Звіт з лабораторної роботи No8

Тема роботи: Ітератори в діях.

Виконав: Кравченко Сергій, ТЦР-12

Дата: 06.06.2025

Мета роботи:

Навчитися працювати з вбудованими ітерабельними об'єктами Python

(list, tuple, set, str) та зрозуміти принцип роботи ітераторів у

простих практичних прикладах.

Хід роботи:

Постановка завдання:

Завдання 1. «Щось про голосні, але не тільки»

Замовник просить «щось зробити» з рядком, але уточнює тільки, що

«треба знайти голосні букви... або приголосні, може, взагалі всі літери... чи

тільки деякі». Результат потрібно вивести, але невідомо як саме: можливо,

просто кількість, або список голосних, чи якось ще.

Код програми:

def process\_string(text, mode='vowels', output\_format='count'):

"""

Обробляє рядок згідно з нечіткими вимогами замовника.

Параметри:

text (str): Вхідний рядок

mode (str): Режим роботи ('vowels', 'consonants', 'all')

output\_format (str): Формат виводу ('count', 'list', 'set')

Повертає:

Результат у вказаному форматі

"""

if not isinstance(text, str):

return "Помилка: очікувався рядок"

vowels = 'аеєиіїоуюяaeiouy' # голосні українською та англійською

consonants = 'бвгґджзйклмнпрстфхцчшщbcdfghjklmnpqrstvwxz' # приголосні

result = []

for char in text.lower():

if mode == 'vowels' and char in vowels:

result.append(char)

elif mode == 'consonants' and char in consonants:

result.append(char)

elif mode == 'all' and char.isalpha():

result.append(char)

if output\_format == 'count':

return len(result)

elif output\_format == 'list':

return result

elif output\_format == 'set':

return list(set(result)) # повертаємо як список для кращої читабельності

else:

return "Невідомий формат виводу"

Завдання 2. «Там якісь числа»

Замовник має два списки з числами, але не каже, що з ними робити: може

об'єднати, або видалити повтори, або відсортувати. Можливо, навіть нічого

не потрібно сортувати. Відомо лише, що результатом має бути «якийсь

Код програми:

def process\_numbers(list1, list2, operation='combine'):

"""

Обробляє два списки чисел згідно з нечіткими вимогами замовника.

Параметри:

list1 (list): Перший список чисел

list2 (list): Другий список чисел

operation (str): Операція ('combine', 'unique', 'sort', 'none')

Повертає:

Результат обробки

"""

# Перевірка на пусті списки або несподівані типи

if not isinstance(list1, list) or not isinstance(list2, list):

return "Помилка: очікувалися два списки"

# Фільтрація нечислових елементів

try:

numbers1 = [x for x in list1 if isinstance(x, (int, float))]

numbers2 = [x for x in list2 if isinstance(x, (int, float))]

except:

return "Помилка при обробці списків"

if operation == 'combine':

result = numbers1 + numbers2

elif operation == 'unique':

result = list(set(numbers1 + numbers2))

elif operation == 'sort':

result = sorted(numbers1 + numbers2)

elif operation == 'none':

return "Замовник не хоче ніяких змін"

else:

return "Невідома операція"

return result if result else "Списки порожні або не містять чисел"

Завдання 3. «Щось із текстом і символами» (найскладніше)

Замовник сказав, що є якийсь текст. Він просить щось з ним зробити:

можливо, порахувати частоту символів, можливо, знайти унікальні

символи, або навіть вивести символи, що зустрічаються певну кількість

разів. Він не визначився, чи враховувати пробіли, цифри та знаки

пунктуації. Також невідомо, як саме він хоче отримати результат: у вигляді

списку, множини чи словника.

Код програми:

def analyze\_text(text, include\_spaces=False, include\_digits=False,

include\_punct=False, output\_format='dict', min\_count=1):

"""

Аналізує текст і повертає інформацію про символи.

Параметри:

text (str): Вхідний текст

include\_spaces (bool): Враховувати пробіли

include\_digits (bool): Враховувати цифри

include\_punct (bool): Враховувати пунктуацію

output\_format (str): Формат виводу ('dict', 'list', 'set')

min\_count (int): Мінімальна кількість входжень для фільтрації

Повертає:

Результат аналізу у вказаному форматі

"""

if not isinstance(text, str):

return "Помилка: очікувався рядок"

freq = {}

for char in text:

# Враховуємо символи згідно з параметрами

if (char == ' ' and not include\_spaces) or \

(char.isdigit() and not include\_digits) or \

(char in '.,!?;:-()\'"' and not include\_punct):

continue

freq[char] = freq.get(char, 0) + 1

# Фільтруємо за мінімальною кількістю

filtered = {k: v for k, v in freq.items() if v >= min\_count}

if output\_format == 'dict':

return filtered

elif output\_format == 'list':

return list(filtered.items())

elif output\_format == 'set':

return set(filtered.keys())

else:

return "Невідомий формат виводу"

Висновки

У ході виконання лабораторної роботи було розглянуто роботу з ітераторами та ітерабельними об'єктами в Python. Було реалізовано три функції, які демонструють різні способи обробки даних згідно з нечітко поставленими умовами:

1. Обробка рядків — функція process\_string дозволяє аналізувати голосні, приголосні або всі літери, надаючи результат у вигляді кількості, списку або множини.

2. Робота зі списками чисел — функція process\_numbers об'єднує, фільтрує унікальні значення або сортує списки, забезпечуючи стабільність навіть при порожніх вхідних даних.

3. Аналіз тексту — функція analyze\_text підраховує частоти символів, враховуючи додаткові умови (пробіли, цифри, пунктуацію), і повертає результат у словнику, списку або множині.