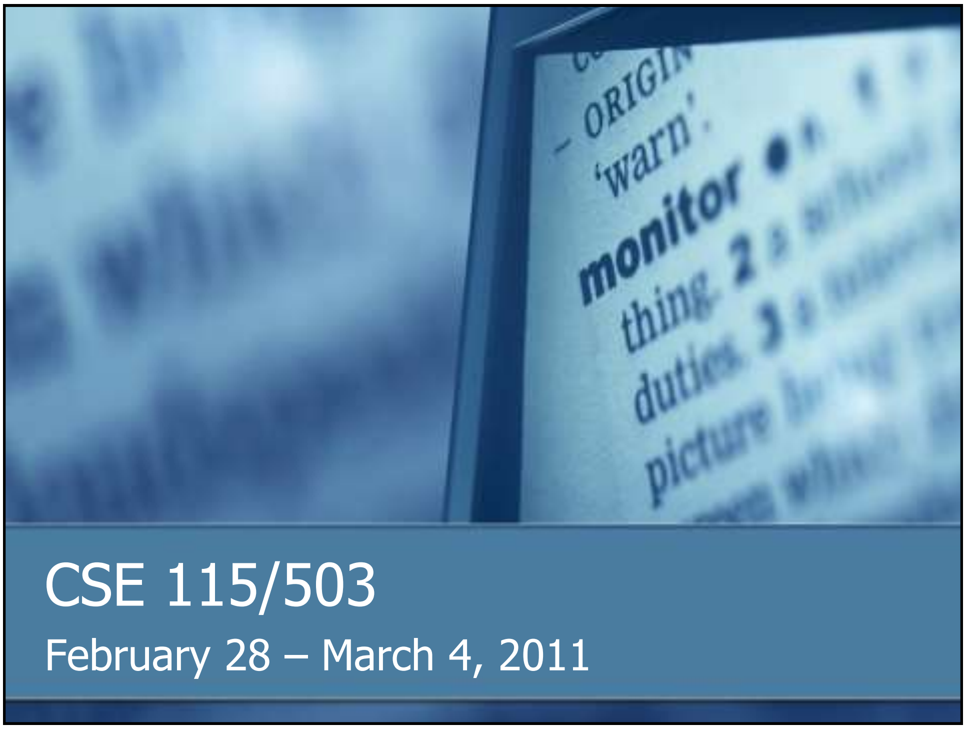
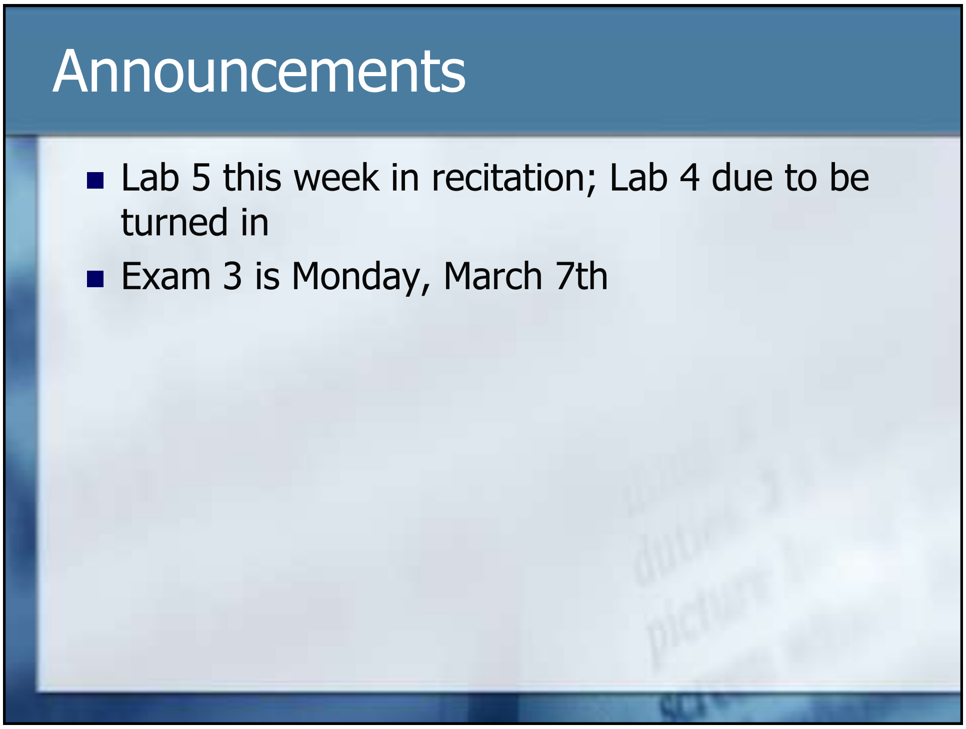


