

## Dokumentationskonzept

Das Dokumentationskonzept dient zur Vereinheitlichung von Kommentierung und Dokumentation der einzelnen Projektphasen und deren Teilergebnisse (Produktergebnisse). Dokumentarisches Festhalten von Struktur, Funktions- und Arbeitsweise, Quellcode und Verhalten bei bestimmten Situationen ist ein Prozess der während der inkrementellen Softwareentwicklung immer zeitnah und analog erfolgen muss. Er ist wichtig, damit Auftraggeber, Anwender (Benutzer) und Entwickler zu einem späteren Zeitpunkt mit dem Produkt arbeiten, es weiterentwickeln oder pflegen/warten können.

Die Dokumentation erfolgt auf zwei Ebenen:

1. interne, quelltextnahe Projektdokumentation
  - a. Quellcode-Komentierung
  - b. Javadoc-Dokumentation
  - c. Dokumentation der verschiedenen Testfälle
2. externe Projektdokumentation
  - a. Spezielle Entwurfs-Dokumentation (Designbeschreibung)
  - b. Kundendokumentation (Installationsanweisung)
  - c. Handbuch

### **1. Interne Dokumentation**

Sie erfolgt zeitnah und innerhalb des Programms bzw. der Software. Zum Einen bezieht sie sich direkt auf den Entwicklungsprozess, zum Anderen auf das Testen der einzelnen Teilergebnisse. Sie ist ein integraler Bestandteil des Programms und soll von ihm eine Kurzbeschreibung liefern.

Um den Teammitgliedern das Dokumentieren zu erleichtern und auch zu gewährleisten, das jeder sich an die gegebenen Standards hält, ist ein Handout erstellt worden, welches alle wichtigen Aspekte in Kurzform darlegt. Dieses Handout wird allen zugänglich gemacht und kann daher gleich ab der ersten Implementierung genutzt werden.

#### **a. Quellcode- Kommentierung**

Der Programmierer (Entwickler) kommentiert seinen entwickelten Code immer zeitnah und nach folgenden Regeln:

- I. Aussagekräftige, sinnvolle, selbsterklärende und kurze Namen für Methoden, Variablen, Klassen, Konstanten
- II. Methoden- und Klassennamen beginnen mit einem Großbuchstaben und Attribut- und Konstantennamen mit einem kleinen Buchstaben.
- III. Bei zusammengesetzten Bezeichnern kann zur besseren Lesbarkeit ab dem zweiten Teilwort jeder Anfangsbuchstabe groß geschrieben werden.
- IV. Einzeilige Kommentare werden mittels

`// Kommentar`

und mehrzeilige Kommentare durch

`/* Kommentar in  
mehreren Zeilen */`

festgehalten.

- V. Jede Methode und Klasse wird zu Beginn in Hinsicht auf Funktionen, Eingabewerte, Voraussetzungen, Ausgabewerte, Fehleranalysen kurz beschrieben. Dabei wird nach Javadoc-Syntax (siehe Punkt b) dokumentiert.

- VI. Die Speicherinhalte aller Attribute und Konstanten werden mit Javadoc-Kommentaren (siehe Punkt b) beschrieben.
- VII. Im Falle einer bedingten Anweisung (zum Beispiel: if) oder Fallunterscheidung (beispielsweise: switch) sind alle möglichen Varianten zu kommentieren, so dass die Bedingung für die entsprechende Auswahl verständlich ist.
- VIII. Zusammengehörige längere Anweisungsblöcke werden durch Einrückung oder Start-Ende signalisiert.
- IX. Jede .java-Datei erhält einen Bearbeitungsstatus (geplant, in Bearbeitung, vorgelegt, akzeptiert, getestet), Versionsnummer und Name des Programmautors.
- X. Entwicklungsentscheidungen werden auch dokumentiert, indem die Designbeschreibung im wöchentlichen Rhythmus angepasst wird (siehe Punkt 2.a).

### **b. Javadoc-Dokumentation**

Javadoc erstellt aus speziell formatierten Kommentaren, die sich im Quelltext befinden, Dokumentationen zu den Programmen, welche dann mit Hilfe eines Browsers zugänglich sind. Im Standard verwendet, erzeugt sie eine HTML-Dokumentation, die den Aufbau von Paketen und Klassen zeigt. Dabei ist bei der Dokumentation zu beachten bzw. zu verwenden:

- I. Alle javadoc-Kommentare beginnen mit `/**` und enden mit `*/`.
- II. Die javadoc-Kommentare werden unmittelbar vor der Deklaration der Programmeinheit stehen, die dokumentiert werden sollen.
- III. Bezüglich auf Methoden/Funktionen werden wir folgende zusätzliche Kommentare verwenden (→ interne Dokumentation)
  - a. `@author` - Name des Autor
  - b. `@version` - Version der Dokumentation
  - c. `@see` - Erzeugen eines Querverweises in HTML
  - d. `@param` - Beschreiben von Parametern einer Methode
  - e. `@return` - Beschreibung der Rückgabewerte einer Methode
  - f. `@exception` - Beschreibung der Ausnahmebehandlung

Die Javadoc wird in unserem Home-Verzeichnis platziert und für jeden zugänglich gemacht. Die als private deklarierten Attribute werden dabei mit in die API-Dokumentation übernommen (javadoc mit Option „-private“). Die Dokumentation muss nach jeder Programmänderung mittels eines Befehles neu erstellt bzw. aktualisiert werden.

