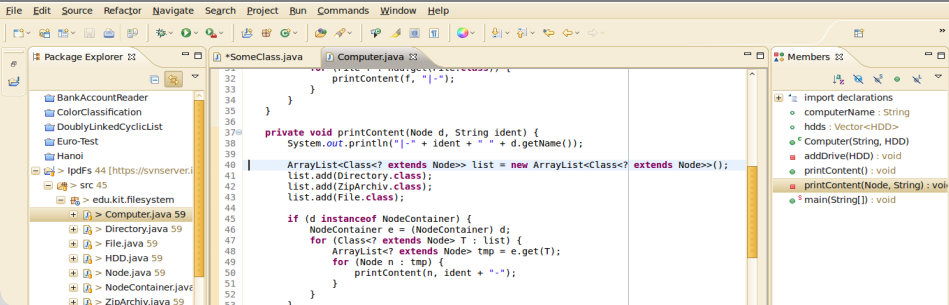


Programmieren-Tutorium Nr. 10 bei Martin Thoma

TODO: Subtitle setzen!

Martin Thoma | 28. Oktober 2012

FAKULTÄT FÜR INFORMATIK



```
File Edit Source Refactor Navigate Search Project Run Commands Window Help

Package Explorer
  BankAccountReader
  ColorClassification
  DoublyLinkedCyclicList
  Euro-Test
  Hanoi
  Ipdfs 44 [https://svnserver.i
  src 45
  edu.kit.filesystem
  Computer.java 59
  Directory.java 59
  File.java 59
  HDD.java 59
  Node.java 59
  NodeContainer.java
  ZipArchiv.java 59

Computer.java
32      printContent(f, "|-");
33    }
34  }
35
36
37  private void printContent(Node d, String ident) {
38    System.out.println(f, ident + " " + d.getName());
39  }
40
41  ArrayList<Class? extends Node> list = new ArrayList<Class? extends Node>();
42  list.add(Directory.class);
43  list.add(ZipArchiv.class);
44  list.add(File.class);
45
46  if (d instanceof NodeContainer) {
47    NodeContainer e = (NodeContainer) d;
48    for (Class? extends Node? T : list) {
49      ArrayList? extends Node? tmp = e.get(T);
50      for (Node n : tmp) {
51        printContent(n, ident + "-");
52      }
53    }
54  }
```

Members

- import declarations
- computerName : String
- hdds : Vector<HDD>
- Computer(String, HDD)
- addDrive(HDD) : void
- printContent() : void
- printContent(Node, String) : void
- main(String[]) : void

- 1 Einleitung
- 2 Wiederholung
- 3 Werbeblock
- 4 Praxis
- 5 Abspann

```
1 public class Bool {
2     public static void main(String[] v) {
3         boolean a = true;
4         boolean b = false;
5         boolean c = true;
6         int d = 0;
7         boolean e = false;
8
9         if (a && b && c) {
10             System.out.println("Alpha");
11         }
12
13         if (d) {
14             System.out.println("Beta");
15         }
16
17         if (a||b && b||e) {
18             System.out.println("Gamma");
19         }
20     }
21 }
```

- Was ist die Ausgabe?
- Gibt es einen Compiler-Fehler?
- Gibt es einen Laufzeit-Fehler?



- Treten bei „offensichtlichen“ Fehlern auf


- Eclipse beschwert sich: .
Häufige Fehler:



- [foo] cannot be resolved to a type : Klassenname nicht definiert?
Falsch geschrieben?
- [foo] cannot be resolved : Die Variable foo wurde nicht deklariert


- Es kann kein gültiger Java Byte-Code erstellt werden


- Treten bei „offensichtlichen“ Fehlern auf
- Eclipse beschwert sich:  .
Häufige Fehler:
 - `[foo]` cannot be resolved to a type : Klassenname nicht definiert?
Falsch geschrieben?
 - `[foo]` cannot be resolved : Die Variable `foo` wurde nicht deklariert
- Es kann kein gültiger Java Byte-Code erstellt werden

