

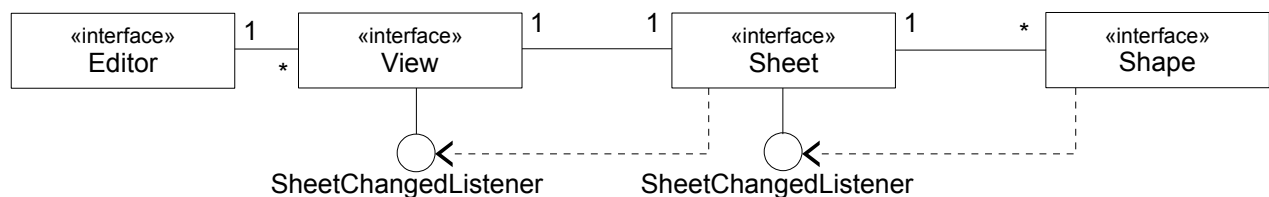
Drawing Shapes

Introduction

This exercise deals with the Observer pattern.

Joint Behavior of Shapes, Sheet and View

When created, a shape is added to a sheet by a tool (indirectly via the editor and view). The following UML class diagram shows the relationship between shapes, a sheet and a view:



A `View` object presents a sheet, that is, a drawing, on the computer's screen. A `Sheet` object contains several `Shape` objects such as lines, ellipses, and boxes. As soon as a `Shape` object changes its state (that is, its position or size) then it notifies the associated sheet. This allows a sheet to take into account the necessary reactions for the shape's state change.

A `Sheet` object, too, notifies its view if it changes its state. Sheet state changes occur, for example, whenever a shape is added or removed, or a shape has changed its state (for example, its color). Upon behalf of this notification, the view will repaint the computer's screen, for example. *More precisely, the view will draw the sheet, and the sheet will draw its content.*

Your Tasks

- Study the framework's interfaces.
- Study the usage of the listeners.
- Implement the registration of listeners for shapes: a sheet must register itself at every added shape
- Implement the management of registered listeners in a new `Sheet` class: a view registers itself at its sheet.
- Ensure that the listeners' method are called appropriately.
- Write specific test classes.
- Execute the JUnit tests you added.

Remarks

- Only with the correct notification of the listeners does the editor display the shapes on the screen!
- Do not add classes to the packages `ch.bfh.due1.jdt.framework` and `ch.bfh.due1.jdt.simple.*`.
- Do not change any classes in the packages `ch.bfh.due1.jdt.framework` and `ch.bfh.due1.jdt.simple.*`.
- Modify and/or extend existing classes in package `ch.bfh.sed.observerpattern`.



References

- Software frameworks: http://en.wikipedia.org/wiki/Software_framework
- Observer pattern (GoF version): http://en.wikipedia.org/wiki/Observer_pattern
- Java events and listeners: <http://chaoticjava.com/posts/event-listener-registrations/>