1.124J Foundations of Software Engineering

Problem Set 5 - Solution

Due Date: Tuesday 10/24/00

Problem 1:[30%]

The provided solution uses AWT ***Applet***. However, use of Swing ***JApplet*** is also an option, probably a better one that someone could have used.

MyPoint.java

```
class MyPoint
{
  double x;
  double y;
  static int numberMyPoints=0;

  MyPoint()
   {
     x = 0.0 ;
     y = 0.0 ;
     numberMyPoints++;
   }

  MyPoint(double x, double y)
   {
     this.x = x;
     this.y = y ;
     numberMyPoints++;
   }

   void move(double dx, double dy)
   {
     x += dx;
```

```
    y += dy ;
   }

  public String toString()
   {
    return ("(x,y) = (" + x + " , " + y + ")");
   }
}
```

ps5_1a.java

```
class ps5_1a
{
  static MyPoint p1, p2;

  public static void main(String args[])
     {
       System.out.println("\n Number of MyPoint objects = "
      + MyPoint.numberMyPoints);
       p1 = new MyPoint();
       System.out.println("\n Number of MyPoint objects = "
      +MyPoint.numberMyPoints);
   System.out.println("\n p1: " + p1);

       p2 = new MyPoint(-4.6,9.5);
       System.out.println("\n Number of MyPoint objects = "
      + MyPoint.numberMyPoints);
       System.out.println("\n p2 = " + p2);

   p1.move(4.5, 0.7);
   System.out.println("\n p1: " + p1);
     }
}
```

ps5_1b.java

```
import java.applet.Applet;
import java.awt.Graphics;
```

```
public class ps5_1b extends Applet
{
  MyPoint p1, p2;

  public void init()
      {
         p1 = new MyPoint();
        p1.move(4.5, 0.7);
         p2 = new MyPoint(-4.6, 9.5);
      }

  public void paint(Graphics g)
      {
        g.drawString("Number of MyPoint objects = "
                    + MyPoint.numberMyPoints, 40,50);
        g.drawString(" p1:  " + p1 , 40, 100);
        g.drawString(" p2:  " + p2 , 40, 150);
      }
}
```

ps5_1b.html

```
<HTML>

   <HEAD>
   <TITLE> Problem set 5: problem 1c</TITLE>
   </HEAD>

   <BODY>
    <h1> Problem Set 5: Problem 1c
    <APPLET CODE="ps5_1c.class" WIDTH=300 HEIGHT=200 align=center>
    </APPLET>
   </BODY>

</HTML>
```

ps5_1c.java

```
import java.applet.Applet;
import java.awt.Graphics;

public class ps5_1c extends Applet
{
  static MyPoint p1, p2;

  public void init()
      {
         p1 = new MyPoint();
         p2 = new MyPoint(-4.6, 9.5);
         p1.move(4.5, 0.7);
      }

  public void paint(Graphics g)
      {
        g.drawString("Number of MyPoint objects = "
                    + MyPoint.numberMyPoints, 40,50);
           g.drawString(" p1:  " + p1 , 40, 100);
           g.drawString(" p2:  " + p2 , 40, 150);
      }

  public static void main(String args[])
      {
        System.out.println("\n Number of MyPoint objects = "
                                  + MyPoint.numberMyPoints);
         p1 = new MyPoint();
         System.out.println("\n Number of MyPoint objects = "
                                    +MyPoint.numberMyPoints);
         System.out.println("\n p1: " + p1);

         p2 = new MyPoint(-4.6,9.5);
         System.out.println("\n Number of MyPoint objects = "
                                  + MyPoint.numberMyPoints);
        System.out.println("\n p2 = " + p2);

         p1.move(4.5, 0.7);
         System.out.println("\n p1: " + p1);
      }
}
```

ps5_1c.html

```
<HTML>

    <HEAD>
    <TITLE> Problem set 5: problem 1c</TITLE>
    </HEAD>

    <BODY>
      <h1> Problem Set 5: Problem 1c
      <APPLET CODE="ps5_1c.class" WIDTH=300 HEIGHT=200 align=center>
      </APPLET>
    </BODY>

</HTML>
```

