

Einstieg in die Informatik mit Java

vordefinierte Datentypen

Gerd Bohlender

Institut für Angewandte und Numerische Mathematik

- 1 Ganzzahlige Typen
- 2 Boolescher Typ
- 3 Gleitkommatypen
- 4 Referenztypen
- 5 `void`-Typ
- 6 Implizite und explizite Typumwandlungen

- 1 Ganzzahlige Typen
- 2 Boolescher Typ
- 3 Gleitkommatypen
- 4 Referenztypen
- 5 void-Typ
- 6 Implizite und explizite Typumwandlungen

Ganzzahlige Typen

Die Darstellung und Genauigkeit der vordefinierten Datentypen ist in Java genau festgelegt.

Typ	Speicherbedarf	Wertebereich
byte	8 bit	von -128 bis 127
short	16 bit	von -32768 bis 32767
int	32 bit	von -2.147.483.648 bis 2.147.483.647
long	64 bit	von -9.223.372.036.854.775.808 bis 9.223.372.036.854.775.807
char	16 bit	von 0 bis 65.535 (von '\u0000' bis '\uffff')

Achtung

char zählt zu den ganzzahligen Typen! Dies ist der einzige vorzeichenlose numerische Datentyp.

- 1 Ganzzahlige Typen
- 2 Boolescher Typ**
- 3 Gleitkommatypen
- 4 Referenztypen
- 5 void-Typ
- 6 Implizite und explizite Typumwandlungen

Boolsche Variablen, d. h. Variablen des Typs `boolean`, können nur die beiden Wahrheitswerte `true` und `false` annehmen.

- 1 Ganzzahlige Typen
- 2 Boolescher Typ
- 3 Gleitkommatypen**
- 4 Referenztypen
- 5 void-Typ
- 6 Implizite und explizite Typumwandlungen

Floating point numbers heißen in ihrer deutschen Übersetzung *Gleitkommazahlen* und sind nach dem IEEE-Standard 754 implementiert.

Typ	Speicherbedarf	Wertebereich (ca.)
float	32 bit	von 3.4028234663852886E+038 bis 1.4012984643248171E-045
double	64 bit	von 1.7976931348623157E+308 bis 4.9406564584124654E-324

Achtung

Obwohl dieser Datentyp in der deutschen Übersetzung das Wort Komma enthält, so werden die Nachkommastellen von Gleitkommazahlen in Java trotzdem mit einem Punkt getrennt!

Tritt „Überlauf“ ein, so wird mit $\pm \text{Infinity}$ weitergerechnet!

Bei einem mathematisch sinnlosen Ergebnis (z. Bsp. $0/0$, $\infty - \infty$, $0 * \infty$) wird NaN („Not a Number“) als Ergebnis eingesetzt. Damit kann das Programm formal fortgesetzt werden.

- 1 Ganzzahlige Typen
- 2 Boolescher Typ
- 3 Gleitkommatypen
- 4 Referenztypen**
- 5 void-Typ
- 6 Implizite und explizite Typumwandlungen

Referenztypen sind das Gegenteil von den bisher genannten *Grundtypen/elementaren Typen* und werden in späteren Kapiteln ausführlich behandelt.

Beispiel

Klassen, Felder, Strings.

- 1 Ganzzahlige Typen
- 2 Boolescher Typ
- 3 Gleitkommatypen
- 4 Referenztypen
- 5 void-Typ**
- 6 Implizite und explizite Typumwandlungen

Dieser Typ ist ein Pseudotyp und tritt bei Methoden auf.
Besitzt eine Methode keinen Rückgabewert, so wird `void`, d. h. „nichts“ als Ergebnis zurückgegeben.

Beispiel

```
public static void main (String [] args) {}
```

- 1 Ganzzahlige Typen
- 2 Boolescher Typ
- 3 Gleitkommatypen
- 4 Referenztypen
- 5 void-Typ
- 6 Implizite und explizite Typumwandlungen

(a) implizite Typumwandlungen

Implizite Typenumwandlungen treten dann auf, wenn ein kleinerer Zahlenbereich in einen größeren Zahlenbereich abgebildet wird. Solche Umwandlungen werden automatisch ausgeführt.

Beispiel

von `byte` nach `short`, von `short` nach `int`, von `int` nach `long`.

(b) explizite Typumwandlungen

Soll ein größerer in einen kleineren Zahlenbereich umgewandelt werden, so muss dies dem Compiler explizit mitgeteilt werden. Die Mitteilung erfolgt durch eine vorangestellte Klammerung des gewünschten Datentyps.

Beispiel

Um den ganzzahligen Anteil einer Gleitkommazahl zu bestimmen, muss diese explizit in einen ganzzahligen Datentyp umgewandelt werden:

```
double a = 5.35;  
int b    = (int) a;
```

Achtung

Ohne obige explizite Typumwandlung wird bei der Kompilierung ein Fehler ausgegeben!