

AWT ARC2D CLASS

http://www.tutorialspoint.com/awt/awt_arc2d_class.htm

Copyright © tutorialspoint.com

Introduction

The Arc2D class is the superclass for all objects that store a 2D arc defined by a framing rectangle, start angle, angular extent *lengthofthearc*, and a closure type *OPEN, CHORD, orPIE*.

Class declaration

Following is the declaration for **java.awt.Arc2D** class:

```
public abstract class Arc2D
    extends RectangularShape
```

Field

Following are the fields for **java.awt.geom.Arc2D** class:

- **static int CHORD** -- The closure type for an arc closed by drawing a straight line segment from the start of the arc segment to the end of the arc segment.
- **static int OPEN** -- The closure type for an open arc with no path segments connecting the two ends of the arc segment.
- **static int PIE** -- The closure type for an arc closed by drawing straight line segments from the start of the arc segment to the center of the full ellipse and from that point to the end of the arc segment.

Class constructors

S.N. Constructor & Description

1

protected Arc2D*inttype*

This is an abstract class that cannot be instantiated directly.

Class methods

S.N. Method & Description

1

boolean contains*doublex, doubley*

Determines whether or not the specified point is inside the boundary of the arc.

2

boolean contains*doublex, doubley, doublew, doubleh*

Determines whether or not the interior of the arc entirely contains the specified rectangle.

3

boolean contains*Rectangle2Dr*

Determines whether or not the interior of the arc entirely contains the specified

rectangle.

4

boolean containsAngle*double angle*

Determines whether or not the specified angle is within the angular extents of the arc.

5

boolean equals*Object obj*

Determines whether or not the specified Object is equal to this Arc2D.

6

abstract double getAngleExtent

Returns the angular extent of the arc.

7

abstract double getAngleStart

Returns the starting angle of the arc.

8

int getArcType

Returns the arc closure type of the arc: OPEN, CHORD, or PIE.

9

Rectangle2D getBounds2D

Returns the high-precision framing rectangle of the arc.

10

Point2D getEndPoint

Returns the ending point of the arc.

11

PathIterator getPathIterator*AffineTransform*

Returns an iteration object that defines the boundary of the arc.

12

Point2D getStartPoint

Returns the starting point of the arc.

13

int hashCode

Returns the hashcode for this Arc2D.

14

boolean intersects*double x, double y, double w, double h*

Determines whether or not the interior of the arc intersects the interior of the specified rectangle.

15

protected abstract Rectangle2D makeBounds*doublex, doubley, doublew, doubleh*

Constructs a Rectangle2D of the appropriate precision to hold the parameters calculated to be the framing rectangle of this arc.

16

abstract void setAngleExtent*doubleangExt*

Sets the angular extent of this arc to the specified double value.

17

void setAngles*doublex1, doubley1, doublex2, doubley2*

Sets the starting angle and angular extent of this arc using two sets of coordinates.

18

void setAngles*Point2Dp1, Point2Dp2*

Sets the starting angle and angular extent of this arc using two points.

19

abstract void setAngleStart*doubleangSt*

Sets the starting angle of this arc to the specified double value.

20

void setAngleStart*Point2Dp*

Sets the starting angle of this arc to the angle that the specified point defines relative to the center of this arc.

21

void setArc*Arc2Da*

Sets this arc to be the same as the specified arc.

22

abstract void setArc*doublex, doubley, doublew, doubleh, doubleangSt, doubleangExt, intclosure*

Sets the location, size, angular extents, and closure type of this arc to the specified double values.

23

void setArc*Point2Dloc, Dimension2Dsize, doubleangSt, doubleangExt, intclosure*

Sets the location, size, angular extents, and closure type of this arc to the specified values.

24

void setArc*Rectangle2Drect, doubleangSt, doubleangExt, intclosure*

Sets the location, size, angular extents, and closure type of this arc to the specified values.

25

void setArcByCenter*doublex, doubley, doubleradius, doubleangSt, doubleangExt, intclosure*

Sets the position, bounds, angular extents, and closure type of this arc to the specified values.

26

void setArcByTangentPoint2Dp1, Point2Dp2, Point2Dp3, doubleradius

Sets the position, bounds, and angular extents of this arc to the specified value.

