

Create a Java interface called "Shape" that defines the following methods:

- `getArea()`: a method that returns a double representing the area of the shape.
- `getPerimeter()`: a method that returns a double representing the perimeter of the shape.

Create a Java enum called "ShapeType" that contains two values: CIRCLE and RECTANGLE.

Create a generic Java class called "ShapeContainer<T extends Shape>" that can store objects that implement the Shape interface. The class should have the following methods:

- `add(T shape)`: a method that adds a shape object to the container.
- `get(int index)`: a method that retrieves the shape object at the given index.
- `size()`: a method that returns the number of shape objects in the container.

Create a Java class called "Circle" that implements the Shape interface. The class should have the following instance variables:

- `radius (double)`: the radius of the circle

The class should have a constructor that takes in the radius and initializes the instance variable.

The Circle class should override the `getArea()` method to calculate and return the area of the circle using the formula:  $\pi * \text{radius}^2$ .

The Circle class should also override the `getPerimeter()` method to calculate and return the perimeter of the circle using the formula:  $2 * \pi * \text{radius}$ .

Create a Java class called "Rectangle" that implements the Shape interface. The class should have the following instance variables:

- `width (double)`: the width of the rectangle
- `height (double)`: the height of the rectangle

The class should have a constructor that takes in the width and height and initializes the instance variables.

The Rectangle class should override the `getArea()` method to calculate and return the area of the rectangle using the formula:  $\text{width} * \text{height}$ .

The Rectangle class should also override the `getPerimeter()` method to calculate and return the perimeter of the rectangle using the formula:  $2 * (\text{width} + \text{height})$ .

Create a Java class called "ShapeContainerTest" that has the following methods:

- `createShapes()`: a method that creates a ShapeContainer object, containing a mixture of Circle and Rectangle objects. This method should use the ShapeType enum to determine which type of shape to create.
- `printShapeInfo(int index)`: a method that takes in an index and prints information about the shape at that index, including its area and perimeter.

The main method of the ShapeContainerTest class should call the `createShapes()` method to create a ShapeContainer object and add several Circle and Rectangle objects to it. Use generics to ensure that only Shape objects can be added to the container. The main method should then call the `printShapeInfo()` method for each object in the container. The `printShapeInfo()` method should handle the exception if the index is out of bounds.

In the `createShapes()` method, use a switch statement to determine which shape to create based on the ShapeType enum. For example:

```
switch(shapeType) {  
    case CIRCLE:  
        // create a new Circle object and add it to the ShapeContainer  
        break;  
    case RECTANGLE:  
        // create a new Rectangle object and add it to the ShapeContainer  
        break;  
    default:  
        // handle invalid input  
}
```