

Exercise Sheet 9

December 9: Decorator and Adapter

Exercise 1

Using decorators to extend `java.io.FilterInputStream`:

1. Create a subclass of `java.io.FilterInputStream` called `LowerCaseInputStream` that reads characters and converts them to lowercase! (Hint: overwrite the `read` methods accordingly!)
2. Write a simple test reading from a text file and outputting to the console (in lower case)!

Exercise 2

Using decorators in the JUnit framework:

1. Have a look into the `junit.extensions` package!
2. Extend `TestDecorator` as `PerfDecorator` that performs a performance logging of test methods executed. Hint: To configure the decorator, write something similar to

```
TestSuite suite = new TestSuite();
...
Test perfTest = new PerfDecorator(new TestSuite(MyTest.class));
suite.addTest(perfTest);
...
junit.textui.TestRunner.run(suite());
```

Exercise 3

Decorating table views:

1. Read the exercise in <http://www.javaworld.com/javaworld/jw-12-2001/jw-1214-designpatterns.html>!
2. Implement all classes and understand the code!
3. Extend `TableBubbleSortDecorator` as `TableLoggingDecorator` which performs a logging to the System console whenever a column is sorted! How do you have to modify the `Test` method in the class `Test` to make use of `TableLoggingDecorator`?

Exercise 4

Analyzing JHotDraw (compare previous exercise sheet)!

1. Find the Decorator Pattern in JHotDraw source code!
2. Why does the Decorator Pattern make sense here?!

Exercise 5

Analyzing Java:

1. Have a look at the `java.io.InputStreamReader(InputStream)` constructor returning a `java.io.InputStreamReader` based on a `java.io.InputStream`! What is the purpose of this constructor and which pattern is at work?
2. Similarly, have a look at the `java.io.OutputStreamWriter(OutputStream)` constructor returning a `java.io.OutputStreamWriter` based on a `java.io.OutputStream`! Again, What is the purpose of this constructor and which pattern is at work?
3. Read the javadoc for the class `java.awt.event.MouseAdapter` carefully! Which patterns are at work?

Hints

- Consult the literature!
- You can work in pairs, if you want!
- If you want to learn a Java API, look into the java docs!
- Always use the same familiar IDE (suggestion Eclipse)!