```java
import java.util.*;

public class App {
  static Scanner scanner = new Scanner(System.in);
  static Scanner scannerStr = new Scanner(System.in);

  public static void main(String[] args) throws Exception {
    // Circle
    System.out.print("Creating circle 1, input radius: ");
    double radiusCircleA = scanner.nextDouble();
    System.out.print("Creating circle 2, input radius: ");
    double radiusCircleB = scanner.nextDouble();
    GeometricObject circleA = new Circle(radiusCircleA);
    GeometricObject circleB = new Circle(radiusCircleB);
    System.out.println("-----------------------");
    System.out.printf("The max circle's radius is %.1f\n", max(circleA, circleB).getRadius());
    System.out.println("========================");
    // Rectangle
    System.out.print("Creating rectangle 1, input width and height: ");
    String[] WHRectangleA = scannerStr.nextLine().split(" ");
    System.out.print("Creating rectangle 2, input width and height: ");
    String[] WHRectangleB = scannerStr.nextLine().split(" ");
    GeometricObject RectangleA = new Rectangle(Double.parseDouble(WHRectangleA[0]),
        Double.parseDouble(WHRectangleA[1]));
    GeometricObject RectangleB = new Rectangle(Double.parseDouble(WHRectangleB[0]),
        Double.parseDouble(WHRectangleB[1]));
    max(RectangleA, RectangleB);
    System.out.println("-----------------------");
    System.out.printf("The max rectangle's width and height are %.1f, %.1f\n", max(RectangleA,
RectangleB).getWidth(),
        max(RectangleA, RectangleB).getHeight());
    System.out.println("========================");
  }

  public static GeometricObject max(GeometricObject o1, GeometricObject o2) {
    if (o1.getArea() > o2.getArea()) {
      return o1;
```

```
    } else {
      return o2;
    }
  }

}
```

```
class Circle extends GeometricObject implements Comparable{
    private double radius;
    private String color;
    private boolean filled;

    public Circle(){
        super("Black",false);
        this.radius = 1;
    }

    public Circle(double radius){
        super("Black",false);
        this.radius = radius;
    }

    public Circle(double radius,String color,boolean filled){
        super(color,filled);
        this.radius = radius;
    }
```

```
//getter

public double getRadius(){
    return radius;
}

public double getHeight(){
    return 0;
}

public double getWidth(){
    return 0;
}

public double getArea(){
    return Math.PI * radius * radius;
}

public double getPerimeter(){
    return 2 * Math.PI * radius;
}

//setter

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

```
    @Override
    public int compareTo(Rectangle o){
        if(getArea()>o.getArea()) return 1;
        else if(getArea()==o.getArea()) return 0;
        else return -1;

    }

    @Override
    public String toString(){
        return super.toString() + "\nRadius: " + String.format("%.2f",radius) +

        "\nPerimeter: " + String.format("%.2f",getPerimeter()) +
        "\nArea: " + String.format("%.2f",getArea());
    }
}
```

```
interface Comparable{
    public int compareTo(Rectangle o);
}
```

```
import java.util.Date;

abstract class GeometricObject{
    private String color;
    private boolean filled;
```

```
private Date dateCreated;

protected GeometricObject(){
    this.color = "Black";
    this.filled = false;
    this.dateCreated = new Date();
}

protected GeometricObject(String color,boolean filled){
    this.color = color;
    this.filled = filled;
    this.dateCreated = new Date();
}

//setter

public void setColor(String color){
    this.color = color;
}

public void setFilled(boolean filled){
    this.filled = filled;
}

//getter

public String getColor(){
    return color;
```

```
    }

    public boolean isFilled(){
        return filled;
    }

    public Date getDate(){
        return dateCreated;
    }

    //abstract method
    abstract public double getArea();
    abstract public double getPerimeter();
    abstract public double getRadius();
    abstract public double getWidth();
    abstract public double getHeight();

    @Override
    public String toString(){
        return "Date Created: "+dateCreated+"\nColor: "+color+"\nFilled: "+filled;
    }
}
```

```
class Rectangle extends GeometricObject implements Comparable{
    private double width,height;
    private String color;
    private boolean filled;
```

```java
public Rectangle(){
    super("Black",false);
    this.width = 1;
    this.height = 1;
}

public Rectangle(double width,double height){
    super("Black",false);
    this.width = width;
    this.height = height;
}

public Rectangle(double width,double height,String color,boolean filled){
    super(color,filled);
    this.width = width;
    this.height = height;
}

//getter

public double getRadius(){
    return 0;
}

public double getWidth(){
    return width;
}

public double getHeight(){
```

```
    return height;
}

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

public double getPerimeter(){
    return width * 2 + height * 2;
}

//setter

public void setWidth(double width){
    this.width = width;
}

public void setHeight(double height){
    this.height = height;
}

@Override
public int compareTo(Rectangle o){
    if(getArea()>o.getArea()) return 1;
    else if(getArea()==o.getArea()) return 0;
    else return -1;

}
```

```
    @Override
    public String toString(){
        return super.toString() + "\nWidth: " + String.format("%.2f",width) +
        "\nHeight: " + String.format("%.2f",height) +
        "\nPerimeter: " + String.format("%.2f",getPerimeter()) +
        "\nArea: " + String.format("%.2f",getArea());
    }
}
```