

Java in 10 Folien

Besser:

Online-Buch „Go to Java 2“

Java Overview

- Java ist **plattformunabhängig**
- Programm besteht aus ≥ 1 Klassen:
Klasse `MeineKlasse`:
Source in Datei `MeineKlasse.java`
Übersetzen: `javac MeineKlasse.java`
 \Rightarrow `MeineKlasse.class` mit Bytecode
- Eine Klasse mit `main`-Methode
Programmstart: `java MeineKlasse`
- `CLASSPATH` setzen

Hello World

```
// Dieses Programm druckt "Hello World!"  
  
public class HelloWorld  
{  
    public static void main(String[] args)  
    {  
        System.out.println("Hello World!");  
    }  
}
```

Variablen, Statements, ...

```
public int meineMethode()  
{  
    int s=0; double d=0.5;  
    for (int k=1; k<=100; k++)  
    { s=s+k; }  
    if (s>500)  
    { s=s*d; }  
  
    while (s>=500)  
    { s=s-1; }  
    return s;  
}
```

Arrays = Sequenzen eines Typs

```
int[] intArray = new int[100];  
  
for (int k=0;k<intArray.length;k++)  
{  
    intArray[k]=k*k;  
}
```

Beispiel eines einfachen Programms

```
/* Listing1610.java */
import java.util.*;

public class Listing1610
{
    public static void main(String[] args)
    {
        final int SIZE = 20;
        int[] values = new int[SIZE];
        Random rand = new Random();
        //Erzeugen und Ausgeben des unsortierten Arrays
        for (int i = 0; i < SIZE; ++i) {
            values[i] = rand.nextInt(10 * SIZE);
        }
        for (int i = 0; i < SIZE; ++i) {
            System.out.println(values[i]);
        }
        //Sortieren des Arrays
        Arrays.sort(values);
        //Ausgeben der Daten
        System.out.println("---");
        for (int i = 0; i < SIZE; ++i) {
            System.out.println(values[i]);
        }
    }
}
```

Objekte = Klassen-Instanzen

```
class A
{ private int myprivar;
  public A(int x) {myprivar = x;}
  public H(int k) {i = i-k;}
}

class B
{ public void mymethod(int m)
  { A myinstance = new A(m);
    myinstance.H(10);
    ... }
}
```

Klassenhierarchien

```
class C extends B
{
    public void neuemethode(int m)
    { mymethod(m/2); }
}
```

```
class D extends C
{
    public void neuemethode(int m)
    { if (m>0) ... }
}
```


Interfaces = Klassen-“Vorlagen“

```
interface I  
{ public void m(int k); }
```

```
class E implements I  
{ public void m(int k) {...}  
}
```

```
Class F implements I {...}
```

...

```
E e1=new E(),e2=new E();F f=new F();  
e1.m(1); e2.m(2);
```

Organisation durch Packages

Zusammenfassung von Klassen (~Directory)

`mein.projekt.Parser`, `mein.projekt.Crawler`

⇒ Im Verzeichnisbaum:

```
mein/projekt/Parser.java
```

⇒ Im Quelltext:

```
package mein.projekt;
```

Externe Pakete verwenden:

- `import java.sql.*;`
- In `CLASSPATH` aufnehmen (z.B. `stemmer.jar`)