

## Programmieren I + II

### *Regeln der Code-Formatierung*

Die hier vorgestellten Regeln zur Formatierung von Java-Programmen basieren auf den „Code Conventions for the Java Programming Language“:

<http://www.oracle.com/technetwork/java/codeconv-138413.html>

Weitere Beispiele kann man z. B. auch dem Buch [1], S. 660 ff., entnehmen.

### 1. Einrückung

Die Einrückung erfolgt ausschließlich mit vier Leerzeichen.

Beispiel :

```
1 public static void beispiel(int x) {  
    System.out.println("Hallo_Welt,_das_ist_ein_Beispiel.");  
3     if(x < 5) {  
        System.out.println("x_ist_kleiner_als_5");  
5     } else {  
        System.out.println("x_ist_groesser_als_5");  
7     }  
}
```

Falsches Beispiel:

```
public static void beispiel(int x) {  
2 System.out.println("Hallo_Welt,_das_ist_ein_Beispiel.");  
    if(x < 5) {  
4 System.out.println("x_ist_kleiner_als_5");  
    } else {  
6 System.out.println("x_ist_groesser_als_5");  
    }  
8 }
```

## 2. Geschweifte Klammern

Die öffnende Klammer kommt in dieselbe Zeile wie der Befehl, mit einem Leerzeichen Abstand. Die schließende Klammer kommt in eine eigene Zeile, eingerückt auf die Höhe des Befehls. Hinter der schließenden Klammer darf kein weiterer Befehl stehen. Ausnahme die do-while-Schleife oder eine else-if-Bedingung. Es werden auch alle einzeiligen Befehle geklammert.

Beispiel:

```
public static void beispiel(int x) {  
2    System.out.println("Hallo_Welt, _das_ist_ein_Beispiel.");  
    if(x < 5) {  
4        System.out.println("x_ist_kleiner_als_5");  
    } else if(x > 9) {  
6        System.out.println("x_ist_groesser_als_5");  
    } else {  
8        System.out.println("x_ist_zwischen_5_und_9");  
    }  
10 }
```

Falsches Beispiel:

```
public static void beispiel(int x)  
2 {  
    System.out.println("Hallo_Welt, _das_ist_ein_Beispiel.")  
4    if(x < 5)  
    {  
6        System.out.println("x_ist_kleiner_als_5");  
    }  
8    else  
    {  
10        System.out.println("x_ist_grsser_als_5");  
    }  
12 }
```

### 3. Nur ein Befehl pro Zeile

Mehrere Befehle pro Zeile werden nicht akzeptiert, Ausnahme else-if-Bedingung.

Beispiel:

```
public static void beispiel(int x) {  
2    System.out.println("Hallo_Welt,_das_ist_ein_Beispiel.");  
    if(x < 5) {  
4        System.out.println("x_ist_kleiner_als_5");  
    } else {  
6        System.out.println("x_ist_groesser_als_5");  
    }  
8 }
```

Falsches Beispiel:

```
public static void beispiel(int x) {  
2    System.out.println("Hallo_Welt,_das_ist_ein_Beispiel.");  
    if(x < 5) { System.out.println("x_ist_kleiner_als_5");  
4 } else { System.out.println("x_ist_groesser_als_5");  
    }  
6 }
```

## 4. Switch-Case-Block

Der Switch-Case-Block wird einmal gesamt geklammert. Die Case-Anweisungen werden eingerückt, die nachfolgenden Befehle werden jeweils in eine neue Zeile geschrieben und ebenfalls eingerückt. Außerdem ist ein default-Case Pflicht. Vor jedem Case und dem Default eine Leerzeile Abstand. Ausnahme ist das erste Case, das darf sofort in der nächsten Zeile angefangen werden.

