| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Line2D.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/geom/IllegalPathStateException.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/geom/Line2D.Double.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/Line2D.html)    [**NO FRAMES**](http://docs.google.com/Line2D.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

## **java.awt.geom**

Class Line2D

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.awt.geom.Line2D**

**All Implemented Interfaces:** [Shape](http://docs.google.com/java/awt/Shape.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [Line2D.Double](http://docs.google.com/java/awt/geom/Line2D.Double.html), [Line2D.Float](http://docs.google.com/java/awt/geom/Line2D.Float.html)

public abstract class **Line2D**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Shape](http://docs.google.com/java/awt/Shape.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

This Line2D represents a line segment in (x,y) coordinate space. This class, like all of the Java 2D API, uses a default coordinate system called *user space* in which the y-axis values increase downward and x-axis values increase to the right. For more information on the user space coordinate system, see the  [Coordinate Systems](http://java.sun.com/j2se/1.3/docs/guide/2d/spec/j2d-intro.fm2.html#61857) section of the Java 2D Programmer's Guide.

This class is only the abstract superclass for all objects that store a 2D line segment. The actual storage representation of the coordinates is left to the subclass.

**Since:** 1.2

| **Nested Class Summary** | |
| --- | --- |
| static class | [**Line2D.Double**](http://docs.google.com/java/awt/geom/Line2D.Double.html)            A line segment specified with double coordinates. |
| static class | [**Line2D.Float**](http://docs.google.com/java/awt/geom/Line2D.Float.html)            A line segment specified with float coordinates. |

