

Kapitel 17.1: Vererbung – Einführung

Beim letzten Mal haben wir gesehen, dass sich die 7 Combo-Klassen sehr ähnlich sind und nur die Berechnung des zufälligen x-Werts und das Erzeugen der Blöcke anders ist. Es wäre doch schön, wenn man das, was gleich ist, auch nur einmal tippen müsste!

Entwickler hassen Copy&Paste! Kopierter Code heißt doppelte Arbeit bei der Fehlersuche, Fehlerbehebung und Weiterentwicklung. Aus diesem Grund ist die sogenannte Vererbung ein wichtiger Bestandteil objekt-orientierter Programmiersprachen. Guck dir mal diese Beispiele an:

```
main.py
from auto import Auto

car = Auto()
car.drucken()

auto.py
class Auto:
    def __init__(self):
        self.name = "Auto"

    def drucken(self):
        print("Ich bin ein " + self.name)
```

```
from mercedes import Mercedes

car = Mercedes()
car.drucken()

auto.py
class Auto:
    def __init__(self):
        self.name = "Auto"

    def drucken(self):
        print("Ich bin ein " + self.name)

mercedes.py
from auto import Auto

class Mercedes(Auto):
    def __init__(self):
        self.name = "Mercedes"
```

```
main.py
from mercedes import Mercedes

car = Mercedes(2019)
car.drucken()

auto.py
class Auto:
    def __init__(self, baujahr):
        self.name = "Auto"
        self.baujahr = baujahr

    def drucken(self):
        print("Ich bin ein " + self.name + " von " + str(self.baujahr))

mercedes.py
from auto import Auto

class Mercedes(Auto):
    def __init__(self, baujahr):
        super().__init__(baujahr)
        self.name = "Mercedes"

    def drucken(self):
        super().drucken()
        print("Aber ich bin auch total cool!")
```

```
main.py
from mercedes import Mercedes
from audi import Audi
from auto import Auto

car1 = Mercedes(2019)
car1.drucken()
car1.hupen()

car2 = Audi(2020)
car2.drucken()
car2.hupen()

car3 = Auto(2018)
car3.drucken()
car3.hupen()

auto.py
class Auto:
    def __init__(self, baujahr):
        self.name = "Auto"
        self.baujahr = baujahr

    def drucken(self):
        print("Ich bin ein " + self.name + " von " + str(self.baujahr))
```

```
def hupen(self):  
    pass  
  
mercedes.py  
from auto import Auto  
  
class Mercedes(Auto):  
    def __init__(self, baujahr):  
        super().__init__(baujahr)  
        self.name = "Mercedes"  
  
    def drucken(self):  
        super().drucken()  
        print("Aber ich bin auch total cool!")  
  
audi.py  
from auto import Auto  
  
class Audi(Auto):  
    def __init__(self, baujahr):  
        super().__init__(baujahr)  
        self.name = "Audi"  
  
    def hupen(self):  
        print("Tuuuuut tuuuut")
```