

---

**Software Practical Course**

<http://proglang.informatik.uni-freiburg.de/teaching/sopra/2004ws>

---

**Example solution of Exercise Sheet 3****Exercise 1**

When a solid rocket engine ignites, the part of its fuel that is exposed to air burns, producing thrust. From ignition to maximum burn rate, the burn area increases from the initial ignition area to the full surface area of the fuel. This maximum rate occurs at time  $t_{peak}$ . At this moment the rocket is at maximum speed. As fuel burns off, the surface area reduces again until the fuel is consumed.

Suppose that the burn area and thrust equation are:

$$rate = 25^{-(t-t_{peak})^2}$$
$$thrust = 1.7(rate/0.6)^{1/0.3}$$

We want to display the thrust/time diagram and the burnrate/time diagram depending on  $t_{peak}$ .  $t_{peak}$  may be set by a slider.

Complete the `slider()` and `stateChanged()` methods so that the diagrams reflect the slider's current value.

Where is an Observer here?

**Solution:**

See attached file *3.1.code.muster.java*

This code completely undoes the intent of Observer. Swing applies Observer so that the slider is not responsible for knowing which clients are interested in it. The Program registers a single dependent object itself, that dispatches changes to interested object. This object takes on responsibility for knowing which clients depend on the slider, instead of letting each dependent object register itself.

**Exercise 2**

Implement a class *Ballistic\_Panel* showing a design that lets each interested object register for slider events. Be sure to account for the label that shows the slider's value. Hint: function `repaint()`;

**Solution:**

To let each interested component register itself to receive the slider's change event, see attached file *3.2.code.muster.java*.

**Exercise 3**

In Marvel, we want to have items which simply provide another view on another existing item.

Say, we have a table with information of countries and want some inspectors to have access to the data without knowing the names of the countries. So we have to generate a new table which accesses the same data as the main table, but hides the countries' names.

Which Design Pattern can be used to implement this functionality? Give reasons for your answer.

**Solution:**

It's Proxy Design Pattern. The Pattern provide a surrogate or placeholder for another object to control access to it. The view is an instance of protection proxy, which controls access to the original object.