Eftekher Husain
Csc 221 1XD

Hesham A Auda

Excersize 1

The Problem

Solution Method for Class Polygon

a. GetArea method

Since the polygon is a regular polygon and the sides and angles are equal, the area of this polygon can be stated as...

Area of Polygon = (¼) * N * (sideLength * sideLength) / (tan(pi/N))

Where **N** is the number of sides and **sideLength** is the length of each side. Since we few things are still unknown, we use a general equation.

Codes Developed

```
//returns the area of Polygon
public double getArea()
{
    double area = (1/4) * N * ((sideLength * sideLength)/(Math.tan(Math.PI/N)));
    return area;
}
```

b. GetPerimeter

To get the perimeter of a regular polygon, all we have to do is multiply its length of each sides by number of its side.

Perimeter = N * sideLength.

Codes Developed

```
//returns the Perimeter of Polygon
public int getPerimeter()
{return N * sideLength;}
```

c. getAngle

In general to get the angle of a side of a regular polygon, we would divide the number of sides by 360 degrees. But to get the interior angle of the regular polygon we would use a general equation.

180 degree - (360 degree / N) or (N-2) * 180 degrees / N.

Codes Developed

```
//returns the interior angle (in degrees) of the Polygon
public double getAngle()
{   double angleInterior = (N-2) * 180/N;
    return angleInterior;
}
```

d. getSide

To get the side of the regular polygon, we first have to declare a variable called side and then we call it through the constructor. Then we use the getSide method which then reuturns the side length of the polygon.

Codes Developed

```
    private int sideLength;


    public Polygon(Color color, int N, int sideLength) {
        super(color);
         this.N = N;
         this.sideLength = sideLength;
    }


//returns the sideLength length of the Polygon
public int getsideLength()
{return sideLength;}
```

e. draw

Codes Developed

Circle Class

1. getArea

To get the area of a circle, we use the general formula of the area of a circle which is …

Area of a cirlce = Pi * radius * radius

Since Java provides Pi from it's Math interface, we use it to get our area for the circle

Codes Developed

```
//returns the area of circle
public double getArea()
{return Math.PI * radius * radius;}
```

2. getPerimeter

To get the perimeter of a circle, we use the general formula of perimeter of a circle which is…

Perimeter of a circle = 2 * Pi * radius

Codes Developed

```
//returns the perimeter of circle
public double getPerimeter()
{return 2 * Math.PI * radius;}
```

3. getRadius

to get the radius of a circle we first have to declare it in our program. Once declared then we call it in our constructor. And then we call it in our getRadius method as the field of out return.

Codes developed

```
private double radius;

public Circle(double x, double y, Color color, double r) {
    super(x, y, color);
    radius = r;
}


public double getRadius()
{return radius;}
```

4. toString

   What the toString method does is print out the objects description as a String. For a circle it will return a string representation of circle object which are radius, perimeter and area.

   Codes developed

```
public String toString()
{   String result = "Circle[ Radius=" +getRadius()+
", Perimeter=" + getPerimeter()+ ", Area = " +getArea()+ "]";
    return result;
}
```

5. draw

   This method will draw the actual circle with a color and according to it's parameter.