- Treten bei „offensichtlichen“ Fehlern auf
- Eclipse beschwert sich:  .
Häufige Fehler:
 - [foo] cannot be resolved to a type : Klassenname nicht definiert?
Falsch geschrieben?
 - [foo] cannot be resolved : Die Variable foo wurde nicht deklariert
- Es kann kein gültiger Java Byte-Code erstellt werden

- Treten bei „offensichtlichen“ Fehlern auf
- Eclipse beschwert sich: .
Häufige Fehler:
 - [foo] cannot be resolved to a type : Klassenname nicht definiert?
Falsch geschrieben?
 - [foo] cannot be resolved : Die Variable foo wurde nicht deklariert
- Es kann kein gültiger Java Byte-Code erstellt werden

- Treten bei „offensichtlichen“ Fehlern auf
- Eclipse beschwert sich:  .
Häufige Fehler:
 - [foo] cannot be resolved to a type : Klassenname nicht definiert?
Falsch geschrieben?
 - [foo] cannot be resolved : Die Variable foo wurde nicht deklariert
- Es kann kein gültiger Java Byte-Code erstellt werden

- Eclipse beschwert sich: 
- Eine Klasse von Compiler-Fehlern

- Eclipse beschwert sich: 
- Eine Klasse von Compiler-Fehlern

- Schwer(er) zu finden
- Gültiger Byte-Code kann erzeugt werden
- Muss nicht immer auftreten
- Eclipse beschwert sich nicht

- Schwer(er) zu finden
- Gültiger Byte-Code kann erzeugt werden
- Muss nicht immer auftreten
- Eclipse beschwert sich nicht

- Schwer(er) zu finden
- Gültiger Byte-Code kann erzeugt werden
- Muss nicht immer auftreten
- Eclipse beschwert sich nicht

- Schwer(er) zu finden
- Gültiger Byte-Code kann erzeugt werden
- Muss nicht immer auftreten
- Eclipse beschwert sich nicht

Bool.java

```
1 public class Bool {  
2     public static void main(String[] v) {  
3         int x = 42;  
4  
5         if (x = x + 42) {  
6             System.out.println("Das Leben");  
7         } else {  
8             System.out.println("Das Universum");  
9         }  
10    }  
11 }
```

- Was ist die Ausgabe?
- Gibt es einen Compiler-Fehler?
- Gibt es einen Laufzeit-Fehler?

Java vs. JavaScript

Java	JavaScript
Keine Scriptsprache	Definitiv Scriptsprache
statisch typisiert	dynamisch typisiert
Klassen	Prototypen
Blockbasiertes Scoping	Funktionsbasiertes Scoping
...	...

Java vs. JavaScript

comparisons.js

```
1 '' == '0'           // false
2 0 == ''             // true
3 0 == '0'            // true
4
5 false == 'false'    // false
6 false == '0'        // true
7
8 false == undefined // false
9 false == null       // false
10 null == undefined  // true
11
12 '\t\r\n' == 0      // true
```

```

9  c = 0;
10 a = b = d = -1;
11 e = [f = [20, 7],
12      [20, 8]
13 ];
14
15 i = Math.random;~
16
17 function n() {
18     if (c && (87 == a || 83 == a)) c = 0, d = 87 == a ? -1 : 1;
19     if (d && (65 == a || 68 == a)) d = 0, c = 65 == a ? -1 : 1;
20     p([j = e[0][0] + c, k = e[0][1] + d])
21     || !j || 39 == j || !k
22     || 10 == k ? b += " | GAME OVER"
23       : (e.unshift([j, k]), p(f)
24          ? (f = [1 + 38 * i() | 0, 1 + 9 * i() | 0], b++)
25            : e.pop());
26     for (m = h = ""; 11 > h; h++) {
27         for (g = 0; 40 > g; g++) l = g + "," + h, m += !g
28           || 39 == g || !h
29           || 10 == h ? "X"
30             : e[0] == 1 ? "0"
31               : p(1) ? "*" : f == 1 ? "%" : " ";
32         m += "\n"
33     }
34     x.innerHTML = m + b;
35     !b.sup && setTimeout(n, 99)
36 }();
37
38 onkeydown = function (o) {

