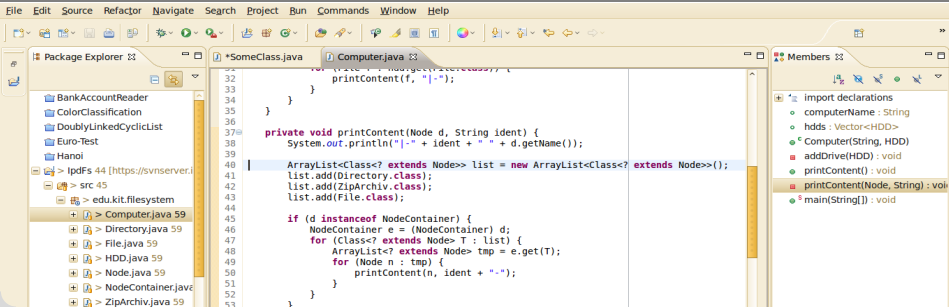


# Programmieren-Tutorium Nr. 10 bei Martin Thoma

TODO: Subtitle setzen!

Martin Thoma | 19. November 2012

FAKULTÄT FÜR INFORMATIK



The screenshot shows an IDE with the following components:

- Package Explorer:** Lists packages like `edu.kit.filesystem` and files like `Computer.java`, `Directory.java`, `File.java`, `HDD.java`, `Node.java`, `NodeContainer.java`, and `ZipArchiv.java`.
- Editor:** Displays the code for `Computer.java`. The code includes a `printContent` method that recursively prints the contents of a `Node` and its children. It uses `ArrayList` to store children and `System.out.println` for output.
- Members:** Shows the class members for `Computer`, including `import declarations`, `computerName : String`, `hdds : Vector<HDD>`, `Computer(String, HDD)`, `addDrive(HDD) : void`, `printContent() : void`, `printContent(Node, String) : void`, and `main(String[]) : void`.

```
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     ArrayList<Class? extends Node> list = new ArrayList<Class? extends Node>();
41     list.add(Directory.class);
42     list.add(ZipArchiv.class);
43     list.add(File.class);
44
45     if (d instanceof NodeContainer) {
46         NodeContainer e = (NodeContainer) d;
47         for (Class<? extends Node> T : list) {
48             ArrayList<? extends Node> tmp = e.get(T);
49             for (Node n : tmp) {
50                 printContent(n, ident + "-");
51             }
52         }
53     }
```

- 1 Einleitung
- 2 Coding Style
- 3 Praxis
- 4 Allgemeine Hinweise zu ÜB
- 5 Hinweise zum ÜB 3
- 6 Abspann

## QuizArray.java

```
1 public class QuizArray {
2     public static void main(String[] args) {
3         String[] myArray1 = { " geh ", "du", "alter", "esel" };
4         String[] myArray2 = myArray1;
5         myArray2[3] = "sack";
6
7         System.out.print("myArray1: ");
8         for (int i = 0; i < myArray1.length; i++) {
9             System.out.print(myArray1[i] + " ");
10        }
11
12        System.out.print("\nmyArray2: ");
13        for (int i = 0; i < myArray2.length; i++) {
14            System.out.print(myArray2[i] + " ");
15        }
16    }
17 }
```

## Quelltext

```
1 public class QuizArray {
2     public static void main(String[] args) {
3         String[] myArray1 = { " geh ", "du", "alter", "esel" };
4         String[] myArray2 = myArray1;
5         myArray2[3] = "sack";
6
7         System.out.print("myArray1: ");
8         for (int i = 0; i < myArray1.length; i++) {
9             System.out.print(myArray1[i] + " ");
10        }
11
12        System.out.print("\nmyArray2: ");
13        for (int i = 0; i < myArray2.length; i++) {
14            System.out.print(myArray2[i] + " ");
15        }
16    }
17 }
```

## Ausgabe

myArray1: geh du alter sack  
myArray2: geh du alter sack

# Shallow copy und deep copy

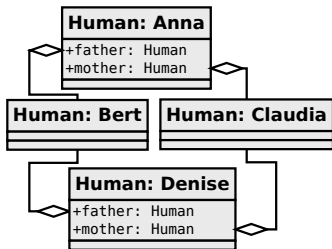
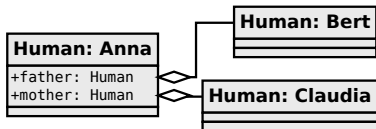


