

## 1. Übung zu Softwaretechnik 2

### Aspektororientierte Programmierung

#### Theorie

Machen sie sich anhand der Vorlesungsfolien und der Quickreference aus dem Aspectj Programmer's Guide mit den Ansatzpunkten (Join Points, PointCuts, Advice, Aspect) aspektororientierter Programmierung vertraut.

(<http://aspectj.org/doc/dist/progguide/apa.html>)

#### Praxis

Entwerfen Sie Aspekte, die

- das Ergebnis von `java.lang.math.sqrt(double a)` überprüfen. (after advice)
- den Aufruf von `java.util.Vector.setSize(int newSize)` überprüft, ob er die momentane `size` überschreitet, und falls das der Fall ist, das letzte Element entsprechend oft einfügt. (siehe API-Beschreibung)
- alle Aufrufe von Funktionen aus dem `java.util`-Paket protokolliert.
- `java.util.Vector` um eine Sortierfunktion erweitert (gewöhnlich sortiert man einen `Vector`, indem man ihn in ein `Array` umwandelt und dessen Sortierfunktion nutzt)

Probieren Sie Ihre Aspekte mit `aspectj` aus. (optional)

#### Entwurf

Beim Entwurf einer Personalverwaltung für die Uni wollen Sie aspektororientierte Programmierung verwenden.

Der Entwurf geht von einem Programm für die Personalverwaltung aus, das

- Adressen erfasst und verwaltet
- Studenten, Mitarbeiter und Professoren unterscheidet
- Hiwis als Subspezies von Studenten und Angestellten kennt.

Als Kernfunktionalität sollen Personen hinzugefügt und entfernt werden können, Verträge erstellt und beendet, Studenten exmatrikuliert und rückgemeldet werden und zusätzlich noch Gehaltshöhe und Zahlungen verwaltet werden.

Diese grundlegende Funktionalität soll jetzt in verschiedene Richtungen erweitert werden. Entwerfen sie jeweils stichwortartig eine Erweiterung, um

- Leistungsabhängige Bezahlung von Professoren
- Eine Warnfunktion, die bei jeder Gehaltszahlung vor auslaufenden Verträgen warnt
- Konfigurationselemente aus Gehalts-, Personal- und Studienverwaltung in einem Fenster für persönliche Einstellungen anzeigt

- Parallele Verarbeitung auf einem zentralen Server zulässt, indem Lockingfunktionen auf den einzelnen Objekten installiert werden.

Die Personalobjekte sollen zur Verwaltung der "Richtlinie Bildschirmarbeitsplätze" durch den Betriebsarzt wiederverwendet werden. Dabei muss jeder Mitarbeiter und Professor alle 3 Jahre zur Augenärztlichen Untersuchung, falls er an einem Bildschirmarbeitsplatz arbeitet. Entwerfen sie Aspekte, die diese Anwendung in das bestehende Personalverwaltungssystem einfügt.

## **Diskussion**

Die Architekturmuster haben sich im Entwurf großer Systeme bewährt. So eignen sich beispielsweise

**Schichtenmodelle** zur Beschreibung für Netzwerk-Kommunikationsstrukturen,

**Pipelines** für Compiler (Lexer - Parser - RTL - Optimizer - Codegeneration)

Wo sehen sie in diesen Mustern oder Anwendungen sinnvolle Ansatzpunkte für die aspektorientierte Programmierung?

—

Bei Fragen wenden sie sich an Holger Cleve [cleve@cs.uni-sb.de](mailto:cleve@cs.uni-sb.de)