# Überschreiben von Methoden bei Vererbung

Wenn in einer Superklasse eine Methode definiert ist,

wird diese in der Subklasse geerbt.

Wenn ich dort eine Methode mit gleichem Namen u. identischer Parameterliste implementiere,

überschreibt die Methode der Subklasse dort die gleichnamige geerbte Methode der Superklasse.

Beispiel:

Habe ich in der Superklasse "Rechteck" eine Methode "schreibe()" definiert,

ist diese auch in der Subklasse "Quader" sichtbar.

Wenn ich in der Klasse "Quader" eine Methode gleichen Namens

und mit identischer Parameterliste implementiere,

überschreibt die Methode der Subklasse in der Subklasse die geerbte.

In der Superklasse bleibt die bisherige Methode unberührt,

da jede Instanz einer Klasse einen vollständigen Satz von

sowohl Attributen als auch Methoden besitzt.

Beim Aufruf unterscheidet der Compiler anhand der Klassenzugehörigkeit der Instanz,

mit dem die Methode aufgerufen wird,

welche der Methoden verwendet wird.

Das ist komfortabel!

Die Methode der Superklasse ist nicht mehr direkt in der Methode der Subklasse sichtbar,

weil die eigene Methode gleichen Namens die Sichtbarkeit der geerbten Methode verdeckt.

Eine ähnliche Situation hatten wir bei der Verkettung von Konstruktoren:

Dort hatten wir mehrere Konstruktoren in einer Klasse,

die sich in den Parameterlisten unterschieden.

Wir wollten in einem speziellen Konstruktor auf den allgemeinsten verzeigern.

Dies war nicht direkt möglich,

da der spezielle Konstruktor,

in dem wir uns befanden,

den Blick auf die anderen gleichen Namens verdeckte.

Dort haben wir das Problem gelöst mit "this",

wodurch wir auf die äußere Schale der Klasse kamen,

in der wir uns befanden.

Hier ist die Situation ähnlich aber doch leicht anders:

Auch hier verdeckt der Methodenname,

in dem wir uns befinden,

die Sichtbarkeit der Methode in der Superklasse.

Hier können wir das Problem lösen,

indem wir uns mit "super." auf die Superklasse beziehen

u. dort die Methode aufrufen.

"super.<Methodenname>",

hier: "super.schreibe()".

# Überschreiben vs. Überladen

In beiden Fällen haben wir mehrere Methoden gleichen Namens.

1. Beim Überladen unterscheiden sie sich in der Parameterliste.  
   Sie sind parallel verfügbar.  
   Je nach Parameterliste wird die passende herausgesucht.
2. Beim Überschreiben sind die Parameterlisten identisch.  
   Die überschriebene Methode ist nicht mehr (direkt) verfügbar.

Je nach Klassenzugehörigkeit des Objektes,  
mit dem die Methode aufgerufen wird,  
wird die Methode der einen oder der anderen Klasse ausgeführt.