

## Münzwerte

```
public class Muenzen1
{
    public static void main(String[] args) {
        int zehner1 = 8; // Anzahl 10 ct Muenzen
        int zwanzger1 = 4; // Anzahl 20 ct Muenzen
        int fuchzger1 = 3; // Anzahl 50 ct Muenzen

        double gesamt = zehner1 * 0.10 +
            zwanzger1 * 0.20 + fuchzger1 * 0.50;

        System.out.print("Gesamtwert = ");
        System.out.println(gesamt);
    }
}
```

Einführung in die Informatik: Programmierung und Softwareentwicklung 31

## Variablen

Das Statement

```
int zehner1 = 8;
```

deklariert eine ganzzahlige **Variable** des Wertes 8.

Vorteil hier: **bessere Dokumentation**.

Semantisch äquivalent:

```
System.out.println(8 * 0.10 + 4 * 0.20 + 3 * 0.50);
```

aber schlechter lesbar.

Einführung in die Informatik: Programmierung und Softwareentwicklung 33

Alles was nach dem // kommt ist **Kommentar**.

Zwei verschiedene Arten von Zahlen:

- **Integers** (ganze Zahlen). In Java int. Werte: 8, 4, 3.
- **Doubles** (Fließkommazahlen mit doppelter Präzision). In Java **double**. Werte: 0.10, 0.20, 0.50, aber auch  $3E9$  ( $=3 \cdot 10^9$ ).

Man sollte Integers für ganzzahlige Größen verwenden, z.B.

Anzahlen, Pixelkoordinaten.

Vorteile: Weniger Speicherplatz, Genauigkeit, vermeidet, „Äpfel mit Birnen zu verwechseln“.

Einführung in die Informatik: Programmierung und Softwareentwicklung 32

## Wertzuweisung

Man kann einer Variablen neue Werte zuweisen:

```
zwanzger1 = 5;
int zw = zwanzger1;
zw = 6;
System.out.print(Wert von \"zwanzger1\");
System.out.println(zwanzger1);
System.out.print(Wert von \"zw\");
System.out.println(zw);
```

Was wird gedruckt?

Einführung in die Informatik: Programmierung und Softwareentwicklung 34

## Antwort

Wert von "zwanzger1": 5  
Wert von "zw": 6

Der Grund ist, dass Integer- und Double-Variablen keine Verweise sind (wie Objektvariablen) sondern den jeweiligen Wert **direkt** enthalten.

Mit anderen Worten: eine Integer-Variablen enthält einen Integer-Wert, eine Objekt-Variablen enthält eine Speicheradresse (unter der sich ein Objekt befindet).

Einführung in die Informatik: Programmierung und Softwareentwicklung 35

## Nochmal Wertzuweisung

Man kann auch schreiben:

```
a = a + 1;
```

Dadurch wird der Wert von `a` um eins erhöht.

Manche Programmierer verwenden dafür die Kurzform

```
a++;
```

Daher auch der Name C++ für „Nachfolger von C“.

Einführung in die Informatik: Programmierung und Softwareentwicklung 37

## Initialisierung

Man muss Variablen nicht initialisieren:

```
int a;  
int b = 4;  
a = b + 2;
```

Sie müssen aber vor der ersten Verwendung einen Wert bekommen:

```
int a;  
int b = 4;  
System.out.println(a);
```

ist ein Programmierfehler (den Java schon beim Compilieren erkennt, C aber nicht.)

Einführung in die Informatik: Programmierung und Softwareentwicklung 36

## Alles ist endlich

Integer rangieren von  $-2.147.483.648$  bis  $2.147.483.647$ , d.h. insgesamt  $2^{32}$  verschiedene Werte.

```
int weltbevoelkerung = 2000000000;  
weltbevoelkerung = 2 * weltbevoelkerung;  
System.out.println(weltbevoelkerung);
```

Ergebnis:

```
-294967296
```

Einführung in die Informatik: Programmierung und Softwareentwicklung 38



## Rundungsfehler

```
double f = 4.35;
int n = (int)(100 * f);
System.out.println(n);
```

Druckt 434.

Grund: In Binärdarstellung ist 4,35 ein **echt periodischer** Bruch.

```
(int)Math.round(100 * f);
```

hat Wert 435.

Einführung in die Informatik: Programmierung und Softwareentwicklung 43

## Arithmetik

Plus + und Mal \* hatten wir schon.

Division wird als / notiert.

**Vorsicht:** Sind beide Operanden von / Integers, so wird **abgerundet**.

```
int s1 = 5;
int s2 = 6;
int s3 = 3;
```

```
double mittelwert = (s1 + s2 + s3) / 3;
```

mittelwert hat den Wert 4 (statt 4.666666...)

Beliebter Programmierfehler.

