



Java Programming Sixth Edition

Chapter 15: Advanced GUI Topics
lec(6)

A Closer Look at Events and Event Handling (cont'd.)

- `ActionEvents` are generated by components that users can click, such as `JButtons`
- `TextEvents` are generated by components into which the user enters text, such as `JTextFields`
- `MouseEvents` include clicking a component, determining the location of the mouse pointer and distinguishing between a single- and double-click using left or right mouse button.

Handling Key Events

- `KeyListener` **interface**
 - Receive actions user initiates from keyboard
- `KeyListener` contains three handlers methods
 - `keyPressed()`
 - `keyTyped()`
 - Discover which character was typed
 - Does not execute calls from action keys, such as a function key.
 - `keyReleased()`
 - Does not take action while user holds down key

JDemoKeyFrame



Handling Mouse Events

- **MouseListener interface** provides you with methods

Method	Description
<code>void mouseClicked(MouseEvent e)</code>	Invoked when the mouse button has been clicked (pressed and released) on a component
<code>void mouseEntered(MouseEvent e)</code>	Invoked when the mouse pointer enters a component
<code>void mouseExited(MouseEvent e)</code>	Invoked when the mouse pointer exits a component
<code>void mousePressed(MouseEvent e)</code>	Invoked when a mouse button has been pressed on a component
<code>void mouseReleased(MouseEvent e)</code>	Invoked when a mouse button has been released on a component

Table 15-6 MouseListener methods

Handling Mouse Events

- **MouseEventListener interface**

provides you with methods that detect the mouse being rolled or dragged across a component surface.

Method	Description
<code>void mouseDragged(MouseEvent e)</code>	Invoked when a mouse button is pressed on a component and then dragged
<code>void mouseMoved(MouseEvent e)</code>	Invoked when the mouse pointer has been moved onto a component but no buttons have been pressed

Table 15-7 MouseEventListener methods

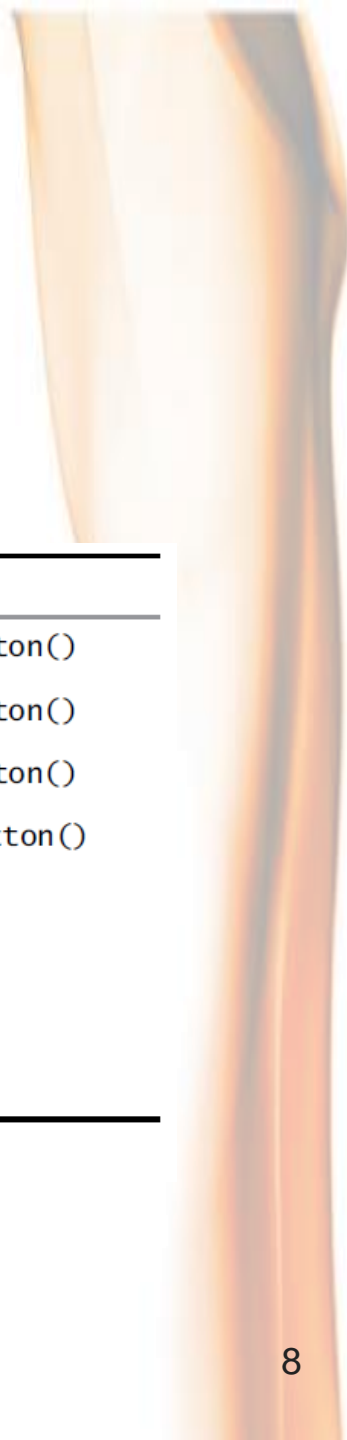
Handling Mouse Events

- **MouseEvent**

- Type of event generated by mouse manipulation
- Contains instance methods and fields useful in describing mouse-generated events

Method	Description
<code>int getButton()</code>	Returns which, if any, of the mouse buttons has changed state; uses fields <code>NOBUTTON</code> , <code>BUTTON1</code> , <code>BUTTON2</code> , and <code>BUTTON3</code>
<code>int getClickCount()</code>	Returns the number of mouse clicks associated with the current event
<code>int getX()</code>	Returns the horizontal x-position of the event relative to the source component
<code>int getY()</code>	Returns the vertical y-position of the event relative to the source component