| **Constructor Summary** | |
| --- | --- |
| protected | [**Line2D**](http://docs.google.com/java/awt/geom/Line2D.html#Line2D())()            This is an abstract class that cannot be instantiated directly. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/java/awt/geom/Line2D.html#clone())()            Creates a new object of the same class as this object. |
| boolean | [**contains**](http://docs.google.com/java/awt/geom/Line2D.html#contains(double,%20double))(double x, double y)            Tests if a specified coordinate is inside the boundary of this Line2D. |
| boolean | [**contains**](http://docs.google.com/java/awt/geom/Line2D.html#contains(double,%20double,%20double,%20double))(double x, double y, double w, double h)            Tests if the interior of this Line2D entirely contains the specified set of rectangular coordinates. |
| boolean | [**contains**](http://docs.google.com/java/awt/geom/Line2D.html#contains(java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p)            Tests if a given Point2D is inside the boundary of this Line2D. |
| boolean | [**contains**](http://docs.google.com/java/awt/geom/Line2D.html#contains(java.awt.geom.Rectangle2D))([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)            Tests if the interior of this Line2D entirely contains the specified Rectangle2D. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getBounds**](http://docs.google.com/java/awt/geom/Line2D.html#getBounds())()            Returns an integer [Rectangle](http://docs.google.com/java/awt/Rectangle.html) that completely encloses the Shape. |
| abstract  [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getP1**](http://docs.google.com/java/awt/geom/Line2D.html#getP1())()            Returns the start Point2D of this Line2D. |
| abstract  [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getP2**](http://docs.google.com/java/awt/geom/Line2D.html#getP2())()            Returns the end Point2D of this Line2D. |
| [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) | [**getPathIterator**](http://docs.google.com/java/awt/geom/Line2D.html#getPathIterator(java.awt.geom.AffineTransform))([AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at)            Returns an iteration object that defines the boundary of this Line2D. |
| [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) | [**getPathIterator**](http://docs.google.com/java/awt/geom/Line2D.html#getPathIterator(java.awt.geom.AffineTransform,%20double))([AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at, double flatness)            Returns an iteration object that defines the boundary of this flattened Line2D. |
| abstract  double | [**getX1**](http://docs.google.com/java/awt/geom/Line2D.html#getX1())()            Returns the X coordinate of the start point in double precision. |
| abstract  double | [**getX2**](http://docs.google.com/java/awt/geom/Line2D.html#getX2())()            Returns the X coordinate of the end point in double precision. |
| abstract  double | [**getY1**](http://docs.google.com/java/awt/geom/Line2D.html#getY1())()            Returns the Y coordinate of the start point in double precision. |
| abstract  double | [**getY2**](http://docs.google.com/java/awt/geom/Line2D.html#getY2())()            Returns the Y coordinate of the end point in double precision. |
| boolean | [**intersects**](http://docs.google.com/java/awt/geom/Line2D.html#intersects(double,%20double,%20double,%20double))(double x, double y, double w, double h)            Tests if the interior of the Shape intersects the interior of a specified rectangular area. |
| boolean | [**intersects**](http://docs.google.com/java/awt/geom/Line2D.html#intersects(java.awt.geom.Rectangle2D))([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)            Tests if the interior of the Shape intersects the interior of a specified Rectangle2D. |
| boolean | [**intersectsLine**](http://docs.google.com/java/awt/geom/Line2D.html#intersectsLine(double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2)            Tests if the line segment from (x1,y1) to (x2,y2) intersects this line segment. |
| boolean | [**intersectsLine**](http://docs.google.com/java/awt/geom/Line2D.html#intersectsLine(java.awt.geom.Line2D))([Line2D](http://docs.google.com/java/awt/geom/Line2D.html) l)            Tests if the specified line segment intersects this line segment. |
| static boolean | [**linesIntersect**](http://docs.google.com/java/awt/geom/Line2D.html#linesIntersect(double,%20double,%20double,%20double,%20double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2, double x3, double y3, double x4, double y4)            Tests if the line segment from (x1,y1) to (x2,y2) intersects the line segment from (x3,y3) to (x4,y4). |
| double | [**ptLineDist**](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDist(double,%20double))(double px, double py)            Returns the distance from a point to this line. |
| static double | [**ptLineDist**](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDist(double,%20double,%20double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2, double px, double py)            Returns the distance from a point to a line. |
| double | [**ptLineDist**](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDist(java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt)            Returns the distance from a Point2D to this line. |
| double | [**ptLineDistSq**](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDistSq(double,%20double))(double px, double py)            Returns the square of the distance from a point to this line. |
| static double | [**ptLineDistSq**](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDistSq(double,%20double,%20double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2, double px, double py)            Returns the square of the distance from a point to a line. |
| double | [**ptLineDistSq**](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDistSq(java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt)            Returns the square of the distance from a specified Point2D to this line. |
| double | [**ptSegDist**](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDist(double,%20double))(double px, double py)            Returns the distance from a point to this line segment. |
| static double | [**ptSegDist**](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDist(double,%20double,%20double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2, double px, double py)            Returns the distance from a point to a line segment. |
| double | [**ptSegDist**](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDist(java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt)            Returns the distance from a Point2D to this line segment. |
| double | [**ptSegDistSq**](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDistSq(double,%20double))(double px, double py)            Returns the square of the distance from a point to this line segment. |
| static double | [**ptSegDistSq**](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDistSq(double,%20double,%20double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2, double px, double py)            Returns the square of the distance from a point to a line segment. |
| double | [**ptSegDistSq**](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDistSq(java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt)            Returns the square of the distance from a Point2D to this line segment. |
| int | [**relativeCCW**](http://docs.google.com/java/awt/geom/Line2D.html#relativeCCW(double,%20double))(double px, double py)            Returns an indicator of where the specified point (px,py) lies with respect to this line segment. |
| static int | [**relativeCCW**](http://docs.google.com/java/awt/geom/Line2D.html#relativeCCW(double,%20double,%20double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2, double px, double py)            Returns an indicator of where the specified point (px,py) lies with respect to the line segment from (x1,y1) to (x2,y2). |
| int | [**relativeCCW**](http://docs.google.com/java/awt/geom/Line2D.html#relativeCCW(java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p)            Returns an indicator of where the specified Point2D lies with respect to this line segment. |
| abstract  void | [**setLine**](http://docs.google.com/java/awt/geom/Line2D.html#setLine(double,%20double,%20double,%20double))(double x1, double y1, double x2, double y2)            Sets the location of the end points of this Line2D to the specified double coordinates. |
| void | [**setLine**](http://docs.google.com/java/awt/geom/Line2D.html#setLine(java.awt.geom.Line2D))([Line2D](http://docs.google.com/java/awt/geom/Line2D.html) l)            Sets the location of the end points of this Line2D to the same as those end points of the specified Line2D. |
| void | [**setLine**](http://docs.google.com/java/awt/geom/Line2D.html#setLine(java.awt.geom.Point2D,%20java.awt.geom.Point2D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p1, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p2)            Sets the location of the end points of this Line2D to the specified Point2D coordinates. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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)) |

