Звіт з практичної роботи No12

Тема роботи: Оптимізація пошуку інформації в списках та словниках.

Виконав: Кравченко Сергій, ТЦР-12

Дата: 02.06.2025

Мета роботи:

Опанувати ефективні прийоми пошуку даних у структурах даних Python,

використовуючи словники, множини та списки, без використання класів і

функцій.

Хід роботи:

Постановка завдання:

Завдання 1. «Швидкий пошук»

● Маємо список міст: ["Київ", "Одеса", "Львів",

"Харків", "Житомир"].

● Перевірте, чи присутнє місто "Одеса" та "Полтава" у списку,

використовуючи множину для прискорення пошуку.

● Виведіть повідомлення про наявність або відсутність кожного міста.

Код програми:

cities = ["Київ", "Одеса", "Львів", "Харків", "Житомир"]

cities\_set = set(cities)

city1 = "Одеса"

city2 = "Полтава"

if city1 in cities\_set:

print(f"Місто {city1} присутнє у списку")

else:

print(f"Місто {city1} відсутнє у списку")

if city2 in cities\_set:

print(f"Місто {city2} присутнє у списку")

else:

print(f"Місто {city2} відсутнє у списку")

Приклад роботи програми:

cities = ["Київ", "Одеса", "Львів", "Харків", "Житомир"]

Місто Одеса присутнє у списку

Місто Полтава відсутнє у списку

Завдання 2. «Пошук за словником»

● Створіть словник студентів та їх оцінок: {"Іван": 80,

"Марія": 95, "Олег": 78, "Анна": 85}.

● Реалізуйте пошук оцінки студента за введеним користувачем ім'ям

(через input()).

Обробіть виняток, якщо введеного студента немає в словнику.

Код програми:

students = {"Іван": 80, "Марія": 95, "Олег": 78, "Анна": 85}

try:

name = input("Введіть ім'я студента: ")

grade = students[name]

print(f"Оцінка студента {name}: {grade}")

except KeyError:

print(f"Студента з ім'ям {name} не знайдено у словнику")

Приклад роботи програми:

students = {"Іван": 80, "Марія": 95, "Олег": 78, "Анна": 85}

Введіть ім'я студента: Марія

Оцінка студента Марія: 95

Введіть ім'я студента: Петро

Студента з ім'ям Петро не знайдено у словнику

Завдання 3. «Оптимізація повторів»

● Маємо великий список чисел (наприклад, від 1 до 1000 з

випадковими повторами).

● Використовуючи словник, підрахуйте кількість повторів кожного

числа.

● Виведіть число, яке повторюється найбільшу кількість разів, та

кількість його повторень.

Код програми:

import random

numbers = [random.randint(1, 1000) for \_ in range(10000)]

count\_dict = {}

for num in numbers:

if num in count\_dict:

count\_dict[num] += 1

else:

count\_dict[num] = 1

max\_num = max(count\_dict, key=count\_dict.get)

max\_count = count\_dict[max\_num]

print(f"Число {max\_num} повторюється найчастіше - {max\_count} разів")

Приклад роботи програми:

numbers = [10, 20, 30, 20, 10, 40, 10, 50, 20, 10] # Приклад згенерованого списку

Число 10 повторюється найчастіше - 4 рази

Висновки

Використання правильних структур даних значно підвищує ефективність обробки інформації, особливо при роботі з великими наборами даних. Множини та словники є потужними інструментами для оптимізації пошуку та аналізу даних у Python.