Kommentare:

1. Nur Blockquotes für Kommentare verwenden:
```
/* Das ist ein Kommentar
*  ….. bla bla
*/
```
2. Linequotes (//) werden nur zum kurzzeitigen auskommentieren von Quellcode genutzt
3. Überflüssige Kommentare an Methoden deren Name bereits aussagekräftig genug ist weglassen
   z.B.
```
/* Returns the name
*   @return – The name
*/
public String getName()
```
4. Kommentare nur an Methoden wo sich der Zustand der Klasse ändert, die Änderung aber nicht über den Methodennamen ersichtlich ist.
5. Kommentare immer in Englisch
6. Kommentare an algorithmisch schwierigen Stellen auch innerhalb von Methoden einfügen
7.

Klammern

1. Block-Klammern (Methoden, Klassen, Schleifen, etc.) immer in die erste Spalte des neuen Blocks und die erste Klammer ist immer in einer neuen Zeile:
```
if ( 1==1)
{
      System.out.println(„Juhu es ist wahr“);
}
```

Klassennamen

1. Aussagekräftige Namen verwenden (lieber zu lang als zu kurz)
2. CamelCase Schreibweise
3. mit großem Buchstaben beginnen

Methodennamen

1. Aussagekräftige Namen verwenden (lieber zu lang als zu kurz)
2. CamelCase Schreibweise
3. mit kleinem Buchstaben beginnen
4. Typische Namen für Zugriff auf gekapselte Membervariablen.
```
int getX();       /* lesender Zugriff auf X */
boolean isX();    /* lesender Zugriff auf X, wenn X vom Typ boolean
                  ist */
void setX(int X); /* schreibender Zugriff auf X */
```

Variablenbezeichner

1. Aussagekräftige Namen verwenden (lieber zu lang als zu kurz)
2. CamelCase Schreibweise
3. mit kleinem Buchstaben beginnen
4. **keine** Typinformationen in den Variablenbezeichner

```
int iNumElements = 0;   /* falsch */
int numElements = 0;    /* richtig */
```

Weiteres

1. `public static` nicht verwenden, notfalls `public static final` nutzen, aber besser ist `private static` und entsprechende getter und setter
2. Möglichst keine Methoden schreiben die die übergebenen Parameter modifiziert. Sollte dies unbedingt nötig sein, dann muss diese Methode mit einem entsprechenden Kommentar versehen werden.
3. Codeleichen vermeiden bzw. nicht mit einchecken:
   Wenn Code kurzzeitig auskommentiert wird z.B. zu Testzwecken oder bei der Fehlersuche, dann sollte bis zum Ende der Arbeitszeit klar sein ob der Code wieder reinkommt oder rausfliegt. Kommentare wie „Überprüfen“ oder „Kann das Weg??“ usw. gelten nicht.
   Solche Codeleichen sollen nicht mit eingecheckt werden ➔ erzeugen nur Fehler bzw. unleserlichen Code.
4. Nur fehlerfreien Code einchecken:
   Mit fehlerfrei ist gemeint, dass der Code keine Syntaxfehler enthält und dass alle vorhandenen Unit-Tests erfolgreich durchgelaufen sind
5.