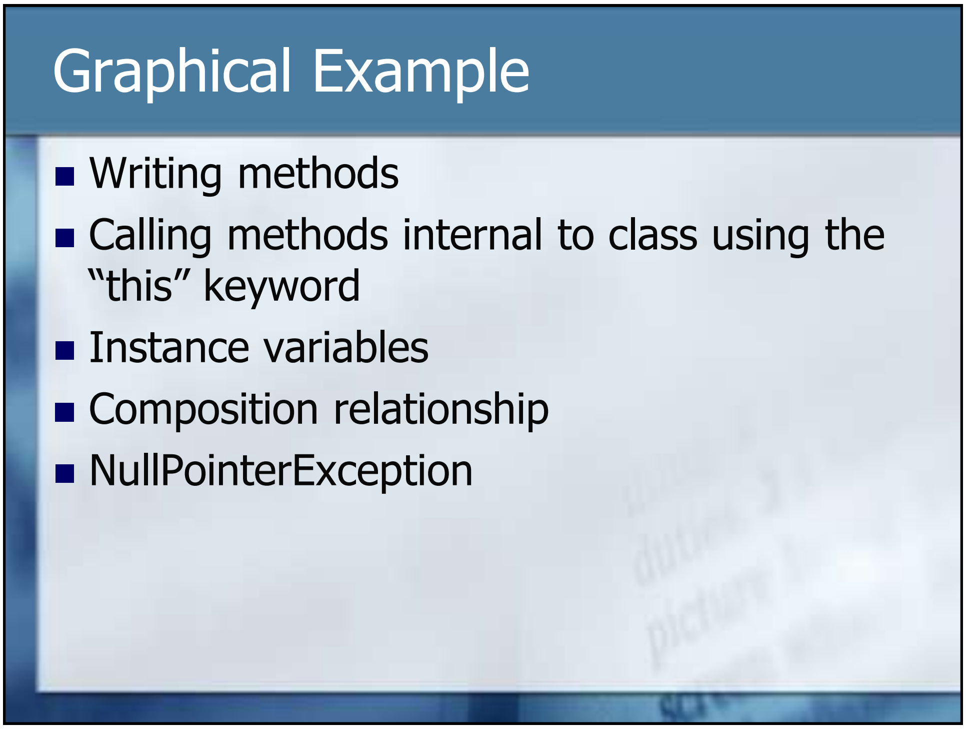
CSE 115/503

February 28 – March 4, 2011



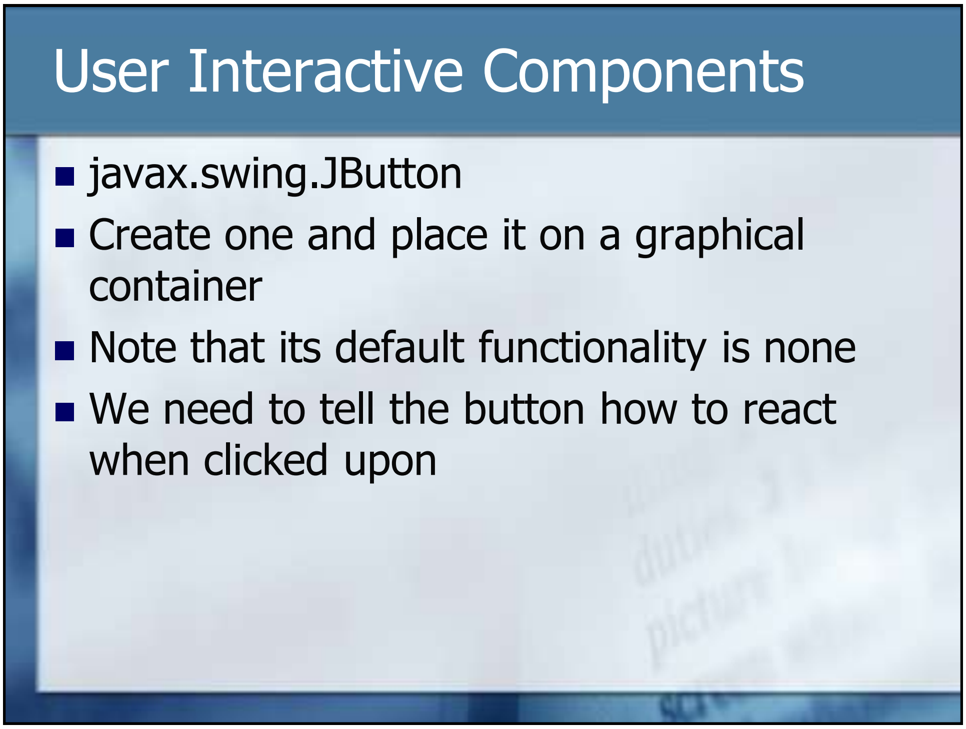
Announcements

- Lab 5 this week in recitation; Lab 4 due to be turned in
- Exam 3 is Monday, March 7th



