

PRACTICA 9: Movimiento y control de un objeto gráfico

Factor de ponderación [0-10]: 9

9.1. Objetivos

- Profundizar los conocimientos sobre la clase `Graphics`.
- Trabajar con la gestión de eventos.
- Practicar el desarrollo de aplicaciones gráficas simples en Java.
- Enfatizar la orientación a objetos en el desarrollo de aplicaciones.

9.2. Descripción

Desarrollar un programa `BolaMovil.java` que muestre en pantalla una ventana rectangular con fondo de color azul sobre la que se verá un círculo centrado en la ventana y un panel con cuatro botones que permiten mover el círculo hacia arriba, abajo, izquierda y derecha.

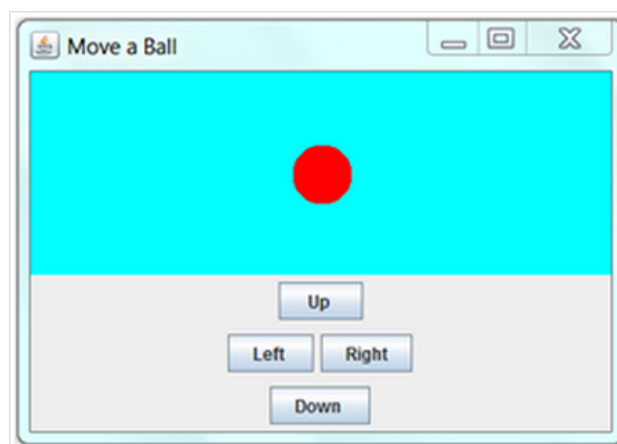


Figura 9.1: Interfaz gráfica del programa

La Figura 9.1 muestra un ejemplo al que debería parecerse la salida del programa a elaborar.

Las siguientes deben tomarse como especificaciones de la aplicación a desarrollar:

- Utilizando `AssertJ`, desarrolle tests unitarios que aseguren que el comportamiento del programa es el deseado.
- Utilice también `JaCoCo` para analizar el cubrimiento que los tests unitarios realizan sobre el código desarrollado.
- La interfaz contendrá los cuatro botones: Arriba, Abajo, Izquierda, Derecha cuyas funciones ya se han comentado.
- Cuando el círculo alcanza cualquiera de los bordes de la ventana, se ha de impedir su movimiento en la dirección del borde alcanzado (los bordes son impenetrables).
- El número de pixeles que se desplaza el círculo por la ventana con cada pulsación de un botón será un parámetro que el programa ha de leer en línea de comandos.
- Si el programa se ejecuta sin pasar ningún parámetro en línea de comandos, se debe escribir un mensaje en la consola indicando la forma correcta de ejecución del programa.
- Se valorará positivamente que el diseño sea efectivamente orientado a objetos, modelando con diferentes clases los distintos elementos que intervienen en la aplicación.
- El programa que aquí se propone se tomará como punto de partida para desarrollar una aplicación similar a la que aquí se propone. Así pues, será beneficioso el realizar un diseño muy modular y flexible.
- Evite utilizar clases o métodos que no hayan sido explicados en las clases de teoría.