LU10c - Vererbung in UML und Python

```
class Transport:
    def __init__(self):
        self._power = 0.0
        self.distance = 0.0
    def accelerate(self):
        pass
    def slowdown(self):
        pass


class Aeroplane(Transport):
    def __init__(self):
        # call constructor of the parent class
        super().__init__()
        self._climb_performance = 0.0
    def take_off(self):
        pass
    def landing(self):
        pass
```

Abb: Vererbung in UML-Darstellung

Sichtbarkeit der Attribute

In früheren Learning Units haben Sie die Sichtbarkeit **+** `public` und **-** `private` kennen gelernt. Bei der Vererbung kommt nun noch die Sichtbarkeit `protected` hinzu. Attribute in der Oberklasse, welche auch in der Unterklasse direkt zugeifbar sein müssen, werden als **protected** (geschützt) deklariert. In UML wird dies mit dem Zeichen **#** signalisiert.

Sichtbarkeit	Symbol	Wer darf zugreifen?
public	+	Jede andere Klasse/Funktion
protected	#	Nur innerhalb der Klasse und deren Kinder
private	-	Nur innerhalb der Klasse, keine Kinder

Im Gegensatz zu anderen Programmiersprachen erzwingt Python keine Sichtbarkeit von Attributen. Die Angaben `protected` und `private` sind mehr als Hinweise für andere Programmierer zu verstehen.

M320-LU10

René Probst, bearbeitet durch Marcel Suter

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

Permanent link:
https://wiki.bzz.ch/de/modul/m320_2024/learningunits/lu10/vererbung?rev=1788091539

Last update: **2026/08/30 14:05**