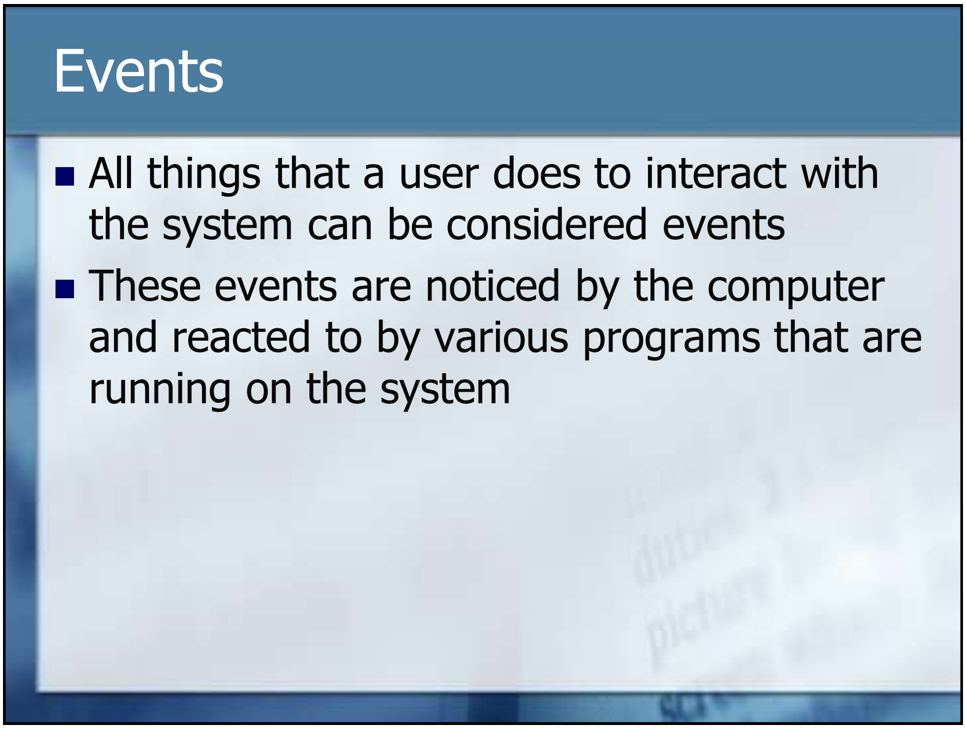
Graphical Example

- Writing methods
- Calling methods internal to class using the "this" keyword
- Instance variables
- Composition relationship
- NullPointerException



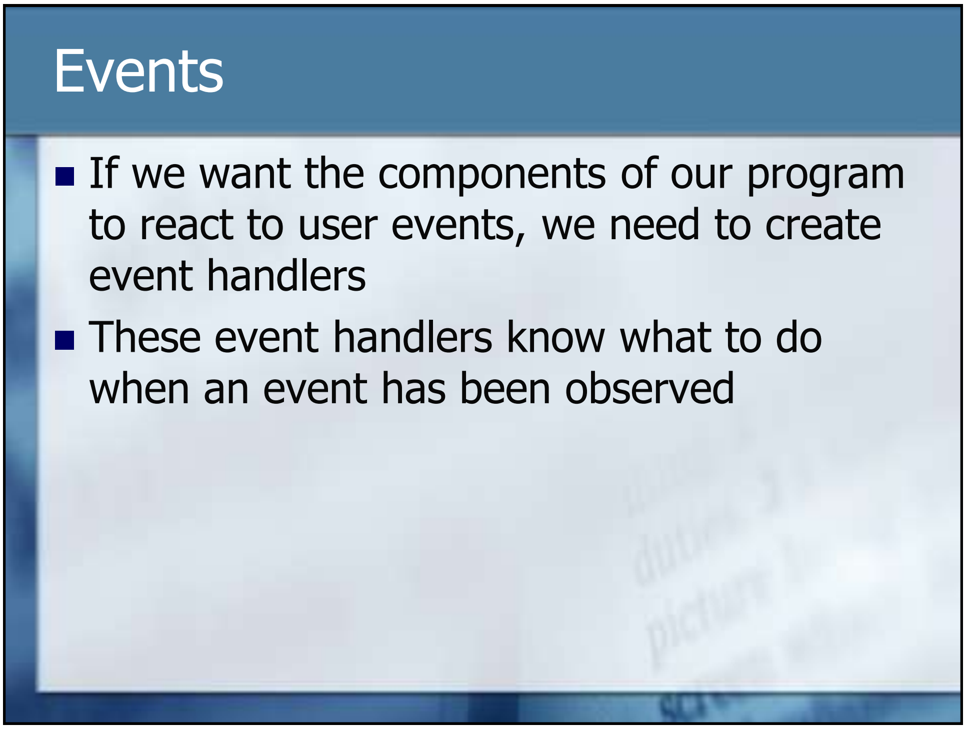
User Interactive Components

- javax.swing.JButton
- Create one and place it on a graphical container
- Note that its default functionality is none
- We need to tell the button how to react when clicked upon