Beispiel:

```
switch(i) {  
2     case 1:  
        System.out.println("i ist: " + i);  
4        i = 99;  
        break;  
6  
        case 2:  
8        System.out.println("i ist: " + i);  
        i = 99;  
10       break;  
12  
        default:  
        System.out.println("default-Case");  
14 }
```

Falsches Beispiel:

```
switch(i) {  
2     case 1:  
        System.out.println("i ist: " + i);  
4     i= 99;  
        break;  
6     case 2: System.out.println("i ist: " + i); i= 99;  
        break;  
8 }
```

## 5. Klassennamen

Die Klassennamen beginnen immer mit einem Großbuchstaben, keine Zahlen, Sonderzeichen oder Schlüsselwörter sind erlaubt. Setzen sich Klassennamen aus mehreren Wörtern zusammen, so sind diese in CamelCase zu schreiben, also jeweils auch wieder groß zu beginnen.

Beispiele:

TestClass Katze HelloWorld

## 6. JavaDoc-Kopf

Jede Klasse und jedes Interface ist mit einem JavaDoc-Kopf zu versehen. Der JavaDoc-Kopf enthält eine Beschreibung der Klasse und zusätzlich noch den Parameter '@author'. Hinter dem '@author' stehen noch Vorname, Name, Matrikelnummer, und Gruppe aller mitwirkenden Programmierer. Der Tag '@version' wird empfohlen, ist aber keine Pflicht.

Beispiel:

```
1  /**
   *   Dies ist ein Klassenkommentar.
   *   Es soll die Klasse beschreiben.
3
5   *   @author Max Mustermann 1234567 Gruppe 0a
   *   @author Michaela von Musterfrau 0123456 Gruppe 0a
7  */
```

## 7. Methoden- und Variablennamen

Die Namen der Methoden oder Variablen beginnen immer mit einem Kleinbuchstaben. Setzt sich ein Name aus mehreren Wörtern zusammen, so ist wieder das CamelCase anzuwenden.

Beispiele:

```
1  i index helloWorld anzListe
```

Die Klammern eines Arrays werden direkt hinter den Typ geschrieben.

Beispiele:

```
1  String [] a;
   MeineKlasse [] b;
```

## 8. Konstanten

Die Konstanten werden ausschließlich mit Großbuchstaben bezeichnet. Bei mehreren Teilwörtern ist ein Unterstrich zu verwenden.

Beispiele:

```
MAXIMUM MIN_VALUE PI NEUE_KONSTANTE
```

## 9. Sprache und Namensgebung

Die Sprache der Variablen- und Methodennamen sowie der Kommentare sind einheitlich zu wählen. Die Namensgebung ist so zu wählen, dass aus dem Bezeichner eindeutig zu erkennen ist, wozu diese Variable eingesetzt wird. Als Sprache ist entweder Deutsch oder Englisch zu wählen und dann konsequent zu benutzen. Wobei z. B. Variablen auf Englisch und Kommentare auf Deutsch erlaubt wären.

Beispiel:

```
1 i anzListe farbe name
```

Falsches Beispiel:

```
1 a1 a2 a3 a4 farbe size
```

## 10. Umlaute und sprachspezifische Sonderzeichen

Sprachspezifische Umlaute wie z. B. 'ä', 'ü', 'ö' oder 'å' sind nicht für die Bezeichner zu benutzen.

Beispiel:

```
1 groesse oeffnen schliessen ueber
```

## 11. Kommentare

Einzeilige Kommentare werden in eine Zeile geschrieben und mit `//` eingeführt. Mehrzeilige stehen direkt vor einem Codefragment und werden mit `/*` eingeführt und mit `*/` beendet. Jede öffentliche Methode ist mit einem JavaDoc-Kommentar zu versehen, der mindestens `@param` und `@return` enthalten muss (sofern Werte übergeben werden oder ein Rückgabewert existiert). Eingeführt wird ein JavaDoc-Kommentar mit `/**` und mit `*/` beendet.