   Codes developed for Circle class

```
import java.awt.Color;

public class Circle extends Shape
{
    private double radius;

    public Circle(int x, int y, Color color, double radius) {
        super(x, y, color);
    }

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


    //get methods
    public int getRadius()
    {return (int) radius;}


    public int getX1()
    {return (int) getX();}

    public int getY1()
    {return (int) getY();}

    //returns the area of circle
    public double getArea()
    {return Math.PI * radius * radius;}

    //returns the perimeter of circle
    public double getPerimeter()
    {return 2 * Math.PI * radius;}



    public String toString()
    {   String result = "Circle[ Radius=" +getRadius()+
    ", Perimeter=" + getPerimeter()+ ", Area = " +getArea()+ "]";
        return result;
    }
```

```
//it paints the drawing canvas in color
  @Override
  public void draw() {
      Circle c = new Circle(this.getX(), this.getY(), this.getColor(), this.getRadius());
      DrawCircle d = new DrawCircle();
      d.setCircle(c);
  }


  public class DrawCircle extends JPanel
  {
      private Circle circle;

      public void setCircle(Circle c)
      {   circle = c; }


      public void painComponent(Graphics g)
      {
          super.paintComponents(g);

          setBackground(Color.WHITE);
          g.setColor(Color.PINK);
          g.fillOval(getRadius() + (circle.getX1()-getRadius()),
                  getRadius() + (circle.getY1()-getRadius()),
                  circle.getRadius() *2,  circle.getRadius() *2);

          g.setColor(Color.GREEN);
          g.fillOval(getRadius() + (circle.getX1()-getRadius()),
                  getRadius() + (circle.getY1()-getRadius()),
                  circle.getRadius() *2,  circle.getRadius() *2);

          g.setColor(Color.BLACK);
          int x1 = getWidth();
          int y1 = getHeight();
          g.drawLine(0, 0, x1, y1);

      }
```

Codes Developed for Shape Class

```
import java.awt.Color;

public abstract class Shape{
    private int x;
    private int y;
    private Color color;
    private int dy;
    private int dx;

    //constructor
    //Point methods
    public Shape(int x2, int y2, Color color) {
        super();
        this.x = x2;
        this.y = y2;
        this.color = color;

    }

    public Shape(Color color) {
        this.color = color;
    }


    //get methods
    public int getX()
    {return x;}

    public int getY()
    {return y;}

    public Color getColor()
    {return color;}

    public int getDy() {
        return dy;
    }

    public int getDx() {
        return dx;
    }
```

```
// set methods
public void setX(int x) {
    this.x = x;
}
public void setY(int y) {
    this.y = y;
}
public void setColor(Color color) {
    this.color = color;
}

public void setDy(int dy) {
    this.dy = dy;
}

public void setDx(int dx) {
    this.dx = dx;
}


//moves point (x, y) by (Δx, Δy);
public void shiftXY(int dx, int dy)
{
    x += dx;
    y =+ dy;
}

//returns the object's description as a String
public String toString()
{   String shape = "Shape[ x =" + getX() + ",y = " + getY() + "color = " + getColor();
    return shape;
}


abstract public void draw();
```

Codes Developed for Polygon Class

```java
import java.awt.Color;

public class Polygon extends Shape {
    private int N;
    private int sideLength;
    private Polygon polygon;

    public Polygon(double x, double y,Color color, int N, int sideLength) {
        super(color);
         this.N = N;
         this.sideLength = sideLength;
    }

    //returns the area of Polygon
    public double getArea()
    {
        double area = (1/4) * N * ((sideLength * sideLength)/(Math.tan(Math.PI/N)));
        return area;
    }

    //returns the Perimeter of Polygon
    public int getPerimeter()
    {return N * sideLength;}

    //returns the interior angle (in degrees) of the Polygon
    public double getAngle()
    {   double angleInterior = (N-2) * 180/N;
        return angleInterior;
    }
```

```
    //returns the Perimeter of Polygon
    public int getPerimeter()
    {return N * sideLength;}


    //returns the interior angle (in degrees) of the Polygon
    public double getAngle()
    {   double angleInterior = (N-2) * 180/N;
        return angleInterior;
    }


    //returns the sideLength length of the Polygon
    public int getsideLength()
    {return sideLength;}


    public String toString() {
        String result = "( side length = " + getsideLength() +
                ", interior angle = " + getAngle()  + ", perimeter = " +
                getPerimeter() + ", area = " + getArea() + ")";
        return result;
    }

  //it paints the drawing canvas in color
@Override
public void draw() {
    // TODO Auto-generated method stub

}
}
```

Codes Developed for main Class