Chapter 27 JavaBeans and Bean Events

1. A JavaBeans component is a serializable public class with a default public constructor. Not every GUI class is a JavaBeans component. For example, java.awt.Color is not a JavaBeans component. Every GUI user interface class is a JavaBeans component. However, a JavaBeans component may be a non-Swing class.

2. By convention, the accessor method is named get<PropertyName>(), which takes no parameters and returns a primitive type value or an object of a type identical to the property type. For a property of boolean type, the accessor method should be named is<PropertyName>(), which returns a boolean value. The mutator method should be named set<PropertyName>(dataType p), which takes a single parameter identical to the property type and returns void.

3. By convention, the registration method is named add<Event>Listener(<Event>Listener listener) and a deregistration method is named remove<Event>Listener(<Event>Listener listener).

4. Since an event class and its listener interface are coexistent, they are often referred to as an event set or event pair. The event listener interface must be named as XListener for the XEvent. For example, the listener interface for ActionEvent is ActionListener.

You can declare a custom event class by extending java.util.EventObject or a subclass of java.util.EventObject.

You can declare a custom event listener interface by extending java.util.EventListener or a subinterface of java.util.EventListener.

5. The Java event model is flexible, allowing modifications and variations. One useful variation of the model is the addition of adapters. When an event occurs, the source object notifies the adapter. The adapter then delegates the handling of the event to the actual processing object, which is referred to as an adaptee.

6. See Section 25.7.

7. There are three types of event adapters: standard adapters, inner class adapters, and anonymous adapters.
   - standard adapter A named class that extends a convenience listener adapter or implements a listener interface.
   - inner class adapter Standard adapters can be shortened using inner classes.
   - anonymous adapter Inner class adapters can be further shortened using anonymous inner classes.