Abbildung : shallow copy

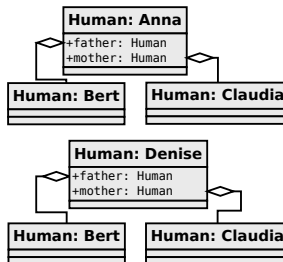


Abbildung : deep copy

Weitere infos auf [stackoverflow.com](https://stackoverflow.com)

- Java macht standardmäßig eine „shallow copy“
- Wenn ihr eine „deep copy“ wollt, müsst ihr das selbst machen

## Lehre aus dem Quiz

Beim kopieren von Objekten (alles außer `boolean`, `byte`, `int`, `long`, `float`, `double` - oder einfacher, alles was mit einem Großbuchstaben beginnt) ist Vorsicht geboten.

# Shallow copy und deep copy

Deep copy sieht im Beispiel so aus:

QuizArray.java

```
1 public class QuizArray {
2     private static String[] deepCopy(String[] original) {
3         String[] copy = new String[original.length];
4         for (int i = 0; i < original.length; i++) {
5             copy[i] = original[i];
6         }
7         return copy;
8     }
9
10    public static void main(String[] args) {
11        String[] myArray1 = { " geh ", "du", "alter", "esel" };
12        String[] myArray2 = deepCopy(myArray1);
13        myArray2[3] = "sack";
14
15        System.out.print("myArray1: ");
16        for (int i = 0; i < myArray1.length; i++) {
17            System.out.print(myArray1[i] + " ");
18        }
19
20        System.out.print("\nmyArray2: ");
21        for (int i = 0; i < myArray2.length; i++) {
22            System.out.print(myArray2[i] + " ");
23        }
24    }
25 }
```

## Definition: Magic numbers

Unique values with unexplained meaning or multiple occurrences which could (preferably) be replaced with named constants.

Warum sind Konstanten besser als magic numbers?

- Code wird lesbarer
- Code wird leichter erweiterbar
- Kein (nennenswerter) Geschwindigkeits- oder Speicherplatznachteil

## Faustregel

Alle Zahlen, bis auf 0, 1 und -1 sollten mittels Konstanten verwendet werden.



# Magic Numbers: Beispiel mit magic numbers

```
1 public class ConnectFourGame implements Cloneable {
2     private Color[] [] board = new Color[7][6];
3     private Color lastMove;
4     private Color winner;
5     private boolean isFinished = false;
6
7     //[...]
8
9     public void printBoard() {
10         for (byte y = 5; y >= 0; y--) {
11             System.out.print(y + " ");
12             for (byte x = 0; x < 7; x++) {
13                 if (board[x][y] == null) {
14                     System.out.print(" ");
15                 } else if (board[x][y] == Color.RED) {
16                     System.out.print("r");
17                 } else {
18                     System.out.print("w");
19                 }
20             }
21             System.out.println("");
22         }
23         System.out.print(" ");
24         for (byte x = 0; x < 7; x++) {
25             System.out.print(x);
26         }
27     }
28 }
```

# Magic Numbers: Beispiel mit magic Konstanten

```
1 public class ConnectFourGame implements Cloneable {
2     public static final int BOARD_HEIGHT = 4;
3     public static final int BOARD_WIDTH = 4;
4     public static final int WINNING_NR = 4;
5
6     private Color[] [] board = new Color[BOARD_WIDTH][BOARD_HEIGHT];
7     private Color lastMove;
8     private Color winner;
9     private boolean isFinished = false;
10
11     //[...]
12
13     public void printBoard() {
14         for (byte y = BOARD_HEIGHT - 1; y >= 0; y--) {
15             System.out.print(y + " ");
16             for (byte x = 0; x < BOARD_WIDTH; x++) {
17                 if (board[x][y] == null) {
18                     System.out.print(" ");
19                 } else if (board[x][y] == Color.RED) {
20                     System.out.print("r");
21                 } else {
22                     System.out.print("w");
23                 }
24             }
25             System.out.println("");
26         }
27         System.out.print(" ");
28         for (byte x = 0; x < BOARD_WIDTH; x++) {
29             System.out.print(x);
30         }
```

Schreiben Sie eine Methode `float getAverage(int[] numbers) { ... }`,  
die den Durchschnitt der Zahlen im Array `numbers` berechnet.

```
1 public class AverageCalculation {
2     public static float getAverage(int[] numbers) {
3         float sum = 0.0f;
4         for (int i = 0; i < numbers.length; i++) {
5             sum += numbers[i];
6         }
7         return sum / numbers.length;
8     }
9
10    public static void main(String[] args) {
11        int[] numbers = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };
12        System.out.println(getAverage(numbers));
13    }
14 }
```

