Drawing Windows

Introduction

In this unit we are going to cover how to build drawing windows.

JFrame Class:

The JFrame class is used to create a window. We will extend this class to create our windows. Below is an overview of the JFrame class.

Constructors:

Constructor	Description
JFrame()	Creates a frame with no title
JFrame(String name)	Creates a frame with a title equal receive name (String)

Methods:

Return Type	Name	Description
void	setSize(int width, int height)	Sets the size of the window to the given width and height
void	setPreferredSize (Dimension d)	Sets the preferred size of the window or sets it back to its defaults if null is sent.
Insets	getInsets()	Returns the insets of the container. The insets are the space used by the title bar and borders.
int	getWidth()	Returns the frame's width.
int	getHeight()	Returns the frame's height.
void	pack()	Causes the window to be sized based on its preferred size.
void	setVisible(boolean show)	Makes the window hidden when show is false or visible when show is true
void	setLayout(LayoutManager manager)	Sets the window's layout manager to the received manager. If null is received no layout manager is used.
void	add(Component c)	Adds the given component to the frames content pane.
void	remove(Component c)	Removes the given component from the frames content pane.
void	paint(Graphics g)	Paints the container and its components.
void	setDefaultCloseOperation(int operation)	This method is used to set what happens when the x of your frame is clicked. We will always set it to JFrame.EXIT_ON_CLOSE. This value will make the x close down your program.
void	setUndecorated (Boolean hide)	When hide is true the frame's boarder and title bar will be hidden, they will be visible when hide is false

Creating a Window Class:

1. **Create a class that extends JFrame**
2. **Create a constructor**
    a. **In the constructor do the following:**
        i. **Call super and set the name of the window**
        ii. **Set the default close operation**
        iii. **Set resizable**
        iv. **Set size**
        v. **Set visible**

Example:

```
import java.awt.*;
import javax.swing.*;

public class DrawingWindow extends JFrame
{
    public DrawingWindow()
    {
        super("First Window");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setResizable(false);
        setSize(400, 400);
        setVisible(true);
    }
}
```

Example Output:

If another file were to create a drawing window, it would produce a 400 by 400 window with the title First Window.

Graphics & Drawing

Now that we have covered how to make a window, we will learn how to draw to the window. To draw to a JFrame override the paint method and then use the Graphics object to draw to the window.

The next page has an overview of the Graphics class.

Methods:

Return Type	Name	Description
void	setColor(Color c)	Changes the drawing color to c.
void	drawArc(int x, int y, int width, int height, int startAngle, int arcAngle)	Draws the outline of a circular or elliptical arc covering the specified rectangle.
void	drawLine(int x1, int y1, int x2, int y2)	Draws a line from (x1,y1) to (x2,y2)
void	drawOval(int x, int y, int width, int height)	Draws the outline of an oval inscribed in an invisible rectangle with a corner at (x,y) and covers the given width/height.
void	drawPolygon(int[] xPoints, int[] yPoints, int nPoints)	Draws the outline of a polygon with points taken from the xPoints and yPoints arrays. nPoints is the length of the arrays.
void	drawPolygon(Polygon p)	Draws the outline of polygon p.
void	drawPolyline(int[] xPoints, int[] yPoints, int nPoints)	Works like drawPolygon, but does not connect the 1st and last points
void	drawRect(int x, int y, int width, int height)	Draws a rectangle with a top left corner at (x,y) and covers the given width/height.
void	drawRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)	Draws a rectangle with rounded corners. The corners are rounded based on arcWidth/arcHeight.
void	fillArc(int x, int y, int width, int height, int startAngle, int arcAngle)	Fills in a circular or elliptical arc covering the specified rectangle.
void	fillOval(int x, int y, int width, int height)	Fills in an oval inscribed in a rectangle with a corner at (x,y) and covers the given width/height.
void	fillPolygon(int[] xPoints, int[] yPoints, int nPoints)	Fills in a polygon with points taken from the xPoints and yPoints arrays. nPoints is the length of the arrays.
void	fillPolygon(Polygon p)	Fills in polygon p.
void	fillRect(int x, int y, int width, int height)	Fills in a rectangle with a top left corner at (x,y) and covers the given width/height.
void	fillRoundRect(int x, int y, int width, int height, int arcWidth, int arcHeight)	Fills in a rectangle with rounded corners. The corners are rounded based on arcWidth/arcHeight.
void	setFont(Font font)	Sets the font to the given font
void	drawString(String str, int x, int y)	Draws the text at (x,y)

Changing the Drawing Color

Method 1 Format:

graphicsObject.setColor(Color.COLOR);

Note: The available colors can be found in the documentation of Color.

Example:

```
g.setColor(Color.RED);        // changes the drawing color to red
```

Method 2 Format:

graphicsObject.setColor(new Color(red, green, blue));

Note: red, green, blue are each int values from 0 to 255

Example:

```
g.setColor(new Color(0,213,230));    // changes the drawing color to teal
```

Drawing Example:

```
import java.awt.*;
import javax.swing.*;

public class XWindow extends JFrame
{
	public XWindow()
	{
		super("X Window");
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
		setResizable(false);
		setSize(400, 400);
		setVisible(true);
	}

	public void paint(Graphics g)
	{
		g.setColor(Color.BLACK);
		g.fillRect(0,0,getWidth(),getHeight());
		g.setColor(Color.RED);
		g.drawLine(0,0,getWidth(),getHeight());
		g.drawLine(getWidth(),0,0,getHeight());
	}
}
```