| **Methods inherited from interface java.awt.**[**Shape**](http://docs.google.com/java/awt/Shape.html) |
| --- |
| [getBounds2D](http://docs.google.com/java/awt/Shape.html#getBounds2D()) |

| **Constructor Detail** |
| --- |

### Line2D

protected **Line2D**()

This is an abstract class that cannot be instantiated directly. Type-specific implementation subclasses are available for instantiation and provide a number of formats for storing the information necessary to satisfy the various accessory methods below.

**Since:** 1.2 **See Also:**[Line2D.Float](http://docs.google.com/java/awt/geom/Line2D.Float.html), [Line2D.Double](http://docs.google.com/java/awt/geom/Line2D.Double.html)

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

### getX1

public abstract double **getX1**()

Returns the X coordinate of the start point in double precision.

**Returns:**the X coordinate of the start point of this Line2D object.**Since:** 1.2

### getY1

public abstract double **getY1**()

Returns the Y coordinate of the start point in double precision.

**Returns:**the Y coordinate of the start point of this Line2D object.**Since:** 1.2

### getP1

public abstract [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) **getP1**()

Returns the start Point2D of this Line2D.

**Returns:**the start Point2D of this Line2D.**Since:** 1.2

### getX2

public abstract double **getX2**()

Returns the X coordinate of the end point in double precision.

**Returns:**the X coordinate of the end point of this Line2D object.**Since:** 1.2

### getY2

public abstract double **getY2**()

Returns the Y coordinate of the end point in double precision.

**Returns:**the Y coordinate of the end point of this Line2D object.**Since:** 1.2

### getP2

public abstract [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) **getP2**()

Returns the end Point2D of this Line2D.

**Returns:**the end Point2D of this Line2D.**Since:** 1.2

### setLine

public abstract void **setLine**(double x1,  
 double y1,  
 double x2,  
 double y2)

Sets the location of the end points of this Line2D to the specified double coordinates.

**Parameters:**x1 - the X coordinate of the start pointy1 - the Y coordinate of the start pointx2 - the X coordinate of the end pointy2 - the Y coordinate of the end point**Since:** 1.2

### setLine

public void **setLine**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p1,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p2)

Sets the location of the end points of this Line2D to the specified Point2D coordinates.

**Parameters:**p1 - the start Point2D of the line segmentp2 - the end Point2D of the line segment**Since:** 1.2

### setLine

public void **setLine**([Line2D](http://docs.google.com/java/awt/geom/Line2D.html) l)

Sets the location of the end points of this Line2D to the same as those end points of the specified Line2D.

**Parameters:**l - the specified Line2D**Since:** 1.2

### relativeCCW

public static int **relativeCCW**(double x1,  
 double y1,  
 double x2,  
 double y2,  
 double px,  
 double py)

Returns an indicator of where the specified point (px,py) lies with respect to the line segment from (x1,y1) to (x2,y2). The return value can be either 1, -1, or 0 and indicates in which direction the specified line must pivot around its first end point, (x1,y1), in order to point at the specified point (px,py).

A return value of 1 indicates that the line segment must turn in the direction that takes the positive X axis towards the negative Y axis. In the default coordinate system used by Java 2D, this direction is counterclockwise.

A return value of -1 indicates that the line segment must turn in the direction that takes the positive X axis towards the positive Y axis. In the default coordinate system, this direction is clockwise.

