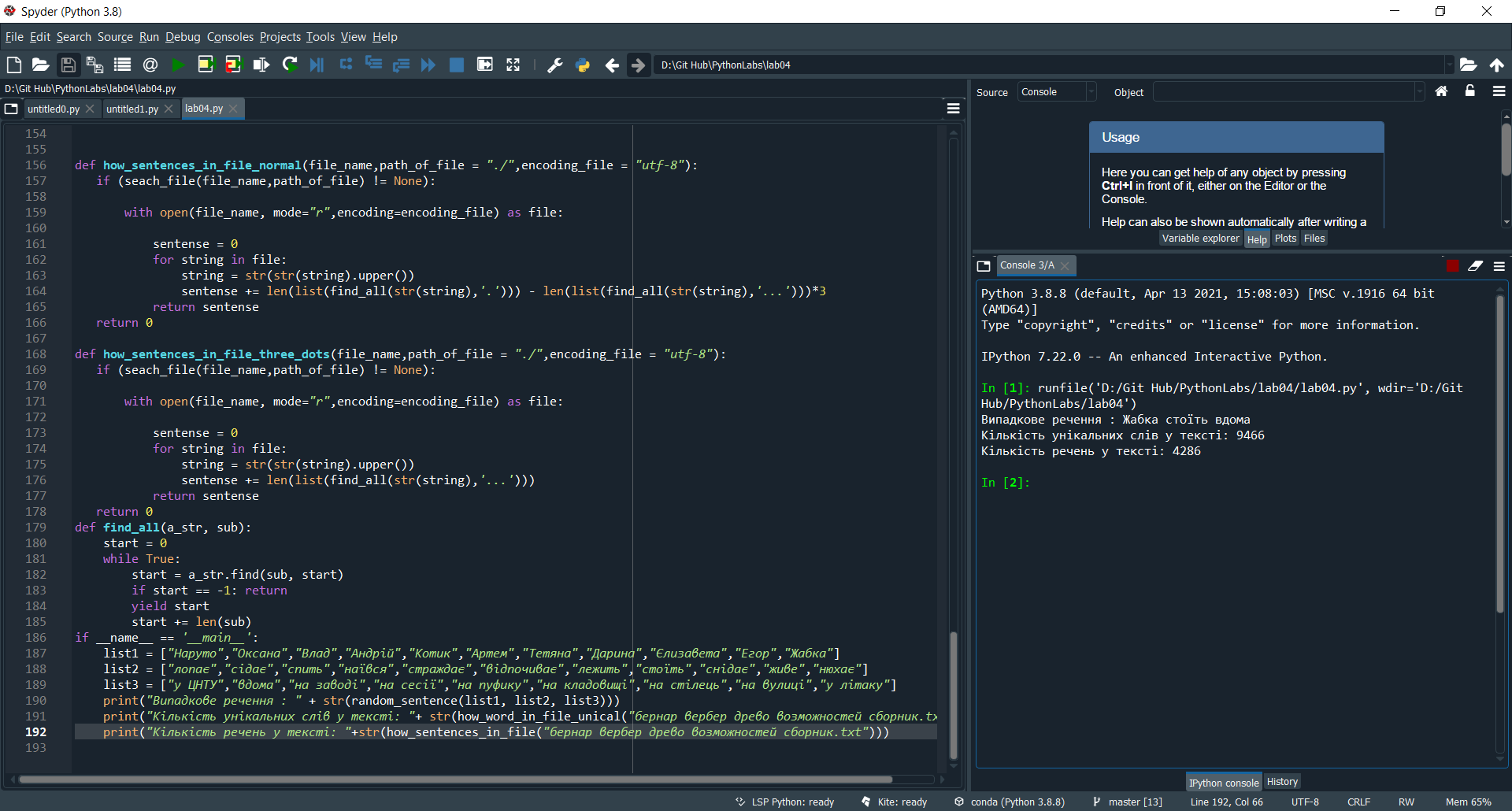


Рисунок 1 – Результат запуску програми