

Vorlesung Aspektorientierte Softwareentwicklung (AOSD)
– Sommersemester 2008 –

Dr. Günter Kniesel, Daniel Speicher

Übungsblatt 1

Abgabe: Dienstag 29.04.2008 23:59:59 per SVN

Bitte fangt **frühzeitig** mit der Bearbeitung des Zettels an, damit wir euch bei Bedarf helfen können!!!!
„Das hat aber nicht funktioniert“, zählt nicht als Entschuldigung, schließlich habt ihr 2 Wochen Zeit,
auch um Probleme **rechtzeitig** zu melden!!!!!!

(SWT-Vorlesung@iai.uni-bonn.de, SWT-tutoren@iai.uni-bonn.de)

Aufgabe 1 (*Hello World*) (4 Punkte)

Installiere, soweit noch nicht geschehen, Eclipse, SVN und AJDT.

<https://sewiki.iai.uni-bonn.de/teaching/lectures/aosd/2008/svn#aspectj>

Öffne ein neues AspectJ-Projekt namens HelloWorld. Füge dort die folgende Klasse ein:

```
public class HelloWorld {
    public static void main(String[] args){
        System.out.println("Hello world!");
    }
}
```

- Schreibe einen Aspekt, der dem Konsolen-Output "Hello AspectJ!" hinzufügt.
- Welche Pointcut-Advice Kombinationen können das gewünschte Ergebnis produzieren?
- Schreibe den Aspekt noch einmal, dieses Mal unter Verwendung eines anderen Primitive Pointcuts und Advice.
- Schreibe einen Aspekt, der die Ausgabe "Hello world" gegen "Hello <DeinName>" ersetzt.

Aufgabe 2 (*Tic Tac Toe*) (9 Punkte)

Unter

https://svn.iai.uni-bonn.de/repos/IAI_Software/se/aosd2008ss/aufgaben/trunk/B01A2_TicTacToe

könnt Ihr eine Version von *TicTacToe* auschecken.

(Hinweis: Ihr müsst das, aus diesem Verzeichnis, ausgecheckte Projekt **IMMER disconnecten** bevor ihr es in euer Gruppen-SVN eincheckt. Siehe:

<https://sewiki.iai.uni-bonn.de/teaching/lectures/aosd/2008/tutorien#uebungsaufgaben>).

Die vorliegende Implementierung ist rein objektorientiert. Importiere das Projekt in Eclipse und konvertiere es zu einem AspectJ-Projekt.

- Schreibe einen Aspekt, der bei den folgenden Ereignissen eine Konsolenausgabe mit einem Zeitstempel produziert :
 - ein Spieler beginnt seinen Zug
 - ein Spieler führt seinen Zug aus
 - das Spiel wird neu gestartet

- b) Nach jedem Spiel soll eine Statistik in der Konsole erscheinen, die
- den Namen des Gewinners
 - die Anzahl der Spielzüge
 - die Dauer des Spiels
- enthält.
- c) Die Turnier-Variante von *Tic Tac Toe* enthält ein Zeitlimit pro Zug von 10 Sekunden. Implementiere diese Funktionalität in einem weiteren Aspekt, so dass sie bei Bedarf in das Programm eingewoben werden kann.

Aufgabe 3 (*Bank Beispiel aus ‚AspectJ in Action‘*) (8 Punkte)

Unter

https://svn.iai.uni-bonn.de/repos/IAI_Software/se/aosd2008ss/aufgaben/trunk/B01A3_BankAccount könnt Ihr das Programm *BankAccount* auschecken.

- a) Die Funktionalität soll um die Eingabe einer PIN bei den folgenden Aktionen erweitert werden:
- Geld wird abgehoben
 - Geld wird von einem Konto auf ein anderes transferiert.
- Implementiere diese Funktionalität in einem Aspekt, so dass eine Aufforderung zur Eingabe einer PIN in der Konsole erscheint.
- b) Wird die PIN falsch eingegeben, wird die Aktion abgebrochen. Die PIN soll dreimal eingegeben werden können. Erst nach dreimaliger Fehleingabe wird die Aktion abgebrochen.
- c) Innerhalb einer Transaktion soll sich der Benutzer nur einmal authentifizieren müssen. D.h. hat sich der User bereits zu Beginn einer Transaktion authentifiziert, soll die Aufforderung nicht ein weiteres Mal bei dem Abheben des Geldes innerhalb der Transaktion erscheinen.

Aufgabe 4 (*cflow*) (5,5 Punkte)

Welche der folgenden Aussagen über *cflow* Pointcuts sind wahr? Finde anderenfalls ein Gegenbeispiel.

- a) `cflow(cflow(ptc))` == `cflow(ptc)`
- b) `cflow(ptc1 || ptc2)` == `cflow(ptc1) || cflow(ptc2)`
- c) `cflow(ptc1 && ptc2)` == `cflow(ptc1) && cflow(ptc2)`
- d) `cflow(cflow(ptc1) || ptc2)` == `cflow(ptc1) || cflow(ptc2)`
- e) `cflow(cflow(ptc1) && ptc2)` == `cflow(ptc1) && cflow(ptc2)`
- f) `cflowbelow(cflow(ptc))` == `cflow(ptc)`
- g) `cflowbelow(cflow(ptc))` == `cflowbelow(ptc)`
- h) `cflow(cflowbelow(ptc))` == `cflow(ptc)`
- i) `cflow(cflowbelow(ptc))` == `cflowbelow(ptc)`
- j) `!cflow(!cflowbelow(ptc1))`
 == `!cflow(ptc2) && (cflowbelow(ptc2) || ptc2)`
- k) `(cflow(ptc1) && !cflow(ptc2)) || (!cflow(ptc1) && flow(ptc2))`
 == `(cflow(ptc1) || cflow(ptc2)) && !(cflow(ptc1) && cflow(ptc2))`