Wir haben ein Array mit IATA-Codes (Flughafenkürzel) gegeben und möchten den Inhalt auf der Konsole ausgeben.

```
1 public class IataCode {
2     public static void printIATACodes(String[] myArray) {
3         for (int i = 0; i < myArray.length; i++) {
4             System.out.println(myArray[i]);
5         }
6     }
7
8     public static void main(String[] args) {
9         String[] iataCodes = new String[4];
10        // Flughafen München
11        iataCodes[0] = "MUC";
12        // Flughafen Berlin Brandenburg
13        iataCodes[1] = "BER";
14        // Flughafen Augsburg
15        iataCodes[2] = "AGB";
16        printIATACodes(iataCodes);
17    }
18 }
```

Kein Compiler- oder Laufzeitfehler, obwohl `iataCodes[3]` nicht initialisiert ist!

## Ausgabe

MUC  
BER  
AGB  
null

# Flughafenkürzel: Umgekehrte Reihenfolge

Modifiziere das Programm so, dass die Kürzel in umgekehrter Reihenfolge ausgegeben werden.

## Lösung von vorhin

```
1 public class IataCode {
2     public static void printIATACodes(String[] myArray) {
3         for (int i = 0; i < myArray.length; i++) {
4             System.out.println(myArray[i]);
5         }
6     }
7
8     public static void main(String[] args) {
9         String[] iataCodes = new String[4];
10        // Flughafen München
11        iataCodes[0] = "MUC";
12        // Flughafen Berlin Brandenburg
13        iataCodes[1] = "BER";
14        // Flughafen Augsburg
15        iataCodes[2] = "AGB";
16        printIATACodes(iataCodes);
17    }
18 }
```



# Flughafenkürzel: Umgekehrte Reihenfolge - Lösung

```
1 public class IataCode {
2     public static void printIATACodes(String[] myArray) {
3         for (int i = 0; i < myArray.length; i++) {
4             System.out.println(myArray[myArray.length - i - 1]);
5         }
6     }
7
8     public static void main(String[] args) {
9         String[] iataCodes = new String[4];
10        iataCodes[0] = "MUC";
11        iataCodes[1] = "BER";
12        iataCodes[2] = "AGB";
13        printIATACodes(iataCodes);
14    }
15 }
```

Modifiziere das Programm so, dass nur jedes dritte Element ausgegeben wird.

## Lösung von vorhin

```
1 public class IataCode {
2     public static void printIATACodes(String[] myArray) {
3         for (int i = 0; i < myArray.length; i++) {
4             System.out.println(myArray[i]);
5         }
6     }
7
8     public static void main(String[] args) {
9         String[] iataCodes = new String[4];
10        // Flughafen München
11        iataCodes[0] = "MUC";
12        // Flughafen Berlin Brandenburg
13        iataCodes[1] = "BER";
14        // Flughafen Augsburg
15        iataCodes[2] = "AGB";
16        printIATACodes(iataCodes);
17    }
18 }
```

```
1 public class IataCode {
2     public static void printIATACodes(String[] myArray) {
3         for (int i = 0; i < myArray.length; i += 3) {
4             System.out.println(myArray[i]);
5         }
6     }
7
8     public static void main(String[] args) {
9         String[] iataCodes = { "MUC", "BER", "AGB", "ABC", "DEF",
10                                "GIH", "JKL", "MNO", "PQR", "STU", "VWX", "YZ" };
11         printIATACodes(iataCodes);
12     }
13 }
```

Gegeben ist ein Schwarz-Weiß Bild `int[][] image`.

- 1 Was repräsentiert `image[2][42]` ?
- 2 Schreiben Sie eine Methode, die die durchschnittliche Farbe von `image` berechnet und zurückgibt.

```
1 public static int getAverageColor(int[] [] image) {  
2     int sum = 0;  
3     for (int x = 0; x < image.length; x++) {  
4         for (int y = 0; y < image[0].length; y++) {  
5             sum += image[x][y];  
6         }  
7     }  
8     return sum / (image.length * image[0].length);  
9 }
```