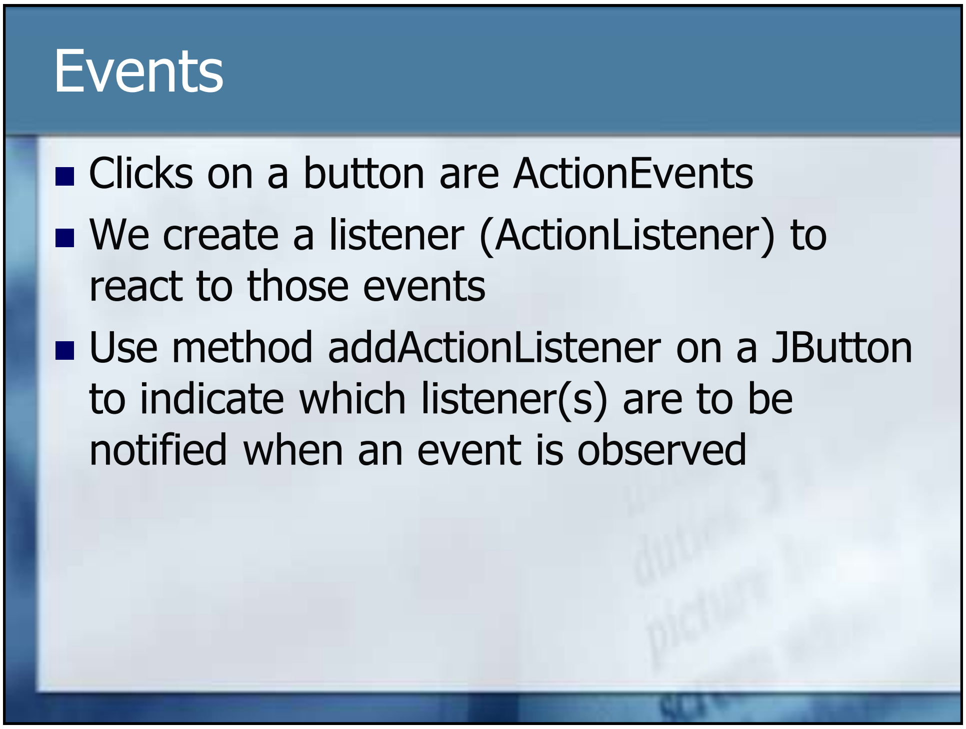
Events

- All things that a user does to interact with the system can be considered events
- These events are noticed by the computer and reacted to by various programs that are running on the system



