# Rückgabewerte von Methoden

Wir haben Methoden mit "Black Boxes" verglichen.

Bisher haben wir nur die Eingabemöglichkeit besprochen,

wie wir der Methode Informationen beim Aufruf übergeben

in Form von Parametern.

Es fehlt die direkte Ausgabemöglichkeit in dem Sinn,

dass die Methode an die aufrufende Stelle etwas zurück gibt,

z.B. den Wert einer Berechnung.

Dies leistet der Rückgabewert einer Methode.

Wie schon bei der Suche nach einer formalen Eingabemöglichkeit bei Methoden,

suchen wir nun, wo Informationen zurückgegeben werden können.

public static void <Methodenname> (<Parameterliste>) {

<Block>

}

Ausgabemöglichkeit für Methoden

Bisher haben wir - ohne wissen, was es bedeutet - "void" vor den Methodennamen geschrieben.

"void" bedeutet "leer" u. sagst hier,

dass die Methode keinen Rückgabewert hat.

Wollen wir eine Methode mit Rückgabewert eines bestimmten Types definieren,

ersetzen wir "void" durch die Angabe des Typs,

der von der Methode zurückgegeben werden soll.

Beispiel:

Eine Methode "max()" soll 2 int-Parameter "a" u. "b" entgegen nehmen

u. einen int-Wert zurückgeben, das Maximum der beiden übergebenen Zahlen.

public static **int** max (int a, int b) { … }

Methode gibt Wert vom Typ "int" an die aufrufende Instanz zurück

Eine Methode kann (auf auf dem Weg) nur maximal einen Wert eines Types zurück geben.

Nun wollen wir die Methode vollständig implementieren.

Dazu fehlt uns noch, wie wir in der Methode anzeigen, welcher Wert zurückgegeben werden soll.

Dazu dient das Schlüsselwort "return".

Der nach dem Wort "return" befindliche Ausdruck

wird an die aufrufende Instanz zurück gegeben

u. die Methode danach verlassen.

Hier die Implementierung der Methode "max()":

public static **int** max (int a, int b) {

if (a>=b)

**return** a;

else **return** b;

}

Wertefluss bei Aufruf mit

System.out.println ( max (3, 5) );

public static **int** max (int a, int b) {

if (a>=b)

**return** a;

else **return** b;

}­

Was passiert beim Aufruf von "max(3,5)"?

Der Compiler sucht, wo die Implementierung der Methode "max()" steht.

Er sieht, dass sie 2 int-Parameter erwartet.

Er übergibt die 3 an a u. die 5 an b.

Dann wird Methode mit diesen Werten durchlaufen.

Kommt der Compiler bei einem return-Statement an,

wird das, was dahinter steht - hier die 3 oder die 5 -

über die Schnittstelle des Rückgabetyps "int"

an die aufrufende Instanz zurück gegeben

u. erscheint dort anstelle des Methodenaufrufes.

Der Rückgabewert wird von der Ausgabeanweisung "…println()" entgegen genommen

u. ausgegeben.

Die Methode funktioniert so,

ist aber stilistisch noch nicht optimal.

Stellen wir uns vor, wir haben eine große Methode mit Rückgabewert

mit sehr vielen return-Statements,

die unerwartet einen unsinnigen Wert zurück gibt.

Dann gestaltet sich die Fehlersuche schwierig,

weil unklar ist, von welchem return-Statement aus

der unsinnige Wert zurückgegeben wurde.

Einfacher wäre es, wenn wir nur genau ein return-Statement hätten,

denn dann wüssten wir, dass die fehlerhafte Ausgabe vor dort erfolgt sein muss.

Dies nennt man auch "single out": Nur ein einziger definierter Ausgang.

Das wäre stilistisch schöner.

Wie können wir das Beispiel in dem Sinn umgestalten?

Wir benutzen eine Hilfsvariable, z.B. "return\_max".

Statt dass wir wie bisher an den beiden Stellen direkt die Methode verlassen,

weisen wir der Hilfsvariablen den Wert zu

u. verlassen die Methode danach mit nur einem return-Statement

unter Rückgabe der Hilfvariablen.

Das kostet uns Platz in Form der Hilfsvariablen,

wir gewinnen aber dadurch an Eleganz u. Wartungsfreundlichkeit.

Wieder haben wir hier den "trade-off",

dass wir uns etwas an einer Stelle erkaufen müssen auf Kosten einer anderen Ressource.

public static int max (int a, int b) {

**int return\_max;**

if (a>=b)

**return\_max =** a;

else **return\_max =** b;

return **return\_max**;

}­