```

String ist eine Java-Klasse, aber ...:

```
1 String myVar  = "Hallo";           // String mit Inhalt
2 String myVar2 = new String();       // leerer String
3 String myVar3 = "";                // auch ein leerer String
```

String ist kein primitiver Datentyp! Trotzdem kann man „rechnen“:

```
1 String myVar = "Hallo";  
2 String myVar2 = "Welt";  
3  
4 // Verknuepfen von Strings  
5 String result1 = myVar + " " + myVar2 ;  
6 String result2 = myVar2 + "+" + myVar2 ;
```

String mit Inhalt `Hallo "Welt"` :

```
1 String = "Hallo \"Welt\"";
```

String mit Inhalt `Hallo "Welt"` :

```
1 String = "Hallo \"Welt\"";
```

String mit Zeilenumbruch:

```
1 String = "Hallo \n Welt";
```

String mit Inhalt `Hallo "Welt"` :

```
1 String = "Hallo \"Welt\"";
```

String mit Zeilenumbruch:

```
1 String = "Hallo \n Welt";
```

String mit Inhalt `Hallo \ Welt` :

```
1 String = "Hallo \\ Welt";
```

Shark.java

```
1 public class Shark {
2     // attributes
3     private Laser laserAttachedToHead;
4
5     // methods
6     public void attack(Enemy e) {
7         this.laserAttachedToHead.fire(e);
8     }
9
10    private void eat() {
11        // rawr!
12    }
13 }
14
15 // invoke method
16 Shark erik = new Shark();
17 erik.attack(somalianPirate);
```

Quelle: kit.trvx.org

Wie kann man Erik auffordern zu essen?

Shark.java

```
1 public class Shark {  
2     private int size;  
3  
4     // constructor  
5     public Shark(int size) {  
6         this.size = size;  
7     }  
8 }  
9  
10 // initializes object  
11 Shark erik = new Shark(25);
```

Quelle: kit.trvx.org

- Was ist die Konstruktorsignatur?
- Was ist shadowing und warum benutzt man 'this'?

Shark.java

```
1 public class Shark {  
2     private int size;  
3  
4     // constructor  
5     public Shark(int size) {  
6         this.size = size;  
7     }  
8 }  
9  
10 // initializes object  
11 Shark erik = new Shark(25);
```

Quelle: kit.trvx.org

- Konstruktorsignatur → `public Shark(int size)`
- Shadowing und 'this' → Variable `size` verbirgt das Attribut `size`
⇒ das Attribut wird über `this.size` aufgerufen

Math.java

```
1 public final class Math {  
2     private Math() {}  
3  
4     public static final double E = 2.7182818284590452354;  
5  
6     public static final double PI = 3.14159265358979323846;  
7  
8     // [...]  
9 }
```

Quelle: docjar.com

Math.java

```
1 public final class Math {  
2     private Math() {}  
3  
4     public static final double E = 2.7182818284590452354;  
5  
6     public static final double PI = 3.14159265358979323846;  
7  
8     // [...]  
9 }
```

Quelle: docjar.com

```
1 int var = Jeep.ANZAHLRAEDER ; // Zugriff wie er sein sollte  
2  
3 Jeep myJeep = new Jeep ( ) ; // Objekterzeugung  
4  
5 int var2 = myJeep.ANZAHLRAEDER ; // Zugriff wie er nicht sein sollte  
6 // ... geth aber
```

- Unbedingt Teilnehmen:
 - Spielerische Einführung ins Lösen algorithmischer Probleme
 - Sehr lehrreich
 - Preise (Sachpreise, Fahrt zur Endrunde, Geld, Studienstiftung)
 - Kontakte
- „Insider-Hinweise“ auf martin-thoma.com
- Offizielles und Aufgaben unter bundeswettbewerb-informatik.de

- Unbedingt Teilnehmen:
 - Spielerische Einführung ins Lösen algorithmischer Probleme
 - Sehr lehrreich
 - Preise (Sachpreise, Fahrt zur Endrunde, Geld, Studienstiftung)
 - Kontakte
- „Insider-Hinweise“ auf martin-thoma.com
- Offizielles und Aufgaben unter bundeswettbewerb-informatik.de

