COMPUTER GRAPHICS

NAME : SHRIRANG. R. MHALGI

CLASS : S.E.

DIV : B

ROLL NO : 222006

PROBLEM STATEMENT :

Write a java program to implement patterns which are given on pg 54.

CODE :

package cgg;

import java.awt.Graphics;

import javax.swing.JFrame;

public class Assignment3Pattern1 extends JFrame{

public void dda(Graphics g, int x1, int y1, int x2, int y2){

double length;

double x, y;

double dx = x2 -x1;

double dy = y2 - y1;

if(Double.compare(dx, dy) == 1)

length = dx;

else

length = dy;

dx = (x2 - x1)/length;

dy = (y2 - y1)/length;

/\*System.out.println("length = " +length);

System.out.println("dx = " +dx);

System.out.println("dy = " +dy);\*/

x = x1 + (0.5 \* Integer.signum((int)dx));

y = y1 + (0.5 \* Integer.signum((int)dy));

//g.fillOval((int)x, (int)y, 2, 2);

int i = 1;

while(i <= length){

g.fillOval((int)x, (int)y, 2, 2);

x = x + dx;

y = y + dy;

i++;

}

}

public void bresenham(Graphics g, int x1, int y1, int x2, int y2){

int s1, s2, x, y, i;

double dx, dy, e, temp;

boolean exchange = false;

dx = Math.abs(x2 - x1);

dy = Math.abs(y2 - y1);

x = x1;

y = y1;

i = 1;

e = -1;

do {

g.fillOval(x, y, 2, 2);

while(e >= 0){

y++;

e = e- 2 \* dx;

}

x++;

e = e + 2 \* dy;

i++;

} while (i <= dx);

}

public static void main(String[] args) {

Assignment3Pattern1 obj = new Assignment3Pattern1();

obj.setSize(1024, 1024);

obj.setVisible(true);

obj.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public void paint(Graphics g) {

//dda(g, 400, 200, 100, 400);

bresenham(g, 400, 400, 800, 400);

dda(g, 800, 400, 800, 600);

bresenham(g, 400, 600, 800, 600);

dda(g, 400, 400, 400, 600);

dda(g, 400, 500, 600, 400);

bresenham(g, 400, 500, 600, 600);

dda(g, 600, 600, 800, 500);

bresenham(g, 600, 400, 800, 500);

dda(g, 500, 450, 500, 550);

bresenham(g, 500, 450, 700, 450);

bresenham(g, 500, 550, 700, 550);

dda(g, 700, 450, 700, 550);

}

}

package cgg;

import java.awt.Graphics;

import javax.swing.JFrame;

public class Assignment4Pattern2 extends JFrame{

public void dda(Graphics g, int x1, int y1, int x2, int y2){

double length;

double x, y;

double dx = x2 -x1;

double dy = y2 - y1;

if(Double.compare(dx, dy) == 1)

length = dx;

else

length = dy;

dx = (x2 - x1)/length;

dy = (y2 - y1)/length;

/\*System.out.println("length = " +length);

System.out.println("dx = " +dx);

System.out.println("dy = " +dy);\*/

x = x1 + (0.5 \* Integer.signum((int)dx));

y = y1 + (0.5 \* Integer.signum((int)dy));

//g.fillOval((int)x, (int)y, 2, 2);

int i = 1;

while(i <= length){

g.fillOval((int)x, (int)y, 2, 2);

x = x + dx;

y = y + dy;

i++;

}

}

void bresenhamCircle(Graphics g, int xc, int yc, int r){

//int xc = 0, yc = 0;

int x = 0;

int y = r;

int p = 3 - 2 \* r;

do{

if(p < 0)

p = p + 4 \* x + 6;

else{

p = p + 4 \* (x - y) + 10;

y--;

}

x++;

g.drawLine(xc + x, yc + y, xc + x, yc + y);

g.drawLine(xc + x, yc - y, xc + x, yc - y);

g.drawLine(xc - x, yc + y, xc - x, yc + y);

g.drawLine(xc - x, yc - y, xc - x, yc - y);

g.drawLine(xc + y, yc + x, xc + y, yc + x);

g.drawLine(xc + y, yc - x, xc + y, yc - x);

g.drawLine(xc - y, yc + x, xc - y, yc + x);

g.drawLine(xc - y, yc - x, xc - y, yc - x);

}while(x < y);

}

public static void main(String[] args) {

Assignment4Pattern2 obj1 = new Assignment4Pattern2();

obj1.setSize(2000, 1524);

obj1.setVisible(true);

obj1.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

}

public void paint(Graphics g) {

bresenhamCircle(g, 1200, 885, 115);//radius of incircle = (root 3 \* side of triangle)/6

bresenhamCircle(g, 1200, 885, 231);//radius of circumcircle = (root 3 \* side of triangle)/3

dda(g, 1000, 1000, 1400, 1000);

dda(g, 1000, 1000, 1200, 654);

dda(g, 1200, 654, 1400, 1000);

}

}

OUTPUT :

