

Übungen zu Programmieren - Code-Formatierung -

Tutoren:

Sven Groppe

groppe@ifis.uni-luebeck.de

Florian Frischat

prog2@ifis.uni-luebeck.de

David Gregorczyk

prog6@ifis.uni-luebeck.de

Dana Linnepe

prog5@ifis.uni-luebeck.de

Sven-Erik Pfeiffer

prog4@ifis.uni-luebeck.de

Markus Weigel

prog1@ifis.uni-luebeck.de

Simon Werner

prog3@ifis.uni-luebeck.de

Code-Formatierung

Regel 1: Einrücken von Codeabschnitten

Beispiel (falsch formatiert)

```
public static void schlechtLesbar(int temperatur) {
System.out.println("Hallo Übungsgruppe.");
if (temperatur > 14)
System.out.println("Angenehme " + temperatur + " C");
else
System.out.println("Oha! Ziemlich kalt. Nur schlappe und unangenehme " +
temperatur + " C");
}
```

- Zusammenhängende Abschnitte **nicht** erkennbar
- Abhilfe durch Einrücken

Code-Formatierung

Regel 1: Einrücken von Codeabschnitten

Beispiel (richtig formatiert)

```
public static void schlechtLesbar(int temperatur) {
    System.out.println("Hallo Übungsgruppe.");
    if (temperatur > 14)
        System.out.println("Angenehme " + temperatur + " C");
    else
        System.out.println("Oha! Ziemlich kalt. Nur schlappe und unangenehme " +
            temperatur + " C");
}
```

- Einrücken von Anweisungen innerhalb der Methode
- Entsprechend der Verschachtelung weitere Einrückungen
- Umgebrochene Zeilen doppelt einrücken (s. *else System.out.println(...);*)

Code-Formatierung

Regel 2: Ein Befehl pro Zeile

Beispiel (falsch formatiert)

```
public static int schlechtLesbar(int zahl1, int zahl2) {
    System.out.println("Hallo Übungsgruppe.");
    if (zahl1 < zahl2) zahl1 = zahl2*zahl1+zahl2;
    else zahl2 = zahl2*zahl2; System.out.println("Zahl2 ist nun: " + zahl2);
    return zahl1+zahl2;
}
```

- Mehr als eine Anweisungen pro Zeile missverständlich
(siehe *System.out.println(...)* in *else*-Zeile)

Code-Formatierung

Regel 2: Ein Befehl pro Zeile

Beispiel (richtig formatiert)

```
public static int schlechtLesbar(int zahl1, int zahl2) {
    System.out.println("Hallo Übungsgruppe.");
    if (zahl1 < zahl2)
        zahl1 = zahl2*zahl1+zahl2;
    else
        zahl2 = zahl2*zahl2;
    System.out.println("Zahl2 ist nun: " + zahl2);
    return zahl1;
}
```

- Nun ersichtlich:
Ausführung von *System.out.println(...)* unabhängig von *if* und *else!!*

Code-Formatierung

Regel 3: Einsatz geschweifter Klammern

Beispiel

```
public static void ordentlichEingerueckt(int z) {
    boolean test1 = false;
    if (test1) {
        for (int i=0; i<z; i++) {
            // irgendwelche Anweisungen
        } //ende for
    } //ende if
    else {
        // noch viel mehr Anweisungen
    } //ende else
} // ende ordentlichEingerueckt
```

- Öffnende geschweifte Klammer am Ende des vorangehenden Befehls
- Schließende geschweifte Klammer bündig unter dazugehörigem Befehl

Code-Formatierung

Regel 4: Einrücken im *switch*-Block

Beispiel

```
switch(i) {
case 1:
    System.out.println("Fall 1");
    break;
case 2:
    System.out.println("Fall 2");
    break;
default:
    System.out.println("Fall default");
    break;
} // ende switch
```

- *case*-Blöcke bündig mit *switch*-Anweisung
- Zusätzliches Einrücken der Befehle eines *case*-Falls
- *case*-Blöcke können optional eingerückt werden.

