# Relevant Classes for the Observer Refactoring Activity

class **PropertyChangeSupport**

- It manages a list of listeners and dispatches PropertyChangeEvents to them. You can use an instance of this class as a member field of your bean and delegate these types of work to it. The PropertyChangeListener can be registered for all properties or for a property specified by name.

Relevant Methods:

public void **addPropertyChangeListener**(String propertyName, PropertyChangeListener listener)

- Add a PropertyChangeListener for a specific property. The listener will be invoked only when a call on firePropertyChange names that specific property. The same listener object may be added more than once. For each property, the listener will be invoked the number of times it was added for that property. If propertyName or listener is null, no exception is thrown and no action is taken.

public void **firePropertyChange**(String propertyName, Object oldValue, Object newValue)

- Reports a bound property update to listeners that have been registered to track updates of all properties or a property with the specified name.

Interface **PropertyChangeListener**

- A "PropertyChange" event gets fired whenever a bean changes a "bound" property. You can register a PropertyChangeListener with a source bean so as to be notified of any bound property updates.

Relevant Methods:

public void **propertyChange**(PropertyChangeEvent evt)

- This method gets called when a bound property is changed.
- A PropertyChangeEvent has the following relevant methods:
  - getPropertyName() → String
  - getSource() → Object
  - getOldValue() → Object
  - getNewValue() → Object