

# **Módulo Programación. 1º DAW.**



PRÁCTICA PROPUESTA 1: Introducción a la herencia.  
UNIDAD DE TRABAJO 5.  
Profesor: Pedro Antonio Santiago Santiago.

Se desea crear un pequeño programa de dibujo que genere imágenes [SVG](#). El programa a de poder crear rectángulos, cuadrados, círculos y elipses (optativo crear rutas (path)).

Las etiquetas básicas y atributos de SVG se pueden encontrar en :[https://www.w3schools.com/graphics/svg\\_rect.asp](https://www.w3schools.com/graphics/svg_rect.asp)

Toda las clase, que se pueda dibujar (Coordenada no se dibuja) han de poder:

- Indicar coordenadas x e y.
- Asignar color de fondo.
- Mover la figura a un lugar.
- Configurar el color del borde y su grosor.
- Asignar las características propias de cada figura, por ejemplo, el círculo tiene un radio, la elipse tiene 2 radios, rx y ry.
- Método para calcular el área.
- Pasar a SVG.

Las clases a crear:

- Coordenada.
- Interfaz IDrawable, con un método public String toSVG();
- Figura (de la que heredan el resto de las clases), con un método abstracto public float calculararea() y que implementa la interfaz IDrawable.
- Rectángulo.
- Cuadrado
- Círculo.
- Elipse.
- Opcionalmente Ruta.

Optimizar las clases para escribir el menos código posible, ver si además de heredad de la clase Figura es posible de heredar de otras ya creadas para optimizar.

- ¿Es razonable crear un objeto de la clase Figura? ¿Cómo se ha de declarar la clase para impedir que se puedan crear objetos de esa clase?

La claseFigura posee una coordenada, con getters y setters, impedir que se puedan sobrecribir los metodos getters y setters de la clase figura en las que lo heredan.

Impedir que se pueda heredar de la clase rectángulo, por todas las clases excepto la de cuadrado.

Crear una clase dibujo que posea un vector de figuras. con:

- Métodos para añadir, borrar y modificar figuras
- Método para almacenar el resultado en un fichero.
- Método para calcular el área, a partir de la clase figura y el vector (pensar en abstract de Figura).

Probar con otra clase o la misma y con un método public static void main..., un ejemplo:

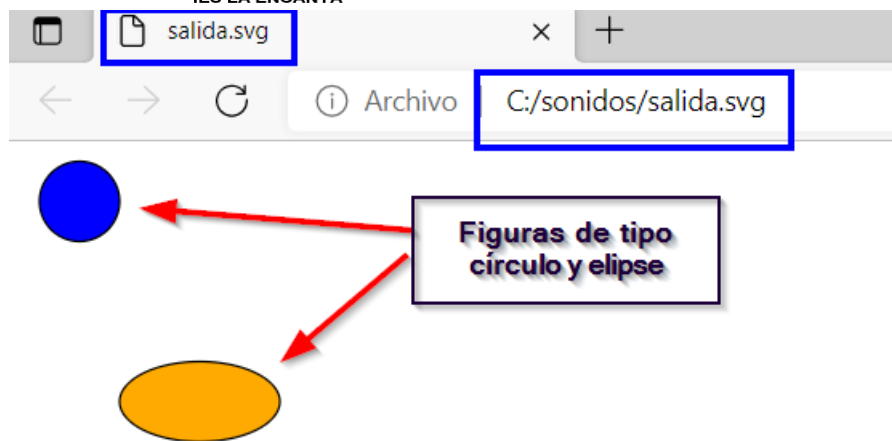
```
public class Principal {  
    public static void main(String[] args){  
        Dibujo d= new Dibujo();  
        Figura tempo=new Circulo(20,40,30);  
        tempo.setStokecolor("black");  
        tempo.setFillcolor("blue");  
        tempo.setStroke_width(1);  
        d.addFigura(tempo);  
  
        tempo=new Elipse(40,20,100,130);  
        tempo.setStokecolor("black");  
        tempo.setFillcolor("orange");  
        tempo.setStroke_width(1);  
  
        d.addFigura(tempo);  
        d.guardarDibujo("C:\\svg\\salida.svg");  
    }  
}
```

Para crear el fichero SVG, en la clase dibujo:

```
public void guardarDibujo(String ruta) {  
    FileWriter fichero = null;  
    try {  
        fichero = new FileWriter(ruta);  
        fichero.write(this.aSVG());  
        fichero.close();  
    } catch (IOException ex) {  
        Logger.getLogger(Dibujo.class.getName()).log(Level.SEVERE, null,  
ex);  
    }  
}
```

El método aSVG() recorre el vector de figuras y llama al método que construye la etiqueta svg de cada figura concreta.

El código anterior genera la siguiente imagen SVG (se puede abrir con cualquier navegador):



Se desea además que se personalicen las clases para:

- Hacer copias profundas de las mismas, por cualquiera.
- Personalizar el método toString() para obtener información de la clase.
- Poder comparar si dos coordenadas o figuras son iguales.
- Sobrecribir usando el IDE el método hashCode().
- Usar el método hashCode para ver si las figuras o coordenadas son iguales. ¿Qué ventaja aporta usar el hashCode().

Por último se quiere poder ordenar el vector de figuras usando el método estático Array.sort de la clase Array, la ordenación se realiza teniendo en cuenta el área de cada una de las figuras. Modificar el menor número de clases para poder usar Array.sort.