# Polymorphismus

Im Beispiel des Dreieckes möchte ich die Größe der Seiten verändern können.

Eine Methode "skaliere\_alle\_Seiten()" bekommt ein int-Parameter übergeben

u. soll alle drei Seiten mit diesem Faktor multiplizieren.

Eine weitere Methode soll implementiert werden,

damit alle Seiten mit einem einzelnen Faktor multipliziert werden können;

Nennen wir sie "skaliere\_Seiten\_einzeln()".

Sie bekommt drei int-Parameter übergeben.

Seite 1 wird mit Faktor 1, Seite 2 mit Faktor 2 u.s.w. multipliziert.

Das funktioniert einwandfrei.

Es besteht jedoch die Fehlerquelle,

dass wir versehentlich die falsch von beiden erwischen

u. 3 anstelle von einem Parameter übergeben oder umgekehrt.

Wir müssen "von Hand" die Zuordnung von Methodennamen

u. benötigter Anzahl Parameter vornehmen.

Frage: Könnte uns der Compiler das nicht abnehmen?

Wenn wir eine Methode aufrufen,

sucht er bei den Methodendeklarationen nach dem Namen der gesuchten Methode.

Wäre es nicht denkbar,

dass wir mehrere Methoden gleich nennen,

die sich nur in der Parameterliste unterscheiden

und er dann neben dem Namen auch nach der passenden Parameterliste sucht?

Allgemein:

Darf es verschiedene Methoden geben,

die gleich heißen, sofern sie sich in der Parameterliste unterscheiden?

Aussage des Polymorphismus (Vielgestaltigkeit):

Es darf mehrere Methoden parallel geben,

die gleich heißen,

sofern sie sich in der Parameterliste unterscheiden.

Der Compiler sucht zur Laufzeit die richtige heraus,

so dass nicht nur der Name,

sondern auch -bei mehreren gleichen Namen-

die Parameterliste übereinstimmt!

Existieren mehrere Methoden mit gleichem Namen  
aber unterschiedlicher Parameterliste,

sagt man auch:

"Die Methode ist überladen".

Was sind gültige Unterscheidungsmerkmale für Parameterlisten

bei überladenen Methoden?

1. Anzahl der Parameter
2. Typ der Parameter
3. Reihenfolge von Parametern

Der Name des Parameters bei sonst gleichen Punkten 1.-3.

ist kein gültiges Unterscheidungskriterium.

Einfaches Beispiel aus dem statischen Kontext:

public class Poly {

public static void main (String[] args) {

**m()**;

**m(17);**

**m(true);**

**m(3,false);**

**m(true,7);**

}

public static void **m()**{

System.out.println ("m ohne Parameter");

}

public static void **m(int i)**{

System.out.println ("m mit 1 \* int-Parameter");

}

public static void **m(boolean b)**{

System.out.println ("m mit 1 \* bool-Parameter");

}

public static void **m(int i, boolean b)**{

System.out.println ("m mit erst int, dann bool");

}

public static void **m(boolean b, int i)**{

System.out.println ("m mit erst bool, dann int");

}

}