

# Programación Orientada a Objetos

Curso 2023/2024

## Sesión 2

### Ejercicios

1. Implementa la clase `circulo` que representa un círculo en el espacio bidimensional de los números enteros. Sitúa esta clase en el paquete `geometria`.

Un círculo se caracteriza por un punto que representa el *centro* y por el *radio* (valor entero). Estas propiedades pueden ser consultadas, pero no modificadas. Además, tiene la propiedad *perímetro* que se calcula como  $2 \cdot \pi \cdot \text{radio}$ . La constante  $\pi$  está disponible en la clase `Math` declarada como `PI`.

Los círculos pueden construirse de tres formas:

- Estableciendo el punto que representa el centro y el radio.
- Tomando como **valores por defecto** el origen de coordenadas para el centro y 5 para el radio. Define dos constantes públicas para representar estos valores:
  - `ORIGEN_COORDENADAS`: representa un punto situado en (0, 0)
  - `RADIO_POR_DEFECTO`: representa el radio por defecto de un círculo. El valor es 5.

La funcionalidad que ofrece esta clase es la siguiente:

- *desplazar*: traslada el círculo una cierta cantidad en el eje X y en el eje Y. Los parámetros de este método son las cantidades de desplazamiento en el eje X y en el eje Y.
- *escalar*: modifica el tamaño del radio en un porcentaje establecido como parámetro. Por ejemplo, escalar al 200% significa duplicar el tamaño del radio. Nótese que el tipo del resultado de una operación depende de los tipos de los operandos. Así, si declaramos las variables `int a = 1; int b = 2;` el resultado de la operación `a/b` será 0 y no 0,5 porque se hace la división entera. Para que el resultado sea un valor real uno de los operandos tiene que serlo (`double` o `float`).

2. Define en el paquete `pruebas` la clase `PruebaCirculo` que incluya un método `main`.

- Declara la variable local *punto1* y asigna un objeto `Punto` en (2, 3).
- Declara la variable *circulo1* que sea asignada por un círculo que tenga como centro *punto1* y radio 3.
- Muestra por la consola las propiedades de *circulo1*.
- Declara la variable *circulo2* que sea asignada por un círculo que se construya con los valores por defecto.
- Muestra por la consola las propiedades de *circulo2*.
- Desplaza *circulo1* 3 unidades en el eje X y 2 unidades en Y.
- Muestra por la consola las propiedades de *circulo1*.
- Escala *circulo2* al 150% y muestra sus propiedades por la consola.

3. **Análisis de las consecuencias del *aliasing*.** Añade al método `main` de la clase `PruebaCirculo` la siguiente funcionalidad.
- Declara la variable `circulo4` que sea asignada por un círculo que tenga de nuevo como centro `punto1` y radio 5.
  - Desplaza `circulo1` 3 unidades en el eje X y 2 unidades en Y.
  - Muestra por la consola las coordenadas del centro de `circulo1` y `circulo4`. Las coordenadas deben ser distintas puesto que el desplazamiento del `circulo1` no debe influir en el `circulo4`.
  - Declara la variable `punto2` de tipo `Punto` y asígnale el resultado de la consulta del centro del `circulo4`.
  - Desplaza el `punto2` una unidad en el eje X.
  - Muestra por la consola las propiedades de la variable `punto2` y también del centro del `circulo4`. Las coordenadas deben ser distintas puesto que el desplazamiento del `centro` no debe influir en el `circulo4`.
4. Reflexiona: ¿Realmente hemos conseguido una constante con la declaración del origen de coordenadas?
5. Implementa una nueva clase que represente puntos como **objetos inmutables**. Adapta la implementación de la clase `Circulo` para que utilice esta versión.