Problem 2:[35%]

ps5_2.java

```
class ps5_2
{
  static final int SIZE = 100;
  static Shape shapes[];


  public static void main(String args[])
      {
        System.out.print("\n Reading the shapes...");
        readShapes();

        System.out.print("\n Printing the shapes...");
        printShapes();

        System.out.print("\n Cleaning-up the shapes...");
        cleanUpShapes();
      }
```

```
static void readShapes()
    {
       shapes = new Shape[SIZE];
       Point p1,p2,p3,p4,p5,p6,p7, p8;

  p1 = new Point(4.1,5.7);
  p2 = new Point(-3.6,-1.2);
  p3 = new Point(2.3,-8.2);
  p4 = new Point(-9.5,3.1);
  p5 = new Point(-5.2,4.2);
  p6 = new Point(-6.2,9.5);
  p7 = new Point(-11.6,8.6);
  p8 = new Point(-9.6, -13.6);

       shapes[Shape.getNumberShapes()] = new Sphere(11);

       ((Sphere)shapes[0]).setRadius(0.25);
       ((Sphere)shapes[0]).setCenter(-6.8,5.3);

       shapes[Shape.getNumberShapes()] = new Triangle(33,p1,p2,p3);

       shapes[Shape.getNumberShapes()] = new Sphere(101);
       shapes[Shape.getNumberShapes()] = new Triangle();
       shapes[Shape.getNumberShapes()] = new Tetrahedron(44,p4,p5,p6,p7);
  shapes[Shape.getNumberShapes()] = new Sphere();
  shapes[Shape.getNumberShapes()-1].setID(147);
  shapes[Shape.getNumberShapes()] = new Tetrahedron();
  shapes[Shape.getNumberShapes()-1].setID(67);
  ((Tetrahedron)shapes[Shape.getNumberShapes()-1]).setVertices(p1,p3,p8,p6);
  shapes[Shape.getNumberShapes()] = new Sphere();
   }


static void printShapes()
    {
        System.out.println("\n Number of shapes: " +
                        Shape.getNumberShapes());

        System.out.print("\n   Number of Spheres: " +
    Sphere.getNumberSpheres());
```

```
  System.out.print("\n   Number of Triangles: " +
                Triangle.getNumberTriangles());
        System.out.println("\n   Number of Tetrahedrons: " +
     Tetrahedron.getNumberTetrahedrons());

          for(int i=0;i<Shape.getNumberShapes();i++)
           {
             System.out.print("\nShapes [ " + (i+1) + " ]: "  );

             if(shapes[i] instanceof Sphere)
               System.out.println("  Sphere " );
             else if(shapes[i] instanceof Triangle)
               System.out.println("  Triangle " );
             else if(shapes[i] instanceof Tetrahedron)
               System.out.println("  Tetrahedron " );

             System.out.println(shapes[i]);
           }
      }

  static void cleanUpShapes()
      {
        System.out.println("\n\n References to shape objects are set to null");
        int n = Shape.getNumberShapes();

        for(int i=0;i<n;i++)
          {
            System.out.print("\nSetting shape [ " + (i+1) + " ]: to null"  );
            shapes[i] = null;
          }

       System.out.println("\n\n Finalizing objects");
       System.runFinalization();

       System.out.println("\n Running the Garbage Collector\n");
       System.gc();
      }
}
```

Shape.java

```
abstract class Shape
{
  private int shapeID;
  private static int numberShapes=0;

  /**************** Constructors ********************/

  Shape()
  {
    shapeID = 0;
 numberShapes++;
  }

  Shape(int id)
  {
    shapeID = id;
 numberShapes++;
  }

  protected void finalize() throws Throwable
        {
           System.out.println("\n\t\t In Shape finalize\n");
           numberShapes --;
           super.finalize();
        }


  /******************  Set methods  ************************/

  void setID(int id)
  {
    shapeID = id;
  }


  /******************** Get methods ************************/
  double getID()
  {
   return shapeID;
  }
```

```
  static int getNumberShapes()
  {
    return numberShapes;
  }


  /******************  toString method  **********************/
   public String toString()
   {
     return " Shape:   ID = " + shapeID;
   }
}
```

