

Program specialization and metaprogramming

Laboratory work No. 7

Introduction into Aspect Oriented Programming with AspectJ

1. Aims

To get acquainted with basic principles of Aspect Oriented programming. To get acquainted with programming in AspectJ.

2. Tasks

Analyse and modify the examples of AspectJ applications.

3. Work

A. Analyse a simple AspectJ program

Run Eclipse IDE. Select Samples on the main screen. Select AspectJ Examples on the Samples screen.

AJDT

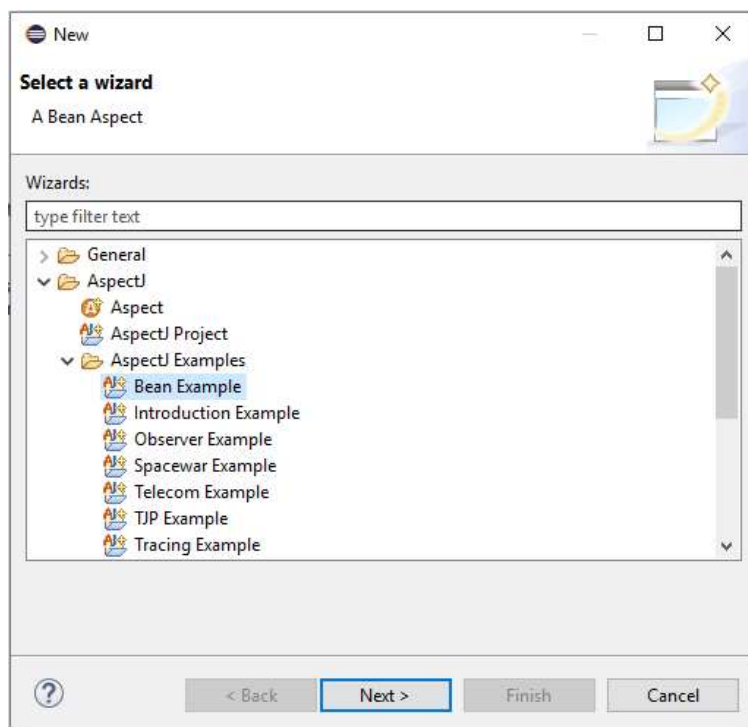
These examples demonstrate some of the capabilities of AspectJ.



AspectJ Examples

By selecting AspectJ, AspectJ Examples in the New Wizard that appears, you can select the example you wish to create.

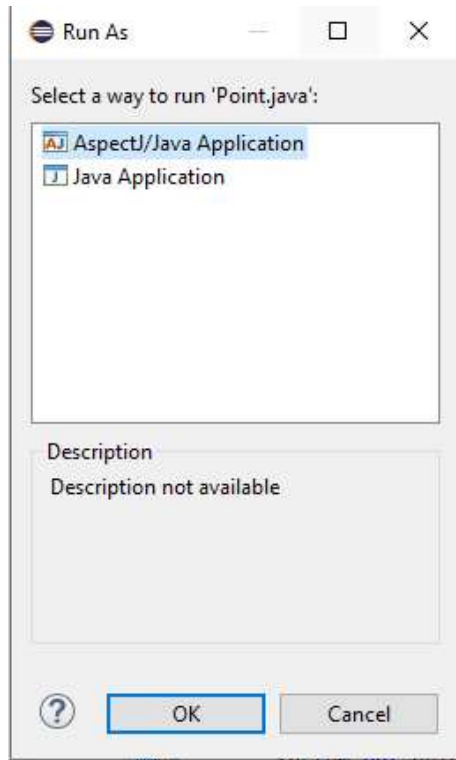
Select AspectJ Examples and in a wizard window select AspectJ Examples -> Introduction Example and press Finish.



Select Workbench at the right end of the toolbar at the top of the Samples window.

Inspect the source files of the project: Point.java, CloneablePoint.aj, ComparablePoint.aj and HashablePoint.aj. **What these files describe?** Check the main(...) methods of each class and aspect, which describe the entry points for program execution. **What operations they describe and what is the expected result?**

Run class Point.java as AspectJ/Java Application.