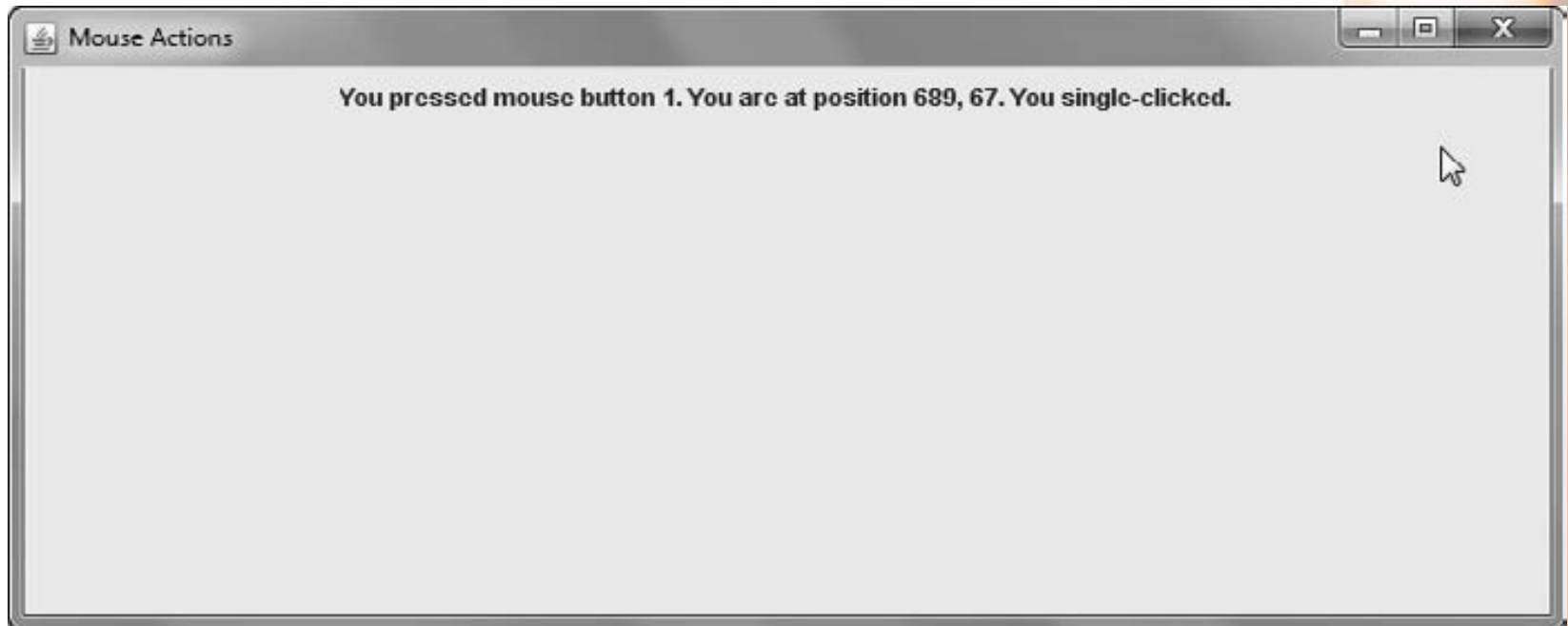
Table 15-8 Some useful `MouseEvent` methods



Field	Description
<code>static int BUTTON1</code>	Indicates mouse button #1; used by <code>getButton()</code>
<code>static int BUTTON2</code>	Indicates mouse button #2; used by <code>getButton()</code>
<code>static int BUTTON3</code>	Indicates mouse button #3; used by <code>getButton()</code>
<code>static int NOBUTTON</code>	Indicates no mouse buttons; used by <code>getButton()</code>
<code>static int MOUSE_CLICKED</code>	The “mouse clicked” event
<code>static int MOUSE_DRAGGED</code>	The “mouse dragged” event
<code>static int MOUSE_ENTERED</code>	The “mouse entered” event
<code>static int MOUSE_EXITED</code>	The “mouse exited” event

Table 15-9 Some useful `MouseEvent` fields

JMouseActionFrame



Using Menus

- **Menus**
 - Horizontal and vertical Lists of user options

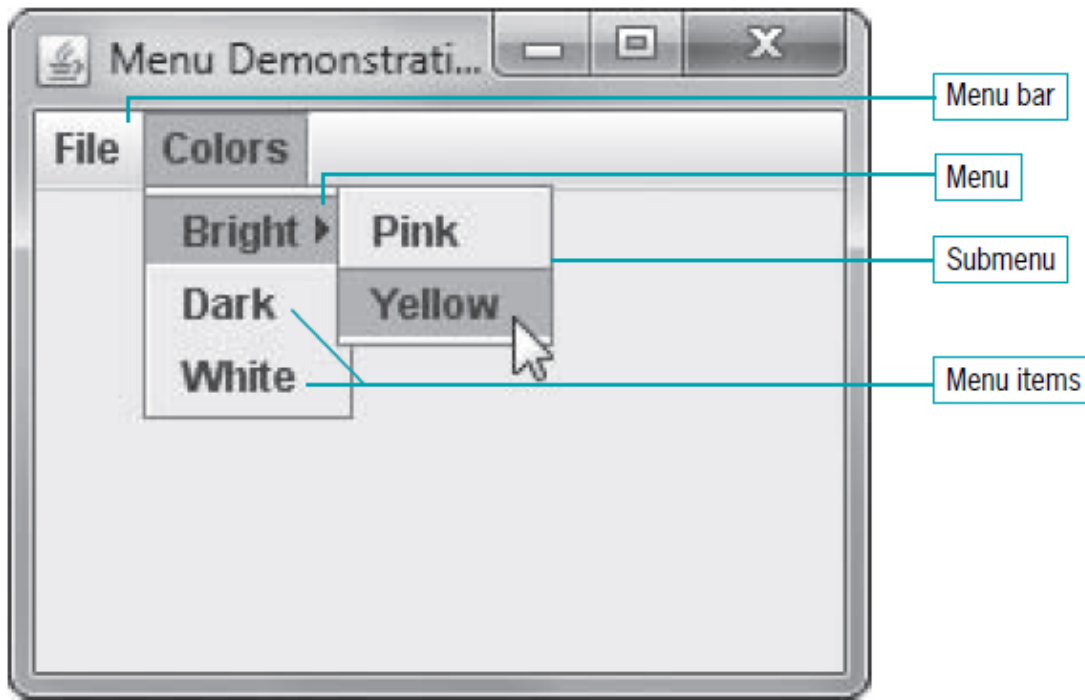


Figure 15-31 A JFrame with a horizontal JMenuBar that holds two JMenus

Java Menu Classes

- **Classes**

- JMenuBar \\ The horizontal list,
- JMenu \\ The vertical list,
- JMenuItem

- **Some methods**

- setJMenuBar ()
 - **Add JMenuBar to a JFrame**
- add ()
 - **Add JMenu to the JMenuBar**

JMenuFrame