Richtig:

```
double mittelwert = (s1 + s2 + s3) / 3.0;
```

Einführung in die Informatik: Programmierung und Softwareentwicklung 45

## Konstanten

```
int flaschen = 3;
int dosen = 5;
int mengeFanta = flaschen * 0.5 + dosen * 0.33;
ist schlechter Stil, da 0.5 und 0.33 einfach so dastehen.
```

Besser:

```
final double FLASCHEN_INHALT = 0.5;
final double DOSEN_INHALT = 0.33;
```

```
double mengeFanta = flaschen * FLASCHEN_INHALT + dosen * DOSEN_INHALT
```

Werte, die mit `final` deklariert werden, können nur einmal initialisiert und danach nicht mehr verändert werden.

Vorteil gegenüber Variablen: Effizienz + Dokumentation.

Numerische Konstanten wie 0.5 **miten im Programm** sind **schlechter Stil**.

Vordefinierte Double-Konstanten: `Math.PI` und `Math.E`.

Einführung in die Informatik: Programmierung und Softwareentwicklung 44

## Was gibt's sonst noch

- Die „Punkt vor Strich“ Regel gilt.
- Mathematische Funktionen sind in der Klasse `Math` definiert.
- Automatische Typkonversionen erfolgen von innen nach außen.

Beispiel: Die „Lösungsformel“:

```
double a;
double b;
double c;
```

```
/* Wertzuweisung */
```

```
x1 = (-b + Math.sqrt(b * b - 4 * a * c)) / (2 * a);
x2 = (-b - Math.sqrt(b * b - 4 * a * c)) / (2 * a);
```

Ebenso: `a * a + b * b - 2 * a * b * Math.sin(phi)`;

Details in `JBUILDER` → `Help`.

Einführung in die Informatik: Programmierung und Softwareentwicklung 46

## Zeichenketten

Der Datentyp `String` besteht aus **Zeichenketten**, d.h. Folgen von Buchstaben und Sonderzeichen.

```
String name = "Matthias";  
String name = "Johanna";  
System.out.println(name);
```

Druckt: Johanna.

```
int n = name.length();
```

Die Variable `n` hat den Wert 7.

Einführung in die Informatik: Programmierung und Softwareentwicklung 47

## Teile einer Zeichenkette

**Antwort:** `sub1` den Wert Hello und `sub2` den Wert o, w. Der Ausdruck `s.substring(7, 12)` hat den Wert World.

Will man alle Zeichen von `anfang` bis zum Ende der Zeichenkette, dann kann man

```
s.substring(anfang, s.length())
```

schreiben. Das letzte Zeichen hat nämlich die Position

```
s.length() - 1.
```

Eine Kurzform dafür ist `s.substring(anfang)`.

Einführung in die Informatik: Programmierung und Softwareentwicklung 49

## Teile einer Zeichenkette

Der Ausdruck

```
s.substring(anfang, endePlusEins)
```

bezeichnet die Teilzeichenkette von `s` angefangen vom Zeichen an der Position `anfang` bis (ausschließlich) zum Zeichen an der Position `endePlusEins`.

Positionen beginnen immer bei Null.

```
String s = "Hello, World!";
```

```
String sub1 = s.substring(0,5);
```

```
String sub2 = s.substring(4,8);
```

Was sind die Werte von `sub1` und `sub2`?

Welcher `substring`-Ausdruck hat den Wert `World` ?

Einführung in die Informatik: Programmierung und Softwareentwicklung 48

## Fehlerbehandlung

Ruft man `s.substring` mit unpassenden Argumenten auf, so gibt es einen Fehler. Z.B.: `s.substring(4,30)` führt zu:

```
Exception in thread "main" java.lang.StringIndexOutOfBoundsException:  
String index out of range: 20  
at java.lang.String.substring(String.java:1473)  
at hamen.main(hamen.java:5)
```

Man sagt: der Ausdruck wirft eine Ausnahme (*throws an exception*).

Es ist möglich, so eine Ausnahme im Programm "aufzufangen" und benutzerdefinierte Befehle auszuführen, z.B. eine ordentliche Fehlermeldung.

Noch besser ist es, das Auftreten solcher Ausnahmen von vornherein zu vermeiden.

Einführung in die Informatik: Programmierung und Softwareentwicklung 50

## Verkettung

Zeichenketten kann man aneinanderhängen ("verketteten", "konkatenerieren" von lat. *catena* = Kette).

In Java verwendet man dafür das Pluszeichen:

Der Ausdruck "Matthias" + "Johanna" hat den Wert MatthiasJohanna.

Der Ausdruck

"Euro" + "s".substring(0,n)

hat den Wert Euro oder Euros ja nachdem, ob n gleich 0 oder 1 ist.

Alle anderen Werte von n sind nicht erlaubt.

Einführung in die Informatik: Programmierung und Softwareentwicklung 51

## Was "sind" Zeichenketten?

Nachteil: JRE muss die nicht mehr benötigten Zeichenketten automatisch "aufräumen" (sog. *garbage collection*), was Zeit kostet.

In C darf man (und muss man) Zeichenketten verändern und explizit Platz für sie reservieren etc.

Das ist die Hauptursache für Sicherheitslücken!

