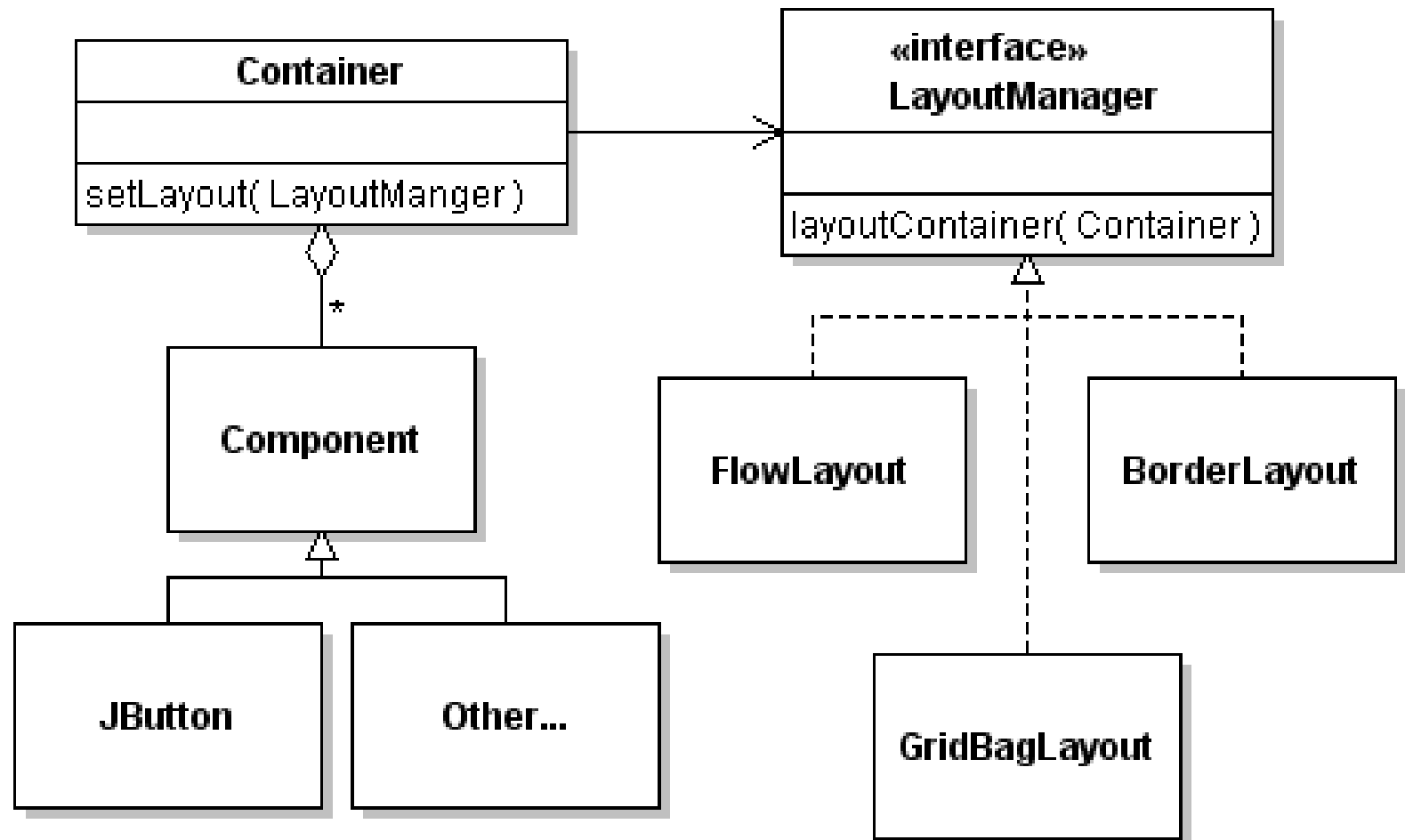




Design Patterns in Swing and AWT

- **Strategy Pattern:** LayoutManagers are *Strategies*
- **Observer Pattern:** event listeners are observers
- **Composite Pattern:** a container can be placed inside another container
- **Decorator Pattern:** you can "decorate" any component by surrounding it with JScrollPane to add scroll bars
- **MVC Pattern:** JTable gets data from a TableModel

LayoutManager - what pattern?





