```
public abstract class Shape implements Comparable<Shape>
{
    public abstract double area( );
    public abstract double perimeter( );

    public int compareTo( Shape rhs )
    {
        double diff = area( ) - rhs.area( );
        if( diff == 0 )
            return 0;
        else if( diff < 0 )
            return -1;
        else
            return 1;
    }

    public double semiperimeter( )
    {
        return perimeter( ) / 2;
    }
}
```

```
public class Circle extends Shape
{
    public Circle( double rad )
    {
        radius = rad;
    }

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

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

    public String toString( )
    {
        return "Circle: " + radius;
    }

    private double radius;
}
```

```
public class Rectangle extends Shape
{
    public Rectangle( double len, double wid )
    {
        length = len; width = wid;
    }
```

```
    public double area( )
    {
        return length * width;
    }

    public double perimeter( )
    {
        return 2 * ( length + width );
    }

    public String toString( )
    {
        return "Rectangle: " + length + " " + width;
    }

    public double getLength( )
    {
        return length;
    }

    public double getWidth( )
    {
        return width;
    }

    private double length;
    private double width;
}
=========================================================================
public class Square extends Rectangle
{
    public Square( double side )
    {
        super( side, side );
    }

    public String toString( )
    {
        return "Square: " + getLength( );
    }
}
=========================================================================
class ShapeDemo
{
    public static double totalArea( Shape [ ] arr )
    {
        double total = 0;

        for( Shape s : arr )
            if( s != null )
                total += s.area( );

        return total;
```

```
    }

    public static double totalArea( java.util.List<? extends Shape> arr )
    {
        double total = 0;

        for( Shape s : arr )
            if( s != null )
                total += s.area( );

        return total;
    }

    public static double totalSemiperimeter( Shape [ ] arr )
    {
        double total = 0;

        for( Shape s : arr )
            if( s != null )
                total += s.semiperimeter( );

        return total;
    }
}
```

==========