LU04c - Destruktor

Destruktoren werden aufgerufen, unmittelbar bevor ein Objekt gelöscht (zerstört) wird.

Beschreibung

Ein Destruktor in Python ist eine spezielle Methode, die aufgerufen wird, wenn ein Objekt einer Klasse zerstört wird, das heisst, wenn seine Lebensdauer endet und es aus dem Speicher entfernt wird. Der Destruktor wird in Python mit der Methode `del()` definiert.

Python hat eine eingebaute Speicherverwaltung und Garbage Collection, weshalb der Einsatz von Destruktoren in vielen Fällen nicht notwendig ist.

Aufruf

Ein Objekt „lebt“ solange mindestens eine Referenzvariable einen Verweis auf das Objekt enthält. Python zählt im Hintergrund, wie viele Referenzen auf ein Objekt existieren. Existiert auf ein Objekt keine Referenz mehr, wird der Dekonstruktor automatisch aufgerufen.

Zweck

Hauptsächlich wird der Destruktor verwendet, um Ressourcen freizugeben, die das Objekt während seiner Lebensdauer verwendet hat (z.B. das Schliessen von Dateien oder Netzwerkverbindungen).

Beispiel

```
class MyClass:
    def __init__(self, name):
        self.name = name
        print(f'Objekt {self.name} wurde erstellt.')

    def __del__(self):
        print(f'Objekt {self.name} wird zerstört.')

# Erstellen eines Objekts
obj = MyClass('TestObjekt')

# Löschen des Objekts
```

```
del obj

# Am Ende des Programms wird der Destruktor automatisch aufgerufen,
# wenn das Objekt nicht mehr referenziert wird.
```

Erklärung des Beispiels

__init__(self, name):

Dies ist der Konstruktor, der beim Erstellen eines Objekts aufgerufen wird. Hier wird das Objekt mit einem Namen initialisiert.

__del__(self):

Dies ist der Destruktor, der aufgerufen wird, wenn das Objekt zerstört wird. In diesem Beispiel gibt der Destruktor einfach eine Meldung aus, wenn das Objekt zerstört wird.

M320-LU04

Marcel Suter

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

Permanent link:
https://wiki.bzz.ch/modul/m320_2024/learningunits/lu04/destruktor

Last update: **2024/08/14 06:38**