A return value of 0 indicates that the point lies exactly on the line segment. Note that an indicator value of 0 is rare and not useful for determining colinearity because of floating point rounding issues.

If the point is colinear with the line segment, but not between the end points, then the value will be -1 if the point lies "beyond (x1,y1)" or 1 if the point lies "beyond (x2,y2)".

**Parameters:**x1 - the X coordinate of the start point of the specified line segmenty1 - the Y coordinate of the start point of the specified line segmentx2 - the X coordinate of the end point of the specified line segmenty2 - the Y coordinate of the end point of the specified line segmentpx - the X coordinate of the specified point to be compared with the specified line segmentpy - the Y coordinate of the specified point to be compared with the specified line segment **Returns:**an integer that indicates the position of the third specified coordinates with respect to the line segment formed by the first two specified coordinates.**Since:** 1.2

### relativeCCW

public int **relativeCCW**(double px,  
 double py)

Returns an indicator of where the specified point (px,py) lies with respect to this line segment. See the method comments of [relativeCCW(double, double, double, double, double, double)](http://docs.google.com/java/awt/geom/Line2D.html#relativeCCW(double,%20double,%20double,%20double,%20double,%20double)) to interpret the return value.

**Parameters:**px - the X coordinate of the specified point to be compared with this Line2Dpy - the Y coordinate of the specified point to be compared with this Line2D **Returns:**an integer that indicates the position of the specified coordinates with respect to this Line2D**Since:** 1.2 **See Also:**[relativeCCW(double, double, double, double, double, double)](http://docs.google.com/java/awt/geom/Line2D.html#relativeCCW(double,%20double,%20double,%20double,%20double,%20double))

### relativeCCW

public int **relativeCCW**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p)

Returns an indicator of where the specified Point2D lies with respect to this line segment. See the method comments of [relativeCCW(double, double, double, double, double, double)](http://docs.google.com/java/awt/geom/Line2D.html#relativeCCW(double,%20double,%20double,%20double,%20double,%20double)) to interpret the return value.

**Parameters:**p - the specified Point2D to be compared with this Line2D **Returns:**an integer that indicates the position of the specified Point2D with respect to this Line2D**Since:** 1.2 **See Also:**[relativeCCW(double, double, double, double, double, double)](http://docs.google.com/java/awt/geom/Line2D.html#relativeCCW(double,%20double,%20double,%20double,%20double,%20double))

### linesIntersect

public static boolean **linesIntersect**(double x1,  
 double y1,  
 double x2,  
 double y2,  
 double x3,  
 double y3,  
 double x4,  
 double y4)

Tests if the line segment from (x1,y1) to (x2,y2) intersects the line segment from (x3,y3) to (x4,y4).

**Parameters:**x1 - the X coordinate of the start point of the first specified line segmenty1 - the Y coordinate of the start point of the first specified line segmentx2 - the X coordinate of the end point of the first specified line segmenty2 - the Y coordinate of the end point of the first specified line segmentx3 - the X coordinate of the start point of the second specified line segmenty3 - the Y coordinate of the start point of the second specified line segmentx4 - the X coordinate of the end point of the second specified line segmenty4 - the Y coordinate of the end point of the second specified line segment **Returns:**true if the first specified line segment and the second specified line segment intersect each other; false otherwise.**Since:** 1.2

### intersectsLine

public boolean **intersectsLine**(double x1,  
 double y1,  
 double x2,  
 double y2)

Tests if the line segment from (x1,y1) to (x2,y2) intersects this line segment.

**Parameters:**x1 - the X coordinate of the start point of the specified line segmenty1 - the Y coordinate of the start point of the specified line segmentx2 - the X coordinate of the end point of the specified line segmenty2 - the Y coordinate of the end point of the specified line segment **Returns:** if this line segment and the specified line segment intersect each other; false otherwise.**Since:** 1.2

### intersectsLine

public boolean **intersectsLine**([Line2D](http://docs.google.com/java/awt/geom/Line2D.html) l)

Tests if the specified line segment intersects this line segment.