Check and analyse the result. **What the program has printed?**

Run aspect CloneablePoint.aj and check the result. **What the program has printed?**

Run aspect ComparablePoint.aj and check the result. **What the program has printed?**

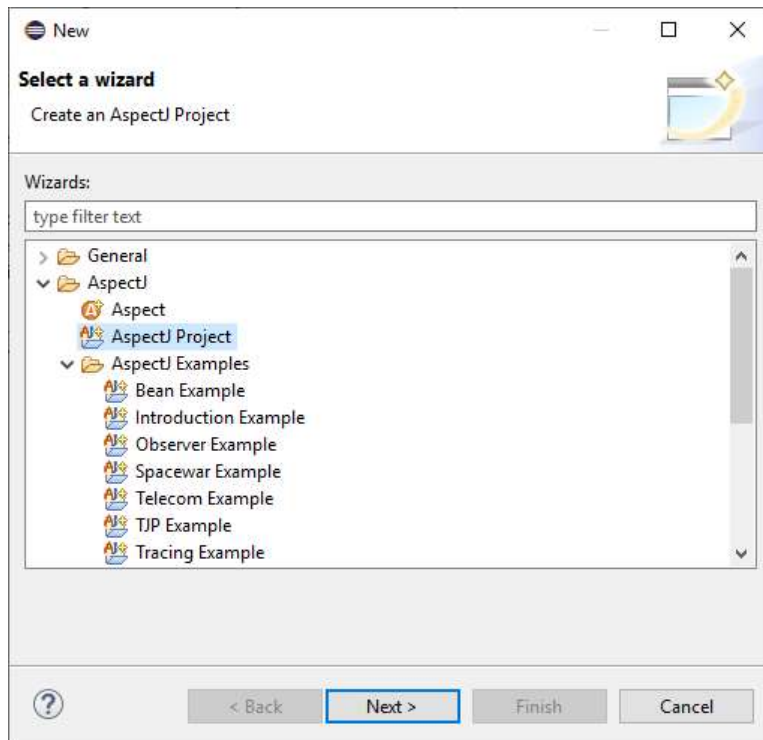
Run aspect HashablePoint.aj and check the result. **What the program has printed?**

How is the behaviour of aspect different from the behaviour of the class?

B. Create simple AspectJ program

Select in the Workbench menu File -> New -> Other

In the Wizard window, select AspectJ Project and press Next.



Enter project name factorial and press Finish. At the tree view of projects (on the left part of Workbench window) select src folder and add new class into project by selecting New -> Class.

Enter class name Math and press finish. Enter the following code into Math.java :

```
package factorial;
public class Math {
    public static int factorial(int n) {
        return 1;
    }
}
```

Add new class. Enter class name Demo and press finish. Enter the following code into Demo.java:

```
package factorial;
public class Demo {
    public static void main(String[] args) {
        System.out.println(Math.factorial(5));
    }
}
```

Add new aspect. Select src folder and add new aspect into project by selecting New -> Other....->Aspect. Enter aspect name Remainder and press finish. Enter the following code into Remainder.aj:

```
package factorial;
public aspect Remainder {
    int around(int n) : execution(int factorial(int)) && args(n) {
        if (n<2)
            return proceed(n);
        else
            return n * Math.factorial(n-1);
    }
}
```

Run the class Demo as AspectJ/Java application. Note the result in the program output window.

How can you explain the result?

What the around method in aspect Remainder does?

What the proceed method in aspect Remainder does?

Change the sentence in class Demo to:

```
System.out.println(Math.factorial(-5));
```

Modify aspect Remainder so that the calculation also works for negative numbers.

What is your solution?

Run modified aspect Remainder and observe the result in program output window.

4. Self-study

Explore, analyse and run the Observer example project.

What this project implements? (hint: The Observer pattern)

What the SubjectObserverProtocol aspect does?

What the SubjectObserverProtocolImpl aspect does?

Modify the project code and observe the results.

5. Report

Prepare a report of your work in this lab using a standard word processing application. The report is a single chapter of your final semester report. Final semester report will have to be uploaded to course Moodle page.

The content of report:

1. Title
2. Aims and tasks
3. Work done, described in steps and illustrated by screenshots and written or modified source code. Provide comments what you have done and what were the results (program outputs). Present answers to the questions given in the lab description.
4. Conclusions – what you have learned?