Graphics Members

Constructors

Name	Description
Graphics()	Constructs a new **Graphics** object.

Methods

Name	Description
clearRect(int, int, int, int)	Clears the specified rectangle by filling it with the background color of the current drawing surface.
clipRect(int, int, int, int)	Intersects the current clip with the specified rectangle.
copyArea(int, int, int, int, int, int)	Copies an area of the component by a distance specified by **dx** and **dy**.
create()	Creates a new **Graphics** object that is a copy of this **Graphics** object.
create(int, int, int, int)	Creates a new **Graphics** object based on this **Graphics** object, but with a new translation and clip area.
dispose()	Disposes of this graphics context and releases any system resources that it is using.
draw3DRect(int, int, int, int, boolean)	Draws a 3-D highlighted outline of the specified rectangle.
drawArc(int, int, int, int, int, int)	Draws the outline of a circular or elliptical arc covering the specified rectangle.
drawBytes(byte[], int, int, int, int)	Draws the text given by the specified byte array, using this graphics context's current font and color.
drawChars(char[], int, int, int, int)	Draws the text given by the specified character array, using this graphics context's current font and color.
drawImage(Image, int, int, Color, ImageObserver)	Draws as much of the specified image as is currently available.
drawImage(Image, int, int, ImageObserver)	Draws as much of the specified image as is currently available.
drawImage(Image, int, int, int, int, Color, ImageObserver)	Draws as much of the specified image as has already been scaled to fit inside the specified rectangle.
drawImage(Image, int, int, int, int, ImageObserver)	Draws as much of the specified image as has already been scaled to fit inside the specified rectangle.
drawImage(Image, int, int, int, int, int, int, int, int, Color, ImageObserver)	Draws as much of the specified area of the specified image as is currently available, scaling it on the fly to fit inside the specified area of the destination drawable surface.
drawImage(Image, int, int, int, int, int, int, int, int, ImageObserver)	Draws as much of the specified area of the specified image as is currently available, scaling it on the fly to fit inside the specified area of the destination drawable surface.

drawLine(int, int, int, int)	Draws a line, using the current color, between the points **(x1, y1)** and **(x2, y2)** in this graphics context's coordinate system.
drawOval(int, int, int, int)	Draws the outline of an oval.
drawPolygon(int[], int[], int)	Draws a closed polygon defined by arrays of *x* and *y* coordinates.
drawPolygon(Polygon)	Draws the outline of a polygon defined by the specified **Polygon** object.
drawPolyline(int[], int[], int)	Draws a sequence of connected lines defined by arrays of *x* and *y* coordinates.
drawRect(int, int, int, int)	Draws the outline of the specified rectangle.
drawRoundRect(int, int, int, int, int, int)	Draws an outlined round-cornered rectangle using this graphics context's current color.
drawString(String, int, int)	Draws the text given by the specified string, using this graphics context's current font and color.
fill3DRect(int, int, int, int, boolean)	Paints a 3-D highlighted rectangle filled with the current color.
fillArc(int, int, int, int, int, int)	Fills a circular or elliptical arc covering the specified rectangle.
fillOval(int, int, int, int)	Fills an oval bounded by the specified rectangle with the current color.
fillPolygon(int[], int[], int)	Fills a closed polygon defined by arrays of *x* and *y* coordinates.
fillPolygon(Polygon)	Fills the polygon defined by the specified Polygon object with the graphics context's current color.
fillRect(int, int, int, int)	Fills the specified rectangle.
fillRoundRect(int, int, int, int, int, int)	Fills the specified rounded corner rectangle with the current color.
finalize()	Disposes of this graphics context once it is no longer referenced.
getClip()	Gets the current clipping area.
getClipBounds()	Returns the bounding rectangle of the current clipping area.
getClipRect()	**Deprecated.**
getColor()	Gets this graphics context's current color.
getFont()	Gets the current font.
getFontMetrics()	Gets the font metrics of the current font.
getFontMetrics(Font)	Gets the font metrics for the specified font.
setClip(int, int, int, int)	Sets the current clip to the rectangle specified by the given coordinates.
setClip(Shape)	Sets the current clipping area to an arbitrary clip shape.

setColor(Color)	Sets this graphics context's current color to the specified color.
setFont(Font)	Sets this graphics context's font to the specified font.
setPaintMode()	Sets the paint mode of this graphics context to overwrite the destination with this graphics context's current color.
setXORMode(Color)	Sets the paint mode of this graphics context to alternate between this graphics context's current color and the new specified color.
toString()	Returns a **String** object representing this **Graphics** object's value.
translate(int, int)	Translates the origin of the graphics context to the point (*x*, *y*) in the current coordinate system.

Graphics.drawPolygon

Syntax 1

public abstract void drawPolygon(int *xPoints[]***, int** *yPoints[]***, int** *nPoints* **)**

Parameters

xPoints
: a an array of **x** coordinates.

yPoints
: a an array of **y** coordinates.

nPoints
: a the total number of points.

Description

Draws a closed polygon defined by arrays of *x* and *y* coordinates. Each pair of (*x*, *y*) coordinates defines a point.

This method draws the polygon defined by **nPoint** line segments, where the first **nPoint - 1** line segments are line segments from **(xPoints[i - 1], yPoints[i - 1])** to **(xPoints[i], yPoints[i])**, for $1 \leq i \leq$ **nPoints**. The figure is automatically closed by drawing a line connecting the final point to the first point, if those points are different.

Graphics.drawOval

Syntax

public abstract void drawOval(int *x*, **int** *y*, **int** *width*, **int** *height* **)**

Parameters

x
: the *x* coordinate of the upper left corner of the oval to be drawn.

y
: the *y* coordinate of the upper left corner of the oval to be drawn.

width
: the width of the oval to be drawn.

height
: the height of the oval to be drawn.

Description

Draws the outline of an oval. The result is a circle or ellipse that fits within the rectangle specified by the **x**, **y**, **width**, and **height** arguments.

The oval covers an area that is **width + 1** pixels wide and **height + 1 pixels tall.**

Graphics.drawArc

Syntax

public abstract void drawArc(int *x*, **int** *y*, **int** *width*, **int** *height*, **int** *startAngle*, **int** *arcAngle* **)**

Parameters

x
: the *x* coordinate of the upper-left corner of the arc to be drawn.

y
: the *y* coordinate of the upper-left corner of the arc to be drawn.

width
: the width of the arc to be drawn.

height
: the height of the arc to be drawn.

startAngle
: the beginning angle.

arcAngle
: the angular extent of the arc, relative to the start angle.

Description

Draws the outline of a circular or elliptical arc covering the specified rectangle.

The resulting arc begins at **startAngle** and extends for **arcAngle** degrees, using the current color. Angles are interpreted such that 0 degrees is at the 3 o'clock position. A positive value indicates a counter-clockwise rotation while a negative value indicates a clockwise rotation.

The center of the arc is the center of the rectangle whose origin is (*x*, *y*) and whose size is specified by the **width** and **height** arguments.

The resulting arc covers an area **width + 1** pixels wide by **height + 1** pixels tall.