### **c. Dokumentation der verschiedenen Testfälle**

Dokumentiert werden die Ergebnisse von allen Komponenten- und Integrationstests sowie dem Systemtest.

Getestet werden alle Methoden, Programme etc.. Mit den Tests werden zwei Dateien angelegt, die eine Auflistung aller durchgeführten Tests beinhalten:

- 1. Datei für erfolgreiche Tests (Datum, Klasse, Test erfolgreich)
- 2. Datei für nicht erfolgreich Tests (Datum, Klasse, Methode, Fehler)

Ein detailliertes Testkonzept liegt bereits vor und wird in die laufenden Projektarbeiten mit einbezogen. Alle Aspekte in Bezug zu Test können im Testkonzept vom Kunden nachgelesen werden.

## 2. Externe Dokumentation

Die externe Dokumentation beinhaltet alle Dokumente, die dem Kunden zugänglich gemacht werden. In diese Form der Dokumentation können Bereiche der internen Dokumentation mit einfließen. Hauptsächlich ist die externe Dokumentation Basis für ein fehlerfreies Arbeiten mit dem Produkt nach erfolgreicher Produktübergabe.

### a. Spezielle Entwurfs-Dokumentation (Designbeschreibung)

Die Designbeschreibung dient zur verallgemeinerten Darstellung des Produktes insbesondere seiner Funktionen, Besonderheiten und seines Aufbaus (Architektur). Dabei wird geachtet auf:

- I. Allgemeines (Charakterisierung, Systemvoraussetzungen, Abgrenzung)
- II. Produktübersicht
- III. Grundsätzliche Designentscheidungen (Fachkonzepte, Aussehen der Anwendung, Ausgabe und Speicherung)
- IV. Paket- und Klassenstruktur

Die Entwurfsbeschreibung erfolgt zeitnah zum Entwurf und den Implementierungsarbeiten am Produkt. Die genaue Aufstellung dieses Dokumentes erfolgt anhand der Dokumentationen aus der ersten Ebene. Zu diesem Zweck werden in regelmäßigen Abständen Gruppentreffen erfolgen, um einen Austausch aller erstellten Teildokumente untereinander zu gewährleisten, diese Zusammenzutragen zu einer einheitlichen Dokumentation zusammenzufügen. Gleichzeitig kann man in diesem Zusammenhang auftretende technische und organisatorische Voraussetzungen und Probleme diskutieren und den weiteren Vorgang im Projekt besprechen.

### b. Kundendokumentation

Die Kundendokumentation zum Produkt „Kooperationsplattform“ besteht aus einer Installationsanweisung, die die Inbetriebnahme des Produktes erleichtern soll. In diesem Zusammenhang wird ein Dokument erstellt, in welchem alle wichtigen technischen Voraussetzungen festgehalten werden und zudem die Vorgehensweise, sowie alle Parameter, beschrieben wird.

Zusätzlich zur Kundendokumentation werden dem Kunden die Dokumente Lastenheft, Pflichtenheft, Glossar (wird während des Projektes immer wieder angepasst) und Entwurfsbeschreibung übergeben. Diese Dokumente ermöglichen dem Kunden immer einen Überblick über den aktuellen Stand des Projektes zu haben.

### c. Handbuch

Das Handbuch zur „Kooperationsplattform“ wird ein pdf- Dokument werden , welches als Download auf den Seiten der „Kooperationsplattform“ niedergelegt wird. Zusätzlich wird im Arbeitsflächenbereich (z.B. Auflistung aller eigener Lehrveranstaltungen, Prüfungen anzeigen etc.) auf der rechten Seite die Möglichkeit gegeben, alle wichtigen Tipps, Informationen und kurze Hilfestellungen nachzulesen, welche sich themenspezifisch ändern.

Das Handbuch wird parallel zur Implementierung und Test geschrieben, um auftretende Probleme sofort zu erkennen und zu beschreiben. Zudem werden durch Screenshots Überblicke gegeben, welche die Bedienbarkeit erleichtern. Bei der Erstellung beider Dokumente, wird auf einfache und formale Sprache geachtet und zusätzlich Fremdwörter vermieden.