

Typumwandlungen mit Java

Implizierte Typumwandlung primitiver Typen

Implizite Typumwandlung geschieht automatisch, falls der Zieldatentyp einen **größeren** Bereich umfasst als der Quelldatentyp.

Beispiel:

```
double d;  
int i = 3;  
d = i; //d erhält den Wert 3.0
```

Explizite Typumwandlung primitiver Typen (Cast-Operator)

Soll eine Variable eines Datentyps einer anderen Variablen **kleineren** Typs zugewiesen werden, gehen ggf. Informationen verloren. Deshalb bedarf es eines speziellen Operators, dem Cast-Operator (**typ**), der unmittelbar vor der umzuwandelnden Variablen steht.

Beispiel:

```
double d = 7.3;  
int i = (int) d; // i erhält den Wert 7  
char z = 'A';  
int a = (int) z; // a erhält den Wert 65 (ASCII-Wert)
```

Umwandlungen eines primitiven Typs in ein Objekt vom Typ String

Mit Hilfe des Operators „+“ lassen sich Zeichenketten verbinden. Verkettet man eine Zeichenkette und den Wert einer Variablen primitiven Typs, so behandelt Java den Variablenwert wie einen vom Typ Zeichenkette.

Beispiel:

```
int i = 2;  
String z = i + ". Platz"; // z hat den Wert "2. Platz"  
double d = 3.14;  
String p = "" + d // p hat den Wert "3.14"
```

Die Klasse String bietet zusätzlich die Klassenmethode `valueOf()`, um den Wert eines primitiven Typs in eine Zeichenkette umzuwandeln.

Beispiel:

```
int i = 7;  
String z1 = String.valueOf(i); // z1 hat den Wert "7"  
double d = 3.14;  
String z2 = String.valueOf(d); // z2 hat den Wert "3.14"
```

Umwandlungen eines Objekts vom Typ String in einen primitiven Typ

Der Wert eines Strings kann ggf. als Zahl, Zeichen oder Wahrheitswert interpretiert werden. Dazu muss man auf die Wrapperklassen `Double`, `Integer`, `Boolean`, `Character`, ... und die zugehörigen Methoden `parseInt(String)`, `parseDouble(String)`, `parseBoolean(String)`, `parseChar(String)`, ... zurückgreifen:

Beispiel:

```
String z1 = "7";  
int i = Integer.parseInt(z1); // i hat den Wert 7  
String z2 = "7.1";  
double d = Double.parseDouble(z2); // d hat den Wert 7.1  
long l = Long.parseLong(z2); // Laufzeitfehler!
```