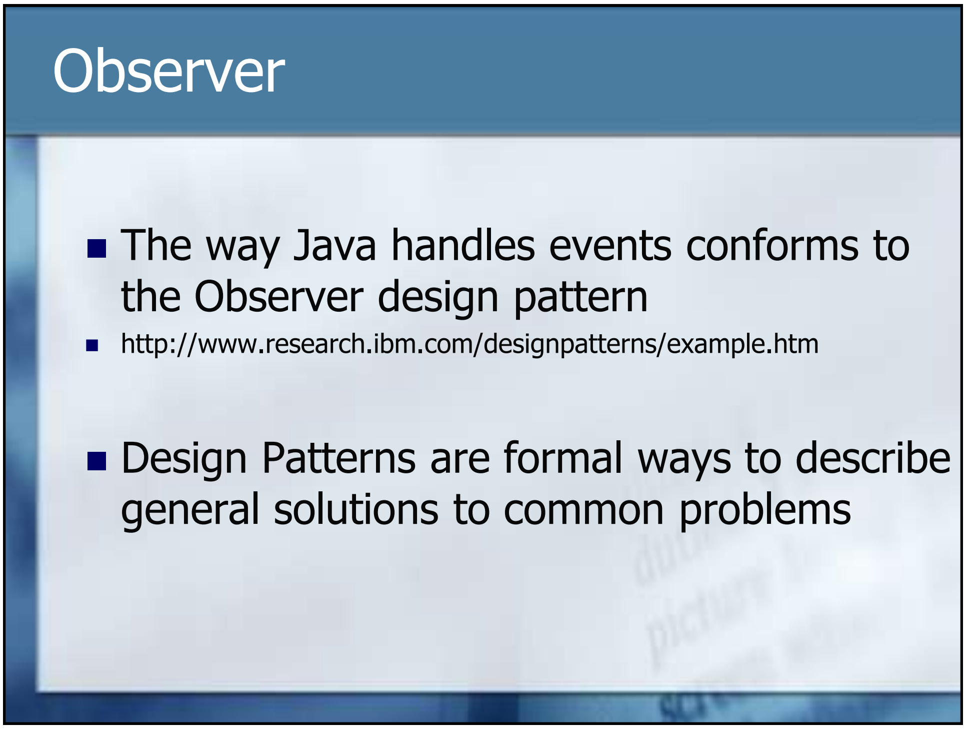
Events

- If we want the components of our program to react to user events, we need to create event handlers
- These event handlers know what to do when an event has been observed



Events

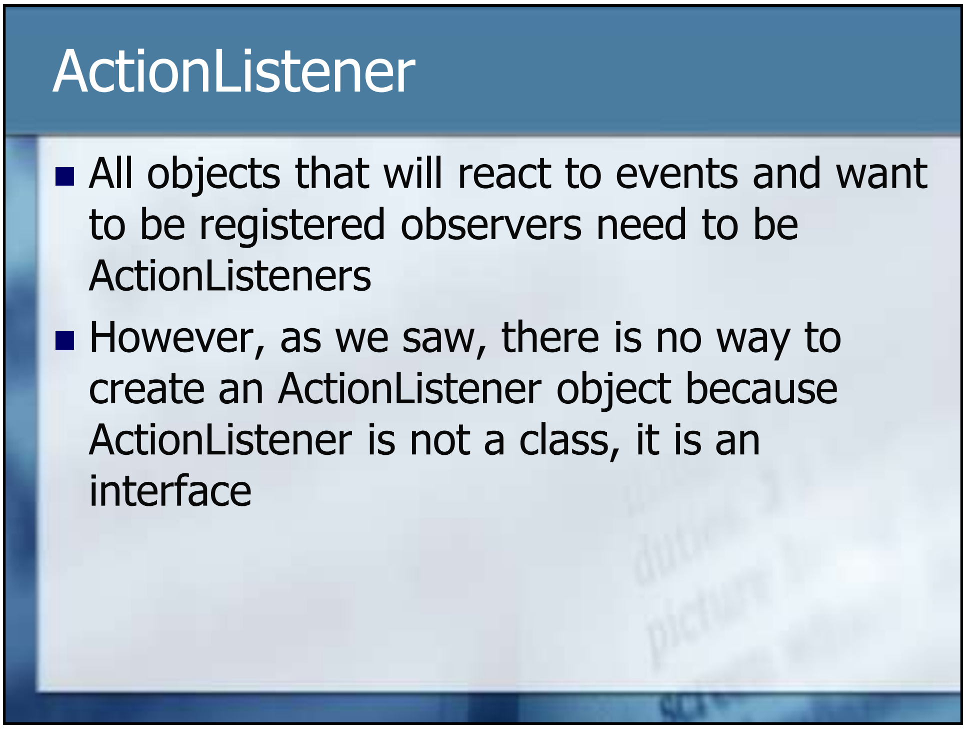
- Clicks on a button are ActionEvents
- We create a listener (ActionListener) to react to those events
- Use method addActionListener on a JButton to indicate which listener(s) are to be notified when an event is observed



