

Vererbung

Ist eine **Klasse** eine Erweiterung einer bereits bestehenden Klasse, so spricht man davon, dass diese Klasse von der anderen Klasse erbt. Die erste nennt man Subklasse (auch abgeleitete Klasse, Unterklasse, untergeordnete Klasse oder Kindklasse), die zweite Superklasse (auch Basisklasse, Oberklasse, übergeordnete Klasse oder Elternklasse).

Diese Vererbung schließt alles von der Superklasse ein, von der man erbt : **Datenfelder**, **Konstruktoren** und **Methoden**

Die Vererbung wird in Java durch das Schlüsselwort *extends* in der Signatur der Klasse definiert, z.B.:

```
public class Dalmatiner extends Hund
```

Man spricht hier auch von einer "Ist ein"-Beziehung. Im Beispiel kann man sagen: "Der Dalmatiner ist ein Hund".

Dabei kann es sich bei der Klasse, von der man erbt, sowohl um eine gewöhnliche als auch eine **abstrakte Klasse** handeln.

Wird bei der Definition einer Klasse das *extends* weggelassen, so erbt die Klasse automatisch von der Klasse *Object* und erbt damit beispielsweise die Methode *toString*. Dadurch existiert in Java eine Vererbungshierarchie, die mit Hilfe eines **Baumdiagramms** dargestellt werden kann mit der Klasse *Object* als Wurzel.

Im Gegensatz zu anderen objektorientierten Programmiersprachen kann eine Klasse nur von einer anderen Klasse direkt erben, eine sogenannte Mehrfachvererbung ist nicht möglich. Allerdings kann eine Klasse mehrere **Interfaces** implementieren.

Überschreibt man eine Methode oder einen Konstruktor, so kann man mit dem Schlüssel *super* weiterhin auf das Element der Superklasse zugreifen, z.B.:

```
public class Dalmatiner extends Hund
{
    private int punkte;
    public Dalmatiner (int punkte)
    {
        super(); // eigentlich nicht nötig, da Java dies automatisch tut!!!
        this.punkte = punkte;
    }

    public void fresse(int futter)
    {
        super.fresse(futter); // ruft die Methode fresse der Superklasse
auf
        this.punkte += punkte; // etwas sinnlos, nur ein Beispiel...
    }
}
```

Bei einem Konstruktor kommt noch folgende Besonderheit dazu: Wird in der ersten Zeile nicht explizit

ein Konstruktor aufgerufen, so ruft Java als erste Anweisung im Konstruktor die Zeile *super()*; hinzu, ruft also den parameterlosen Konstruktor der Superklasse auf.

From:
<https://www.info-bw.de/> -

Permanent link:
<https://www.info-bw.de/faecher:informatik:oberstufe:glossar:vererbung:start>

Last update: **05.04.2022 13:04**

