COMPUTER GRAPHICS

NAME : SHRIRANG. R. MHALGI

CLASS : S.E.

DIV : B

ROLL NO : 222006

PROBLEM STATEMENT :

Write a java program to implement cohen sutherland algorithm. Use mouse interface to draw the polygon.

CODE :

**package** cgg;

**import** java.awt.\*;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**import** java.util.logging.Level;

**import** java.util.logging.Logger;

**import** javax.swing.JFrame;

**public** **class** CohenSutherland **extends** JFrame{

//to store all input lines

**private** ArrayList<Line> lines=**new** ArrayList();

//to store window coordinates

**private** **int** xw,yw,width,height;

//help in creating outcode

**private** **int** xMin,xMax,yMin,yMax;

**private** **static** **final** **int** ***INSIDE*** = 0;

**private** **static** **final** **int** ***LEFT*** = 1;

**private** **static** **final** **int** ***RIGHT*** = 2;

**private** **static** **final** **int** ***BOTTOM*** = 4;

**private** **static** **final** **int** ***TOP*** = 8;

**public** **void** input(Graphics g){

Scanner scanner=**new** Scanner(System.***in***);

System.***out***.println("Enter the no of lines");

**int** no=scanner.nextInt();

**for**(**int** i=0;i<no;i++){

Line line=**new** Line();

System.***out***.println("Enter coordinates of starting point :-");

System.***out***.print("Enter x : ");

line.setX1(scanner.nextInt());

System.***out***.print("Enter y : ");

line.setY1(scanner.nextInt());

System.***out***.println("Enter coordinates of end point :-");

System.***out***.print("Enter x : ");

line.setX2(scanner.nextInt());

System.***out***.print("Enter y : ");

line.setY2(scanner.nextInt());

lines.add(line);

g.drawLine(line.getX1(), line.getY1(), line.getX2(), line.getY2());

}

/\*System.out.println("Enter coordinates of top left corner of window: -");

System.out.println("Enter x: ");

xw=scanner.nextInt();

System.out.println("Enter y: ");

yw=scanner.nextInt();

System.out.println("Enter width: ");

width=scanner.nextInt();

System.out.println("Enter height: ");

height=scanner.nextInt();\*/

xw=100; yw=100; width=300; height=300;

drawWindow(g, xw , yw, width ,height);

}

**public** **void** output(Graphics g){

drawWindow(g, xw , yw, width ,height);

**for**(Line line:lines)

drawclipedLine(g,line.getX1(),line.getY1(),line.getX2(),line.getY2());

}

**private** **void** drawWindow(Graphics g, **int** xw, **int** yw, **int** width, **int** height) {

g.setColor(Color.***red***);

g.drawRect(xw, yw, width, height);

xMin=xw;

yMax=yw;

xMax=xw+width;

yMin=yw+height;

g.setColor(Color.***BLACK***);

}

**private** **int** computeOutCode(**int** x,**int** y){

**int** outcode=***INSIDE***;

**if**(x<xMin)

outcode= outcode | ***LEFT***;

**else** **if**(x>xMax)

outcode=outcode | ***RIGHT***;

**if**(y > yMin)

outcode=outcode | ***BOTTOM***;

**else** **if**(y < yMax)

outcode= outcode | ***TOP***;

**return** outcode;

}

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

**int** outcode1=computeOutCode(x1,y1);

**int** outcode2=computeOutCode(x2,y2);

**while**(**true**){

**if**((outcode1 | outcode2)==0){

//completely visible

g.drawLine(x1,y1,x2,y2);

**break**;

}

**else** **if**((outcode1 & outcode2)!=0){

//completely invisble

**break**;

}

**else**{

//partially visible or completely invisible

**int** outcode= (outcode1!=0) ? outcode1 :outcode2;

**int** x,y;

**if** ((outcode & ***TOP***) != 0) {

x = x1 + (x2 - x1) \* (yMax - y1) / (y2 - y1);

y = yMax;

}

**else** **if** ((outcode & ***BOTTOM***) != 0) {

x = x1 + (x2 - x1) \* (yMin - y1) / (y2 - y1);

y = yMin;

}

**else** **if** ((outcode & ***RIGHT***) != 0) {

y = y1 + (y2 - y1) \* (xMax - x1) / (x2 - x1);

x = xMax;

}

**else** {

y = y1 + (y2 - y1) \* (xMin - x1) / (x2 - x1);

x = xMin;

}

**if** (outcode == outcode1) {

x1 = x;

y1 = y;

outcode1 = computeOutCode(x1, y1);

} **else** {

x2 = x;

y2 = y;

outcode2 = computeOutCode(x2, y2);

}

}

}

}

@Override

**public** **void** paint(Graphics g){

input(g);

**try** {

Thread.*sleep*(4000);

} **catch** (InterruptedException ex) {

Logger.*getLogger*(CohenSutherland.**class**.getName()).log(Level.***SEVERE***, **null**, ex);

}

g.clearRect(0, 0, 800, 800);

output(g);

}

**public** **static** **void** main(String[] args) {

// **TODO** code application logic here

CohenSutherland a=**new** CohenSutherland();

a.setSize(800,800);

a.setDefaultCloseOperation(JFrame.***EXIT\_ON\_CLOSE***);

a.setVisible(**true**);

}

}

**class** Line {

**private** **int** x1,x2,y1,y2;

**public** **void** setX1(**int** x1){

**this**.x1=x1;

}

**public** **void** setX2(**int** x2){

**this**.x2=x2;

}

**public** **void** setY1(**int** y1){

**this**.y1=y1;

}

**public** **void** setY2(**int** y2){

**this**.y2=y2;

}

**public** **int** getX1(){

**return** x1;

}

**public** **int** getX2(){

**return** x2;

}

**public** **int** getY1(){

**return** y1;

}

**public** **int** getY2(){

**return** y2;

}

}

OUTPUT :