Z.B. *buffer overflow*.

Aktuelle Forschung: Speicherkonzept wie in Java aber gleichzeitig Effizienz von C.

Einführung in die Informatik: Programmierung und Softwareentwicklung 53

## Was "sind" Zeichenketten?

Eine Zeichenkette ist ein Objekt.

Es versteht u.a. die Methoden `length` und `substring`.

Die Methode `length` liefert die Länge zurück.

Die Methode `substring` ein **neues** String-Objekt, das den jeweiligen Teilstring enthält.

Man kann eine Zeichenkette nie verändern (im Gegensatz zu veränderlichen Objekten wie `Rectangles`).

Im Computer ist ein String eine Speicheradresse. Unter dieser Adresse befindet sich die Länge, z.B. *n*. In den *n* darauffolgenden Speicherstellen befinden sich dann die Zeichen.

In der Sprache Java gibt es keine Möglichkeit, diese Zeichen oder die Länge zu verändern, obwohl die Maschinsprache das im Prinzip zuließe.

Vorteil: Weniger Programmierfehler.

Einführung in die Informatik: Programmierung und Softwareentwicklung 52

## Verkettung mit Zahlwerten

```
double betrag = 34.99;
int nummerMahnung = 2;
String anweisung = nummerMahnung + ". Mahnung: " +
    "Bitte zahlen Sie " + betrag + " EUR.";
```

`System.out.println(anweisung);`

Druckt: 2. Mahnung: Bitte zahlen Sie 34.99 EUR.

Einführung in die Informatik: Programmierung und Softwareentwicklung 54

## Verkettung mit Zahlwerten

Ist ein Operand von + eine Zeichenkette, so wird der andere automatisch in eine Zeichenkette umgewandelt. Das ist **keine**

Typkonversion:

```
String betrag = 34.99 * 2;
```

löst aus:

```
incompatible types
```

```
found   : double
```

```
required: java.lang.String
```

```
String betrag = 34.99 * 2;
```

Einführung in die Informatik: Programmierung und Softwareentwicklung 55

## Abhilfe: Formatierte Ausgabe

```
import java.text.NumberFormat;
```

```
...
```

```
NumberFormat formatierer = NumberFormat.getInstance();
```

```
formatierer.setMaximumFractionDigits(2);
```

```
formatierer.setMinimumFractionDigits(2);
```

```
double betrag = 34.99;
```

```
int nummerMahnung = 2;
```

```
String anweisung = nummerMahnung + ". Mahnung: " +
```

```
"Bitte zahlen Sie " + formatierer.format(betrag/3) + " EUR.";
```

Ergebnis:

2. Mahnung: Bitte zahlen Sie 11,66 EUR.

Sogar Tausendertpunkte werden eingesetzt, z.B.: 1.192.279,33 EUR.

Einführung in die Informatik: Programmierung und Softwareentwicklung 57

## Verkettung mit Zahlwerten

Man kann schreiben ""+x um x in eine Zeichenkette umzuwandeln.

Aber Vorsicht:

```
String anweisung = nummerMahnung + ". Mahnung: " +
```

```
"Bitte zahlen Sie " + betrag/3 + " EUR.";
```

```
System.out.println(anweisung);
```

Ergebnis:

2. Mahnung: Bitte zahlen Sie 11.663333333333334 EUR.

Einführung in die Informatik: Programmierung und Softwareentwicklung 56

## Benutzernamen

Wir möchten aus dem ersten und letzten Buchstaben des Namens und einer laufenden Nummer einen Benutzernamen erzeugen:

```
String name = "Johanna";
```

```
int lfdNo = 1728;
```

Der Benutzername sollte Ja1728 sein.

Einführung in die Informatik: Programmierung und Softwareentwicklung 58

## Kein Problem

```
String benutzerName;  
benutzerName = name.substring(0,1) +  
name.substring(name.length() - 1) + lfdNo;
```

Einführung in die Informatik: Programmierung und Softwareentwicklung 59

## Parsing von Doubles

```
...geht analog mit Double.parseDouble, z.B.:  
double c = Double.parseDouble("2.97E9"); /* oder so */  
Aber Vorsicht: eine "deutsche" Zahl, wie 1.234,59 kann parseDouble  
nicht verarbeiten.
```

Wie das geht, siehe Java Reference Manual

## Parsing von Zeichenketten

Wie erhalten wir aus einem Benutzernamen die laufende Nummer?  
`benutzerName.substring(2)`

enthält zwar die Ziffern der lfd. Nr. ist aber immer noch ein String.

Die Lösung:

```
int nummer = Integer.parseInt(benutzerName.substring(2));  
parseInt ist eine statische Methode der Klasse Integer und dient  
dazu, eine Zeichenkette in einen integer umzuwandeln.
```

Statische Methoden werden nicht an ein Objekt geschickt, sondern  
können "einfach so" ausgeführt werden.

Details später.

Einführung in die Informatik: Programmierung und Softwareentwicklung 60