Chapter 14 Event-Driven Programming

1. A WindowEvent is generated by an instance of the Window class or its subclass. JButton is not a subclass of Window, therefore, it cannot generate the WindowEvent. JButton can generate MouseEvent and ActionEvent.

2. A listener must have a correct method to handle the event. To ensure that a listener have the method, a listener must be an instance of a listener interface, where a method is defined.

   To register a listener object, you invoke the source object's add*X*Listener's method; for example, button.AddActionListener(this). To implement a listener interface, you add implements *X*listener and implement all the handlers in the listener's object.

3. Yes. Yes. Yes.

4. To override a method defined in the listener interface, you provide the code in the body of the method for handling the event. You need to override all the methods defined in the listener interface. If a handler is not used in the program, you can give it an empty body.

5. Objects of an inner class are often created in the outer class. But you can also create an object of an inner class from another class. If the inner class is nonstatic, you must first create an instance of the outer class, then use the following syntax to create an object for the inner class:

```
OuterClass.InnerClass innerObject = outerObject.new InnerClass();
```

If the inner class is static, use the following syntax to create an object for it:

```
OuterClass.InnerClass innerObject = new OuterClass.InnerClass();
```

6. Yes.

7. If class A is an inner class in class B, the .class file for A is B$A.class. If class B contains two anonymous inner classes, the .class file names for these two classes are B$+1 and B$+2.

8. (a)
   1. java.awt.event package should be imported to use ActionEvent and ActionListener in this program;

2. java.swing should be javax.swing
3. actionPerform should be actionPerformed;
5. jbtOK is not in the scope of the actionPerformed method;
5. no listener is registered with jbtOK.

(b) missing) in Line ;

9. Use getSource() in the EventObject class to get the source of an event. Use getWhen() in the ActionEvent class to get the timestamp for an action event, use getWhen() in the InputEvent class to get the timestamp a mouse event or a key event. Use getPoint(), or getX() and getY() to get the mouse point position for a mouse event. Use getKeyChar() to get the key character for a key event.

10. The listener interface for move pressed, released, clicked, entered, and exited is MouseListener. The listener interface for mouse moved and dragged is MouseMotionListener.

11. Not every key in the keyboard has a Unicode. For example, the function keys, modifier keys, action keys, and control keys are not Unicode characters. The key code in the KeyEvent class are defined as constants. Their value is the same as the Unicode value if the key is Unicode character.

12. The keyPressed handler is invoked after a key is pressed. The keyReleased handler is invoked after a key is released. The keyTyped handler invoked after *a Unicode character* key is typed.

13. You create a timer using the constructor

```
public Timer(int delay, ActionListener listener)
```

Use the start method to start the timer, and the stop method to stop the timer.

14. The Timer class does not have a no-arg constructor. You can add multiple listeners by invoking the timer.addActionListener(listener).