

SESSION 5: STATEFUL ASPECTS IN JASCO EXERCISES

Using the JAsCo-plugin for Eclipse.

For this exercise session, the Eclipse IDE and the JAsCo plugin are employed. Short User Guide:

- For writing a JAsCo-enabled application, create a new JAsCo project. The base behaviour of a JAsCo project (i.e. the application itself) is implemented by making use of regular Java classes.
- For implementing a JAsCo aspect bean, add a “JAsCo aspect” to the project.
- To connect the aspects to the concrete classes, add a “JAsCo connector” to the project.
- To enable aspect application, just run the project using a JAsCo runner (run → run... → JAsCo Application → new), the aspects are automatically inserted at load-time. Make sure to employ Java 1.5 as running JRE!
- Use help → help contents → JAsCo help, for help for using the JAsCo IDE and for viewing the language reference.

Ex1: Protocol Checking

a) Download the stateful.zip file and import it into a new project. Try to launch the application and observe its behaviour. Also notice there is already an aspect integrated which provides content for the logging console.

b) Make a stateful aspect that prints a message whenever the protocol ComponentX.a, ComponentX.b, ComponentX.c has been encountered.

For printing a message, use the following method:

```
util.Logger.getInstance().logImportant(s);
```

c) Make a stateful aspect that prints a message whenever the protocol ComponentX.a, ComponentX.b, ComponentX.c has been encountered without allowing any intermediate transitions at all! *Use strict*

For printing a message, use the following method:

```
util.Logger.getInstance().logImportant(s);
```

d) Make a stateful aspect that prints a message whenever a method is executed on ComponentX outside of the protocol ComponentX.a, ComponentX.b, ComponentX.c.

Use complement

For printing a message, use the following method:

```
util.Logger.getInstance().logError(s);
```