Звіт з практичної роботи No13

Тема роботи: Робота з текстовими файлами у Python: створення, читання,

запис та видалення даних.

Виконав: Кравченко Сергій, ТЦР-12

Дата: 02.06.2025

Мета роботи:

Опанувати навички роботи з файлами у Python, навчитися зчитувати,

записувати та видаляти інформацію з текстових файлів, використовувати

менеджери контексту.

Хід роботи:

Постановка завдання:

Завдання 1. "Простий текстовий редактор"

● Створіть програму, яка запитує користувача про створення нового

текстового файлу.

● Дозвольте користувачу додати кілька рядків тексту до файлу.

● Після завершення введення тексту файл повинен автоматично

зберігатися.

● Виведіть вміст створеного файлу на екран для перевірки.

Код програми:

def simple\_text\_editor():

filename = input("Введіть назву файлу для створення (з розширенням .txt): ")

print("Введіть текст (для завершення введіть пустий рядок):")

lines = []

while True:

line = input()

if line == "":

break

lines.append(line + "\n")

with open(filename, 'w', encoding='utf-8') as file:

file.writelines(lines)

print("\nФайл успішно збережено. Вміст файлу:")

with open(filename, 'r', encoding='utf-8') as file:

print(file.read())

# Виклик функції

simple\_text\_editor()

Приклад роботи програми:

Введіть назву файлу для створення (з розширенням .txt): example.txt

Введіть текст (для завершення введіть пустий рядок):

Це перший рядок.

Це другий рядок.

Тут може бути будь-який текст.

Файл успішно збережено. Вміст файлу:

Це перший рядок.

Це другий рядок.

Тут може бути будь-який текст.

Завдання 2. "Аналіз вмісту файлу"

● Напишіть програму, що зчитує вміст текстового файлу, підраховує

кількість слів, символів та рядків у ньому.

● Виведіть цю інформацію у зручному форматі (таблиця або

структурований текст).

Код програми:

def analyze\_file():

filename = input("Введіть назву файлу для аналізу: ")

try:

with open(filename, 'r', encoding='utf-8') as file:

content = file.read()

lines = content.split('\n')

words = content.split()

characters = len(content)

print("\nРезультати аналізу файлу:")

print(f"Кількість рядків: {len(lines)}")

print(f"Кількість слів: {len(words)}")

print(f"Кількість символів: {characters}")

print("\nЗведена таблиця:")

print("+------------------+-----------+")

print("| Характеристика | Кількість |")

print("+------------------+-----------+")

print(f"| Рядки | {len(lines):<9} |")

print(f"| Слова | {len(words):<9} |")

print(f"| Символи | {characters:<9} |")

print("+------------------+-----------+")

except FileNotFoundError:

print("Файл не знайдено!")

# Виклик функції

analyze\_file()

Приклад роботи програми:

Введіть назву файлу для аналізу: example.txt

Результати аналізу файлу:

Кількість рядків: 3

Кількість слів: 10

Кількість символів: 58

Зведена таблиця:

+------------------+-----------+

| Характеристика | Кількість |

+------------------+-----------+

| Рядки | 3 |

| Слова | 10 |

| Символи | 58 |

+------------------+-----------+

Завдання 3. "Пошук і заміна"

● Створіть програму, яка дозволяє користувачу відкрити текстовий

файл, знайти задане слово або фразу та замінити його на інше слово

або фразу.

● Збережіть зміни у новому файлі, щоб зберегти оригінал.

Код програми:

def find\_and\_replace():

filename = input("Введіть назву файлу для редагування: ")

search\_text = input("Введіть текст для пошуку: ")

replace\_text = input("Введіть текст для заміни: ")

new\_filename = input("Введіть назву нового файлу (з розширенням .txt): ")

try:

with open(filename, 'r', encoding='utf-8') as file:

content = file.read()

modified\_content = content.replace(search\_text, replace\_text)

with open(new\_filename, 'w', encoding='utf-8') as new\_file:

new\_file.write(modified\_content)

print("\nЗміни успішно збережено у новому файлі.")

print(f"Оригінальний файл: {filename} (не змінено)")

print(f"Новий файл: {new\_filename}")

print("\nПорівняння вмісту:")

print("Оригінал (перші 3 рядки):")

print('\n'.join(content.split('\n')[:3]))

print("\nНова версія (перші 3 рядки):")

print('\n'.join(modified\_content.split('\n')[:3]))

except FileNotFoundError:

print("Файл не знайдено!")

# Виклик функції

find\_and\_replace()

Приклад роботи програми:

Введіть назву файлу для редагування: example.txt

Введіть текст для пошуку: рядок

Введіть текст для заміни: речення

Введіть назву нового файлу (з розширенням .txt): modified.txt

Зміни успішно збережено у новому файлі.

Оригінальний файл: example.txt (не змінено)

Новий файл: modified.txt

Порівняння вмісту:

Оригінал (перші 3 рядки):

Це перший рядок.

Це другий рядок.

Тут може бути будь-який текст.

Нова версія (перші 3 рядки):

Це перший речення.

Це другий речення.

Тут може бути будь-який текст.

Висновки

У ході виконання практичної роботи було опановано роботу з текстовими файлами в Python. Виконані завдання охопили:

- Створення, запис та зчитування файлів з використанням менеджера контексту (with open).

- Аналіз вмісту (підрахунок рядків, слів, символів) з поданням результатів у зручному форматі.

- Пошук і заміну тексту зі збереженням змін у новому файлі.

Програми коректно обробляють помилки (наприклад, відсутність файлу) та надають інтерактивний інтерфейс для користувача. Отримані навички є важливими для подальшої роботи з обробкою даних у Python.