Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра Бі та Т

Група: ЗІПЗ-21-1

Програмування мовою Python

Лабораторна робота № 6

«Основи мови Python»

Виконав: Ліман П. О.

Прийняв: Морозов Д. С.

**Мета роботи:** ознайомитися з засобами роботи з файлами в мові Python, діями над ними.

**Виконання роботи:**

**1. Створіть новий файл numbers.txt у текстовому редакторі і запишіть у нього 10 чисел, кожне з нового рядка. Напишіть програму, яка зчитує ці числа з файла і обчислює їх суму, виводить цю суму на екран і, водночас, записує цю суму у інший файл з назвою sum\_numbers.txt.**

import os

if os.path.exists('numbers.txt'):

myFile = open('numbers.txt', 'r')

sum = 0

for number in myFile:

try:

sum += int(number)

except:

print("Помилка! У файлі є символи, що не являються числами")

myFile.close()

print(sum)

myFile = open('sum\_numbers.txt', 'w+', encoding='UTF-8')

myFile.write(f"Сума чисел з файлу numbers.txt = {sum}")

myFile.close()

else:

print("Помилка!!! Файлу не існує")

**2. Реалізуйте програму, яка зчитує довільну кількість цілих чисел, що вводяться з командного рядка, і записує у тестовий файл інформацію, щодо парності або непарності чисел.**

print("Введіть числа через пробіл: ")

numbers = list(map(int, input().split()))

myFile = open('result.txt', 'w+', encoding='UTF-8')

for i in numbers:

if i % 2 == 0:

myFile.write(f"Число {i} парне\n")

else:

myFile.write(f"число {i} не парне\n")

myFile.close()

**3. Створіть новий файл у текстовому редакторі і напишіть кілька рядків тексту у ньому про можливості Python. Кожен рядок повинен починатися з фрази: «Python можна використати для ...» . Збережіть файл з ім’ям learning\_python.txt. Напишіть програму, яка зчитує файл і виводить текст з перебором рядків файла і зі збереженням рядків у списку з подальшим сортуванням списку за довжиною рядків в ньому найбільшого до найменшого.**

myFile = open('learning\_python.txt', encoding='utf-8')

text = []

for line in myFile:

text.append(line.replace("\n", ""))

print(line.replace("\n", ""))

text.sort(key=len, reverse=True)

print('\nВідсортований список рядків від найбільшого до найменшого:')

print('------------------------------------')

#print(text)

for line in text:

print(line)

myFile.close()

**4. Прочитайте кожен рядок зі створеного у попередньому завданні файла learning\_python.txt і замініть слово Python назвою іншої мови, наприклад C при виведенні на екран. Отриманий файл має бути створений в новому каталозі, що розміщується в поточному. Відкрийте файл пострічково і дайте**

**можливість користувачеві визначити які змінені фрази є актуальними, наприклад для мови С, а які ні. Всі хибні твердження запишіть в інший файл, а істинні – в поточний.**

text.append(line.replace("\n", ""))

text = [i.replace("Python", "Swift") for i in text]

print(text)

MyFile.close()

try:

os.mkdir("newdir")

except:

print("Файл вже створений!!!")

MyFile = open('./newdir/changLang', 'w', encoding='utf-8')

for line in text:

MyFile.write(line + "\n")

MyFile.close()

MyFile = open('./newdir/changLang', 'r', encoding='utf-8')

true = []

false = []

print("Фраза про мову є правдивою [так] [ні]: ")

for line in MyFile:

print(line, end="")

print("-->", end=" ")

answer = input()

if answer == "так":

true.append(line.replace("\n", ""))

elif answer == "ні":

false.append(line.replace("\n", ""))

else:

print("Помилка!!! Введіть так або ні")

MyFile.close()

FileFalse = open('./newdir/falseOutput', 'w', encoding='utf-8')

FileTrue = open('./newdir/changLang', 'w', encoding='utf-8')

for line in true:

FileTrue.write(line + "\n")

for line in false:

FileFalse.write(line + "\n")

FileFalse.close()

FileTrue.close()

**5. Створіть порожній файл guest\_book.txt у текстовому редакторі. Напишіть програму, яка запитує у користувачів імена. При введенні кожного імені виведіть на екран рядок з вітанням для користувача і запишіть рядок вітання у файл з ім’ям guest\_book.txt. Простежте за тим, щоб кожне повідомлення розміщувалося в окремому рядку файла з зазначенням часу внесення цього повідомлення. Передбачте зазначення в файлі часу його створення і вказання в ньому часу останніх внесених змін.**

from datetime import datetime

import os

def fileRead(path, time):

with open(path, 'r', encoding='utf-8') as questBook:

text = ""

for line in questBook:

if line[0] == '2':

line = f"2. Час внесення останніх змін: {time} \n"

text += line

questBook.close()

return text

def fileCreate(path, record, time):

with open(path, 'w', encoding='utf-8') as questBook:

questBook.write(f"1. Час створення файлу: {time} \n")

questBook.write(f"2. Час внесення останніх змін: {time} \n\n")

questBook.write(f"{record} | час добавлення: {time} \n")

def fileWrite(path, text, record, time):

text += f"{record} | час добавлення: {time} \n"

with open(path, 'w', encoding='utf-8') as questBook:

questBook.write(text)

path = 'quest\_book.txt'

print("Введіть їм'я: ")

name = input()

greeting = f"Доброго дня {name}"

print(greeting)

time = datetime.now()

if os.path.isfile(path):

text = fileRead(path, time)

fileWrite(path, text, greeting, time)

else:

fileCreate(path, greeting, time)

