Schleifen und Arrays
Javakurs

Robert Buchholz
Maik Pflugradt

http://freitagsrunde.org/Javakurs
10. April 2007

Wollen wir das?

```
System.out.println(1);
System.out.println(2);
System.out.println(3);
System.out.println(4);
System.out.println(5);
```

Ausgabe:
1
2
3
4
5

Aufgabe:
Gib die Zahlen 1 bis 5 auf der Konsole aus.

Schönere Lösung als Schleife

```
int i = 1;
while (i <= 5) {
  System.out.println(i);
  i = i + 1;
}
```

Ausgabe:
1
2
3
4
5

- Keine Zeilen gespart
- Kein doppelter Code

Ein einfaches Beispiel

```
int n = 5;
int x = 0;
int xFak = 1;
while (x < n) {
    x = x + 1;
    xFak = xFak * x;
}
System.out.println(n + "! = " + xFak);
```

Aufgabe:
Berechne die Fakultät von n.
(n! = 1 * 2 * 3 * ... * n)

Ausgabe:
5! = 120

```
int n = 5;
int zahl = 1;
while (zahl < n) {
  if (zahl % 2 == 1) {
    // zahl ist ungerade
    System.out.println(zahl);
  }
  zahl = zahl + 1;
}
```

Initialisierung

Bedingung

Inkrement

Aufgabe:
Gib alle ungeraden
Zahlen kleiner als n aus.

Schleifen mit „for“

```
int n = 5;
for (int zahl = 1; zahl < n; zahl = zahl + 1) {
    if (zahl % 2 == 1) {
        // zahl ist ungerade
        System.out.println(zahl);
    }
}
```

Trennung von Zähler und Rumpf.

Aufgabe:
Vereinfache die Schleife von der vorherigen Folie.

for vs. while

```
for («Init»; «Condition»; «Update») {
    «Body»
}
```

Yeah!

...

Geht beides!

```
«Init»
while («Condition») {
    «Body»
    «Update»
}
```

```
double d = 0.0;
while (d != 1.1) {
  System.out.println(d);
  d = d + 0.1;
}
```

```
// geht auch mit for

for (double d = 0.0;  d != 1.1; d = d + 0.1) {
  System.out.println(d);
}
```

Aufgabe:
Zähle eine Zahl in Schritten von 0.1 bis 1 hoch.

```
double
while (
  Syste
  d = d
}
```

```
// geh

for (d
  Syst
}
```

Ausgabe:

```
0.0
0.1
0.2
0.3000000000
0.4
0.5
0.6
0.7
0.7999999999999999
0.8999999999999999
0.9999999999999999
1.0999999999999999
1.2
1.3
1.4000000000000001
```

Rundungsfehler bei reellen Zahlen.

```
l) {
```

Aufgabe:
Zähle eine

och.

```
double d = 0.0;
while (d < 1.1) {
  System.out.println(d);
  d = d + 0.1;
}
```

Aufgabe:
Zähle d in Schritten von 0.1 bis 1 hoch.

