

## Praktikumsanleitung – Versuch 2

### Software-Pflege, Wiederverwendung, Nachdokumentation

Praktikumstermine: Mo, 05.04.2004 / Di, 06.04.2004 / Do, 08.04.2004

#### Einführung Teil 1

Eine Steigerung der Produktivität bei der Entwicklung von Software ist nur in beschränktem Masse möglich, weil wesentliche Teile des Prozesses kreative, nicht automatisierbare Arbeiten sind. Wenn möglich werden aus diesem Grund Software-Komponenten nicht neu entwickelt, sondern beschafft.

Die Packages “remote”, “sopra” und “util” des Systems “Börsencafé” wurde von auswärtigen Entwicklern bereits erstellt. Sie werden diese Komponenten später in Ihrem System verwenden müssen.

Immer wieder kommt es vor, dass der Quellcode die einzige Dokumentation einer Software ist. Solch spärlich kommentierter Quellcode ohne jegliche weitere Dokumentation, ist äusserst schwer zu verstehen und zu pflegen. Bevor solcher Code systematisch geändert werden kann, z.B. um Fehler zu korrigieren oder Erweiterungen vorzunehmen, ist es notwendig, bestimmte Dokumente nachzuführen, die es ermöglichen, zu verstehen, was die betreffende Software tut und wie sie arbeitet. Entwurfsdokumente und auch eine ausreichende Kommentierung des Quellcodes müssen nachträglich erstellt werden, wobei zur Erstellung dieser Dokumente lediglich der Quellcode zur Verfügung steht.

Inhalt dieses Versuches ist es, eine solche Nachdokumentation exemplarisch durchzuführen.

Gegenstand der Nachdokumentation sind die Packages “remote”, “sopra” und “util” des Börsencafé-Projekts. Gegeben ist jeweils der Quellcode. Der vorgegebene Code ist ausführbar. Für die Packages existieren weder Spezifikation noch Entwurf, auch der Quellcode ist nur äusserst unvollständig kommentiert. Sie sollen den vorgegebenen Quellcode nachdokumentieren.

#### Vorbereitung des Versuchs

Um diesen Versuch in der zur Verfügung stehenden Zeit durchführen zu können, ist es notwendig, dass Sie sich *vor* Versuchsbeginn mit der Programmiersprache Java [Camp96,

Flan96, Gosl96] und den Java-Entwicklungsrichtlinien [Bern98] vertraut machen. Sie müssen die Programmiersprache Java keineswegs perfekt beherrschen, aber sie sollten soweit mit den wichtigsten Sprachelementen vertraut sein, dass sie den Quellcode der Software lesen und verstehen können. Hinweis: Unbekannte Systemklassen können in der Java API Dokumentation [JAPI97] nachgeschlagen werden.

## **Durchführung des Versuchs Teil 1**

### **Aufgabe 2.1: Verstehen des Quellcodes**

- a) Stellen Sie fest, aus welchen Klassen die nachzudokumentierende Packages besteht und wie die Klassen zusammenhängen. Wie sind die Benutzt-Beziehungen? Wie ist die Vererbungshierarchie?
- b) Ermitteln Sie die Funktionalität der Komponente und beschreiben Sie diese in Form einer Übersicht. Verwenden Sie als Grundlage das Ergebnis von Teil a) dieser Aufgabe. Ordnen Sie die Funktionalität gezielt den einzelnen Klassen zu. Die Beschreibung soll insgesamt max. 2 A4 Seite Text umfassen.

### **Aufgabe 2.2: Dokumentieren von Programmcode**

Dokumentieren Sie die Packages „remote“, „sopra“ und „util“ nach den Vorgaben der Java-Entwicklungsrichtlinien [Bern98], d.h. erstellen Sie den Klassenkommentar, die Kommentare für Klassen- und Instanzvariablen sowie die Kopf- und Rumpfkomentar für die Methoden der Klasse.

### **Aufgabe 2.3: Darstellen der logischen Struktur**

- a) Stellen sie die logische Struktur der Klassen des Packages “remote” mittels eines Klassendiagramms dar.
- b) Fügen Sie die Ergebnisse der Teilaufgaben 2.1 b) und 2.3 a) so zusammen, dass sie das Kapitel “Übersicht“ in einem nachträglich zu erzeugenden Entwurfsdokument ergeben.

