# Vererbung

Ausgangssituation:

Für ein Unternehmen sollen mehrere Typen von Mitarbeitern erfasst werden.

Sie sollen dies implementieren.

Wir erstellen ein Klassendiagramm:

**…**

name

vorname

email

tel

gebdatum

gebort

…

**Praktikant**

name

vorname

email

tel

gebdatum

gebort

pauschale

**Selbständiger**

name

vorname

email

tel

gebdatum

gebort

stundensatz

**Angestellter**

name

vorname

email

tel

gebdatum

gebort

monatsgehalt

…

Uns fällt auf:

Die meisten Attribute haben haben alle Typen von Mitarbeitern gemeinsam.

Würden wir sie so direkt implementieren,

hätten wir eine Codeduplizierung.

Auch wäre die Realisierung schwer wartbar:

Soll ein gemeinsames Attribut zu allen Mitarbeitern dazu kommen,

z.B. eine Faxnummer,

müsste es aufwändig u. fehleranfällig in allen Klassen einzeln ergänzt werden.

**…**

name

vorname

email

tel

gebdatum

gebort

…

**Praktikant**

name

vorname

email

tel

gebdatum

gebort

pauschale

**Selbständiger**

name

vorname

email

tel

gebdatum

gebort

stundensatz

**Angestellter**

name

vorname

email

tel

gebdatum

gebort

monatsgehalt

…

Wie können wir die Codeduplizierung verhindern u. das System übersichtlicher gestalten?

Wir wollen die gemeinsamen Attribute nur einmal schreiben müssen u. an eine Stelle isolieren,

ähnlich wie ausgelagerte Methoden, die mehrfach benutzt werden,

um Änderungen, die wir dort vornehmen,

automatisch an alle entsprechenden Stellen weiterzugeben.

Wir verschieben die gemeinsamen Attribute in eine neue Klasse

u. positionieren sie darüber:

**Mitarbeiter**

name

vorname

email

tel

gebdatum

gebort

**…**

name

vorname

email

tel

gebdatum

gebort

…

**Praktikant**

name

vorname

email

tel

gebdatum

gebort

pauschale

**Selbständiger**

name

vorname

email

tel

gebdatum

gebort

stundensatz

**Angestellter**

name

vorname

email

tel

gebdatum

gebort

monatsgehalt

­

…

**Mitarbeiter**

name

vorname

email

tel

gebdatum

gebort

**…**

…

**Praktikant**

pauschale

**Selbständiger**

stundensatz

**Angestellter**

monatsgehalt

­ …

Um die Zugehörigkeit zu kennzeichnen u. auszudrücken,

dass alle Klassen in der unteren Reihe die Attribute der Klasse darüber bestitzen sollen,

kennzeichne ich dies mit einem Pfeil.

Dadurch ist das Ziel erreicht:

1. Gemeinsame Attribute brauchen nur einmal angegeben werden
2. Ein Attribut, das für alle Typen von Mitarbeitern dazukommen soll,  
   muss nur an einer Stelle ergänzt werden,  
   nämlich in der Klasse in der oberen Reihe.

Das nennt man "Vererbung"!

"Vokabeln" dazu:

Die obere Klasse, die die gemeinsamen Attribute enthält, nennt man,   
im Gegensatz zu den Klassen, die die Attribute übernehmen auch:

* Superklasse/Subklasse
* Eltern/Kindklasse
* Ober/Unterklasse
* generalisierte/spezialisierte Klasse
* vererbende/erbende Klasse
* Basisklasse/abgeleitete Klasse

Für den Begriff "vererben" gibt es Synonyme:

* ableiten
* erweitern
* generalisieren bzw. spezialisieren

Formulierungen z.B.:

* Die Subklasse erbt von der Superklasse.
* Die Kindklasse ist abgeleitet aus der Elternklasse.
* Die erbende Klasse spezialisiert die vererbende Klasse.
* Die Unterklasse erweitert die Oberklasse.

Vererbung heißt hier,

dass alle Eigenschaften der Oberklasse auch in den Unterklassen verfügbar sind.

Vererbung bezieht sich sowohl auf Attribute wie auch auf Methoden.

Vererbt die Superklasse etwas an eine oder mehrere Subklassen,

dann bleiben die vererbten Eigenschaften nach wie vor in der Superklasse verfügbar.

Den Begriff "vererben" kann man sowohl im monetären Sinn sehen wie auch im genetischen.

Je nach Kontext passt der eine oder andere besser.

Hier in der letzten Aussage passt der genetische Besser:

Vererben die Eltern ihre Gene an die Kinder,

haben die Eltern immer noch ihre Gene.

Übertragen:

Vererbt eine Superklasse an eine Subklasse,

ist die Superklasse immer noch im Besitz des Vererbten.

Vererbung in mehreren Generationen:

Vererbung kann in mehreren Generationen erfolgen:

Eine Klasse A kann an eine Klasse B vererben u. diese wieder an Klasse C.

Dadurch werden die Begriffe relativ:

Sie sind gegenüber Ihren Eltern in der Kindrolle.

Bekommen Sie eigene Kinder, nehmen Sie diesen gegenüber die Elternrolle ein.

Somit entsteht eine hierarchische Vererbungsstruktur,

wie wir eine ähnliche schon von verschachtelten Methodenaufrufen  
kennengelernt haben.

Mehrfachvererbung:

Gegensatz dazu: "Mehrfachvererbung"

Diese würde bedeuten,

dass sie gleichzeitig von mehreren Klassen erben.

Dies gibt es in Java (aus gutem Grunde) nicht.

­

Vorteile der Vererbung:

1. (wie eben schon gesagt): Codeduplizierung vermeiden
2. " Eigenschaften brauchen nur an einer Stelle eingefügt werden,  
   werden aber an mehreren wirksam
3. Durch Vererbung werden objektorientierte Softwaresysteme
   1. leicht änderbar
   2. leicht erweiterbar
   3. leicht wartbar

Dadurch steigt die Wiederverwendbarkeit

Das ist für Softwareentwicklung "im Großen" wichtig!

In Java wird die Vererbungsbeziehung mit dem Schlüsselwort "extends" ausgedückt.

Diese hat abstrakt folgende Syntax:

class <Subklasse> **extends** <Superklasse>;

Die Subklasse erweitert die Superklasse.

Damit hat die Subklasse alles was auch die Superklasse hat,

erweitert dies aber mit eigenen Attributen u. Methoden.

Somit ist die Subklasse größer als die Superklasse.

Der Name "Sub" u. "Super" bezieht sich also nicht auf die Größe,

sondern auf die Stellung zueinander:

Die Superklasse steht über der Subklasse.