- Unbedingt Teilnehmen:
 - Spielerische Einführung ins Lösen algorithmischer Probleme
 - Sehr lehrreich
 - Preise (Sachpreise, Fahrt zur Endrunde, Geld, Studienstiftung)
 - Kontakte
- „Insider-Hinweise“ auf martin-thoma.com
- Offizielles und Aufgaben unter bundeswettbewerb-informatik.de

- Unbedingt Teilnehmen:
 - Spielerische Einführung ins Lösen algorithmischer Probleme
 - Sehr lehrreich
 - Preise (Sachpreise, Fahrt zur Endrunde, Geld, Studienstiftung)
 - Kontakte
- „Insider-Hinweise“ auf martin-thoma.com
- Offizielles und Aufgaben unter bundeswettbewerb-informatik.de

- Unbedingt Teilnehmen:
 - Spielerische Einführung ins Lösen algorithmischer Probleme
 - Sehr lehrreich
 - Preise (Sachpreise, Fahrt zur Endrunde, Geld, Studienstiftung)
 - Kontakte
- „Insider-Hinweise“ auf martin-thoma.com
- Offizielles und Aufgaben unter bundeswettbewerb-informatik.de

- Unbedingt Teilnehmen:
 - Spielerische Einführung ins Lösen algorithmischer Probleme
 - Sehr lehrreich
 - Preise (Sachpreise, Fahrt zur Endrunde, Geld, Studienstiftung)
 - Kontakte
- „Insider-Hinweise“ auf martin-thoma.com
- Offizielles und Aufgaben unter bundeswettbewerb-informatik.de

- Unbedingt Teilnehmen:
 - Spielerische Einführung ins Lösen algorithmischer Probleme
 - Sehr lehrreich
 - Preise (Sachpreise, Fahrt zur Endrunde, Geld, Studienstiftung)
 - Kontakte
- „Insider-Hinweise“ auf martin-thoma.com
- Offizielles und Aufgaben unter bundeswettbewerb-informatik.de

Aufgabe 1a)

Entwerfen Sie eine Klasse Baby mit den Attributen Name, Gewicht, Größe und Lautstärke (eventuell fallen Ihnen weitere sinnvolle Attribute ein).

Aufgabe 1a) - Lösung

Baby.java

```
1 public class Baby {  
2     /** Der Vor- und Nachname des Babys */  
3     public String name;  
4  
5     /** Gewicht in Gramm */  
6     public int weight;  
7  
8     /** Größe in cm */  
9     public int size;  
10  
11     /** Lautstärke in dB */  
12     public float loudness;  
13 }
```

Aufgabe 1b)

Schreiben Sie einen Konstruktor für ihre Baby-Klasse. Der Name soll nicht in den Konstruktor.

Aufgabe 1b) - Lösung

Baby.java

```
1 public class Baby {
2     /**
3      * Write something in the constructor
4      * @param weight set weight in gramm
5      * @param size the size in cm
6      * @param loudness the loudnes in dB
7      */
8     public Baby(int weight, int size, float loudness) {
9         super();
10        this.name = name;
11        this.weight = weight;
12        this.size = size;
13        this.loudness = loudness;
14    }
15
16    /** Der Vor- und Nachname des Babys */
17    public String name;
18
19    /** Gewicht in Gramm */
20    public int weight;
21
22    /** Größe in cm */
23    public int size;
24
25    /** Lautstärke in dB */
26    public float loudness;
27 }
```

Aufgabe 1c)

Schreiben Sie eine `main`-Methode und instanzieren Sie Babies.

Aufgabe 1d)

Erweitern Sie die Klasse Baby um ein Attribut „Geschlecht“.
Jedes Mädchen ohne Taufe (also ohne weitere Namenszuweisung) soll bei der Geburt
(also der Objekterstellung) den Namen „Anna“ bekommen, jeder Junge den
Namen „Bob“.

Habt ihr ...

- ... den Disclaimer abgeben? Deadline: Freitag, 02. November 2012
- ... euch für die Klausur angemeldet? vernünftige Deadline: Noch dieses Jahr!
- ... das Übungsblatt angefangen? Deadline: Montag, 05. November 2012, 13 Uhr

Vielen Dank für eure Aufmerksamkeit!