Code-Formatierung

Regel 5: Namensgebung Klassen

Beispiel

```
• class Student  
• class Tier  
• class MusikVerwaltung
```

- Jedes Wort im Klassennamen beginnt mit einem Großbuchstaben
- Klassenname repräsentativ für Funktionalität der Klasse

Code-Formatierung

Regel 6: Namensgebung Variablen und Methodennamen

Beispiele für Variablen

- `int count`
- `String nameOfArtist`
- `float bigFloatingPoint`

Beispiele für Signaturen

- `private String getName()`
- `private void setNameOfArtist(String name)`
- `public int calculateSum(int summand1, int summand2)`

- Jede Variable/Signatur beginnt mit einem Kleinbuchstaben
- Folgewörter beginnen mit einem Großbuchstaben
- Bezeichnungen repräsentativ für...
 - ... Inhalt der Variablen
 - ... Funktion der Methode

Code-Formatierung

Regel 7: Namensgebung Konstanten

Beispiele für Konstanten

```
• final private double PI = 3.14159;  
• final private String EMAIL_UNSERERER_GRUPPE = "ifis_prog3@ifis.uni-luebeck.de";
```

- Komplette Großschreibung
- Folgewörter durch Unterstriche separieren
- Bezeichnungen repräsentativ für Inhalt der Konstanten

Code-Formatierung

Häufig verwendete Code-Fragmente

for-Anweisung (einzeilig)

```
for (int i=0; i<50; i++)  
    System.out.println(i);
```

for-Anweisung (mehrzeilig)

```
for (int i=0; i<50; i++){  
    i = i + 2;  
    System.out.println(i);  
}
```

do-while-Anweisung (einzeilig)

```
do  
    System.out.println(i++);  
while(i<10);
```

do-while-Anweisung (mehrzeilig)

```
do {  
    System.out.println(i++);  
    System.out.println("i = i+1.");  
} while(i<10);
```

Code-Formatierung

Häufig verwendete Code-Fragmente

if-Anweisung (einzeilig)

```
if (esIstNeujahr)
    weihnachtenIstVorbei = true;
else if (esIstHimmelfahrt)
    esIstVatertag = true;
else
    System.out.println("Genug
        gefeiert!");
```

if-Anweisung (mehrzeilig)

```
if (esIstNeujahr){
    weihnachtenIstVorbei = true;
    System.out.println(
        "Frohes Neues!");
}
else if (esIstHimmelfahrt) {
    esIstVatertag = true;
    System.out.println(
        "Bollerwagen beladen");
}
else {
    feiern = false;
    System.out.println(
        "Genug gefeiert!");
}
```

Code-Formatierung

Häufig verwendete Code-Fragmente

Typischer Klassenaufbau

```
public class MeineKlasse{
    // Konstanten
    public final int KONSTANTE = -1;

    // globale Variabeln
    private int variable;

    // Konstruktor(en)
    public MeineKlasse(<Parameterliste>){
        // Anweisungen
    }

    // Methoden
    public int berechneVariable(<Parameterliste>) {
        // Anweisungen
    }
}
```

Code-Formatierung

Häufig verwendete Code-Fragmente

Typischer Methodenaufbau

```
public int berechneProdukt(int multiplikant, int multiplikator) {  
    // lokale Variablen  
    // (stehen ausschließlich dieser Methode zur Verfügung!!!)  
    int produkt;  
  
    // Anweisungen...  
    // z.B. umständliche Berechnung des Produkts durch eine Schleife  
    for (int i=0; i<multiplikant; i++) {  
        produkt = produkt + multiplikator;  
    }  
  
    // Rückgabe des Ergebnisses (falls benötigt)...  
    // z.B. das berechnete Produkt  
    return produkt;  
}
```

Code-Formatierung

Quellen

- Dietmar Ratz, Jens Scheffler, Detlef Seese:
„**Grundkurs Programmieren in Java: Band 1**“, Hanser, 2001,
ISBN: 3-446-21813-0

Für weiterführende Informationen unbedingt einen Blick auf folgende Seite werfen:

- „**Code Conventions for the Java Programming Language**“, Sun Microsystems,
<http://java.sun.com/docs/codeconv/>