**Parameters:**l - the specified Line2D **Returns:**true if this line segment and the specified line segment intersect each other; false otherwise.**Since:** 1.2

### ptSegDistSq

public static double **ptSegDistSq**(double x1,  
 double y1,  
 double x2,  
 double y2,  
 double px,  
 double py)

Returns the square of the distance from a point to a line segment. The distance measured is the distance between the specified point and the closest point between the specified end points. If the specified point intersects the line segment in between the end points, this method returns 0.0.

**Parameters:**x1 - the X coordinate of the start point of the specified line segmenty1 - the Y coordinate of the start point of the specified line segmentx2 - the X coordinate of the end point of the specified line segmenty2 - the Y coordinate of the end point of the specified line segmentpx - the X coordinate of the specified point being measured against the specified line segmentpy - the Y coordinate of the specified point being measured against the specified line segment **Returns:**a double value that is the square of the distance from the specified point to the specified line segment.**Since:** 1.2 **See Also:**[ptLineDistSq(double, double, double, double, double, double)](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDistSq(double,%20double,%20double,%20double,%20double,%20double))

### ptSegDist

public static double **ptSegDist**(double x1,  
 double y1,  
 double x2,  
 double y2,  
 double px,  
 double py)

Returns the distance from a point to a line segment. The distance measured is the distance between the specified point and the closest point between the specified end points. If the specified point intersects the line segment in between the end points, this method returns 0.0.

**Parameters:**x1 - the X coordinate of the start point of the specified line segmenty1 - the Y coordinate of the start point of the specified line segmentx2 - the X coordinate of the end point of the specified line segmenty2 - the Y coordinate of the end point of the specified line segmentpx - the X coordinate of the specified point being measured against the specified line segmentpy - the Y coordinate of the specified point being measured against the specified line segment **Returns:**a double value that is the distance from the specified point to the specified line segment.**Since:** 1.2 **See Also:**[ptLineDist(double, double, double, double, double, double)](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDist(double,%20double,%20double,%20double,%20double,%20double))

### ptSegDistSq

public double **ptSegDistSq**(double px,  
 double py)

Returns the square of the distance from a point to this line segment. The distance measured is the distance between the specified point and the closest point between the current line's end points. If the specified point intersects the line segment in between the end points, this method returns 0.0.

**Parameters:**px - the X coordinate of the specified point being measured against this line segmentpy - the Y coordinate of the specified point being measured against this line segment **Returns:**a double value that is the square of the distance from the specified point to the current line segment.**Since:** 1.2 **See Also:**[ptLineDistSq(double, double)](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDistSq(double,%20double))

### ptSegDistSq

public double **ptSegDistSq**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt)

Returns the square of the distance from a Point2D to this line segment. The distance measured is the distance between the specified point and the closest point between the current line's end points. If the specified point intersects the line segment in between the end points, this method returns 0.0.

**Parameters:**pt - the specified Point2D being measured against this line segment. **Returns:**a double value that is the square of the distance from the specified Point2D to the current line segment.**Since:** 1.2 **See Also:**[ptLineDistSq(Point2D)](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDistSq(java.awt.geom.Point2D))

### ptSegDist

public double **ptSegDist**(double px,  
 double py)

Returns the distance from a point to this line segment. The distance measured is the distance between the specified point and the closest point between the current line's end points. If the specified point intersects the line segment in between the end points, this method returns 0.0.

**Parameters:**px - the X coordinate of the specified point being measured against this line segmentpy - the Y coordinate of the specified point being measured against this line segment **Returns:**a double value that is the distance from the specified point to the current line segment.**Since:** 1.2 **See Also:**[ptLineDist(double, double)](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDist(double,%20double))

### ptSegDist

public double **ptSegDist**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt)

Returns the distance from a Point2D to this line segment. The distance measured is the distance between the specified point and the closest point between the current line's end points. If the specified point intersects the line segment in between the end points, this method returns 0.0.

