

Exercise 5

1. Active values are objects that inform other objects about state changes. An example is a temperature sensor coupled with a plotter. Every time the sensor changes its temperature value, all objects interested in the information are informed. Another receiver of the temperature could be an alarm device, which raises an alert when temperature exceeds 40 degrees celsius.
 - a. Implement the sensor and the plotter without events.
 - b. Implement the sensor as an active value with events and add a corresponding plotter. For the plotter it should be possible to register with the active value. The sensor informs the registered objects about changes.
 - c. Extend both solutions with a temperature alert.
 - d. Compare both solutions.
 - e. Draw the object diagram with interactions for both solutions.
2. The „Factory Method“ pattern is used to let the framework create an object without specifying the exact class to create it. You can describe the behavior in abstract classes in the framework, but the concrete implementation is left to be filled later. Instead of creating the object directly, you call a dynamically bound method to create the object for you, which returns the object. This method is called a factory method. By overriding this method in subclasses, someone can return more specialized objects without changing the rest of the implementation. With this pattern, the behavior of a framework can be adapted.

An example for this pattern is an application handling many different documents.

Write a small framework consisting of the application and document, and write a dummy application based on this, e.g. DrawingApplication and ImageDocument.

At least the framework should offer functions to create a new document, to open an existing document, and to list all open documents.

A Document offers functions to create, open, save, and close a file.

Draw the class diagram for the application.

Look at the following page for the factory method pattern:

<http://c2.com/cgi/wiki?FactoryMethodPattern>