Schleifenbedingungen

```
int i = 11;

while (i != 9) {
    i = i + 1;
}
```

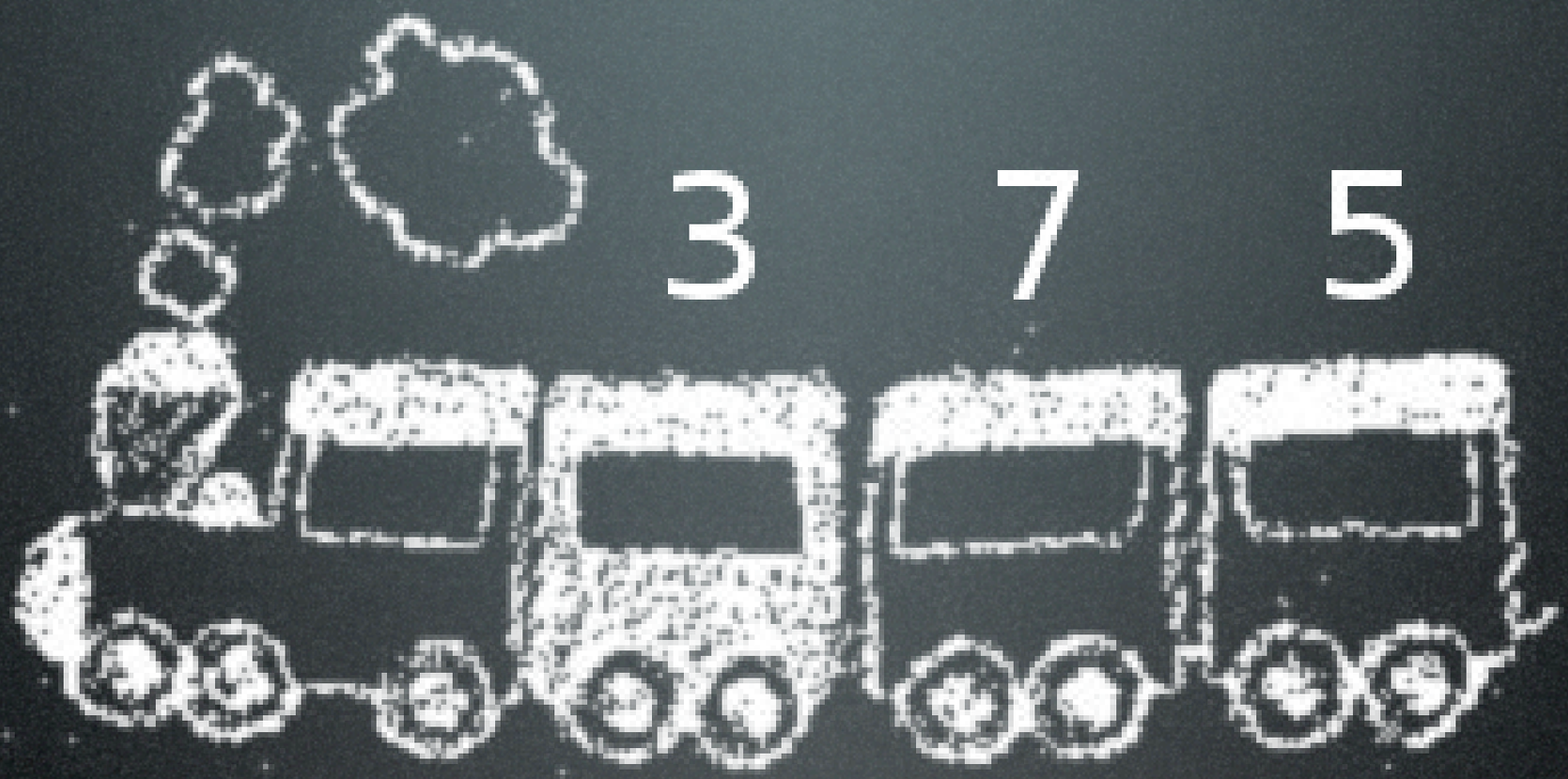
```
int i = 11;

while (i < 9) {
    i = i + 1;
}
```

Der Rumpf wird nie ausgeführt.

Aufgabe:
Zähle i in Einerschritten bis 9 hoch.

Nächstes Thema: Arrays



Aufgabe:
Denke an einen Zug.
Er hat Waggons, darin sind Fahrgäste.

Zug mit Variablen

```
// Wir modellieren einen Zug
int waggon1 = 3;
int waggon2 = 7;
int waggon3 = 5;
```

Das geht nur für genau drei Waggons. :-/

Frage:
Wie viele Leute sitzen im Zug?

```
int gaeste = waggon1 + waggon2 + waggon3;
System.out.println(gaeste);
```

Einzelne Variablen vs. Datenstruktur

Zug mit Array

```
// Wir modellieren einen Zug
// zweiter Versuch
zug[0] = 3;
zug[1] = 7;
zug[2] = 5;
```

Der Typ fehlt...

Frage:
Wie viele Leute sitzen im Zug?

```
int gaeste = 0;
for (int i = 0; i < zug.length; i++) {
  gaeste = gaeste + zug[i];
}
System.out.println(gaeste);
```

Zug, jetzt richtig

```
int [] zug = new int[3];
zug[0] = 3;
zug[1] = 7;
zug[2] = 5;
```

Die Länge in eckigen Klammern.

Alle Elemente haben denselben Typ.

Indizierung von 0 bis n-1.

Das Array verrät seine Länge

```
int [] zug = new int[3];
zug[0] = 3;
zug[1] = 7;
zug[2] = 5;
```

```
System.out.println("Länge: " + zug.length);
```

Ausgabe:
Länge: 3

Don‘t #1

```
int zug[];
zug[0] = 3; // Fehler!
```

Initialisierung vergessen!

```
$ javac ArrayFehler.java
ArrayFehler.java:6: variable zug
might not have been initialized
            zug[0] = 3;
            ^
1 error
```

```
int zug[] = new int[3];
zug[0] = 3;
```

Don‘t #2

Über die Grenze.

