**Practical No. 29: Write a program to create animated shape using graphics and applets. You may use following shapes:**

1. **Lines and Rectangles.**
2. **Circles and Ellipses.**
3. **Arcs.**
4. **Polygons with fill Polygon method.**
5. **Practical Significance:**

The graphics class of java provides methods for drawing many different types of shapes. Students will be able to use different methods of graphics programming to draw simple lines, figures of different shapes, images and text in different fonts and styles with different appearance of colors.

1. **Relevant Course Outcome:**

Develop a program using graphics and applet.

1. **Practical Outcome:**

Write a program to create animated shape using graphics and applets. You may use following shapes:

1. Lines and Rectangles.
2. Circles and Ellipses.
3. Arcs.
4. Polygons with fill Polygon method.
5. **Minimum Theoretical Background:**

Every applet has its own area of the screen known as canvas, where it creates its display.

The size of an applet’s space is decided by the attributes of the <APPLET…>tag.

**Drawing Methods of the Graphics Class:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr.**  **No.** | **Method** | **Description** | **Syntax** |
| 1 | drawArc() | Draw a hollow arc | void draw Arc(int top, int left, int width, int heigt, intstartAngle) |
| 2 | drawLine() | Draw a straight line | void drawLine(int startX, int startY, int endX, int endY) |
| 3 | drawOval() | Draw a hollow oval | void drawOval(int top, int left, int width, int height) |
| 4 | drawPolygon() | Polygon | void drawPolygon(int, [], int y[], int numPointer) |
| 5 | drawRect() | Draws a rectangle | void drawRect(int top, int left, int width, int hight) |
| 6 | drawRoundRect() | Draw a hollow rectangle with rounded corners | void drawRoundRect(int top,int left, int width, int height, int Xdiam, intYDiam) |
| 7 | drawString() | Displays a text string | void drawstring(String str, int x, int y) |
| 8 | fillArc() | Draw filled arc | void fillArc(int top, int left, int width, int height, int startAngle, int sweetAngle) |
| 9 | fillOval() | Draws a filled oval | void fillOval(int top, int left, int width, int height) |
| 10 | fillPolygon() | Draws a filled polygon | void fillpolygon(int, [], int y[], int numPointer) |
| 11 | fillRect() | Draws a filled rectangle | void fillRect(int top, int left, int width, int height) |
| 12 | fillRoundRect() | Draws a filled rectangle with rounded corners | void fillRoundRec(int top, int left, int width, int height, int Xdiam, int YDiam) |
| 13 | getColor() | Retrieves the current drawing color | void getColor(String nm) |
| 14 | getFont() | Retrieves the currently used font | void gertFont() |
| 15 | getFontMetrics() | Retrieves the information about the | void getFontMetrics() |
| 16 | setColor() | Sets drawing color | void setColor(Color c) |
| 17 | setFont() | Sets the font | void setFont() |

1. **Program Code:**

import java.awt.\*;

import java.applet.\*;

public class Shapes extends Applet

{

public void paint(Graphics g)

{

g.setFont(new Font("Cambria", Font.BOLD,15));

g.drawString("Different Shapes", 15, 15);

g.drawLine(10,20,50,60);

g.drawRect(10,70,40,40);

g.setColor(Color.ORANGE);

g.fillOval(60,20,30,90);

g.fillArc(60,135,80,40,180,180);

g.fillRoundRect(20,120,60,30,5,5);

}

}

/\*

<applet code="Shapes" width=200 height=200>

</applet>

\*/

1. **Practical Related Questions:**
2. **Differentiate between executing the applet with appletviewer and HTML file.**

An applet requires an HTML file before its execution. The browser, in fact, requires a Java plugin to run an applet. ... Applet programs, on the other hand, are also compiled using the “javac” command but are executed either by using the “appletviewer” command or using the web browser.

1. **Explain methods required to draw different shapes with different colors?**

**Graphics.setColor():**This is the setColor() method which is the Graphics class method imported by the java.awt.\*; package. This method sets the color for the object by specified color. Here is the syntax of the setColor() method :

g.setColor(Color.color\_name);

**Graphics.fillOval():**This is the fillOval() method used to fill the color inside the oval by specified color. Here is the syntax of the fillColor() method :

g.fillColor(Color.color\_name);

**Graphics.fillRect():**  
This is the fillRect() method used to fill the color inside the rectangle by specified color. Here is the syntax of the fillRect() method :

g.fillRect(int X\_coordinate, int Y\_coordinate, int Wdth, int height)

1. **Differentiate between setforeground() and setColor() methods.**

**setcolor()** will set the **pen color of the drawing**which isgoing to be drawn

**Example:**

setColor(Color.BLUE);

drawOval(20,20,100,100);

fillOval(20, 20, 100, 100);

**setforeground():**

Rather setforeground() corresponds to **color of the text**and to be clear it is no way related to the background color.

**Example:**

setForeground(Color.red);

drawString("Hello", 100, 80);

The text  "Hello" will appear in red.

1. **Differentiate between applets and applications.**

Applications are just like a Java programs that can be execute independently without using the web browser. Applets are small Java programs that are designed to be included with the HTML web document. They require a Java-enabled web browser for execution. Applets can only access the browser specific services.

1. **Exercise:**
2. **Develop a program to draw a polygon.**

import java.awt.\*;

import java.applet.\*;

public class polygon extends Applet

{

int a1[]={20, 120, 220, 20};

int b1[]={20, 120, 20, 20};

int n1=4;

int a2[]= {120, 220, 220, 120};

int b2[]= {120, 20, 220, 120};

int n2=4;

public void paint(Graphics g)

{

g.drawPolygon(a1,b1,n1);

g.fillPolygon(a2,b2,n2);

}

}

/\*

<html>

<head>

</head>

<body>

<applet code = "polygon.class" width = "480" height = "360"></applet>

</body>

</html>

\*/

1. **Develop an applet for drawing a human face.**

import java.applet.\*;

import java.awt.\*;

public class Human\_Face extends Applet

{

public void init()

{

setBackground(Color.white);

}

public void paint(Graphics g)

{

Color clr=new Color(255,179,86);

g.setColor(clr);

g.drawOval(100,100,250,300);

g.fillOval(100,100,250,300);

g.setColor(Color.black);

g.drawOval(160,185,40,25);

g.fillOval(160,185,40,25);

g.drawOval(250,185,40,25);

g.fillOval(250,185,40,25);

g.drawArc(160,170,35,10,0,180);

g.drawArc(250,170,35,10,0,180);

g.drawLine(210,265,210,275);

g.drawLine(240,265,240,275);

g.drawArc(210,275,30,10,0,-180);

g.drawArc(175,300,100,50,0,-180);

}

}

/\*

<applet code = Human\_Face.class width=500 height=500>

</applet>

\*/

1. **Write output of the program.**

import java.awt.\*;

import java.applet.\*;

/\*

<applet code="Arcs" width=300 Heigth=300>

<applet>

\*/

public class Acrs extends Applet

{

public void paint(Graphics g)

{

g.drawArc(10,40,70,70,0,75);

g.fillArc(100,40,70,70,0,75);

g.drawArc(10,100,70,80,0,175);

g.fillArc(100,100,70,90,0,270);

g.drawArc(200,80,80,80,0,180);

}

}