LU02.L10 - Immutable Dataclass

car.py

```
"""
This module contains the Car dataclass.
"""

from dataclasses import dataclass, field
from typing import List


@dataclass(frozen=True)
class Car:
    """
    A simple Car class that contains the brand, mileage, and service
dates.
    """

    brand: str
    mileage: int
    service_dates: List[str] = field(default_factory=list)
```

main.py

```
"""
This script demonstrates the usage of the Car class and the
functions add_mileage and check_service.
"""

from car import Car


def add_mileage(car, distance):
    """
    Returns a new Car instance with updated mileage and possibly an
updated service date.
    """
    new_mileage = car.mileage + distance
    new_service_dates = car.service_dates[:]

    if new_mileage > 10000 and (car.mileage <= 10000):
        new_service_dates.append('2024-08-28')

    return Car(brand=car.brand, mileage=new_mileage,
service_dates=new_service_dates)
```

```python
def check_service(car):
    """
    Checks if the car's mileage exceeds 10,000 km.
    """
    return car.mileage > 10000


if __name__ == '__main__':
    my_car = Car(brand='Toyota', mileage=9500)
    print(f'Vor der Fahrt: {my_car}')

    my_car = add_mileage(my_car, 600)  # Sollte den Service hinzufügen
    print(f'Nach der ersten Fahrt: {my_car}')

    if check_service(my_car):
        print('Service erforderlich!')

    my_car = add_mileage(my_car, 100)  # Keine Änderung bei Service-Daten
    print(f'Nach der zweiten Fahrt: {my_car}')
    print(f'Service-Daten: {my_car.service_dates}')
```

© Kevin Maurizi

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

Permanent link:
https://wiki.bzz.ch/modul/m323/learningunits/lu02/loesungen/dataclass1

Last update: **2024/08/28 11:40**