

Zadanie: Równoległe pobieranie i przetwarzanie danych pogodowych z użyciem klas danych i wielowątkowości

Cel dydaktyczny

Celem zadania jest integracja kilku kluczowych umiejętności z zakresu programowania w języku Python, w tym:

- projektowania klas danych z wykorzystaniem dekoratora `@dataclass`,
- definiowania metod klasowych (`@classmethod`),
- wykonywania operacji wejścia/wyjścia (zapis i odczyt danych do/z pliku JSON),
- przetwarzania równoległego z użyciem `concurrent.futures.ThreadPoolExecutor`.

Zadanie ma również na celu rozwijanie umiejętności organizowania kodu w sposób czytelny i rozszerzalny, a także symulację prostego systemu pozyskiwania danych.

Opis ogólny

Twoim zadaniem jest stworzenie systemu do pobierania, przechowywania i przetwarzania danych pogodowych dla zestawu miast. System ma działać równoległe, dzięki czemu dane będą pobierane szybciej. Dane mają być symulowane, ale realistyczne — np. losowa temperatura, wilgotność oraz aktualna data i czas.

Wymagania funkcjonalne

1. Model danych pogodowych

- Zdefiniuj klasę reprezentującą dane pogodowe, zawierającą:
 - nazwę miasta,
 - temperaturę (liczba zmiennoprzecinkowa),
 - wilgotność (liczba całkowita),
 - znacznik czasu (czas pobrania danych).
- Użyj dekoratora `@dataclass` do zdefiniowania tej klasy.
- Dodaj metodę umożliwiającą przekształcenie obiektu do formatu słownika (`dict`) w celu zapisu do pliku.
- Dodaj metodę klasową (`@classmethod`), która tworzy obiekt klasy na podstawie słownika (np. wczytanego z pliku JSON).

2. Symulacja pobierania danych pogodowych

- Zaimplementuj funkcję, która symuluje pobieranie danych pogodowych dla danego miasta.
- Funkcja powinna w losowy sposób generować temperaturę i wilgotność oraz ustawić aktualny czas jako znacznik pobrania.

- Symuluj opóźnienie (np. losowe między 0.5 a 2 sekund), aby zasymulować rzeczywiste warunki sieciowe.

3. Równoległe pobieranie danych

- Wykorzystaj `concurrent.futures.ThreadPoolExecutor`, aby równoległe pobierać dane pogodowe dla wielu miast.
- Przykładowa liczba miast: co najmniej 5 (np. Kraków, Warszawa, Gdańsk, Wrocław, Poznań).
- Zwróć listę obiektów reprezentujących dane pogodowe.

4. Zapis danych do pliku

- Zaimplementuj funkcję, która zapisze dane pogodowe do pliku w formacie JSON.
- Plik powinien zawierać listę słowników odpowiadających kolejnym obiektom pogodowym.

5. Odczyt danych z pliku

- Zaimplementuj funkcję, która wczyta dane z pliku JSON i przekształci je w listę obiektów klasy reprezentującej dane pogodowe (z użyciem metody klasowej `from_dict`).

6. Prezentacja wyników

- Po wczytaniu danych z pliku, wyświetl je w konsoli w czytelnej formie

Wersja rozszerzona (dla chętnych)

- Dodaj filtr wyświetlający tylko miasta z temperaturą powyżej określonej wartości.
- Dodaj możliwość eksportu danych do pliku CSV.
- Zamiast symulacji – zintegrować się z publicznym API pogodowym (np. OpenWeatherMap), używając biblioteki `requests`.