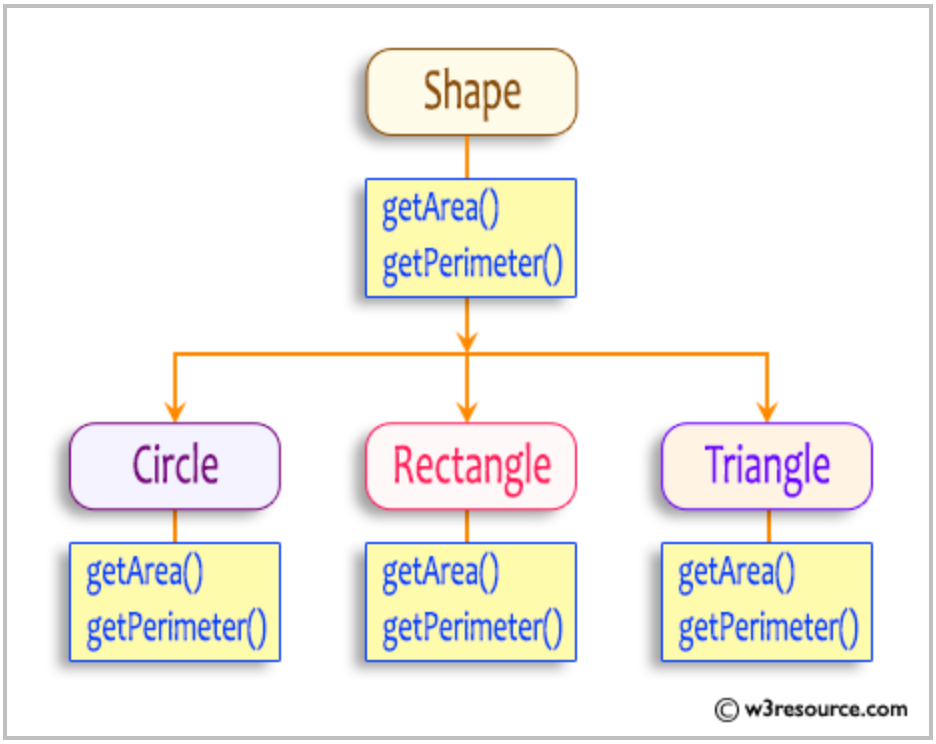
Write a Java program to create a class Shape with methods getArea() and getPerimeter(). Create three subclasses: Circle, Rectangle, and Triangle. Override the getArea() and getPerimeter() methods in each subclass to calculate and return the area and perimeter of the respective shapes.

In the given exercise, here is a simple diagram illustrating polymorphism implementation:



In the above diagram, the Circle, Rectangle, and Triangle classes have the getArea() and getPerimeter() methods. These methods allow them to calculate and return the area and perimeter specific to each shape.

Sample Solution:

Java Code:

```
// Shape.java
// Base class Shape
abstract class Shape {  // Declare an abstract class Shape
    public abstract double getArea();  // Declare an abstract method getAre
    public abstract double getPerimeter();  // Declare an abstract method g
}
```

```
// Circle.java
// Subclass Circle
```

```
class Circle extends Shape {  // Declare a subclass Circle that extends the
    private double radius;  // Declare a private double variable radius

    public Circle(double radius) {  // Define a constructor that takes a do
        this.radius = radius;  // Initialize the radius variable with the p
    }

    @Override  // Override the getArea method from the Shape class
    public double getArea() {  // Define the getArea method
        return Math.PI * radius * radius;  // Calculate and return the area
    }

    @Override  // Override the getPerimeter method from the Shape class
    public double getPerimeter() {  // Define the getPerimeter method
        return 2 * Math.PI * radius;  // Calculate and return the perimeter
    }
}
```

```
// Rectangle.java
// Subclass Rectangle
class Rectangle extends Shape {  // Declare a subclass Rectangle that exter
    private double length;  // Declare a private double variable length
    private double width;  // Declare a private double variable width

    public Rectangle(double length, double width) {  // Define a constructo
        this.length = length;  // Initialize the length variable with the p
        this.width = width;  // Initialize the width variable with the prov
    }

    @Override  // Override the getArea method from the Shape class
    public double getArea() {  // Define the getArea method
        return length * width;  // Calculate and return the area of the rec
    }

    @Override  // Override the getPerimeter method from the Shape class
    public double getPerimeter() {  // Define the getPerimeter method
        return 2 * (length + width);  // Calculate and return the perimeter
    }
}
```

```java
// Triangle.java
// Subclass Triangle

class Triangle extends Shape {  // Declare a subclass Triangle that extends
    private double side1;  // Declare a private double variable side1
    private double side2;  // Declare a private double variable side2
    private double side3;  // Declare a private double variable side3

    public Triangle(double side1, double side2, double side3) {  // Define
        this.side1 = side1;  // Initialize the side1 variable with the prov
        this.side2 = side2;  // Initialize the side2 variable with the prov
        this.side3 = side3;  // Initialize the side3 variable with the prov
    }

    @Override  // Override the getArea method from the Shape class
    public double getArea() {  // Define the getArea method
        double s = (side1 + side2 + side3) / 2;  // Calculate the semi-peri
        return Math.sqrt(s * (s - side1) * (s - side2) * (s - side3));  //
    }

    @Override  // Override the getPerimeter method from the Shape class
    public double getPerimeter() {  // Define the getPerimeter method
        return side1 + side2 + side3;  // Calculate and return the perimete
    }
}
```

```java
// Main.java
// Main class
public class Main {  // Declare the Main class
    public static void main(String[] args) {  // Define the main method
        double r = 4.0;  // Initialize a double variable r with the value 4
        Circle circle = new Circle(r);  // Create an instance of the Circle

        double rs1 = 4.0, rs2 = 6.0;  // Initialize double variables rs1 ar
        double ts1 = 3.0, ts2 = 4.0, ts3 = 5.0;  // Initialize double varia

        Rectangle rectangle = new Rectangle(rs1, rs2);  // Create an instar
        Triangle triangle = new Triangle(ts1, ts2, ts3);  // Create an inst

        System.out.println("Radius of the Circle"+r);  // Print the radius
```

```
            System.out.println("Area of the Circle: " + circle.getArea());  //
            System.out.println("Perimeter of the Circle: " + circle.getPerimete

            System.out.println("\nSides of the rectangle are: "+rs1+','+rs2);
            System.out.println("Area of the Rectangle: " + rectangle.getArea()
            System.out.println("Perimeter of the Rectangle: " + rectangle.getPe

            System.out.println("\nSides of the Triangle are: "+ts1+','+ts2+','+
            System.out.println("Area of the Triangle: " + triangle.getArea());
            System.out.println("Perimeter of the Triangle: " + triangle.getPeri
        }
    }
```

Output:

```
Radius of the Circle4.0
Area of the Circle: 50.26548245743669
Perimeter of the Circle: 25.132741228718345

Sides of the rectangle are: 4.0,6.0
Area of the Rectangle: 24.0
Perimeter of the Rectangle: 20.0

Sides of the Traiangel are: 3.0,4.0,5.0
Area of the Triangle: 6.0
Perimeter of the Triangle: 12.0
```

Flowchart: