

AspectJ-Umgebung, Dokumentation, Logging (bis zum 29.10.)**Notenkriterien**

- **Klausur:** Di 29. Jan. 2008 , 8-10h oder 10-12h (angefragt) ohne Unterlagen!
- **Übung:** Jede 2. Woche eine Aufgabe, Gutschrift von Klausurpunkten bei fristgerechter Abgabe (alle=10%). Verhinderungen werden für die Gutschrift nicht anerkannt! Abgabe einzeln oder in fester 2-er-Gruppe während der Übungszeit, Anwesenheitspflicht beider Partner bei der Abgabe, bei (Stil-) Mängeln 1 Woche Nachfrist für Überarbeitung, späteste Abgabemöglichkeit Di 29. Jan. 2008.

Aufgabe 1: Installation und Signatur-Logging

- Im SWE-Labor ist Eclipse 3.3.0 mit dem Feature AJDT 1.5.0 installiert.
Um zu Hause dieselben Versionen zu installieren, benötigen Sie Datei eclipse-java-europa-3.3.0-win32.zip, die Sie von mir beziehen können, da sie unter <http://www.eclipse.org/downloads/> nicht mehr zu finden ist.
Diese entpacken Sie in ein Verzeichnis und rufen eclipse.exe auf. Dann installieren Sie AJDT über Eclipse-Menüpunkt Help > Software Updates > Find and Install ... > Search for new features to install > Europa Discovery Site > Finish. Wählen Sie einen deutschen Mirror aus. Wählen Sie Europa Discovery Site > Java Development aus. Darin wählen Sie Eclipse Plug-in Development Environment und AspectJ Development Tools (AJDT) aus.
- Erzeugen Sie sich im Startmenü eine Verknüpfung, die auf die Datei index.html im Verzeichnis *Eclipse-Installationsverzeichnis\plugins\org.aspectj.ajdt_1.5.4.zeitstempel\doc* zeigt, so dass Sie die AspectJ-Dokumentation immer griffbereit haben. Drucken Sie ein Bildschirmfoto vom Explorer, der den Inhalt dieses Verzeichnisses anzeigt.
- Legen Sie sich auch im Labor eine Verknüpfung auf die AspectJ-Doku an.
- Legen Sie im Navigator ein neues AspectJ-Projekt in Eclipse an.
In dieses importieren Sie das Ihnen zur Verfügung gestellte Paket telecom und führen die Hauptklasse BasicSimulation aus. Es erscheint ein Protokoll, wer wen anruft.
- Legen Sie ein Paket aspects mit einem Aspekt Uebung1 an. Dieser Aspekt soll alle in Böhm, Kapitel 3.2 besprochenen Joinpoints protokollieren. Schreiben Sie den Advice darin wie folgt:

```

before(): !within(aspects.*) {
    System.out.println(" " + thisJoinPointStaticPart);
}

```

Speichern Sie das Protokoll in einer Datei not-within-aspects.log.
- Im Protokoll sehen Sie die Schlüsselwörter und Signaturen aller primitiven Joinpoints. Nun definieren Sie einen Pointcut interesting() als Veroderung vieler Pointcuts, die jeweils eine Art Joinpoint ansprechen. Endlosrekursionen müssen Sie dabei ausschließen durch Pointcuts wie `pointcut callMethod(): call(* *(..)) && within(telecom.*);`
Erfassen Sie damit alle Joinpoints, damit das erzeugte Protokoll identisch zu dem aus Teilaufgabe e) ist. Sie dürfen keine Negation dabei verwenden! Beide Protokolle dürfen beim Vergleich mit dem Eclipse Compare Editor keine Unterschiede aufweisen!

Das in der Konsole erscheinende Protokoll sieht in Teilen wie folgt aus:

```

execution(void telecom.AbstractSimulation.say(String))
get(PrintStream java.lang.System.out)
call(void java.io.PrintStream.println(String))
jim calls mik...
call(Call telecom.Customer.call(Customer))
execution(Call telecom.Customer.call(Customer))
call(boolean java.lang.Object.equals(Object))
call(telecom.Call(Customer, Customer))

```

```
staticinitialization(telecom.Call.<clinit>)
```