

Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

Dokumentationskonzept

Damit Programme gut lesbar sind, müssen sie durch den Einsatz von Kommentaren geeignet dokumentiert und optisch strukturiert werden. Alle Programmierer haben sich u.a. an die Regeln für guten Java-Code nach Balzert zu halten. Die Vereinbarungen im Folgenden.

Richtlinien zur Versionskennzeichnung:

1. Jedes Dokument ist mit einer Versionsnummer zu kennzeichnen.
2. Die Versionsnummer besteht aus einem „V“ gefolgt von der Release-Nummer, die durch einen Punkt von der Level-Nummer getrennt wird. Die Level-Nr. wird erhöht, wenn eine kleine Änderung stattfindet, die Release-Nr. bei größeren Änderungen oder Erweiterungen, wobei gleichzeitig die Level-Nr. genullt wird. (z.B.: V 1.2)

Tests:

1. Für jede Klasse sind Testklassen bzw. -funktionen mit Beschreibung anzulegen.
2. Test- und Produktionscode sollen abwechselnd entwickelt werden, sofern sinnvoll.
3. Nach Implementierung der Klassen, ist diese mit JUnit zu überprüfen.
4. Die Klasse wird erst in das Gesamtsystem integriert, wenn die Tests erfolgreich waren.

Bezeichner-Richtlinien:

1. Bezeichner bestehen aus problemnahen Namen bzw. verständlichen Abkürzungen.
2. Jeder Bezeichner beginnt mit einem Buchstaben und wird Leerzeichenlos geschrieben.
3. Es wird englische Namensgebung verwendet.
4. Besteht ein Bezeichner aus mehreren Worten, dann beginnt jedes Wort mit einem Großbuchstaben. (z.B.: *numberOfWords*)
5. Klassennamen/Schnittstellennamen...
 - a) beginnen immer mit einem Großbuchstaben.
 - b) bestehen aus einem Substantiv, oder einem Substantiv mit Adjektiv
6. Objektnamen...
 - a) beginnen immer mit einem Kleinbuchstaben.
 - b) von GUI-Interaktionselementen beginnen klein geschrieben mit zugeordneten Attributnamen gefolgt von ihrer Gui-Klass (z.B.: *saveButton*)
7. Attributnamen...
 - a) beginnen immer mit einem Kleinbuchstaben
 - b) als Konstanten sind komplett groß zu schreiben und durch Unterstriche zu trennen
8. Operationsnamen...
 - a) beginnen immer mit einem kleinen Verb, evtl. gefolgt von einem Substantiv
 - b) heißen *getAttributeName*, wenn nur ein Attributwert eines Objektes gelesen wird.
 - c) heißen *setAttributeName*, wenn nur ein Attributwert eines Objektes gespeichert wird.
 - d) heißen *isAttributeName*, wenn das Ergebnis nur true oder false sein kann
9. Pakete...
 - a) sind mit dem Präfix „*de.uni_leipzig.swtp.tr06_2*“ zu versehen.
 - b) Paketnamen sind komplett klein zu schreiben.

Softwaretechnik-Praktikum SS 2006

tr-06-2

Projektleiter: Marcus Lechner

Formatierungs-Richtlinien:

1. Leerzeichen
 - a) Bei binären Operatoren werden Operanden und Operator durch jeweils ein Leerzeichen getrennt. (z.B.: $2 + 4 * 2$)
 - b) Keine Leerzeichen bei der Punktnotation: *Objekt.Operation*
 - c) Zwischen Operationsname und Klammer und nach der öffnenden und vor der schließenden Klammer steht kein Leerzeichen (z.B.: *setColor(Color.blue)*)
 - d) Nach Schlüsselwörtern steht immer ein Leerzeichen
2. Einrücken und Klammern von Strukturen
 - a) Paarweise geschweifte Klammern stehen immer in derselben Spalte untereinander.
 - b) In der Zeile, in der eine geschweifte Klammer steht, steht sonst nichts mehr.
 - c) Alle Zeilen innerhalb eines Klammerpaares sind jeweils um 4 Zeichen eingerückt.
 - d) Tabulatoren sind nicht zu verwenden.
3. Längenbegrenzungen
 - a) Methoden sind in der Regel auf 10 bis 20 Zeilen zu begrenzen (ohne Kommentare).
 - b) Klassen sind als kleinstmögliche zusammengehörige Einheit zu entwickeln.

Java-Dokumentations-Richtlinien:

1. Ein Programm hat Kurzbeschreibung und Verwaltungsinformationen zu beinhalten.
2. Javadoc-Kommentare (*/**Kommentar*/*) und Formatanweisungen sind zu verwenden.
3. Die Kurzbeschreibung und die Verwaltungsinformationen werden in einem Programmvorspann zusammengefasst:
 - a) Programmname: möglichst genauer Name.
 - b) Aufgabe: Kurze Beschreibung des Programms
 - c) Zeit- und Speicherkomplexität des Programms
 - d) *@author* Name des Autors
 - e) *@version* Versionsnummer und Datum
4. Auch der Quellcode selbst muss dokumentiert werden. Die Aufgabenbeschreibung jeder Operation und die Bedeutung der Parameter müssen kommentiert werden. Dabei sind folgende Parameter einzuarbeiten:
 - @param* Name und Beschreibung der Parameter
 - @return* Beschreibung des Rückgabewertes
 - @exception* Name und Beschreibung von Ausnahmen
 - @see* Verweis auf Klasse

Verantwortlichkeiten der Quellcodedokumentation und Test:

Bereits beim Entwerfen des Codes ist vom Autor auf gute Dokumentation und Kommentierung zu achten. Der Verantwortliche für Dokumentation und Qualitätssicherung wird die Einhaltung der hier festgehaltenen Qualitätsstandards und die Lesbarkeit der Dokumentation regelmäßig überprüfen.