Observer

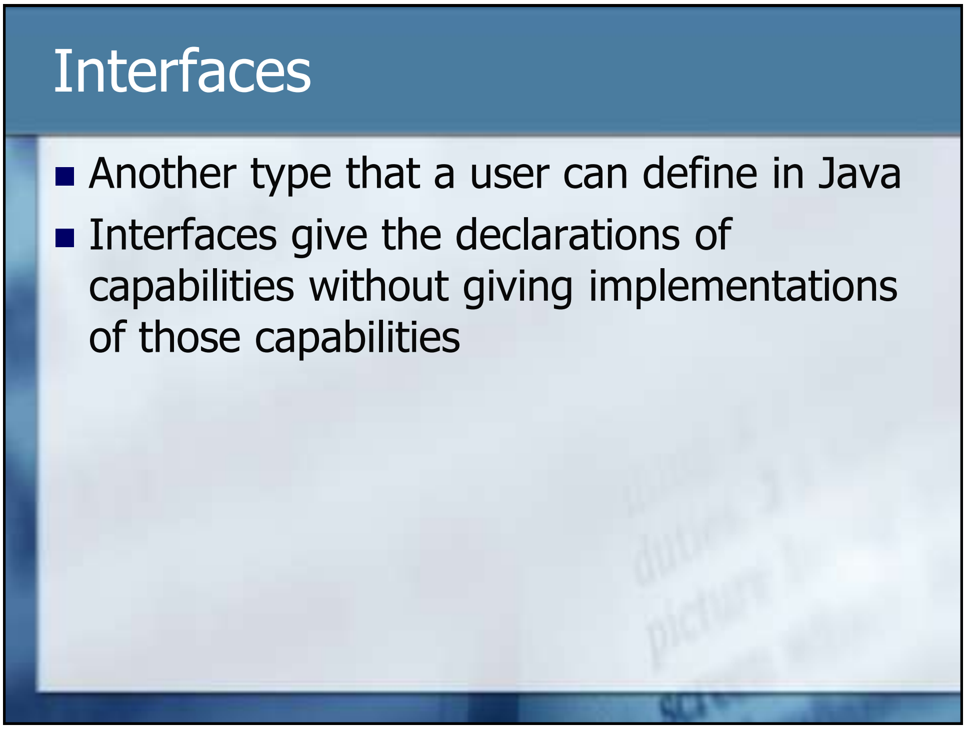
- The way Java handles events conforms to the Observer design pattern
- http://www.research.ibm.com/designpatterns/example.htm

- Design Patterns are formal ways to describe general solutions to common problems



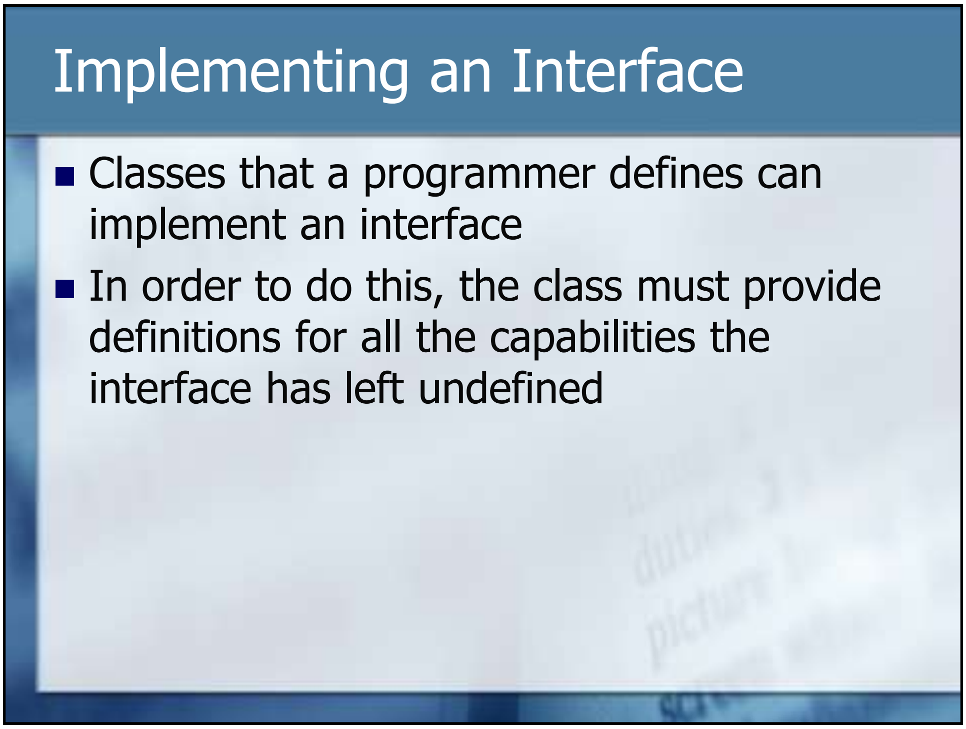
ActionListener

- All objects that will react to events and want to be registered observers need to be ActionListeners
- However, as we saw, there is no way to create an ActionListener object because ActionListener is not a class, it is an interface



