Softwaretechnik-Praktikum SS 2007

GR-07-2
Projektleiter: Michael Hütel , Autor:NK, JN, DS

Dokumentationskonzept

1.Einleitung

Eine umfassende Dokumentation ist ein wichtiger Bestandteil für den Erfolg eines Softwareprojektes. Dazu ist es notwendig sich innerhalb des Teams auf zuvor festgelegte Standards für Codekonventionen und Dokumentation zu einigen und diese einzuhalten(Qualitätssicherungskriterien). Dabei ist sowohl die technische als auch die Benutzerdokumentation von großer Bedeutung, da letzteres für den Endanwender vorgesehen ist und es neuen Programmierern leichter fällt sich in ein gut dokumentiertes Softwareprojekt einzuarbeiten. Die Dokumentation ist parallel zur Implementierung zu erstellen, da ein nachträgliches erarbeiten und ändern der Dokumentationskonzepte mit großen zeitlichen und finanziellen Aufwand verbunden ist und Informationen zu Prozessen verloren gehen könnten. Daher ist auf die adäquate Dokumentation von jedem Teammitglied selbst zu achten, um einen möglichst hohen Grad an Wiederverwendbarkeit und Änderbarkeit zu garantieren.

2.Interne Dokumentation

Eine akkurate Dokumentation des Quellcodes ist die Basis für eine effektive Arbeit an demselben. Daher ist es notwendig, Konventionen zur Formatierung und Dokumentation des Quellcodes zu vereinbaren.

Diese Code-Konventionen sind folgendermaßen:

- eine Codezeile sollte nicht länger als 80 Zeichen sein. Dies erhöht die Lesbarkeit des Codes, vermeidet Inkompatibilitätsprobleme mit diversen Tools und macht den Code leserlich druckbar.
- Nach Kommata oder vor Operator-Zeichen kann ggf. ein Zeilenumbruch stehen.
- Nach einem Umbruch sollte der restliche Ausdruck so eingerückt werden, dass er direkt unter dem Anfang des Ausdrucks unter der Zeile darüber steht.
- Lange Strings können via + als Konkatenation über mehrere Zeilen verteilt werden. Jede Zeile eines Texts sollte in der selben spalte beginnen.
- Jede Variablendeklarationen sollte in einer separaten Zeile stehen, um dahinter jeweils Kommentare einfügen zu können.
- Variablennamen sollten klein geschrieben werden. Klassennamen groß. Konstanten sollten in Großbuchstaben geschrieben werden.
- Bei zusammengesetzten Bezeichnern, sollten neue Wörter groß beginnen. Zum Beispiel `anzahlStudenten` statt `anzahlstudenten`.
- Bezeichner sollten stets deskriptiv sein. Ihr Inhalt oder Zweck sollte aus dem Namen ersichtlich werden.
- Ein Bezeichner kann aus den Buchstaben von a bis z und zusätzlich aus Ziffern von 0 bis 9 bestehen, muss aber mit einem Buchstaben beginnen
- Generell dürfen keine Sonderzeichen verwendet werden
- Anweisungsblöcke sollten in einer neuen Zeile beginnen. Ihr Inhalt sollte mit 4 Leerzeichen sauber eingerückt sein. Bleibt aufgrund der Einrückung zu wenig Platz um noch übersichtlichen Code zu schrieben, sollte über das Auslagern von Funktionalität in eine neue Methode nachgedacht werden. Sonst kann man die Einrückung auf 2 Leerzeichen reduzieren.
- Unnötig komplexer/komplizierter Code (z.B. Fakultätsberechnung in nur einer Codezeile) ist zu vermeiden. (Solche Minimierungen werden vom Compiler erledigt.)

Softwaretechnik-Praktikum SS 2007

GR-07-2
Projektleiter: Michael Hütel , Autor:NK, JN, DS

Konventionen zum Kommentieren von Code:

Maßgebend für Kommentare ist die Spezifikation für **Javadoc.**

- Der Code muss jederzeit von jedem Programmierer ohne Hilfe des Autors nachvollzogen werden können. Der Autor ist für die Dokumentation im Code verantwortlich.
- Vor jeder Klasse und Methode muss ein Kommentar stehen, der die Funktionen treffend beschreibt und Eingabe/Ausgabewerte-Werte von Methoden erklärt.
- Diese sind zudem mit den Javadoc-Kommentarzeilen
  - `* @param parametername Erklärung`
  - `* @return variablenname Erklärung`
  - `* @see Verweis auf andere Klassen oder Methoden(meistens automatisch erzeugt)`
- Damit Kommentare vom Javadoc-Tool erfasst werden können, müssen sie mit /** beginnen.
- Jede weitere Zeile ist mit einem führenden, eingerücktem * einzuleiten
- Zudem muss bei Klassen autor und Version (als Datum der letzten Änderung) angegeben sein
- auf korrekte Klammerung ist zu achten
- Geschweifte Klammern werden immer in die entsprechende Zeile, also hinter die einen Klassen- oder Methodennamen, geschrieben.
-
- Beispiel:

```
/** Kodierer
* Der Kodierer verschlüsselt den übergebenen Textstring.
* @author Tim Taler
* @version 31.4.1592 6.53 Uhr
*/
```

  Unverständliche Codefragmente sind per `//...`
  oder in einer neuen Zeile darüber/darunter per
  `/* ... */`
  zu kommentieren mehrzeilige Kommentare haben die Form:

```
/*
*
.
.
*/
```

- Für Testdokumentation siehe Testkonzept

3.Externe Dokuementation

Designdokumentation

die Designdokumentation ist hauptsächlich vom Verantwortlichen für Modellierung zu erstellen. Unterstützt wird er vom Verantwortlichen für Recherche. Beide können Aufgaben delegieren. Als Ergebnis muss ein Modell bereitstehen, dass dem Team einen Überblick über die Architektur der Software bietet und über die Funktion deren Teile informiert.

Benutzer-Dokumentation
Für die Benutzer sind Dokumente zu erstellen, die ausführlich und leicht verständlich sind.
Es wird neben einem Benutzerhandbuch auch ein Online Hilfesystem angeboten.
Beides steht dem Benutzer zur Verfügung und soll alle Funktionalitäten verständlich und

umfassend beschreiben. Es werden Informationen über Systemvoraussetzungen, Installation und Handhabung des Produktes beschrieben. In einem Glossar sollen programmrelevante Begriffe erklärt werden. Auch häufig gestellte Fragen (FAQ) sollten im Handbuch eingebunden sein.
Da das OLAT ein Open-Source-Projekt ist, sind auch andere Programmierer Nutzer. Daher ist nicht nur ein Handbuch nötig, das die Benutzung der Funktionen aus der Sicht verschiedener Akteure wie Administratoren, Dozenten, Studenten beschreibt, sondern auch eine genaue Dokumentation des Quellcodes, insbesondere der Schnittstellen der Software.

Verantwortlich sind hierfür der Verantwortliche für Dokumentation. Die Dokumentation der Schnittstellen und des Codes ist zusätzlich die Aufgabe der Implementierer selbst.