**6. Збережіть в тектовому файлі публікацію про Python на 3000 слів англійською мовою. Напишіть програму, що аналізуватиме частоту з якою в тексті зустрічастимуться окремі літери чи слова незалежно від їх регістру. Результат робот програми має виводитись в консоль і зберігатись в окремому файлі з зазначенням часу його стоврення, часу виконнання окремих змін, результатів пошкуку і часу, що знадовся на виконнання цього пошуку.**

from datetime import datetime

import time

with open('PublicationPython.txt', 'r', encoding='UTF-8') as pubPyFile:

text = pubPyFile.read()

def strRefactor(word):

symbols = ['.', ',', ';', '!', '?', '(', ')']

for symbol in symbols:

word = word.replace(symbol, '')

return word.lower()

words = list(map(strRefactor, text.split()))

temp = []

for word in words:

if word not in temp:

temp.append(word)

with open('result.txt', 'w', encoding='UTF-8') as resultFile:

timeCreate = datetime.now()

start = time.time()

for word in temp:

countWord = words.count(word)

resultFile.write(f"{word} - Кількість повторень: {countWord} | час зміни у файлі: {datetime.now().strftime('%H:%M:%S:%f')}\n")

print(f"{word} - Кількість повторень: {countWord}")

end = time.time() - start

resultFile.write("-----------------------------------------------------------------------------\n")

resultFile.write(f"Файл було створено: {timeCreate}\n")

resultFile.write(f"Час на виконання пошуку: {end} секунд\n")

resultFile.write("-----------------------------------------------------------------------------\n")

print(len(words))

**7. Завантажте файл marks.csv і визначте кількість студентів, що**

**проходили тестування. Виведіть інформацію про те яку оцінку набрали відповідна кільксть студентів. Виведіть інформацію яку середню оцінку отримував студент за певний час виконання КМР (крок – 1 хв). Створіть текстовий файл і запишіть в нього статистику по правильним відповідям для кожного окремого питання (який відсоток правильних і неправильних відповідей на питання дали студенти). В цей же файл внесіть інформацію про 5 найкращих оцінок в співвідношенні оцінка/час витрачений складання КМР.**

NUMBER\_QUESTIONS = 20

def is\_number(string):

try:

float(string)

return True

except ValueError:

return False

def check\_repeat\_grade(grade\_check, grades):

count = 0

for gradeSt in grades:

if float(grade\_check) == float(gradeSt):

count = count + 1

return count

def separation():

return "========================================================================\n"

with open('marks.lab6.csv', encoding='UTF-8') as fileMarks:

studentGrades = []

countStudent = 0

averageGrades = []

statsStudentResult = [0] \* NUMBER\_QUESTIONS

leadTimes = []

for entry in fileMarks:

student = entry.split(',', 4)

resultTest = student[4].replace('\",\"', " ").replace('\"', "").replace(',-', ' ').split()

grade = resultTest[0].replace(',', '.')

resultTest.pop(0)

studentGrades.append(grade)

leadTimes.append(student[3].replace(' хв ', '.').replace(' сек', '').replace(' хв', ''))

# Розрахунок середньої оцінки с кроком 1 хвилина

averageGrades.append(float(grade) / (len(resultTest) - 1))

# Лічильник студентів

countStudent += 1

i = 0

while i < len(resultTest):

if resultTest[i] == "0,50":

statsStudentResult[i] = statsStudentResult[i] + 1

i = i + 1

print(f'Кількість студентів що проходили тестування: {countStudent}\n')

grade = 0

while grade <= 10:

print(f"Оцінку {grade} отримало = {check\_repeat\_grade(grade, studentGrades)} студ.")

grade = grade + 0.50

print(separation())

for student in range(1, countStudent + 1):

print(f'Середня оцінка студента {student} с кроком 1 хв. = {averageGrades[student - 1]:.2f}')

print(separation())

with open('result.txt', 'w', encoding='UTF-8') as resultFile:

resultFile.write("Статистика правильних відповідей:\n")

for question in range(NUMBER\_QUESTIONS):

result = statsStudentResult[question] / countStudent \* 100

resultFile.write(f"\tПитання {question + 1} = {result:.2f} %\n")

resultFile.write(separation())

resultFile.write("Статистика неправильних відповідей:\n")

for question in range(NUMBER\_QUESTIONS):

result = 100 - statsStudentResult[question] / countStudent \* 100

resultFile.write(f"\tПитання {question + 1} = {result:.2f} %\n")

fiveBestGrades = [8] \* 5

gradesRatioTimes = []

i = 0

while i < len(studentGrades):

gradesRatioTimes.append(float(studentGrades[i]) / float(leadTimes[i]))

i = i + 1

resultFile.write(separation())

resultFile.write("Найкращі 5 оцінок в співвідношенні оцінка/час:\n")

i = 0

while i < 5:

student = gradesRatioTimes.index(max(gradesRatioTimes))

resultFile.write(f"\tСтудент {student + 1} : Оцінка = {studentGrades[student]}: Час виконання = {leadTimes[student]}\n")

del gradesRatioTimes[student]

i = i + 1

**Результат виконання програми:**

**завдання 1:**





**завдання 2:**





**завдання 3:**





**завдання 4:**







**завдання 5:**





**завдання 6:**





**завдання 7:**







**Висновок:** Опрацьовані засоби роботи з файлами в мові Python, діями над ними.