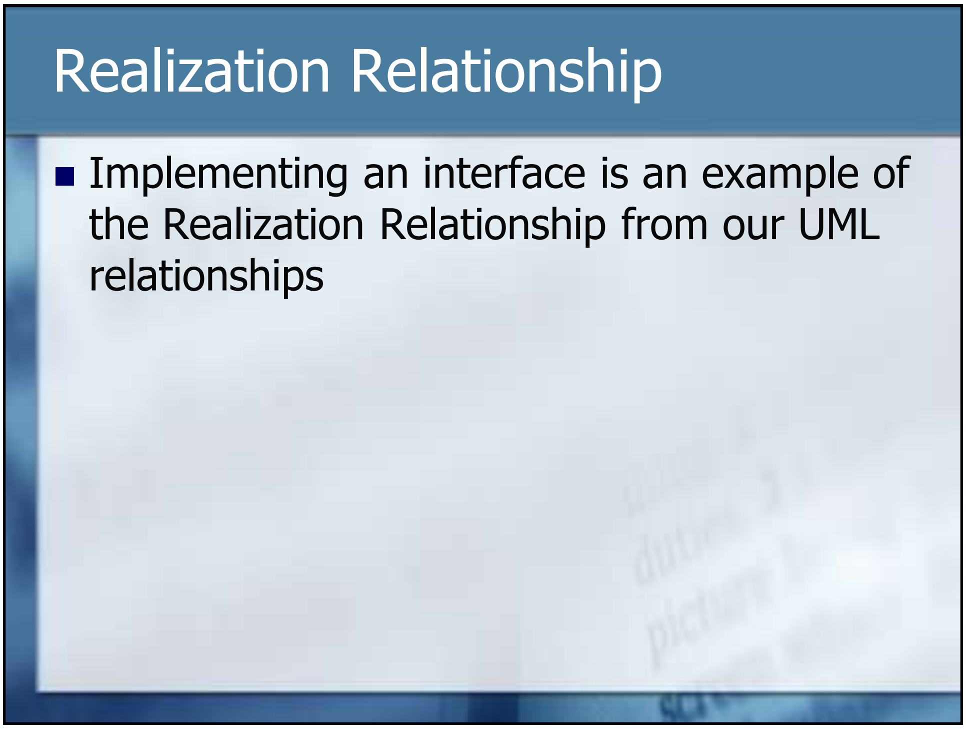
Interfaces

- Another type that a user can define in Java
- Interfaces give the declarations of capabilities without giving implementations of those capabilities



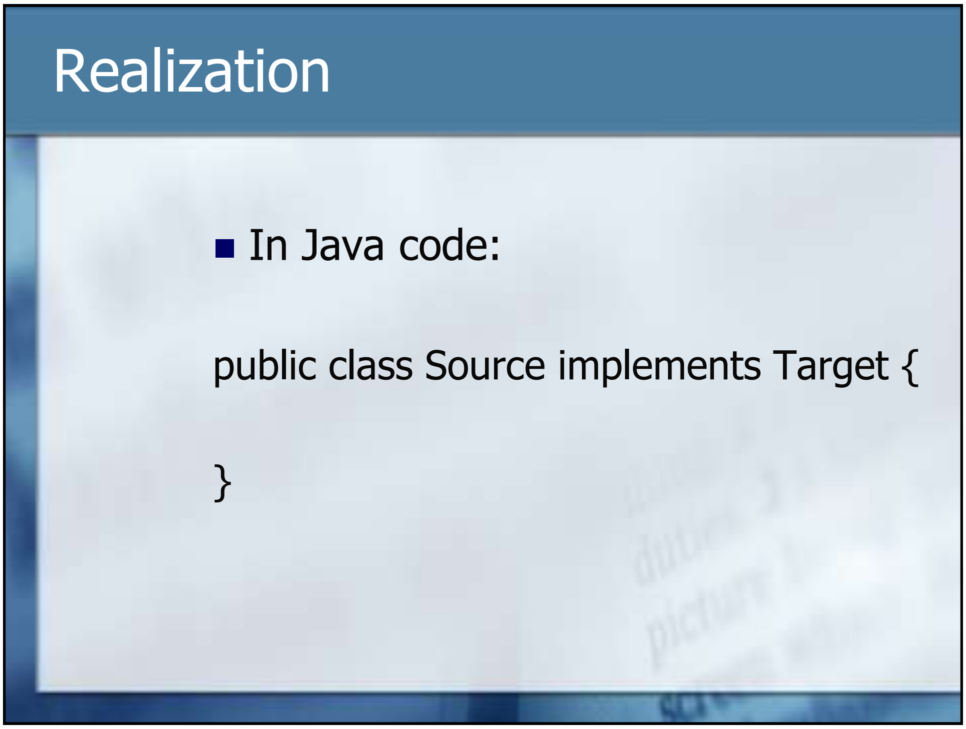
Implementing an Interface

- Classes that a programmer defines can implement an interface
- In order to do this, the class must provide definitions for all the capabilities the interface has left undefined



Realization Relationship

- Implementing an interface is an example of the Realization Relationship from our UML relationships

