| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MouseInfo.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/MenuShortcut.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/MultipleGradientPaint.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/MouseInfo.html)    [**NO FRAMES**](http://docs.google.com/MouseInfo.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **java.awt**

Class MouseInfo

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.awt.MouseInfo**

public class **MouseInfo**extends [Object](http://docs.google.com/java/lang/Object.html)

MouseInfo provides methods for getting information about the mouse, such as mouse pointer location and the number of mouse buttons.

**Since:** 1.5

| **Method Summary** | |
| --- | --- |
| static int | [**getNumberOfButtons**](http://docs.google.com/java/awt/MouseInfo.html#getNumberOfButtons())()            Returns the number of buttons on the mouse. |
| static [PointerInfo](http://docs.google.com/java/awt/PointerInfo.html) | [**getPointerInfo**](http://docs.google.com/java/awt/MouseInfo.html#getPointerInfo())()            Returns a PointerInfo instance that represents the current location of the mouse pointer. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Method Detail** |
| --- |

### getPointerInfo

public static [PointerInfo](http://docs.google.com/java/awt/PointerInfo.html) **getPointerInfo**()  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Returns a PointerInfo instance that represents the current location of the mouse pointer. The GraphicsDevice stored in this PointerInfo contains the mouse pointer. The coordinate system used for the mouse position depends on whether or not the GraphicsDevice is part of a virtual screen device. For virtual screen devices, the coordinates are given in the virtual coordinate system, otherwise they are returned in the coordinate system of the GraphicsDevice. See [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) for more information about the virtual screen devices. On systems without a mouse, returns null.

If there is a security manager, its checkPermission method is called with an AWTPermission("watchMousePointer") permission before creating and returning a PointerInfo object. This may result in a SecurityException.

**Returns:**location of the mouse pointer **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if a security manager exists and its checkPermission method doesn't allow the operation**Since:** 1.5 **See Also:**[GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html), [SecurityManager.checkPermission(java.security.Permission)](http://docs.google.com/java/lang/SecurityManager.html#checkPermission(java.security.Permission)), [AWTPermission](http://docs.google.com/java/awt/AWTPermission.html)

### getNumberOfButtons

public static int **getNumberOfButtons**()  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Returns the number of buttons on the mouse. On systems without a mouse, returns -1.

**Returns:**number of buttons on the mouse **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**Since:** 1.5

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MouseInfo.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/MenuShortcut.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/MultipleGradientPaint.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/MouseInfo.html)    [**NO FRAMES**](http://docs.google.com/MouseInfo.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).