LayoutManager is a Strategy

Strategy Pattern

Name in Pattern	Name in this Example
Content	Container
Strategy	LayoutManager
Concrete Strategy	FlowLayout, GridBagLayout, ..
setStrategy	setLayout(LayoutManager)
doWork()	layoutContainer()

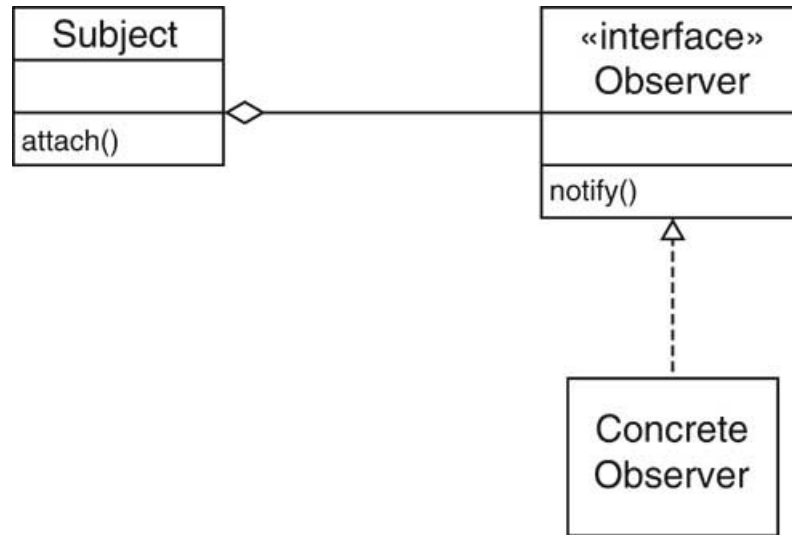


Benefit of LayoutManager

- What are the benefits of separating LayoutManager from the container classes?

Why don't we put the layout code inside each container?

Observer Pattern



Name In Observer Pattern

Subject

Observer

Concrete Observer

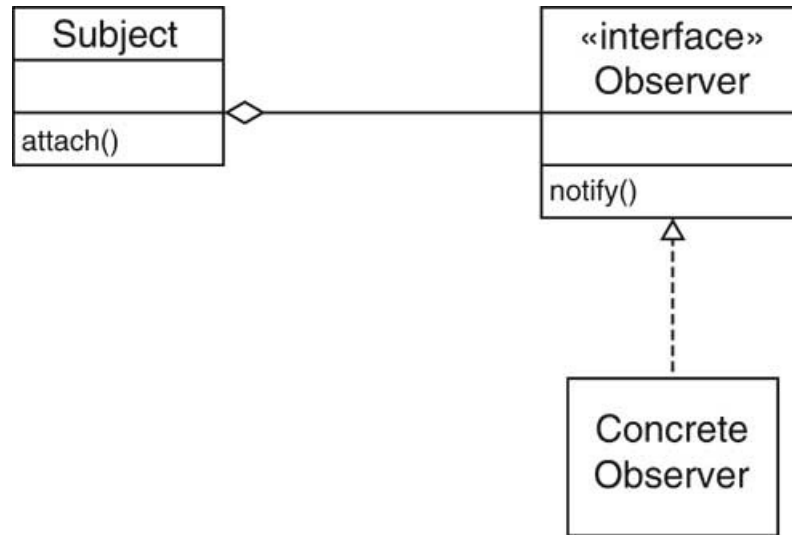
attach()

notify()

Name in Swing graphics



Observer Pattern



Name In Observer Pattern

Subject

Observer

Concrete Observer

attach()

notify()

Name in Swing graphics

JButton, JMenuItem, JCheckBox

ActionListener

your class implementing ActionListener

addActionListener(observer)

actionPerformed()