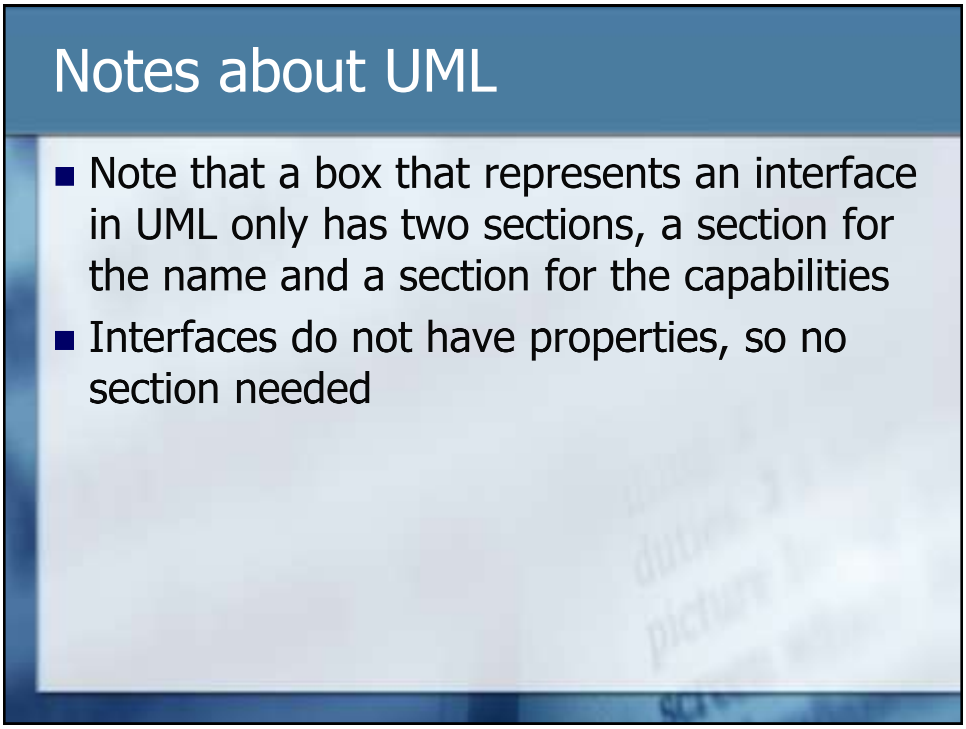


Realization

- In Java code:

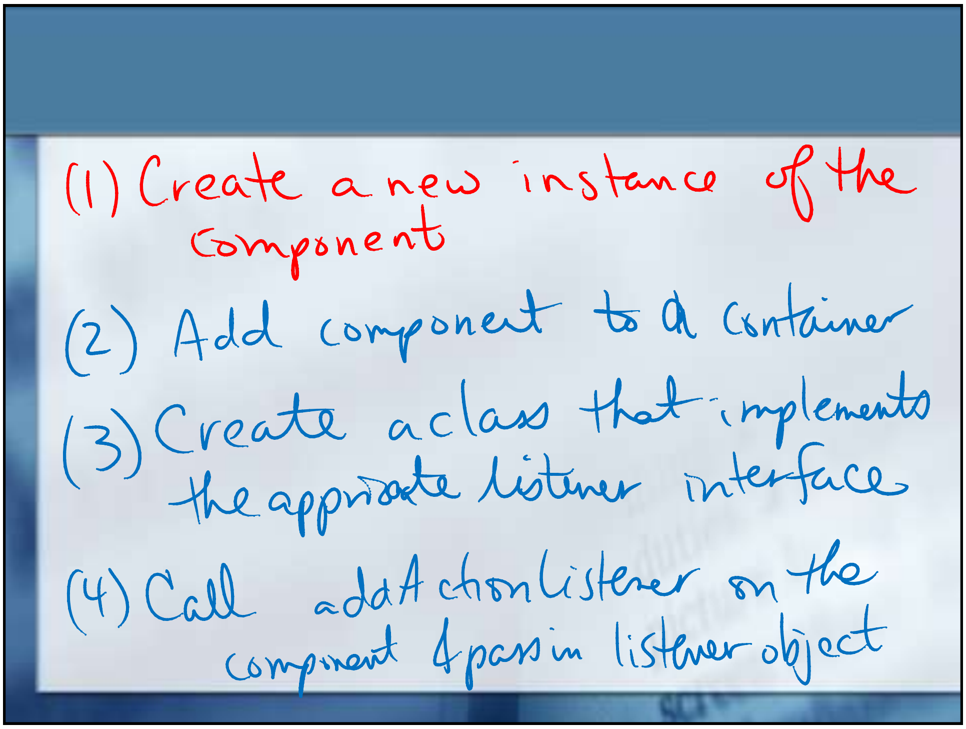
```
public class Source implements Target {

}
```



Notes about UML

- Note that a box that represents an interface in UML only has two sections, a section for the name and a section for the capabilities
- Interfaces do not have properties, so no section needed



(1) Create a new instance of the component

(2) Add component to a container

(3) Create a class that implements the appropriate listener interface

(4) Call addActionListener on the component & pass in listener object