

Java:
Casten (1)
implizites / explizites Casten

"Casten"

= Umwandlung eines Datentyps in einen anderen

auch: Typumwandlung, Typanpassung,
Konvertierung

1) Implizites Casten

Daten des kleineren Datentyps werden durch den Compiler automatisch dem größeren angepasst

→ nur dann, wenn kein Datenverlust!

(Frage: Welcher Datentyp ist "mächtiger"?)

1) Implizites Casten

```
int einInteger = 20;  
long einLong = 1000;
```

```
long neuerLong = einInteger;  
                // funktioniert
```

```
int neuerInt = einLong;  
             // funktioniert NICHT (Datenverlust!)
```

```
double einDouble = 1.24;  
float einFloat = 4.28F;
```

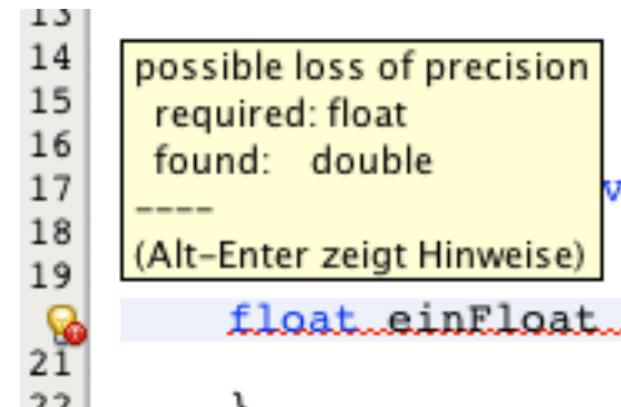
```
double neuDouble = einFloat;  
                // funktioniert
```

```
float neuFloat = einDouble;  
            // funktioniert NICHT (Datenverlust!)
```

1) Implizites Casten

```
double einDouble = 1.24;  
float einFloat = 4.28;  
    // funktioniert nicht  
    // Dezimalzahl wird als Double  
interpretiert; Umwandlung in Float zöge  
Genauigkeitsverlust nach sich!
```

```
float nochEinFloat = 4.28F;
```



Zur Wiederholung: Wertebereiche

Ganzzahltypen

short -2^{15} bis $2^{15}-1$ (-32768 bis 32767)

int -2^{31} bis $2^{31}-1$ (-2.147.483.648 bis 2.147.483.647)

long -2^{63} bis $2^{63}-1$ (-9 Trillionen bis +9 Trillionen) // L anhängen!

Gleitpunkttypen

float $-3.40... * 10^{38}$ bis $3.4.... * 10^{38}$ // F anhängen!

double $-1.79... * 10^{308}$ bis $1.79.... * 10^{308}$ // höhere Genauigkeit!

1) Implizites Casten

Automatische Umwandlung von

short → int, long, float, double
int → long, float, double
long → float, double
float → double

```
int einInteger = 20;
```

```
long neuerLong = einInteger;  
// funktioniert
```

```
float neuerFloat = einInteger;  
// funktioniert
```

```
short neuerShort = einInteger;  
// funktioniert NICHT (Datenverlust!)
```

1) Implizites Casten: Übung

Übung Casting-1: 6 Fehler im Listing

```
int alter = 83;
float schuhgroesse = 39.5F;
float kontostand = 1925.30F;
float zinssatz = 3.5F;
long anzahlGehirnzellenBeiGeburt = 5000000000L;
int verlustGehirnzellenProJahr = 500000;
double atomGewichtSynapsen = 1.34823e-300;

int anzahlGehirnzellenHeute1 = anzahlGehirnzellenBeiGeburt - verlustGehirnzellenProJahr * alter;
long anzahlGehirnzellenHeute2 = anzahlGehirnzellenBeiGeburt - verlustGehirnzellenProJahr * alter;
double anzahlGehirnzellenHeute3 = anzahlGehirnzellenBeiGeburt - verlustGehirnzellenProJahr * alter;

double zusammenhangHirnSchuhe1 = schuhgroesse / anzahlGehirnzellenBeiGeburt;
float zusammenhangHirnSchuhe2 = schuhgroesse / anzahlGehirnzellenBeiGeburt;
long zusammenhangHirnSchuhe1 = schuhgroesse / anzahlGehirnzellenBeiGeburt;

double kontostandIn10Jahren1 = 10 * zinssatz*kontostand;
int kontostandIn10Jahren2 = 10 * zinssatz*kontostand;
long kontostandIn10Jahren3 = 10 * zinssatz*kontostand;

long gesamtGewicht1 = anzahlGehirnzellenBeiGeburt*atomGewichtSynapsen;
double gesamtGewicht2 = anzahlGehirnzellenBeiGeburt*atomGewichtSynapsen;
float gesamtGewicht3 = anzahlGehirnzellenBeiGeburt*atomGewichtSynapsen;
```

2) Explizites Casten

Gewünschter neuer Datentyp wird angegeben

→ Datenverlust wird in Kauf genommen!

```
int einInteger = 20;
```

```
long neuerLong    = einInteger;  
                // funktioniert
```

```
float neuerFloat  = einInteger;  
                // funktioniert
```

```
short neuerShort = (short) einInteger;  
                // funktioniert (aber mit Datenverlust!)
```

2) Explizites Casten

Weitere Beispiele:

```
int einInteger = 20;  
long einLong = 1000;
```

```
long neuerLong = einInteger;  
                // funktioniert  
int neuerInt   = (int) einLong;  
                // funktioniert (Datenverlust!)
```

```
double einDouble = 1.24;  
float einFloat = 4.28F;
```

```
double neuDouble = einFloat;  
                // funktioniert  
float neuFloat = (float) einDouble;  
                // funktioniert (Datenverlust!)
```

2) Explizites Casten

Achtung: Explizites Casten hat hohe Priorität!

Deshalb: Klammern verwenden

```
int schuhgroesseUSA = (int) 1.353 * 1.22;  
    // funktioniert NICHT (1.22 wird nicht gekastet!)
```

```
int schuhgroesseUSA = (int) (1.353 * 1.22);  
    // funktioniert
```