

# Softwaretechnik-Praktikum SS06

## Dokumentationskonzept

---

Eine Dokumentation ist ausschlaggebend für die Qualität des Endprodukts. Die Qualitätssicherung setzt die Einhaltung des Dokumentationskonzeptes voraus. Es wird durch einen einheitlichen Stil von Quelltexten und Quelltextdokumentationen ein gute Wartbarkeit und die Anwendung von Dokumentations- und Testwerkzeugen überhaupt gegeben. Ein ausschlaggebender Punkt hierfür ist, dass Wartung&Pflegenphase ungefähr 80% des Softwarelebenszyklus einnehmen. Zudem wird der Aufwand in die Einarbeitung in das Projekt dadurch minimal gehalten.

## Interne Dokumentation

---

### ► Quellcode

**Der Aufbau des Java Source ist wie folgend festgelegt:**

- Deklaration von Paket und Import-Statements
- Deklaration der Klasse oder des Interface:
  1. Attribute deklarieren
  2. Konstruktoren
  3. Setter- und Getter-Mehoden
  4. andere Methoden

### Kommentare:

Es gibt zwei Arten von Kommentaren. Diese werden vom Compiler ignoriert. Sie dienen dazu, den Quelltext zu dokumentieren und werden grundsätzlich in Deutsch formuliert.

// Kommentar : dienen als kurzen Kommentar für den Programmierer

/\*  
\* Kommentar : sie werden für die Dokumentation mit javadoc genutzt (siehe  
\* Quellcodedokumentation)  
\*/

### if-else - Anweisungen:

Die zueinander gehörende if-else haben untereinander zustehen.

```
if (a<2)
{
    anweisung;
}
else return;
```

### allgemein:

- Variablen- und Methodennamen werden in deutsch benannt
- Operanden und Operator werden durch ein Leerzeichen getrennt  
2 + x
- vor Punktnotation stehen keine Leerzeichen  
objekt.mehtode()
- zwischen Methodename und den in Klammern übergebenen Attribute stehen keine Leerzeichen  
methode(int zahl)
- nach einem Komma wird ein Leerzeichen geschrieben  
int zahl, int wert
- zueinander gehörende öffnende und schließende geschweifte Klammern haben untereinander zu stehen und der darin enthaltene Block wird eingerückt  
public void methode()  
{  
 methodenrumpf;  
}
- für bessere Erkennbarkeit fügen wir an gegebenen Stellen Klammern ein  
statt: (a ==b && b==c) schreiben wir: ((a ==b) && (b==c))
- für Variablen und Methoden sind aussagegebende Namen zu verwenden  
erfasseZeit(), int bruttoBetrag

# Softwaretechnik-Praktikum SS06

## Dokumentationskonzept

---

ein „sauberer“ Quelltext würde so aussehen:

```
import java.*;

public Auto
{
    private int anzahlReparaturen;

    public Auto()
    {
        anzahlReparaturen = 0;
    }

    public void setReparatur(int anzahl)
    {
        anzahlReparaturen = anzahl;
    }

    public int getReparatur()
    {
        return anzahlReparaturen;
    }
}
```

### ► Quellcodekommentierung und javadoc

Damit javadoc aus dem Quellcode ein html-Dokument erstellen kann, müssen folgende Regeln eingehalten werden:

- javadoc-Kommentare beginnen mit `/**` und enden mit `*/`
- vor jeder Zeile innerhalb des Kommentaren kann auch ein `*` stehen
- der Kommentar muss unmittelbar vor der Deklaration der Programmeinheit stehen
- es sind auch innerhalb der Kommentare html-Tags(wie `<h2>Überschrift</h2>`) möglich, Überschriften sollten aber vermieden werden da javadoc selbst schon Formatierung dafür an den Kommentaren vornimmt
- `@` ist ein reserviertes Zeichen, dass von javadoc erkannt wird und dementsprechend im html-Dokument gekennzeichnet werden:
  - `@autor name`: schreibt den Namen des Autors in die Dokumentation
  - `@version version`: schreibt sie Version in die Dokumentation
  - `@see verweis`: erzeugt einen Querverweis im Dokument
  - `@param name beschreibung`: damit können die Parameter einer Methode beschrieben werden
  - `@return beschreibung`: die Beschreibung für den Rückgabewert wird übernommen
  - `@exception klasse beschreibung`: die von der Methode ausgelöste Ausnahme wird in die Dokumentation geschrieben

beispielsweise:

```
/**
 * Kurzbeschreibung
 *
 * @author Implementierer
 * @version 4.1
 * @see http://pcai042.informatik.uni-leipzig.de/swp-06/index.html
 */
public ....
```

### ► Testdokumentation

Jeder Test bedarf einer Dokumentation.

Sie beinhaltet:    Genauere Beschreibung der Tests.  
                    Wozu dient der Test?  
                    Worauf wurde getestet?  
                    Ergebnis

# Softwaretechnik-Praktikum SS06

## Dokumentationskonzept

---

### ► Formate von Dokumenten

Jedes Dokument hat die Schriftart Arial mit Schriftgröße 11. Der Dokumentenkopf und -fuß ist diesem zu entnehmen. Die Gliederung ist sinnvoll und übersichtlich zu wählen. Alle Dokumente werden im pdf-Format verteilt. Die Sprache hierfür wird deutsch sein.

### ► Versionsnummern

Jede Implementierung (Klasse/Interface) hat eine Versionsnummer x.y zutragen. x steht für das Release und wird bei jeder größeren Änderung fortlaufend geändert. y ist das Level und steht für kleinere Änderungen auch diese wird fortlaufend geändert. Die Implementierungsarbeiten beginnen mit 0.1. Zu jeder Änderung ist die Änderungsnummer, der Änderungsgrund und eine kurze Beschreibung zu schriftlich festzuhalten.  
Die erste lauffähige Version trägt die Nummer 1.0.

## Externe Dokumentation

---

### ► Designbeschreibung

Die Designbeschreibung legt die Klassen- und Paketstrukturen dar. Die Veranschaulichung soll durch UML-Diagramme erfolgen. Dadurch sollen Fremdprogrammierer einen Überblick über das Softwarepaket bekommen.

### ► Handbuch und Onlinehilfe

Das Handbuch soll dem Benutzer eine Übersicht über die Software geben und alle Funktionen begreifbar erklären. Weithin werden Angaben über die Installation und Systemvoraussetzungen gegeben. In einem Glossar sollen programmrelevante Begriffe erklärt werden. Auch häufig gestellte Fragen (FAQ) sollten im Handbuch eingebunden sein. Dem Benutzer steht weiterhin die Onlinehilfe zur Verfügung.