**Parameters:**pt - the specified Point2D being measured against this line segment **Returns:**a double value that is the distance from the specified Point2D to the current line segment.**Since:** 1.2 **See Also:**[ptLineDist(Point2D)](http://docs.google.com/java/awt/geom/Line2D.html#ptLineDist(java.awt.geom.Point2D))

### ptLineDistSq

public static double **ptLineDistSq**(double x1,  
 double y1,  
 double x2,  
 double y2,  
 double px,  
 double py)

Returns the square of the distance from a point to a line. The distance measured is the distance between the specified point and the closest point on the infinitely-extended line defined by the specified coordinates. If the specified point intersects the line, this method returns 0.0.

**Parameters:**x1 - the X coordinate of the start point of the specified liney1 - the Y coordinate of the start point of the specified linex2 - the X coordinate of the end point of the specified liney2 - the Y coordinate of the end point of the specified linepx - the X coordinate of the specified point being measured against the specified linepy - the Y coordinate of the specified point being measured against the specified line **Returns:**a double value that is the square of the distance from the specified point to the specified line.**Since:** 1.2 **See Also:**[ptSegDistSq(double, double, double, double, double, double)](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDistSq(double,%20double,%20double,%20double,%20double,%20double))

### ptLineDist

public static double **ptLineDist**(double x1,  
 double y1,  
 double x2,  
 double y2,  
 double px,  
 double py)

Returns the distance from a point to a line. The distance measured is the distance between the specified point and the closest point on the infinitely-extended line defined by the specified coordinates. If the specified point intersects the line, this method returns 0.0.

**Parameters:**x1 - the X coordinate of the start point of the specified liney1 - the Y coordinate of the start point of the specified linex2 - the X coordinate of the end point of the specified liney2 - the Y coordinate of the end point of the specified linepx - the X coordinate of the specified point being measured against the specified linepy - the Y coordinate of the specified point being measured against the specified line **Returns:**a double value that is the distance from the specified point to the specified line.**Since:** 1.2 **See Also:**[ptSegDist(double, double, double, double, double, double)](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDist(double,%20double,%20double,%20double,%20double,%20double))

### ptLineDistSq

public double **ptLineDistSq**(double px,  
 double py)

Returns the square of the distance from a point to this line. The distance measured is the distance between the specified point and the closest point on the infinitely-extended line defined by this Line2D. If the specified point intersects the line, this method returns 0.0.

**Parameters:**px - the X coordinate of the specified point being measured against this linepy - the Y coordinate of the specified point being measured against this line **Returns:**a double value that is the square of the distance from a specified point to the current line.**Since:** 1.2 **See Also:**[ptSegDistSq(double, double)](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDistSq(double,%20double))

### ptLineDistSq

public double **ptLineDistSq**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt)

Returns the square of the distance from a specified Point2D to this line. The distance measured is the distance between the specified point and the closest point on the infinitely-extended line defined by this Line2D. If the specified point intersects the line, this method returns 0.0.

**Parameters:**pt - the specified Point2D being measured against this line **Returns:**a double value that is the square of the distance from a specified Point2D to the current line.**Since:** 1.2 **See Also:**[ptSegDistSq(Point2D)](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDistSq(java.awt.geom.Point2D))

### ptLineDist

public double **ptLineDist**(double px,  
 double py)

Returns the distance from a point to this line. The distance measured is the distance between the specified point and the closest point on the infinitely-extended line defined by this Line2D. If the specified point intersects the line, this method returns 0.0.

**Parameters:**px - the X coordinate of the specified point being measured against this linepy - the Y coordinate of the specified point being measured against this line **Returns:**a double value that is the distance from a specified point to the current line.**Since:** 1.2 **See Also:**[ptSegDist(double, double)](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDist(double,%20double))

### ptLineDist

public double **ptLineDist**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt)

Returns the distance from a Point2D to this line. The distance measured is the distance between the specified point and the closest point on the infinitely-extended line defined by this Line2D. If the specified point intersects the line, this method returns 0.0.

