LU02.A10 - Immutable Dataclass



Implementieren Sie eine Funktion, die eine neue Instanz einer Dataclass erstellt, anstatt die Originalinstanz zu verändern, und nutzen Sie dabei die Prinzipien der funktionalen Programmierung.

Aufgabenstellung

Im Rahmen der funktionalen Programmierung ist es von entscheidender Bedeutung, Datenstrukturen unveränderlich (immutable) zu gestalten. Indem wir immutable Dataclasses verwenden, vermeiden wir unbeabsichtigte Seiteneffekte und sorgen dafür, dass unsere Datenstrukturen nur durch explizite, kontrollierte Operationen verändert werden. In dieser Aufgabe werden Sie lernen, wie man eine Dataclass verwendet, um eine unveränderliche (immutable) Datenstruktur zu erstellen, und wie man diese im Kontext der funktionalen Programmierung manipuliert.

- 1. Definieren Sie eine @dataclass(frozen=True) namens Car mit den Attributen brand (str), mileage (int) und service dates (List[str]).
- 2. Implementieren Sie eine Funktion add_mileage(car: Car, distance: int) → Car, die eine neue Instanz der Dataclass Car erstellt, mit einer aktualisierten mileage und einer neuen Service-Datum, wenn die Laufleistung über 10.000 km steigt.
 - 1. Die neue Instanz soll das aktualisierte Datum in der Liste service_dates speichern. Das Datum kann dabei ein fester Wert wie 2024-08-28 sein (für Testzwecke).
- 3. Implementieren Sie eine zweite Funktion check_service(car: Car) → bool, die prüft, ob die Laufleistung über 10.000 km liegt und entsprechend True oder False zurückgibt.
- 4. Nutzen Sie die Funktionen, um mehrere Änderungen an einer Instanz von Car durchzuführen, und drucken Sie die Liste der Service-Daten am Ende aus.

Schritt für Schritt

- 1. Definieren Sie die immutable Dataclass Car in einer separaten Datei car.py.
- 2. Implementieren Sie die Funktion add_mileage, die eine neue Instanz zurückgibt, wenn die Laufleistung aktualisiert wird, und fügt gegebenenfalls ein Service-Datum hinzu.
- 3. Implementieren Sie die Funktion check service, die die Laufleistung überprüft.
- 4. Führen Sie mehrere Funktionsaufrufe durch, um die Änderungen zu testen und die Liste der Service-Daten zu überprüfen.



From:

https://wiki.bzz.ch/ - BZZ - Modulwiki

Permanent link:

https://wiki.bzz.ch/modul/m323/learningunits/lu02/aufgaben/dataclass1

Last update: 2024/08/28 11:39



https://wiki.bzz.ch/ Printed on 2025/11/29 11:39