// Define an interface

interface Shape {

double getArea();

double getPerimeter();

}

// Implement the interface in a class

class Circle implements Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double getArea() {

return Math.PI \* radius \* radius;

}

@Override

public double getPerimeter() {

return 2 \* Math.PI \* radius;

}

}

class Rectangle implements Shape {

private double width;

private double height;

public Rectangle(double width, double height) {

this.width = width;

this.height = height;

}

@Override

public double getArea() {

return width \* height;

}

@Override

public double getPerimeter() {

return 2 \* (width + height);

}

}

public class InterfaceExample {

public static void main(String[] args) {

Shape circle = new Circle(5.0);

Shape rectangle = new Rectangle(4.0, 3.0);

// Calculate and display the area and perimeter of the circle

System.out.println("Circle - Area: " + circle.getArea() + ", Perimeter: " + circle.getPerimeter());

// Calculate and display the area and perimeter of the rectangle

System.out.println("Rectangle - Area: " + rectangle.getArea() + ", Perimeter: " + rectangle.getPerimeter());

}

}

In this program:

1. We define an interface called **Shape**, which includes two methods: **getArea()** and **getPerimeter()**.
2. We implement the **Shape** interface in two classes, **Circle** and **Rectangle**.
3. The **Circle** and **Rectangle** classes provide concrete implementations for the **getArea()** and **getPerimeter()** methods.
4. In the **InterfaceExample** class, we create instances of **Circle** and **Rectangle**, and then we call the **getArea()** and **getPerimeter()** methods on them.

This demonstrates how Java interfaces can be used to define a contract that classes must adhere to by implementing the interface's methods.