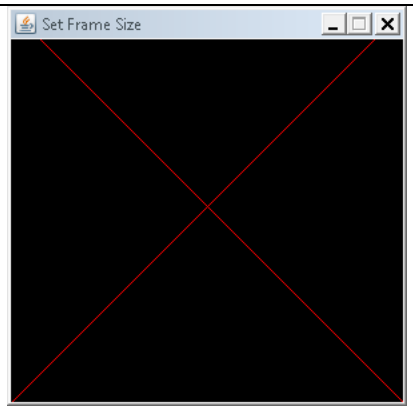
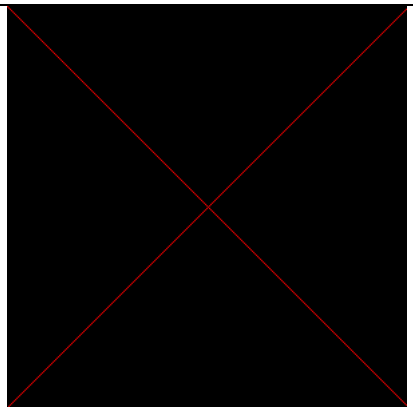
Drawing Output:

If another file were to create a drawing window, it would create a 400 by 400 window with the title "X window". The window would be black with a red X on it.

JPanel Class:

If you ran the last example you may have noticed that the red x did not look right. The reason for this is that (0, 0) of the JFrame is underneath its title bar and even though we cannot see it, it is a valid drawing coordinate.

If we run the example without decorations you will see where the x is drawing to. Below are the two versions of the JFrame with a red x.

Decorated JFrame	Undecorated JFrame
 **Code: Code Examples\Frame Problem B**	 **Code: Code Examples\Frame Problem A**

If you want a decorated JFrame and for (0, 0) to be a visible coordinate then we must create a JFrame with a centered JPanel. Then we draw to the JPanel. Before we get into how to accomplish this, here is an overview on the JPanel class.

Constructors:

Constructor	Description
JPanel()	Creates a JPanel

Methods:

Return Type	Name	Description
void	setSize(int width, int height)	Sets the size of the window to the given width and height
int	getWidth()	Returns the panel's width.
int	getHeight()	Returns the panel's height.
void	add(Component c)	Adds the given component to the panel
void	remove(Component c)	Removes the given component from the panel
void	paint(Graphics g)	Paints the panel and its components.

Decorated Frame/Panel Solutions:

In this section we will go over three procedures for creating a decorated JFrame with a panel.

Creating a frame/panel of with a set frame size:

1. In the main file do the following:
    a. Create an object of your frame
2. In the frame file do the following:
    a. Call super
    b. Set the size
    c. Pack the window
    (sets up the window, so that its insets are known)
    a. Find the Frame Insets
    d. Calculate the amount of room left over for the panel
    (Frame Size – Frame Insets)
    e. Create the panel
    f. Add the panel
    g. Show the window
3. In the panel file do the following:
    a. Call super
    b. Set the size

Code: Code Examples\Set JFrame Size

Creating a frame/panel with a set panel size:

1. In the main file do the following:
    a. Create an object of your frame
2. In the frame file do the following:
    a. Call super
    b. Pack the window
    (sets up the window, so that its insets are known)
    c. Create the panel
    d. Find the Frame Insets
    e. Calculate the amount of room needed for the frame
    (Panel Size + Frame Insets)
    f. Set the Frames preferred size
    g. Add the panel
    h. Pack the window
    (Adjusts the window's size to its preferred size)
    i. Show the window
3. In the panel file do the following:
    a. Call super
    b. Set the size

Code: Code Examples\Set JPanel Size

Creating a maximized Window:

1. In the main file do the following:
    a. Create an object of your frame
2. In the frame file do the following:
    a. Call super
    b. Pack the window
    (sets up the window, so that its insets are known)
    c. Find the Screens Size
    d. Find the Desktop Insets
    e. Calculate the amount of room need for the frame
    (Screen Size - Desktop Insets)
    f. Set the size
    g. Pack the window
    (Adjusts the window's size to its preferred size)
    h. Find the Frame Insets
    i. Calculate the amount of room left over for the panel
    (Frame Size – Frame Insets)
    j. Create the panel
    k. Add the panel
    l. Show the window
3. In the panel file do the following:
    a. Call super
    b. Set the size

Code: Code Examples\Maximized Window

Full Screen Solutions:

In this section we will go over the steps to create a full screen window.

Creating a Full Screen Window with an undecorated Frame:

1. In the main file do the following:
    a. Create an object of your frame
2. In the frame file do the following:
    a. Call super
    b. Set undecorated to true
    c. Find the Screen Size
    d. Set the Frame Size to the Screen Size
    e. Show the window

Code: Code Examples\Full Screen Undecorated

Creating a Full Screen Window with a Given Resolution:

1. In the main file do the following:
    a. Create an object of your frame
2. In the frame file do the following:
    a. Call super
    b. Set full screen mode
    c. Set the display mode
    d. Set undecorated to true
    e. Show the window

Code: Code Examples\Full Screen Your Resolution

Terms

Frame Borders	The thin bars that frame out a window.
Decorated	When the frame's borders and title bar are shown.
Resolution	The number of pixels being display horizontally and vertically.
Title Bar	The bar at the top of a frame that has the title of the window, the minimize button, the maximize button and the close button.
Undecorated	When the frame's borders and title bar are hidden.