Beispiel einzeiliger Kommentar:

```
1 public void beispiel(int x) {  
    System.out.println("Hallo_Welt, _das_ist_ein_Beispiel.");  
3    if(x < 5) { // Ueberpruefung, ob x kleiner als 5 ist  
        System.out.println("x_ist_kleiner_als_5");  
5    } else {  
        System.out.println("x_ist_groesser_als_5");  
7    }  
}
```

Beispiel mehrzeiliger Kommentar:

```
public void beispiel(int x) {  
2    System.out.println("Hallo_Welt, _das_ist_ein_Beispiel.");  
    /*  
4        Ueberpruefung ob x  
        kleiner als 5 ist  
6    */  
    if(x < 5) {  
8        System.out.println("x_ist_kleiner_als_5");  
    } else {  
10        System.out.println("x_ist_groesser_als_5");  
    }  
12 }
```

Beispiel JavaDoc Kommentar:

```
/**
2  Diese Methode ist eine Beispielmethode und ueberprueft, ob
   x groesser ist als 5
4  @param x und die Beschreibung zu x
   @return Einen String mit dem Ergebnis ("groesser/kleiner")
6  */
public String beispiel(int x) {
8      System.out.println("Hallo_Welt, _das_ist_ein_Beispiel.");
      if(x < 5) {
10         System.out.println("x_ist_kleiner_als_5");
         return "kleiner";
12     } else {
        System.out.println("x_ist_groesser_als_5");
14         return "groesser";
    }
16 }
```

oder

```
/**
2  * Diese Methode ist eine Beispielmethode und ueberprueft, ob
   * x groesser ist als 5
4  * @param x und die Beschreibung zu x
   * @return Einen String mit dem Ergebnis ("groesser/kleiner")
6  */
public String beispiel(int x) {
8      System.out.println("Hallo_Welt, _das_ist_ein_Beispiel.");
      if(x < 5) {
10         System.out.println("x_ist_kleiner_als_5");
         return "kleiner";
12     } else {
        System.out.println("x_ist_groesser_als_5");
14         return "groesser";
    }
16 }
```

## 12. Weitere Beispiele für Code-Fragmente

a) for-Anweisung, einzeilig:

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



b) for-Anweisung, mehrzeilig:

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

c) if-Anweisung, einzeilig:

```
if (i < 10) {  
    i = 10;  
} else if (i > 100) {  
    i = 100;  
} else {  
    System.out.println("Alles O.K.");  
}
```

d) if-Anweisung, mehrzeilig:

```
if (i < 10) {  
    System.out.println("Erhöhe i");  
    i = 10;  
} else if (i > 100) {  
    System.out.println("Erniedrige i");  
    i = 100;  
} else {  
    System.out.println("Alles O.K.");  
    System.out.println("Keine Änderungen");  
}
```

e) try-catch-Block:

```
try {  
    i = i / j;  
} catch (NumberFormatException e) {  
    System.out.println("Achtung: j == 0");  
    i = 100;  
} finally {  
    System.out.println("Fertig");  
}
```

f) switch-Anweisung:

```
switch(i) {  
    case 1:  
        j = 17;  
        break;  
  
    case 2:  
        j = 23;  
        break;  
  
    default:  
        j = 0;  
}
```

g) do-while-Schleife, einzeilig:

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

h) do-while-Schleife, mehrzeilig:

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

i) Variablendeklaration:

```
int i = 0;  
String name = "Werner";  
Object objekt = new Object();
```

j) Klassendeklarationen:

```
public class KlassenName {  
    // ...  
}
```

k) Konstantendeklarationen:

```
public final static int MINIMUM = 10;  
public final float MAX_VALUE = 100.0f;  
final double KONSTANTE = 23.7;
```

## Literatur

- [1] RATZ, Dietmar; SCHEFFLER, Jens; SEESE, Detlef; WIESENBERGER, Jan: *Grundkurs Programmieren in Java*. 6. aktualisierte und erweiterte Auflage. München Wien: Hanser Verlag, 2011