Einführung in Java

Stephan Diehl

- altenglisches Wort für Kaffee
- Entwicklung: 1990-1995, SUN, Erfolg erst duch World Wide Web
- einfach (keine Zeiger, keine mehrfache Vererbung, keine Strukturen)
- objektorientiert
- robust (keine Zeiger, automatische Speicherverwaltung, strenges Typsystem)
- portierbar, da architekturunabhängig
- schnell, da vorkompiliert in Byte-Code
- sicher (Byte-Code-Verifizierer)
- Threads
- dynamische Bindung, dyn. Laden
- überschaubarer Sprachumfang
- umfangreiche Klassenbibliothek (API)

Datentypen und Variablen

Standard-, Feld- und Klassentypen

```
int zaehler;
int alter=27;
int sinPiHalbe = Math.sin(Math.PI/2);
```

mehrdimensionale Felder

```
int meinFeld[][][];


int x=3, int y=5;
int meinFeld[][][]
     = new int [3][4][Math.max(x,y)];
```

Index beginnt bei 0

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

Kontrollanweisungen

```
if (...) ...  else
for (...;...;...) ...
while (...) ...
do ... while (...);

switch (E)
 { case K_1: ... break;
   ...
   case K_k: ... break;
   default: A }

schleife:
 for (var i=0; i<100; i++)
  { for (var j=0; j<100; j++)
     { if (j>=i) break schleife;
       document.write("Die Zahl ",j,
                      " ist kleiner ",i);
     }
  }
```

Klassen

```
public class K extends O
               implements S_1, ..., S_k
 { // Variablen- und Methodendefinitionen
 }
```

Beispiel:

```
// Datei Adresse.java

public class Adresse
 { public String Strasse;
   public String Ort;

   public void print()
    { System.out.println(Strasse);
      System.out.println(Ort);
    }
 }
```

Vererbung

Datei Adresse.java

```
public class Adresse
 { public String Strasse;
   public String Ort;

   public void print()
    { System.out.println(Strasse);
      System.out.println(Ort); }
 }
```

Datei Privatadresse.java

```
public class Privatadresse extends Adresse
 { public String Vorname;
   public String Nachname;

   public void print()
    { System.out.print(Vorname+" ");
      System.out.println(Nachname);
      super.print(); }
 }
```

Objekte

Instantiierung

```
Privatadresse adr = new Privatadresse;
```

Zugriff auf Variablen

```
adr.Vorname = "Fred";
adr.Nachname = "Feuerstein";
adr.Strasse = "Im Steinbruch 4";
adr.Ort = "99999 Dinodorf";
```

Methodenaufruf

```
adr.print();
```

Konstruktoren

Datei Adresse.java

```
public class Adresse
 { public String Strasse;
   public String Ort;

   public Adresse(String s, String o)
    { Strasse=s;
      Ort=o; }
 }
```

Datei Privatadresse.java

```
public class Privatadresse extends Adresse
 { public String Vorname;
   public String Nachname;

   public Privatadresse(String v, String n;
                        String s, String o)
    { super(s,o);
      Vorname=v;
      Nachname=n; }
 }
```

Initialisierung per Konstruktor

```
Privatadresse adr
 = new Privatadresse("Fred",
                     "Feuerstein",
                     "Im Steinbruch 4",
                     "99999 Dinodorf");
```

Polymorphismus

Inklusionspolymorphismus

```
public void printAll(Adresse adr)
 { adr.print();
 }
```

Methodenüberladung

```
public void printAll(Adresse adr)
 { adr.print();
 }


public void printAll(String strasse,
                     String ort)
 { System.out.println(strasse);
   System.out.println(ort);
 }
```