

Programmier-Richtlinien

vgl. auch <http://java.sun.com/docs/codeconv/>

Programmier-Richtlinien

Bemerkung

- Programme werden häufiger gelesen als geschrieben
- Programmtexte müssen gut lesbar und nachvollziehbar sein

Bemerkung

Nicht die konkrete Ausprägung der Programmier-Richtlinien sondern ihr generelles Vorliegen ist wichtig

Feststellung

- Aufstellen und Einhalten von Programmier-Richtlinien erleichtert das Lesen, Einarbeiten, und Erweitern von Programmtexten

Hinweis

- Das Einhalten der Programmier-Richtlinien ist (durch geeignete Werkzeuge) zu überprüfen

[Balzert 2001, S. 1068ff]

Programmier-Richtlinien

Bezeichner

- Bezeichner für Klassen, Methoden, Attribute etc. müssen "sprechend" und selbsterklärend sein
- Bezeichner enthalten keine Leerzeichen
- in zusammengesetzten Bezeichnern beginnt jedes Wort mit einem Grossbuchstaben (CamelCase)
- Bezeichner sollten durchgängig deutsch oder (bevorzugt) englisch sein, Umlaute und Sonderzeichen sind zu vermeiden

Programmier-Richtlinien

Klassenbezeichner, Interfacebezeichner

- Klassen und Interfaces werden durch Substantive im Singular bezeichnet
- Klassen- und Interfacebezeichner beginnen mit Grossbuchstaben

```
class DigitalLibrary;  
interface Collection;
```

Attributbezeichner

- Attribute werden durch Substantive mit evtl. vorangestelltem Adjektiv im Singular bezeichnet
- Attributbezeichner beginnen mit einem Kleinbuchstaben

```
int i;  
Author theAuthor;  
float currentSpeed;
```

Programmier-Richtlinien

Methodenbezeichner

- Methoden werden durch Verbalphrasen bezeichnet
- Methodenbezeichner beginnen mit einem Kleinbuchstaben
- Methoden, die auf Attribute lesend (getter) oder schreibend (setter) zugreifen werden durch den Attributbezeichner mit vorangestelltem "get" oder "set" benannt

```
setName();  
getName();  
printName();
```

Programmier-Richtlinien

Konstantenbezeichner

- Konstanten werden durch Substantive in GROSSBUCHSTABEN beschrieben
- Teilworte werden durch "_" voneinander getrennt

```
int MAX_AUTHORS = 5;  
int MIN_WIDTH = 4;
```

- generell sind sinnvoll bezeichnete Konstanten der Verwendung konstanter Werte vorzuziehen
 - bessere Lesbarkeit
 - höhere Wartungsfreundlichkeit

Programmier-Richtlinien

Struktur (für Java-Programme)

- .java-Dateien enthalten genau eine Klasse oder genau ein Interface
 - einleitender Kommentar
 - Package-Angabe
 - Import Statement
 - *eine* Klasse/Interface

Bemerkung

die Komponenten sind in der angegebenen Reihenfolge zu notieren

- Klassen/Interfaces bestehen aus
 - einleitender Kommentar
 - Attribute
 - statische (Klassen-) Variablen (public, protected, private)
 - Instanzvariablen (public, protected, private)
 - Methoden
 - Konstruktoren
 - vergleichende Methoden
 - Methoden mit schreibendem Zugriff
 - Methoden mit lesendem Zugriff
 - restliche Methoden

Programmier-Richtlinien

generelle Formatierungsrichtlinien

- Dateien dürfen nicht mehr als 2000 Zeilen besitzen (2000 Zeilen ist schon viel zu viel)
- Zeilen dürfen nicht mehr als 80 Zeichen besitzen
- Verfeinerungen (Blöcke) werden 4 Zeichen eingerückt
- pro Zeile soll höchstens
 - eine Variable deklariert werden
 - ein Programmstatement notiert werden
- Leerzeilen trennen
 - Variablen von Methoden
 - Methoden von Methoden
 - (lokale) Variablen vom ersten Programmstatement
- Leerzeichen trennen
 - Schlüsselworte von "("
 - Operanden von Operatoren (außer ".")

```
int i;
```

```
int j;
```

```
i++;
```

```
j--;
```

```
while (i != 5) {
```

```
...
```

```
}
```


Programmier-Richtlinien

Klammerregeln für Methoden

- zwischen Methodennamen und "(" steht kein Leerzeichen
- Blöcke werden in "{" und "}" eingeschlossen
- öffnende Klammer "{" erscheint in der Zeile des zusammengesetzten Statements
- schließende Klammer "}" erscheint in einer eigenen Zeile auf der selben Einrückungsstufe wie das zusammengesetzte Statement

```
class MyDemo {  
    int ivar;  
  
    doIt(int i, int j) {  
        ivar = i + j;  
    }  
  
    doSmtElse(int i){  
        if (i == 0) {  
            System.out.println("Zero");  
        }  
        else {                                // ( i != 0)  
            System.out.println("NonZero");  
        }  
    }  
}
```

Programmier-Richtlinien

Dokumentation

- Klassen werden durch eine Präambel eingeleitet
 - Autoren
 - Datum der initialen Erstellung
 - Versionsnummer
 - Bearbeitungszustand
(geplant, in Bearbeitung, freigegeben)
 - Änderungshistorie
 - kurze Aufgabenbeschreibung

Programmier-Richtlinien

Dokumentation

- Methoden werden durch eine Präambel eingeleitet
 - Autoren
 - Datum der initialen Erstellung
 - Versionsnummer
 - Bearbeitungszustand (geplant, in Bearbeitung, freigegeben)
 - Änderungshistorie
 - kurze Aufgabenbeschreibung der Methode
 - für Methoden
 - Anforderungen an Eingabedaten
 - Anforderungen an Ausgabedaten (incl. Bezug zu Eingabedaten)
- im Code
 - ist jede Schleife durch ihre Invariante zu dokumentieren
 - ist jeder else-Teil einer Verzweigung mit dem hier zutreffenden Prädikat zu kommentieren