Sphere.java

```
class Sphere extends Shape
{
  private Point center;
  private double radius;

  private static int numberSpheres=0;


  /**************** Constructors ********************/

  Sphere()
  {
    super();
    center = new Point();
  radius = 0.0;
    numberSpheres++;
  }

  Sphere(int id)
  {
    super(id);
    center = new Point();
  radius = 0.0;
```

```
    numberSpheres++;
  }

  Sphere(int id, double x, double y, double radius)
  {
    super(id);
    center = new Point(x,y);
  this.radius = radius;
  }

  Sphere(int id, Point p, double radius)
  {
    super(id);
    center = new Point(p);
  this.radius = radius;
  }
  protected void finalize() throws Throwable
        {
           System.out.println("\n    In Sphere finalize");
           numberSpheres --;
           super.finalize();
        }


public void setRadius(double radius)
    {
   this.radius = radius;
    }

 public void setCenter(double x, double y)
    {
     center = new Point(x,y);
    }

  public void setCenter(Point p)
    {
     center = new Point(p);
    }

  static int getNumberSpheres()
  {
    return numberSpheres;
```

```
  }



  /******************  toString method  ************************/
   public String toString()
   {
 return  super.toString()+ "\n\t Radius = " + radius +
   "\n\t Center: " + center ;
   }
}
```

Triangle.java

```
class Triangle extends Shape
{
  private Point a, b, c;

  private static int numberTriangles=0;


  /***************** Constructors ********************/

  Triangle()
  {
    super();
    a = new Point();
    b = new Point();
    c = new Point();
    numberTriangles++;
  }

  Triangle(int id, Point v1, Point v2, Point v3)
  {
    super(id);
  a = new Point(v1);
  b = new Point(v2);
  c = new Point(v3);
    numberTriangles++;

  }
```

```
  protected void finalize() throws Throwable
        {
           System.out.println("\n    In Triangle finalize");
           numberTriangles --;
           super.finalize();
        }


 public void setVertices(Point v1, Point v2, Point v3)
  {
  a = new Point(v1);
  b = new Point(v2);
  c = new Point(v3);
  }


  static int getNumberTriangles()
  {
    return numberTriangles;
  }



  /*******************  toString method  ************************/
   public String toString()
   {
  return  super.toString()+ "\n\t Vertex a: " + a +
       "\n\t Vertex b: " + b + "\n\t Vertex c: " + c;
   }
}
```

Tetrahedron.java

```
class Tetrahedron extends Shape
{
  private Point a, b, c, d;

  private static int numberTetrahedrons=0;
```

```
/***************** Constructors ********************/

Tetrahedron()
{
  super();
  a = new Point();
  b = new Point();
  c = new Point();
  c = new Point();
  numberTetrahedrons++;
}

Tetrahedron(int id, Point v1, Point v2, Point v3, Point v4)
{
  super(id);
 a = new Point(v1);
 b = new Point(v2);
 c = new Point(v3);
 d = new Point(v3);
  numberTetrahedrons++;
}

 protected void finalize() throws Throwable
       {
          System.out.println("\n    In Tetrahedron finalize");
          numberTetrahedrons --;
          super.finalize();
       }


public void setVertices(Point v1, Point v2, Point v3, Point v4)
 {
 a = new Point(v1);
 b = new Point(v2);
 c = new Point(v3);
 d = new Point(v3);
 }


 static int getNumberTetrahedrons()
 {
  return numberTetrahedrons;
```

```
  }



  /******************** toString method ************************/
   public String toString()
   {
  return  super.toString()+ "\n\t Vertex a: " + a +
          "\n\t Vertex b: " + b + "\n\t Vertex c: "
               + c + "\n\t Vertex d: " + d;
   }
}
```

Point.java

```
class Point
{
  double x;
  double y;
  static int numberPoints=0;

  Point()
   {
     x = 0.0 ;
     y = 0.0 ;
     numberPoints++;
   }

  Point(double x, double y)
   {
     this.x = x;
     this.y = y ;
     numberPoints++;
   }

   Point(Point p)
   {
     this.x = p.x;
     this.y = p.y ;
     numberPoints++;
   }
```

```
  public double getX()
 {
   return x;
 }

 public double getY()
 {
   return y;
 }

   public void move(double dx, double dy)
    {
      x += dx;
      y += dy ;
    }

  public String toString()
     {
       return ("(x,y) = (" + x + " , " + y + ")");
     }
}
```