## Mögliche Probleme:

- Ein `int` für `sum` ist eventuell zu klein
- 255 Grauwerte, `int` hat maximalwert von 2.147.483.64

$$\Rightarrow \frac{2147483647}{255} \approx 8.4 \cdot 10^6$$

$\Rightarrow$  maximal 8.4 Millionen Pixel

$$\Rightarrow \sqrt{8.4 \cdot 10^6} \approx 2900$$

$\Rightarrow$  maximale Auflösung von  $2900 \times 2900$  Pixel

- Eventuell `int[][] image = new int[0][0];`

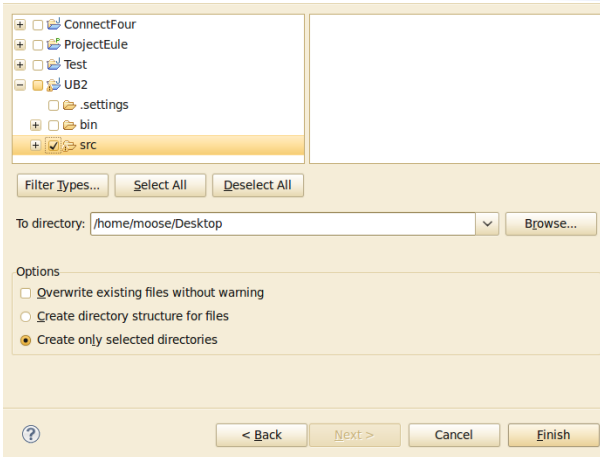
Implementieren Sie das Sieb des Erasthostenes.

- Fragen frühzeitig stellen!
- Wenn ihr Pakete verwendet, am Anfang eurer Dateien also `package AB_2;` oder ähnliches steht, müsst ihr die korrekte Ordnerstruktur in den Praktomat hochladen
- In Eclipse im Fenster „Navigation“ den Ordner „src“ auswählen
- `File` » `Export...` » `File system`
- Zip-Datei aus dem exportiertem src-Ordner erstellen
- Zip-Datei hochladen

# Projekt mit Eclipse exportieren

## File system

Export resources to the local file system.



Project tree:

- ConnectFour
- ProjectEule
- Test
- UB2
  - .settings
  - bin
  - src**

Buttons: Filter Types..., Select All, Deselect All

To directory: /home/moose/Desktop Browse...

Options:

- ☐ Overwrite existing files without warning
- ☐ Create directory structure for files
- ☒ Create only selected directories

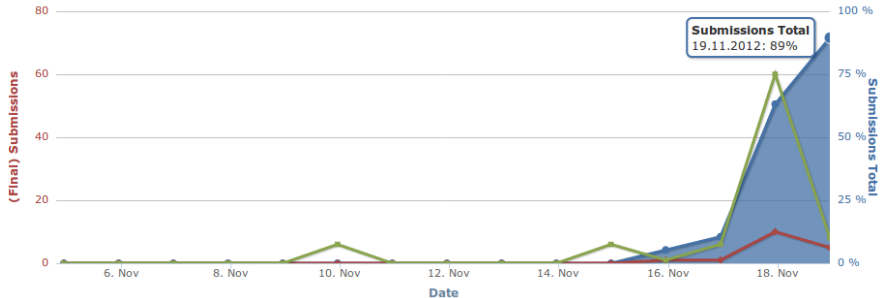
Navigation: ? < Back Next > Cancel Finish



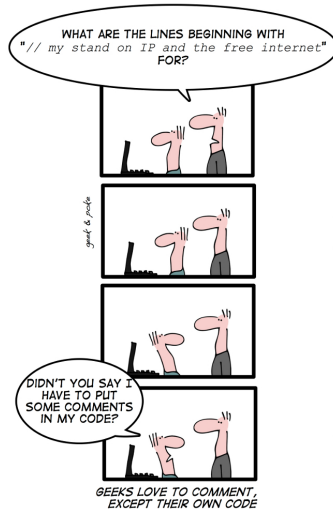
- Eclipse-Warnungen beherzigen und die Ursache dafür beseitigen
- Formattieren: `Strg` + `A` und `Strg` + `⇧` + `F`
- Encoding: `Project` » `Properties` » `Ressource` » `Text file encoding` » `UTF-8`

- A1:
  - Pro Aufgabe jeweils eine Textdatei( A1.txt , B1.txt )
  - Schaut euch meinen Artikel [Java Puzzle #1: Pre- and Postincrement](#) an
- A2: Buchstaben „d“ und „f“ nicht verwenden

# Abgaben um 11:55 Uhr



# Vielen Dank für eure Aufmerksamkeit!



<http://geek-and-poke.com/2010/01/what-geeks-love.html>