Denken Sie daran, dass Sie die vorgegebenen Klassen in Ihrem System verwenden müssen. Kommentieren Sie so, dass Ihnen die Kommentare auch Ende des Semesters noch verständlich sind.

## **Einführung Teil 2**

Die Validierung von Spezifikationen muss in Zusammenarbeit mit dem Kunden geschehen. Es soll sichergestellt werden, dass die erhobenen Anforderungen den Bedürfnissen des Kunden

entsprechen. Zu diesem Zweck steht Ihnen der Kunde (Wirt) im Verlauf dieses Versuchs für ein ungefähr 35 minütiges Gespräch zur Verfügung. Ziel dieses Versuchs ist es, die in Versuch 1 erstellten Anwendungsfälle zu validieren und allfällige Unklarheiten zu beseitigen.

## **Durchführung des Versuchs Teil 2**

### **Aufgabe 2.4: Diskussion mit dem Kunden**

Diskutieren Sie die in Versuch 1 erstellten Anwendungsfälle mit dem Wirt. Nutzen Sie die Zeit, um die bestehenden Anwendungsfälle zu validieren und allfällige bisher vergessene Szenarien zu diskutieren. Führen Sie diese Diskussion sorgfältig und werten Sie die Ergebnisse aus.

Je besser Ihre Anwendungsfälle den Vorstellungen des Auftraggebers entsprechen, desto weniger Probleme haben Sie in den nächsten Versuchen.

### **Aufgabe 2.5: Verbesserung der Anwendungsfälle**

Modifizieren und ergänzen Sie auf Grund des Gesprächs ihr Pflichtenheft aus Versuch 1.

## **Abgabe**

- Gemäss Styleguide [Bern98] kommentierter Code der Packages “remote”, “sopra” und “util”
- Ein sinnvoll gegliedertes Dokument (.pdf oder .doc) enthaltend:
  - Zusammenfassung der Funktionalität der Komponenten (max. 2 A4 Seiten)
  - Klassendiagramm des Packages “remote”
- Korrigiertes und ergänztes Pflichtenheft aus Versuch 1

Die Abgabe erfolgt im OLAT Gruppenordner in einem von Ihnen anzulegenden Verzeichnis mit der Bezeichnung „Versuch02\_Lsg“.

## **Abgabetermin**

- Klasse Montag: Do, 08.04.04, 24:00
- Klasse Dienstag: Fr, 09.04.04, 24:00
- Klasse Donnerstag: So, 11.04.04, 24:00

## **Unterlagen in OLAT:**

- Quellcode für die Packages “remote”, “sopra” und “util” (Datei vorgabe.zip)
- Intro Together
- Intro Threads

- Intro Observer-Pattern
- Intro UML
- Eclipse Starthilfe
- [Bern98] S. Berner, M. Glinz, S. Joos, N. Schett (1998): *Entwicklungsrichtlinien für Java-Software – Version 3.0.1* Institut für Informatik der Universität Zürich, 1998.

**Referenzen:**

- [Camp96] M. Campione, K. Walrath: *The Java Tutorial: Object-Oriented Programming for the Internet*. Java Series. N.Y: Addison Wesley Inc., 1996. (-> HBI)
- [Flan96] D. Flanagan: *Java in a Nutshell - a desktop quick reference for Java*. Sebastopol CA: O'Reilly & Associates, 1996. (-> IFI Bibliothek)
- [Früh95] K. Frühauf, J. Ludewig, H. Sandmayr: *Software Prüfung*. Zürich: vdf, Hochschulverl. an der ETH Zürich. 1995. (-> IFI Bibliothek)
- [Glin98] M. Glinz: Vorlesungsskript WS 98/99 , *Software Engineering I*. Institut für Informatik der Universität Zürich.
- [Gosl96] J. Gosling, B. Joy, G. Steele: *The Java Language Specification*. Java Series. N.Y: Addison Wesley Inc., 1996. (-> IFI Bibliothek)
- [JAPI97] J. Gosling, F. Yellin, and the Java Team (1997): *Application Programming Interface*. Java Series. N.Y: Addison Wesley Inc., 1997. (-> Java API Dokumentation ist als HTML auch auf den Workstations)
- [Sedg92] Sedgewick, R. (1992): *Algorithmen*. Bonn; München; Reading Mass [u.a.]: Addison Wesley, 1992. (-> IFI Bibliothek)