Rheinische Friedrich-Wilhelms-Universität Bonn
Institut für Informatik III
Prof. Dr. A. B. Cremers

Aspect-Oriented Software Development
Course in the summer semester 2009
Dr. Günter Kniesel, Daniel Speicher

Exercise Sheet 7

Due: Sunday 14.6.2009, 23:59:59 via SVN

For help, contact aosd-staff@lists.iai.uni-bonn.de (staff only) or aosd-course@lists.iai.uni-bonn.de (staff and participants).

Please start working on the exercises early enough so that you can contact us in time in case of problems. Don't expect us to be available during weekend!

LogicAJ

For this and the next exercise sheets you will need LogicAJ. A tutorial how to install LogicAJ is provided on the following website: http://sewiki.iai.uni-bonn.de/research/logicaj/installation.

Please be aware that you can NOT install LogicAJ and AspectJ within the same Eclipse instance. (AspectJ substitutes the Eclipse Java-Compiler with its own compiler, so LogicAJ won't work when you have AspectJ installed.)

Exercise 1: *"Generic Contracts"* (4 Points)

Use LogicAJ to implement generic *"declare warning"* contracts that check that

a) All methods and field names in a program start with a lower case letter.
b) All type names start with a capital letter.

Hint: Use the "predicate pointcuts" of LogicAJ for manipulating String literals and Lists.

(continued on page 2)

Exercise 2: *"Law of Demeter"* (8 Points)

The "Law of Demeter" is a style rule for object oriented software that advises you to let your methods only "talk to their friends". In particular, a method *m* of an object of type *T* should only invoke methods on a objects whose type is

1) *T* (or a subtype of thereof)
   or
2) the type of a parameter type of *m* (or a subtype thereof)
   or
3) the type of a field of *T* (or a subtype thereof)
   or
4) a type that is instantiated by any method or constructor within *T* (or a subtype thereof).

a) Your task is to write with LogicAJ a *"declare warning"* that checks if code of the project **"E07_E02_LawOfDemeter"** in your repository abides by the "Law of Demeter".
   Tip1: Your pointcut should have the form

```
call(…) &&
withincode(…) &&
!( rule1 || rule2 || rule3 || rule4 )
```

   Tip2: For rule 4 you need explicit join point variables (see slide 7-42).

b) If you had to write the same *"declare warning"* in AspectJ how would it look? What would be the main obstacles in using AspectJ?