

# Práctica 3. Dibuja círculos.

## Objetivos

- Practicar la realización de un diagrama de clases sencillo.
- Familiarizarse con los conceptos de clase y objeto.
- Seguir a familiarizándose con el uso de clases y objetos en lenguaje Java.

## 1. Documento de requisitos

### Descripción general

Se desea realizar una aplicación que permita mostrar un dibujo compuesto por un conjunto de círculos.

Para cada círculo se almacena su radio y las coordenadas x e y de su centro.

Se deben proporcionar operaciones para añadir círculos al dibujo y para mostrar el dibujo en una ventana.

### Hacer:

1. Diseño (diagrama de clases) de la aplicación.
2. Descarga el código proporcionado y comprueba que coincide con el diseño realizado.
3. Ejecuta el código proporcionado (el `main()` se encuentra definido en la clase `GUIMueveCirculos`).
4. Modifica el programa principal para crear otros círculos en diferentes posiciones y añádelos al dibujo (debes tener en cuenta que, en la ventana, la coordenada x crece hacia la derecha y la coordenada y hacia abajo).

## 2. Ampliación: color de los círculos

1. Añade un atributo privado de tipo `Color` a la clase `Círculo` (la clase enumerada `Color` está definida en el paquete `java.awt` por lo que, para poder usar dicha clase, deberás añadir la línea `"import java.awt.Color;"` justo después de la línea `"package pract03;"`)
2. Añade un parámetro al constructor de la clase `Círculo` que permita dar valor al atributo `color` del círculo que se está creando.
3. Crea el método "observador" `color()` que retorne el valor del atributo `color`.
4. Modifica el programa principal para que asigne color a los círculos. Para ello será necesario importar el tipo `color` (`"import java.awt.Color;"`). Los valores del tipo `color` se obtienen poniendo `Color.BLUE`, `Color.RED`, etc.

### 3. Ampliación: movimiento de los círculos

1. Añade dos nuevos atributos de tipo `double` a la clase `Círculo` que representen las componentes de la velocidad en cada eje (`velocidadX` y `velocidadY`). En el constructor estos atributos se inicializan a 0. Ambas velocidades se miden en pixels/s.
2. Añade un método `asignaVelocidad()` a la clase `Círculo` que permita dar valor a dichos atributos.
3. Añade un método `mueve()` a la clase `Círculo` que reciba como parámetro el intervalo de tiempo en segundos (un número real) durante el que se realiza el movimiento. El método deberá actualizar las coordenadas del centro (`centroX` y `centroY`) en función de la velocidad (`velocidadX` y `velocidadY`) y del intervalo de tiempo.
4. Escribe el código del método `mueveCirculos()` de la clase `Dibujo` de forma que llame al método `mueve()` para todos los círculos.
5. En el programa principal, añade un lazo infinito en que se llame al método `ventanaCirculos.redibuja()`.

### 4. Ampliación opcional: rebote contra las paredes

Modifica la aplicación para que los círculos reboten contra los bordes de la pantalla.

Una posible forma de hacerlo sería modificando el método `mueveCirculos()` de forma que, para cada círculo, compruebe si se sale de los límites de la pantalla y, en ese caso, invierta la componente de la velocidad en el eje correspondiente.

### Entregar

- Diagrama de clases (en papel).
- Código desarrollado (a través del moodle).