Analog Clock

Input :

import java.awt.BasicStroke;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.Graphics2D;

import java.awt.RenderingHints;

import java.awt.Stroke;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Calendar;

import javax.swing.JComponent;

import javax.swing.JFrame;

import javax.swing.Timer;

class Clock extends JFrame

{

Clock()

{

super("Clock");

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setContentPane(new SwingCanvas());

pack();

setVisible(true);

}

public static void main(String[] args)

{

Runnable r = new Runnable()

{

@Override

public void run()

{

new Clock();

}

};

EventQueue.invokeLater(r);

}

}

class SwingCanvas extends JComponent

{

private final static int BORDER\_WIDTH =10;

private BasicStroke bs;

private Calendar cal;

private Dimension d;

private Font font;

private int width;

SwingCanvas()

{

bs = new BasicStroke(2.5f);

cal = Calendar.getInstance();

d = new Dimension(400, 400);

font = new Font("Arial", Font.BOLD, 14);

width = d.width-2\*BORDER\_WIDTH;

ActionListener al;

al = new ActionListener()

{

@Override

public void actionPerformed(ActionEvent ae)

{

cal.setTimeInMillis(System.currentTimeMillis());

repaint();

}

};

new Timer(50, al).start();

}

@Override

public Dimension getPreferredSize()

{

return d;

}

@Override

public void paint(Graphics g)

{

Graphics2D g2d = (Graphics2D) g;

g2d.setRenderingHint(RenderingHints.KEY\_ANTIALIASING,

RenderingHints.VALUE\_ANTIALIAS\_ON);

g2d.translate(BORDER\_WIDTH, BORDER\_WIDTH);

Stroke stroke = g2d.getStroke();

// Paint oval.

g2d.setStroke(bs);

g2d.drawOval(0, 0, width, width);

g2d.setStroke(stroke);

// Paint tick marks.

int tickEnd = width/2;

for (int second = 0; second < 60; second++)

{

int tickStart;

if (second%5 == 0)

tickStart = tickEnd-26; // long tick

else

tickStart = tickEnd-13; // short tick

drawSegment(g2d, second/60.0, tickStart, tickEnd);

}

// Paint hour labels.

g2d.setFont(font);

for (int hour = 1; hour <= 12; hour++)

{

double angle = (hour-3)\*2\*Math.PI/12;

int x = (int) (Math.cos(angle)\*(width/2-35))+width/2-5;

int y = (int) (Math.sin(angle)\*(width/2-35))+width/2+5;

g2d.drawString(""+hour, x, y);

}

// Paint hands.

int hour = cal.get(Calendar.HOUR);

int min = cal.get(Calendar.MINUTE);

int sec = cal.get(Calendar.SECOND);

int ms = cal.get(Calendar.MILLISECOND);

int secHandMaxRad = width/2-5;

double fracSec = (sec+ms/1000.0)/60.0;

g2d.setColor(Color.RED);

drawSegment(g2d, fracSec, 0, secHandMaxRad);

g2d.setColor(Color.BLACK);

int minHandMaxRad = width/3-10;

double fracMin = (min+fracSec)/60.0;

drawSegment(g2d, fracMin, 0, minHandMaxRad);

int hrHandMaxRad = width/4;

drawSegment(g2d, (hour+fracMin)/12.0, 0, hrHandMaxRad);

}

private void drawSegment(Graphics2D g2d, double fraction, int start, int end)

{

double angle = fraction\*Math.PI\*2-Math.PI/2.0;

double \_cos = Math.cos(angle);

double \_sin = Math.sin(angle);

double minx = width/2+\_cos\*start;

double miny = width/2+\_sin\*start;

double maxx = width/2+\_cos\*end;

double maxy = width/2+\_sin\*end;

g2d.drawLine((int) minx, (int) miny, (int) maxx, (int) maxy);

}

}

Output :