**Parameters:**pt - the specified Point2D being measured **Returns:**a double value that is the distance from a specified Point2D to the current line.**Since:** 1.2 **See Also:**[ptSegDist(Point2D)](http://docs.google.com/java/awt/geom/Line2D.html#ptSegDist(java.awt.geom.Point2D))

### contains

public boolean **contains**(double x,  
 double y)

Tests if a specified coordinate is inside the boundary of this Line2D. This method is required to implement the [Shape](http://docs.google.com/java/awt/Shape.html) interface, but in the case of Line2D objects it always returns false since a line contains no area.

**Specified by:**[contains](http://docs.google.com/java/awt/Shape.html#contains(double,%20double)) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Parameters:**x - the X coordinate of the specified point to be testedy - the Y coordinate of the specified point to be tested **Returns:**false because a Line2D contains no area.**Since:** 1.2

### contains

public boolean **contains**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) p)

Tests if a given Point2D is inside the boundary of this Line2D. This method is required to implement the [Shape](http://docs.google.com/java/awt/Shape.html) interface, but in the case of Line2D objects it always returns false since a line contains no area.

**Specified by:**[contains](http://docs.google.com/java/awt/Shape.html#contains(java.awt.geom.Point2D)) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Parameters:**p - the specified Point2D to be tested **Returns:**false because a Line2D contains no area.**Since:** 1.2

### intersects

public boolean **intersects**(double x,  
 double y,  
 double w,  
 double h)

Tests if the interior of the Shape intersects the interior of a specified rectangular area. The rectangular area is considered to intersect the Shape if any point is contained in both the interior of the Shape and the specified rectangular area.

The Shape.intersects() method allows a Shape implementation to conservatively return true when:

* there is a high probability that the rectangular area and the Shape intersect, but
* the calculations to accurately determine this intersection are prohibitively expensive.

This means that for some Shapes this method might return true even though the rectangular area does not intersect the Shape. The [Area](http://docs.google.com/java/awt/geom/Area.html) class performs more accurate computations of geometric intersection than most Shape objects and therefore can be used if a more precise answer is required.

**Specified by:**[intersects](http://docs.google.com/java/awt/Shape.html#intersects(double,%20double,%20double,%20double)) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Parameters:**x - the X coordinate of the upper-left corner of the specified rectangular areay - the Y coordinate of the upper-left corner of the specified rectangular areaw - the width of the specified rectangular areah - the height of the specified rectangular area **Returns:**true if the interior of the Shape and the interior of the rectangular area intersect, or are both highly likely to intersect and intersection calculations would be too expensive to perform; false otherwise.**Since:** 1.2 **See Also:**[Area](http://docs.google.com/java/awt/geom/Area.html)

### intersects

public boolean **intersects**([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)

Tests if the interior of the Shape intersects the interior of a specified Rectangle2D. The Shape.intersects() method allows a Shape implementation to conservatively return true when:

* there is a high probability that the Rectangle2D and the Shape intersect, but
* the calculations to accurately determine this intersection are prohibitively expensive.

This means that for some Shapes this method might return true even though the Rectangle2D does not intersect the Shape. The [Area](http://docs.google.com/java/awt/geom/Area.html) class performs more accurate computations of geometric intersection than most Shape objects and therefore can be used if a more precise answer is required.

**Specified by:**[intersects](http://docs.google.com/java/awt/Shape.html#intersects(java.awt.geom.Rectangle2D)) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Parameters:**r - the specified Rectangle2D **Returns:**true if the interior of the Shape and the interior of the specified Rectangle2D intersect, or are both highly likely to intersect and intersection calculations would be too expensive to perform; false otherwise.**Since:** 1.2 **See Also:**[Shape.intersects(double, double, double, double)](http://docs.google.com/java/awt/Shape.html#intersects(double,%20double,%20double,%20double))

### contains

public boolean **contains**(double x,  
 double y,  
 double w,  
 double h)

Tests if the interior of this Line2D entirely contains the specified set of rectangular coordinates. This method is required to implement the Shape interface, but in the case of Line2D objects it always returns false since a line contains no area.

