

Graphics in awt

java.awt.Component class

The Component class provides 1 method

1. **public void paint(Graphics g):** is used to paint the Applet. It provides Graphics class object that can be used for drawing oval, rectangle, arc etc.

Graphics

java.awt.Graphics class provides many methods for graphics programming.

Commonly used methods of Graphics class:

1. **public abstract void drawString(String str, int x, int y):** is used to draw the specified string.
2. **public void drawRect(int x, int y, int width, int height):** draws a rectangle with the specified width and height.
3. **public abstract void fillRect(int x, int y, int width, int height):** is used to fill rectangle with the default color and specified width and height.
4. **public abstract void drawOval(int x, int y, int width, int height):** is used to draw oval with the specified width and height.
5. **public abstract void fillOval(int x, int y, int width, int height):** is used to fill oval with the default color and specified width and height.
6. **public abstract void drawLine(int x1, int y1, int x2, int y2):** is used to draw line between the points(x1, y1) and (x2, y2).
7. **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.
8. **public abstract void drawArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used draw a circular or elliptical arc.
9. **public abstract void fillArc(int x, int y, int width, int height, int startAngle, int arcAngle):** is used to fill a circular or elliptical arc.
10. **public abstract void setColor(Color c):** is used to set the graphics current color to the specified color.
11. **public abstract void setFont(Font font):** is used to set the graphics current font to the specified font.

Example

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class GraphicsDemo extends Frame
```

```
{
```

```

GraphicsDemo()

{

    super("Graphics Example");

    setLayout(null);

    addWindowListener(new WindowAdapter()

    {

        public void windowClosing(WindowEvent e)

        {

            System.exit(0);

        }

    });

    setSize(400, 400);

    setVisible(true);

}

public void paint(Graphics g)

{

    g.setColor(Color.red);

    g.drawString("Welcome",50, 50);

    g.drawLine(20,30,20,300);

    g.drawRect(70,100,60,30);

    g.fillRect(170,100,30,30);

    g.drawOval(70,200,30,60);

```

```

        g.setColor(Color.blue);

        g.fillOval(170,200,30,30);

        g.drawArc(90,150,30,30,30,270);

        g.fillArc(270,150,30,30,0,180);

    }

    public static void main(String args[])

    {

        GraphicsDemo gd = new GraphicsDemo();

    }

}

```

Displaying Image in AWT

The java.awt.Graphics class provide a method drawImage() to display the image.

Syntax of drawImage() method:

1. **public abstract boolean drawImage(Image img, int x, int y, ImageObserver observer):** is used draw the specified image.

Toolkit

This class is the abstract superclass of all actual implementations of the Abstract Window Toolkit. Subclasses of the `Toolkit` class are used to bind the various components to particular native toolkit implementations

Example

```

import java.awt.*;
import java.awt.event.*;
public class DisplayImage extends Frame
{
    DisplayImage()
    {
        super("Graphics Example");
        setLayout(null);
        addWindowListener(new WindowAdapter()

```

```

        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
        setSize(400, 400);
        setVisible(true);
    }
    public void paint(Graphics g)
    {
        Toolkit t=Toolkit.getDefaultToolkit();
        Image i=t.getImage("Desert.jpg");
        g.drawImage(i, 120,100,this);
    }
    public static void main(String args[])
    {
        DisplayImage di = new DisplayImage();
    }
}

```

In the above example, drawImage() method of Graphics class is used to display the image. The 4th argument of drawImage() method of is ImageObserver object. The Component class implements ImageObserver interface. So current class object would also be treated as ImageObserver.

Animation

For this animation image is required to be moved.

Example of animation in awt:

```

import java.awt.*;
import java.awt.event.*;
public class AnimationExample extends Frame
{
    AnimationExample()
    {
        super("Animation Example");
        setLayout(null);
        addWindowListener(new WindowAdapter()
        {

```

```

        public void windowClosing(WindowEvent e)
        {
            System.exit(0);
        }
    });
    setSize(400, 400);
    setVisible(true);
}
public void paint(Graphics g)
{
    Toolkit t=Toolkit.getDefaultToolkit();
    Image i=t.getImage("bike.jpg");
    for(int j=0;j<500;j++)
    {
        g.drawImage(i, j,30,this);
        try
        {
            Thread.sleep(20);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
}
public static void main(String args[])
{
    AnimationExample ae = new AnimationExample();
}
}

```