

Qualitätssicherungskonzept

1 Dokumentation

Das Dokumentationskonzept dient der einheitlichen Festlegung von Qualitätssicherungskriterien, die eine einfache Wartbarkeit und übersichtliche Anwendung des Softwareprodukts im inkrementellen Softwareentwicklungs- und Pflegemodell ermöglichen. Die Einhaltung der festgelegten Qualitätssicherungskriterien garantiert die rechtzeitige Verfügbarkeit aller Dokumentationsartefakte mit minimalem Aufwand, da diese im Laufe des Softwareentwicklungsprozesses von den zuständigen Spezialisten produziert werden.

1.1 Quellcode- Konventionen

1.1.1 Javadoc

Javadoc ist ein Software-Dokumentationswerkzeug, das aus Java-Quelltexten automatisch HTML-Dokumentationsdateien erstellt. Die Dokumentation kann somit durch spezielle Kommentare im Quelltext erstellt werden. Dadurch können Beschreibungen für Interfaces, Klassen, Methoden definiert werden.

Namenskonventionen

- Allgemein sind für Klassen, Interfaces, Methoden und Variablen “sprechende” Namen zu wählen.
- Die Benennung der Klassen, Interface, Methoden und Variablen erfolgt einheitlich in englischer Sprache.
- Klassen und Interfaces beginnen grundsätzlich mit einem Großbuchstaben.
- Methoden und Variablen beginnen mit einem Kleinbuchstaben.
- Für zusammengesetzte Wörter gilt folgende Regelung: Jedes Neue Wort beginnt mit einem Großbuchstaben.

Implementierungskommentare Jede nichttriviale Codezeile bedarf eines Kommentars. Zeilen-Kommentare beginnen mit `//` und enden am Zeilenende. Block-Kommentare beginnen mit `/*` und enden mit `*/`

Deklarationen

- Eine Zeile pro Deklaration.
- Variablen möglichst bei Deklaration gleich initialisieren.
- Die Deklarationen sollten immer am Anfang eines Blockes stehen.
- Bei Schnittstellen- und Klassendeklarationen gelten folgende Regeln:
 - Kein Leerzeichen zwischen dem Methodennamen und der aufgehenden Klammer.
 - Öffnende geschwungene Klammern sollten in der selben Zeile stehen.
 - Schließende geschwungene Klammern sollten in einer eigenen Zeile stehen, außer bei leeren Statements, die sofort geschlossen werden.

Entwurfsdokumentation

Inhalt: Übersicht, Abstraktionen, High-Level Funktionalität, Design-Entscheidungen, Anwendungsbereiche, Einschränkungen

Im Rahmen einer Softwaremodellierung wird eine Design-Beschreibung angelegt, die einen Einblick in die Architektur des zu entwickelnden Systems gewährt. Es ermöglicht einerseits anderen projekt-fremden Entwicklern bzw. dem Wartungspersonal sich in kurzer Zeit mit dem Projekt vertraut zu machen und sich leichter einzuarbeiten. Andererseits bekommt der Kunde damit schon im frühen Entwicklungsstadium die Übersicht über die Grundlegenden Eigenschaften des Systems. Konzeptionelle Fehler sollen dadurch vermieden werden.

2 Testkonzept

JUnit-Framework JUnit ist ein Framework zum Testen von Java-Programmen, das besonders für automatisierte Unit-Tests einzelner Units (meist Klassen oder Methoden) geeignet ist.

Die Komponenten von JUnit-Tests:

TestCase jeder TestCase stellt einen Test mit Testfällen dar Trennung von Applikations- und Test-Code

junit.framework.TestCase

TestFixture Stellt allgemeine Objektkonfiguration für die Tests bereit Instanziert Objekte, Erzeugt DB-Connection, ...

TestSuite Fasst einzelne Tests zu einer logischen Einheit zusammen *junit.framework.TestSuite* kann bspw. alle "testXXX" Methoden eines TestCases ausführen.

TestRunner um Tests auszuführen

3 Organisatorische Festlegungen

3.1 Namenskonventionen

Es werden die Vorgaben aus 1.1 streng umgesetzt.

3.2 Implementierungskommentare

Jeder Entwickler soll sich in die Rolle eines Fremden, der den eigenen Quellcode bearbeiten soll, versetzen und dementsprechen nicht-offensichtliche Passagen nach deren Funktionalität kommentieren.

3.3 Deklaration

Es werden die Vorgaben aus 1.1 streng umgesetzt.

3.4 Entwurfsdokumentation

Vor Beginn der Implementierungsphase wird vom gesamten Team eine schematische Übersicht des Projekts erstellt.

3.5 Testkonzept

Jeder Klasse werden TestCases hinzugefügt, um wichtige Funktionalität überprüfen zu können.

4 Verantwortlichkeiten

Quellcodes: Sebastian Kruse

Entwurfsdokumentation: Gesamtes Team

Test: Onon Ganzorig