

FIGURAS

Se va a diseñar una aplicación que rellenará un vector de figuras (ej. 4 figuras) con círculos y rectángulos.

1. Todas las figuras tienen:

- color
- posición: se utilizará un vector de 2 elementos para guardar el valor de la x y la y del centro de la figura.

1.1. Para cada atributo se tendrán el getter y el setter correspondiente

1.2. Al crear una figura se le puede pasar el color y la posición o sólo la posición (en ese caso el color será negro por defecto).

1.3. Todas las figuras tendrán un método que calcule el área y un método que calcule el perímetro.

1.4. Todas las figuras tendrán un método con la siguiente interfaz que dirá si la figura es menor que aquella que se le pase como parámetro. Teniendo en cuenta que se comparan por su área.

Public boolean menorQue(Figura f)

1.5. Sobrecribirá además el método toString para mostrar el centro y el color

1.6. Teniendo en cuenta que una figura como tal no existe y no sabemos cómo se implementan los métodos perímetro y área, ¿de qué tipo será la clase Figura?

2. El círculo tendrá además del centro, el radio que se proporcionará en el momento de crearlo, así como su centro.

3. El rectángulo tendrá la longitud de sus dos lados, que se proporcionarán en el momento de crearlo, así como su centro.

4. Tanto el círculo como el rectángulo sobrescribirán los métodos área y perímetro. También sobrescribirán el método toString para mostrar los datos generales (color y posición) y los particulares de cada figura (además del área y el perímetro).

	Área	Perímetro
Círculo	$\text{Math.PI} * \text{radio} * \text{radio}$	$2 * \text{Math.PI} * \text{radio}$
rectángulo	$\text{base} * \text{altura}$	$2 * (\text{base} + \text{altura})$

Prueba el funcionamiento con la siguiente aplicación:

```
public class AplicacionFiguras {  
    public static void main(String[] args) {  
        Figura[] vec=new Figura[4];  
        vec[0]=new Circulo("rojo",4,5,3);  
        vec[1]=new Rectangulo(3,6,10,12);  
        vec[2]=new Rectangulo("azul",2,5,3,5);  
        vec[3]=new Circulo(42,35,4);  
        for(int i=0;i<4;i++){  
            System.out.println(vec[i].area());  
        }  
        System.out.println("Ordenados:" + estanEnOrden(vec));  
    }  
}
```

Implementa el método **estanEnOrden** que devolverá true o false si las figuras en el vector están en orden creciente por tamaño de área.

Para esto, Figura debe implementar la interfaz Comparable(busca información en internet) y por tanto desarrollar el método **CompareTo** devolviendo -1 si el área de la figura es menor que el de la figura que le pasan como parámetro, 0 si las áreas son iguales o 1 si su área es mayor que la del parámetro.