**Specified by:**[contains](http://docs.google.com/java/awt/Shape.html#contains(double,%20double,%20double,%20double)) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Parameters:**x - the X coordinate of the upper-left corner of the specified rectangular areay - the Y coordinate of the upper-left corner of the specified rectangular areaw - the width of the specified rectangular areah - the height of the specified rectangular area **Returns:**false because a Line2D contains no area.**Since:** 1.2 **See Also:**[Area](http://docs.google.com/java/awt/geom/Area.html), [Shape.intersects(double, double, double, double)](http://docs.google.com/java/awt/Shape.html#intersects(double,%20double,%20double,%20double))

### contains

public boolean **contains**([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)

Tests if the interior of this Line2D entirely contains the specified Rectangle2D. This method is required to implement the Shape interface, but in the case of Line2D objects it always returns false since a line contains no area.

**Specified by:**[contains](http://docs.google.com/java/awt/Shape.html#contains(java.awt.geom.Rectangle2D)) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Parameters:**r - the specified Rectangle2D to be tested **Returns:**false because a Line2D contains no area.**Since:** 1.2 **See Also:**[Shape.contains(double, double, double, double)](http://docs.google.com/java/awt/Shape.html#contains(double,%20double,%20double,%20double))

### getBounds

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getBounds**()

Returns an integer [Rectangle](http://docs.google.com/java/awt/Rectangle.html) that completely encloses the Shape. Note that there is no guarantee that the returned Rectangle is the smallest bounding box that encloses the Shape, only that the Shape lies entirely within the indicated Rectangle. The returned Rectangle might also fail to completely enclose the Shape if the Shape overflows the limited range of the integer data type. The getBounds2D method generally returns a tighter bounding box due to its greater flexibility in representation.

**Specified by:**[getBounds](http://docs.google.com/java/awt/Shape.html#getBounds()) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Returns:**an integer Rectangle that completely encloses the Shape.**Since:** 1.2 **See Also:**[Shape.getBounds2D()](http://docs.google.com/java/awt/Shape.html#getBounds2D())

### getPathIterator

public [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) **getPathIterator**([AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at)

Returns an iteration object that defines the boundary of this Line2D. The iterator for this class is not multi-threaded safe, which means that this Line2D class does not guarantee that modifications to the geometry of this Line2D object do not affect any iterations of that geometry that are already in process.

**Specified by:**[getPathIterator](http://docs.google.com/java/awt/Shape.html#getPathIterator(java.awt.geom.AffineTransform)) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Parameters:**at - the specified [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) **Returns:**a [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) that defines the boundary of this Line2D.**Since:** 1.2

### getPathIterator

public [PathIterator](http://docs.google.com/java/awt/geom/PathIterator.html) **getPathIterator**([AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) at,  
 double flatness)

Returns an iteration object that defines the boundary of this flattened Line2D. The iterator for this class is not multi-threaded safe, which means that this Line2D class does not guarantee that modifications to the geometry of this Line2D object do not affect any iterations of that geometry that are already in process.

**Specified by:**[getPathIterator](http://docs.google.com/java/awt/Shape.html#getPathIterator(java.awt.geom.AffineTransform,%20double)) in interface [Shape](http://docs.google.com/java/awt/Shape.html) **Parameters:**at - the specified AffineTransformflatness - the maximum amount that the control points for a given curve can vary from colinear before a subdivided curve is replaced by a straight line connecting the end points. Since a Line2D object is always flat, this parameter is ignored. **Returns:**a PathIterator that defines the boundary of the flattened Line2D**Since:** 1.2

### clone

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

Creates a new object of the same class as this object.

**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone of this instance. **Throws:** [OutOfMemoryError](http://docs.google.com/java/lang/OutOfMemoryError.html) - if there is not enough memory.**Since:** 1.2 **See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Line2D.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/geom/IllegalPathStateException.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/geom/Line2D.Double.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/geom/Line2D.html)    [**NO FRAMES**](http://docs.google.com/Line2D.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#17dp8vu) |

[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).