Assignment-6

Title: Program to Calculate Area of Different Shapes Using Interfaces in Java

Objective:

To implement a Java program that demonstrates the use of interfaces to calculate the area of different shapes such as Circle, Rectangle, and Triangle.

Theory:

In Java, an interface is a reference type that acts as a contract for classes. It can contain abstract methods, and any class implementing the interface must provide concrete implementations of these methods. This approach promotes polymorphism and code reusability.

In this lab, we will create an interface Shape with a method calculateArea(). The classes Circle, Rectangle, and Triangle will implement this interface and provide their specific implementations for calculating areas.

Algorithm:

1. Start
2. Define an interface Shape with an abstract method calculateArea().
3. Create classes Circle, Rectangle, and Triangle that implement the Shape interface.
4. In each class, define the required parameters (e.g., radius for Circle, length and width for Rectangle, base and height for Triangle).
5. Implement the calculateArea() method in each class using the appropriate formula:
    1. Circle: Area = π * radius * radius
    2. Rectangle: Area = length * width
    3. Triangle: Area = 0.5 * base * height
6. In the Main class, create objects for each shape and initialize them with specific values.
7. Call the calculateArea() method for each object and display the results.
8. End

Program Code:

```
// Shape.java
interface Shape {
   double calculateArea();
}

// Circle.java
class Circle implements Shape {
   private double radius;
```

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

    @Override
    public double calculateArea() {
        return Math.PI * radius * radius;
    }
}

// Rectangle.java
class Rectangle implements Shape {
    private double length;
    private double width;

    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }

    @Override
    public double calculateArea() {
        return length * width;
    }
}

// Triangle.java
class Triangle implements Shape {
    private double base;
    private double height;

    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }

    @Override
    public double calculateArea() {
        return 0.5 * base * height;
    }
}

// Main.java
public class Main {
    public static void main(String[] args) {
        Shape circle = new Circle(5);
        Shape rectangle = new Rectangle(4, 6);
        Shape triangle = new Triangle(3, 8);

        System.out.println("Area of Circle: " + circle.calculateArea());
        System.out.println("Area of Rectangle: " + rectangle.calculateArea());
        System.out.println("Area of Triangle: " + triangle.calculateArea());
    }
}
```

Expected Output:

```
Area of Circle: 78.53981633974483
Area of Rectangle: 24.0
Area of Triangle: 12.0
```

Explanation:

1. The Shape interface defines the calculateArea() method.
2. Each shape class implements the Shape interface and provides its specific formula to calculate the area.
3. In the Main class, objects of each shape are created, and their areas are calculated and displayed.

Conclusion:

This lab demonstrated how to use interfaces in Java to achieve polymorphism. By implementing a common interface, different classes provided their unique implementations while maintaining a consistent method signature.