

IPD
Universität Karlsruhe
AVG, 2. OG
Adenauerring 20a
76131 Karlsruhe

Prof. Dr. Gerhard Goos

Volker Kuttruff

Zi. 201
ggoos@ipd.info.uni-karlsruhe.de

Zi. 1.1.35 (FZI) Tel. 0721/9654-640
kuttruff@fzi.de

Übungsblatt: 6

Ausgabe: 30.6. 2004

Besprechung: 7.7. 2004

Aufgabe: 6.1 (Beispiele für Kompositionsfilter)

Diskutieren Sie, wie die folgenden Aspekte mit Kompositionsfiltern umgesetzt werden können:

- Protokollierung (Logging)
- Authorisierung
- Caching (von Aufrufergebnissen)
- Synchronisation
- Datenanpassung
- Protokollprüfung

Diskutieren Sie die Eigenschaften der Filter und machen Sie Vorschläge zur Einteilung. Diskutieren Sie mögliche Interaktionen zwischen den Filtern.

Aufgabe: 6.2 (Eine Architektur für Kompositionsfilter)

Wir haben das Paket `java.lang.reflect` bereits erfolgreich eingesetzt, um Fernaufrufe in Java umzusetzen. Derselbe Mechanismus kann auch allgemeine Kompositionsfilter unterstützen.

- Geben sie an, wie Kompositionsfilter mit Reflektion in Java umgesetzt werden können.
- Diskutieren Sie das Problem der echten Delegation.
- Setzen Sie einen Rahmen für Kompositionsfilter um.

Aufgabe: 6.3 (Umsetzung von Kompositionsfiltern)

Setzen Sie folgende Aspekte als Kompositionsfilter in Java um:

- Protokollierung (Logging)
- Authorisierung
- Caching
- Synchronisation

Aufgabe: 6.4 (AspectJ)

In der Vorlesung wurde die Sprache `AspectJ` vorgestellt, die es erlaubt Aspektcode in Java-Klassen einzuweben.

1. Geben Sie eine Implementierung des Beobachtermusters als abstrakten Aspekt in `AspectJ` an.
2. In der Vorlesung wurde als Beispiel einer Aspektimplementierung in `AspectJ` das Aktualisieren einer Anzeige (`Display`) bei der Veränderung geometrischer Objekte vorgeführt. Implementieren Sie diesen Aspekt als Beobachtermuster, indem Sie ihn aus dem abstrakten Aspekt der vorangegangenen Teilaufgabe ableiten.