Dokumentationkonzept

Gruppe SK-1

Projektleiter: Rudi Kallenberg

Einleitung

Zur unserem Dokumentationskonzept gehören die folgenden Komponenten:
- Entwurfbeschreibung (siehe externes Dokument)
- Quelltext und Kommentierung
- JavaDoc

Damit wird sicher gestellt, das die Einarbeitung von Außenstehenden in das System möglich einfach ist. Somit muss sich das komplette Entwicklungsteam an Regeln halten, die in unserem Dokumentationskonzept vorgestellt werden.

Quelltext und Kommentierung („Lieber 10 Kommentare zuviel, als 1 Kommentar zu wenig“)

Dazu gehören 2 miteinander verknüpfte Arten der Dokumentation im Quelltext. Einerseits die Kommentare direkt im Quelltext, welche der Verständigung und zum nachvollziehen der Algorithmik der Implementierer dienen.
Andererseits die indirekte Kommentierung durch „sprechende Variablen und dem Einhalten der Regeln für guten Code", und dessen gute, übersichtliche Strukturierung.
Bei der Kommentierung im Quelltext hält sich unser Team einerseits an die Code Conventions von Sun (http://java.sun.com/docs/codeconv/), und andererseits primär an die „allgemeinen Prinzipien“ von Balzert:

Verbalisierung
- Aussagekräftige, mnemonische Namensgebung
- Geeignete Kommentare
- Selbstdokumentierende Programmiersprache
- Funktion bzw. Aufgabe zum Ausdruck bringen, durch geeignete Namenswahl
- Kurze Bezeichner nicht aussagekräftig und wegen geringer Redundanz, leicht Tippfehler
- Erhöhte Schreibaufwand durch lange Bezeichner wird durch die Vorteile mehr als ausgeglichen.
- Den else-Teil jeder Auswahl kommentieren (... else //A >= 5 { ...} ...)
- Kurzkommentare vermeiden
- Information besser in Namen unterbringen: Lagermenge = Lagermenge + 1;
- Wichtige Kommentare: In Kommentarkasten

--> Leichte Einarbeitung in fremde Programme bzw. Wiedereinarbeitung in eigene Programme
--> erleichtert »code reviews«, Modifikationen und Wartung, verbesserte Lesbarkeit der Programme

Datentypen
- Daten- und Kontrollstrukturen eines Problems sollen sich in der programmiersprachlichen Lösung möglichst unverfälscht widerspiegeln.
- Aufgabe des Programmierers: Angebot an Konzepten einer Programmiersprache optimal zur problemnahen Lösungsformulierung verwenden.
- Typen von Attributen durch elementare Klassen modellieren, gilt auch für Aufzählungstypen

--> Verständliche, leicht lesbare, selbstdokumentierende und wartbare Programm
--> statische und dynamische Typprüfungen verbessern die Qualität des jeweiligen Programms
--> die Daten des Problems werden 1:1 in Datentypen des Programms abgebildet, d. h. Wertebereiche werden weder über- noch unterspezifiziert.

Verfeinerung
- Dient dazu, ein Programm durch Abstraktionsebenen zu strukturieren.
- Verfeinerungsstruktur kann auf 2 Arten im Quellprogramm sichtbar gemacht werden.
- Die oberste Verfeinerungsebene – bestehend aus abstrakten Daten und abstrakten Anweisungen – ist kompakt beschrieben.
- Die Realisierung jeder Verfeinerung wird an anderer Stelle beschrieben. Alle Verfeinerungsebenen sind substituiert
- Die übergeordneten Verfeinerungen werden als Kommentare gekennzeichnet.

--> Entwicklungsprozess im Quellprogramm dokumentiert
--> leichtere und schnellere Einarbeitung in ein Programm
--> Details können zunächst übergangen werden
--> Entwicklungsentscheidungen können besser nachvollzogen werden
--> Obere Verfeinerungsschichten können in natürlicher Sprache formuliert werden.
--> Ein Programm wird zweidimensional strukturiert: sowohl durch Kontrollstrukturen als auch durch Verfeinerungsschichten.

Prinzip der integrierten Dokumentation
- Bei Nachdokumentation am Ende der Codeerstellung sind wichtige Informationen, die während der Entwicklung angefallen sind, oft nicht mehr vorhanden.
- Entwicklungsentscheidungen (z. B.: Warum wurde welche Alternative gewählt?) müssen dokumentiert werden, um bei Modifikationen und Neuentwicklungen bereits gemachte Erfahrungen auswerten zu können.
- Aufwand für die Dokumentation wird reduziert, wenn zu dem Zeitpunkt, an dem die Information anfällt von demjenigen, der sie erzeugt oder verarbeitet, auch dokumentiert wird.

--> reduziert den Aufwand zur Dokumentenerstellung
--> stellt sicher, dass keine Informationen verloren gehen
--> garantiert die rechtzeitige Verfügbarkeit der Dokumentation
--> erfordert die entwicklungsbegleitende Dokumentation
--> leichte Einarbeitung in ein Produkt bei Personalwechsel oder durch neue Mitarbeiter;
--> gute Wartbarkeit des Produkts.

JavaDoc

Javadoc ist ein Programm, das aus Java-Quelltexten Dokumentationen im HTML-Format erstellt, die ausführliche Informationen über alle Klassen, Methoden und Attribute enthalten. Es verwendet dafür die öffentlichen Klassen-, Interface- und Methodendeklarationen und fügt zusätzliche Informationen aus eventuell vorhandenen Dokumentationskommentaren hinzu.
Darüber hinaus wird noch eine Übersicht über alle Klassen und Pakete, ein Vererbungsbaum erstellt.
Da bereits bewährte Systeme zur Dokumentation mit Hilfe von javadoc existieren besteht kein Grund neue Richtlinien zu entwerfen.