

Dokumentationskonzept

Geo01

Inhaltsverzeichnis

Dokumentationskonzept	2
Zum Programm	2
Zu Rational Rose	2
Zum Quellcode	2
Zur Hilfe	3

GEO01 - SOFTWARE SOLUTIONS

04109 Leipzig • Augustusplatz 10
mail: softwaresolutions@everyday.com

Erstellt von: Madlen Hartmann, Dominic Rose



Dokumentationskonzept

Das Dokumentationskonzept definiert Richtlinien zur Dokumentationserstellung unseres Programms. Allen Mitarbeitern sollen möglichst ausführliche Informationen über unser Programm zur Verfügung stehen, um effizientes Arbeiten zu ermöglichen. Diese Dokumente erleichtern es auch dem Auftraggeber das Programm zu benutzen und zu verstehen.

Zum Programm

Der wesentliche Inhalt und Aufbau des Programms wird in der Designbeschreibung festgehalten. Während der Entwicklung ist eine Anwender- und Installationsdokumentation schrittweise mit wachsendem Funktionsumfang des Programms zu erstellen. Die Anwenderdokumentation ermöglicht den direkten Einstieg in das Programm und erleichtert ungeübten Benutzern die Handhabung des Programms. Die Installationsdokumentation beschreibt die Systemvoraussetzungen sowie nötige Schritte um das Programm starten zu können. Der Quellcode selbst dient auf unterster Ebene der Dokumentation.

Zu Rational Rose

Mit Rational Rose erzeugte Diagramme (egal ob UML, Use-Case etc.) sind ausreichend und sinnvoll zu kommentieren, damit bereits geleistete Arbeitsschritte von nicht direkt beteiligten Personen verstanden werden können und spätere Erweiterungen für jeden möglich und nachvollziehbar sind.

Hierzu besteht zum einen die Möglichkeit Notizen in Rational Rose selbst einzufügen, zum anderen, speziell bei Klassendiagrammen, ist das `javadoc` Format zu nutzen. Dies hat den Vorteil, dass all diese Kommentare im später zu generierenden Code-Rahmen miteingebunden werden. Bei besonders schwierigen Sachverhalten ist es durchaus sinnvoll auch externe Dokumentationsdokumente zu verfassen.

Zum Quellcode

Quellcode ist in sinnvoller Weise mit Blockkommentaren und Inline-Kommentaren zu versehen. Diese sind wie folgt aufgebaut:

```
/* bei mehrzeiligen Kommentaren */  
// bei einzeiligen Kommentaren
```

Man orientiere sich hierbei an den im [Balzert] angegebenen Programmierrichtlinien.

Außerdem ist das `javadoc` Format zu benutzen.

Der Rahmen eines solchen Kommentars sieht folgendermaßen aus:

```
/**  
 * Hier steh der Kommentar  
 */
```

Kommentare müssen jeweils vor jeder Klasse und jeder Methode sowie vor jedem Attribut und Konstruktor stehen.

Zum Kommentar gehören generell (in folgender Reihenfolge):

- eine detaillierte Beschreibung der Funktion des folgenden Codes
- der Autor (@autor), wobei mehrere Autoren durch Kommas getrennt werden

GEO01 - SOFTWARE SOLUTIONS

04109 Leipzig • Augustusplatz 10
mail: softwaresolutions@everyday.com

Erstellt von: Madlen Hartmann, Dominic Rose



- evtl. die Versionsnummer (@version)

Zum Kommentar einer Methode gehören außerdem (in folgender Reihenfolge):

- die Parameter (@param), wobei zuerst der Name, dann die Beschreibung des Parameters folgt
- der Rückgabewert (@return)
- die Exceptions (@throw), wobei der Name und dann eine kurze Beschreibung der Exception folgt

Zur Hilfe

Während der Benutzung des Programms wird es später möglich sein eine Hilfsfunktion aufzurufen. Diese wird nicht nach Beendigung des Programms geschrieben, sondern parallel zum programmieren entwickelt und ergänzt.

Sobald also im Iterationsprozess der Programmentwicklung eine Funktion integriert wurde, so sollte in einem separaten Dokument festgehalten werden, wie man diese Funktion nutzen kann. Diese Dokumente sind dann Grundlage der Hilfe. Die Hilfe selbst wird ebenfalls entsprechend den Zeitscheiben implementiert.

Außerdem wird (nicht im Programm, sondern als separates Dokument) eine Installationshilfe mitgeliefert, in dieser sind die Systemvoraussetzungen sowie nötige Schritte um das Programm starten zu können, beschrieben.