27

void setArcTypeinttype

Sets the closure type of this arc to the specified value: OPEN, CHORD, or PIE.

28

void setFramedoublex, doubley, doublew, doubleh

Sets the location and size of the framing rectangle of this Shape to the specified rectangular values.

Methods inherited

This class inherits methods from the following classes:

- java.awt.geom.RectangularShape
- java.lang.Object

Arc2D Example

Create the following java program using any editor of your choice in say **D:/ > AWT > com > tutorialspoint > gui >**

AWTGraphicsDemo.java

```
package com.tutorialspoint.gui;

import java.awt.*;
import java.awt.event.*;
import java.awt.geom.*;

public class AWTGraphicsDemo extends Frame {

    public AWTGraphicsDemo(){
        super("Java AWT Examples");
        prepareGUI();
    }

    public static void main(String[] args){
        AWTGraphicsDemo awtGraphicsDemo = new AWTGraphicsDemo();
        awtGraphicsDemo.setVisible(true);
    }

    private void prepareGUI(){
        setSize(400,400);
        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
                System.exit(0);
            }
        });
    }

    @Override
    public void paint(Graphics g) {
        Arc2D.Float arc = new Arc2D.Float(Arc2D.PIE);
        arc.setFrame(70, 200, 150, 150);
        arc.setAngleStart(0);
        arc.setAngleExtent(145);
        Graphics2D g2 = (Graphics2D) g;
```

```
g2.setColor(Color.gray);
g2.draw(arc);
g2.setColor(Color.red);
g2.fill(arc);
g2.setColor(Color.black);
Font font = new Font("Serif", Font.PLAIN, 24);
g2.setFont(font);
g.drawString("Welcome to Tutorialspoint", 50, 70);
g.drawString("Arc2D.PIE", 100, 120);
}
}
```

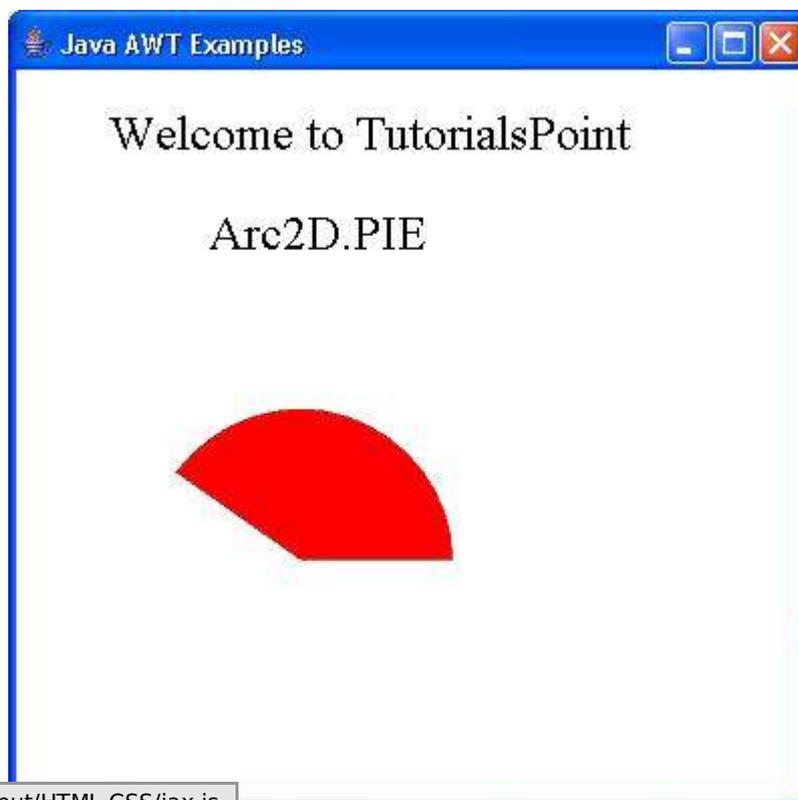
Compile the program using command prompt. Go to **D:/ > AWT** and type the following command.

```
D:\AWT>javac com\tutorialspoint\gui\AwtGraphicsDemo.java
```

If no error comes that means compilation is successful. Run the program using following command.

```
D:\AWT>java com.tutorialspoint.gui.AwtGraphicsDemo
```

Verify the following output



Loading [Mathjax]/jax/output/HTML-CSS/jax.js