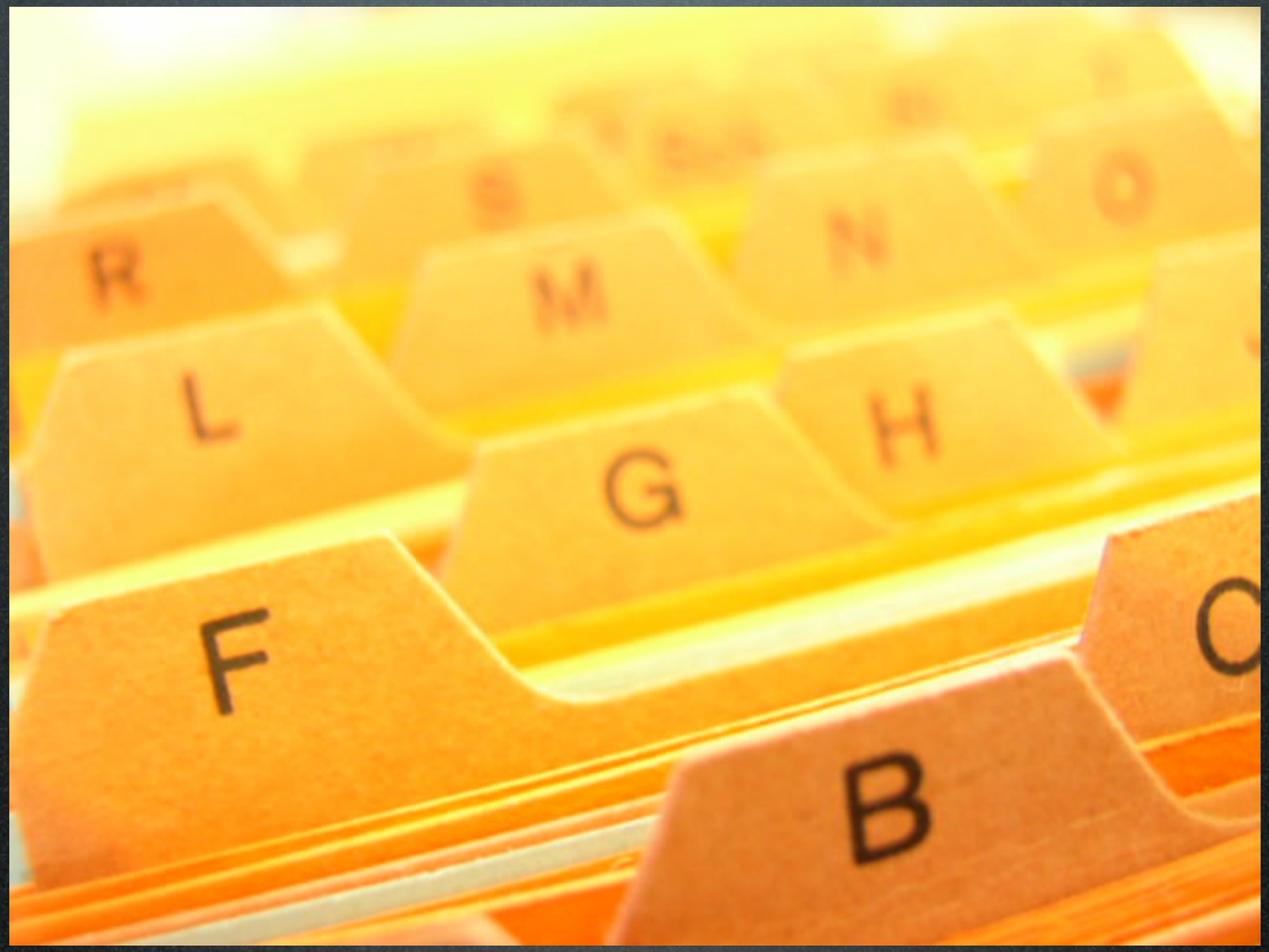
```
int zug[] = new int[3];
zug[3] = 10; // Fehler!
```

```
$ javac ArrayFehler.java
$

$ java ArrayFehler
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException: 3
        at ArrayFehler.main(ArrayFehler.java:6)
```

```
int zug[] = new int[4];
zug[3] = 10;
```

Telefonkartei



Aufgabe:
Organisiere deine Telefonnummern
in einem Array.

Telefonkartei als Array

```
int [] nummern = new int[4];
nummern[0] = 92211;
nummern[1] = 110;
nummern[2] = 2342;
nummern[3] = 31421386;
```

Aufgabe:
Finde heraus, wie oft eine bestimmte Nummer vorkommt.

```
int gesucht = 112;
int gefundene = 0;
for  (int i = 0; i < nummern.length; i++) {
  if  (nummern[i] == gesucht) {
    gefundene++;
  }
}
System.out.println("Die Nummer kommt "
                    + gefundene + " mal vor.");
```

Ausgabe:
Die Nummer kommt 0 mal vor.

Aufgabe:
Finde heraus, wie oft eine bestimmte Nummer vorkommt.

```
...
String haeufigkeit;
if (gefundene == 0) {
    haeufigkeit = "gar nicht";
} else {
    haeufigkeit = gefundene + " mal";
}
System.out.println("Die Nummer kommt "
                   + haeufigkeit + " vor.");
```

Ausgabe:
Die Nummer kommt gar nicht vor.

Aufgabe:
Finde heraus, wie oft eine bestimmte Nummer vorkommt.
Und gib es „in schön“ aus.

Was haben wir gelernt?

- Schleifen können Code mehrmals ausführen.
- Arrays enthalten Daten gleichen Typs.

Aufgabe:
Macht die Übungsaufgaben.
Und habt Spaß dabei :-)