Problem 3:[35%]

ps5_3.java

```
import java.applet.Applet;
import java.awt.Graphics;


public class ps5_3 extends Applet
{
  Rectangle r[];

  public void init()
      {
    r = new Rectangle[10];
```

```
  Point p;

  p = new Point(75,130);
  r[Rectangle.getNumberRectangles()] = new Rectangle(p,40,30);

  p = new Point(100,75);
  r[Rectangle.getNumberRectangles()] = new Rectangle(p,30,25);

  p = new Point(165,155);
  r[Rectangle.getNumberRectangles()] = new Rectangle(p,45,45);

  p = new Point(195,85);
  r[Rectangle.getNumberRectangles()] = new Rectangle(p,30,50);
     }

public void paint(Graphics g)
     {
    Point c = getCentroid();
    g.drawString("Total Area: " + getArea() , 50,20);
    g.drawString("Centroid: " + c, 50,40);

 for(int i=0; i<Rectangle.getNumberRectangles(); i++)
   {
    g.drawRect((int)(r[i].getXc()-r[i].getWidth()/2),
          (int)(r[i].getYc()-r[i].getHeight()/2),
         (int)(r[i].getWidth()),(int)(r[i].getHeight()));

     g.drawString("R"+(i+1),(int)(r[i].getXc()-r[i].getWidth()/4),
           (int)(r[i].getYc()+r[i].getHeight()/4));
   }
  g.fillOval((int)(c.getX()-2),(int)(c.getY()-2),4,4);
    }


double getArea()
{
 double sa=0.0;

 for(int i=0; i<Rectangle.getNumberRectangles(); i++)
       sa += r[i].area();

 return sa;
```

```
  }

  Point getCentroid()
   {
      double x, y, sx, sy, sax, say, sa;
      x = y = sx = sy = sax = say = sa = 0.0;

    for(int i=0; i<Rectangle.getNumberRectangles(); i++)
      {
        sx += r[i].getXc();
        sy += r[i].getYc();
        sa += r[i].area();
        sax += (r[i].area()*r[i].getXc());
        say += (r[i].area()*r[i].getYc());
      }
   return new Point(sax/sa,say/sa);
    }
}
```

Rectangle.java

```
class Rectangle
{
  private Point center;
  private double width, height;

  private static int numberRectangles=0;

  Rectangle(Point c, double w, double h)
  {
    center = new Point(c);
  width = w;
  height = h;
    numberRectangles++;
  }

  protected void finalize() throws Throwable
        {
           numberRectangles --;
           super.finalize();
```

```
        }

  static int getNumberRectangles()
  {
     return numberRectangles;
  }

  public double getXc()
  {
   return center.getX();
  }

  public double getYc()
  {
   return center.getY();
  }

  public double getWidth()
  {
   return width;
  }

  public double getHeight()
  {
   return height;
  }

  public double area()
  {
 return width * height;
  }

}
```

Point.java

```
// This class is used in both problems 2 and 3 of PS5
import java.text.*;
```

```
class Point
{
  double x;
  double y;
  static int numberPoints=0;

  Point()
   {
    x = 0.0 ;
    y = 0.0 ;
    numberPoints++;
   }

  Point(double x, double y)
   {
    this.x = x;
    this.y = y ;
    numberPoints++;
   }

   Point(Point p)
   {
    this.x = p.x;
    this.y = p.y ;
    numberPoints++;
   }

 public double getX()
 {
  return x;
 }

 public double getY()
 {
  return y;
 }

  public void move(double dx, double dy)
   {
    x += dx;
    y += dy ;
   }
```

```
  public String toString()
    {
   DecimalFormat df = new DecimalFormat("##0.##");
     return ("(x,y) = (" + df.format(x) +
    " , " +  df.format(y) + ")");
    }
}
```

ps5_3.html

```
<HTML>

    <HEAD>
    <TITLE> Problem set 5: problem 3</TITLE>
    </HEAD>

    <BODY>
     <h1>
     <APPLET CODE="ps5_3.class" WIDTH=300 HEIGHT=200 align=center>
     </APPLET>
    </BODY>

</HTML>
```

© 1.124J Foundations of Software Engineering