

Programación Orientada a Objetos

Curso 2017/2018

Sesión 2

Ejercicios

Nota: los nuevos tipos de datos que se desarrollen en esta sesión serán añadidos al paquete `geometria` creado en la sesión 1.

1. Define el enumerado `Direccion` con los valores `ARRIBA`, `ABAJO`, `DERECHA` e `IZQUIERDA`. Este enumerado representa las cuatro orientaciones en el plano XY.
2. Modifica la clase `Punto` desarrollada en la sesión 1 para añadir la siguiente funcionalidad:
 - `distancia`: calcula la distancia euclídea entre un punto y otro que es establecido como parámetro. Aplica la siguiente fórmula:

$$dist = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

La clase `Math` de la librería de Java define el método de clase (`static`) `sqrt` para calcular la raíz cuadrada y `pow(double x, double y)` que devuelve el resultado de x^y .

- `desplazar` (versión sobrecargada): esta operación desplaza el punto una unidad según la dirección establecida como parámetro.
 - `mayorDistancia`: recibe como parámetro una colección de puntos (argumento variable) y devuelve aquel que esté más alejado del origen de coordenadas.
3. Implementa la clase `Circulo` que representa un círculo en el espacio bidimensional de los números enteros.

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 π está disponible en la clase `Math` declarada como `PI`.

Los círculos pueden construirse de dos 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. Los valores por defecto deben definirse como constantes.
- Utilizando un constructor de copia.

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.
- *desplazar* (versión sobrecargada): desplaza el círculo de modo que su centro coincida con el parámetro que se establece en la operación.
- *escalar*: modifica el tamaño del radio en un porcentaje establecido como parámetro. Por ejemplo, escalar al 200% significa duplicar el tamaño del radio.

4. Define en el paquete `pruebas` la clase `PruebasSesion2` que incluya un método `main`. El código del programa debe estar documentado con comentarios que expliquen los pasos que se van realizando.
- Declara la variable local *punto1* y asigna un objeto `Punto` en (2, 3).
 - Desplaza *punto1* en la dirección hacia arriba.
 - Muestra las propiedades de *punto1* por la consola.
 - Declara la variable local *punto2* y asigna un objeto `Punto` en (4, 2).
 - Calcula la distancia entre *punto1* y *punto2*. Muestra el resultado en la consola.
 - Utiliza el método `mayorDistancia` para comprobar cuál de los dos puntos (*punto1*, *punto2*) está más alejado del eje de coordenadas. Muestra el resultado en la consola.
 - Declara la variable *circulo1* que sea asignada por un círculo que tenga como centro *punto1* y radio 3.
 - Declara la variable *circulo2* 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 *circulo2*. Las coordenadas deben ser distintas puesto que el desplazamiento del *circulo1* no debe influir en el *circulo2*.
 - Declara la variable *centro* y asigne el resultado de la consulta del centro del *circulo2*.
 - Desplaza el *centro* hacia abajo.
 - Muestra por la consola las coordenadas del *centro* y del centro del *circulo2*. Las coordenadas deben ser distintas puesto que el desplazamiento del *centro* no debe influir en el *circulo2*.
 - Escala *circulo2* al 150% y muestra sus propiedades por la consola.