

CASO PRÁCTICO 6

• TÍTULO: Programación orientada a objetos con Java

SITUACIÓN

Tenemos que resolver los siguientes problemas para la empresa de programación para la que trabajamos.

INSTRUCCIONES

Implementar las clases Punto, Círculo y Cilindro con el protocolo que aparece en el siguiente diagrama de clases.

Círculo

response to the string of the string of the string of the string of the string that the string that the string that the string of the string o

centro(p: Punto):void centro():punto radio():float radio(r:float):void toString():String trasladar(a:float, b:float):void Circulo() Circulo(p:Punto, r:float)

altura: float

altura(a:float):void
altura():float
base(b:Circulo):void
toString():String
trasladar(a:float,
b:float):void
Cilindro()
Cilindro(b:Circulo,
a:float)

Tener en cuenta que la clase Circulo va a heredar de la clase Punto, y, a su vez, la clase Cilindro va a heredar de la clase Circulo.

Por otra parte, hay que tener en cuenta también, que no hace falta ningún atributo *centro* en la clase Circulo porque las coordenadas del centro las hereda de la clase Punto (atributos x, y).

Tampoco hace falta ningún atributo *base* en la clase Cilindro porque el círculo que actúa de base de este Cilindro lo hereda de la clase Circulo (en concreto, el radio y el centro del círculo).



Probar el funcionamiento de las clases con el siguiente método main:

```
public class TestCilindro {
    public static void main(String[] args) {
        Punto centroBase = new Punto(3.0f, 5.0f);
        Circulo base = new Circulo(centroBase, 4.0f);
        Cilindro miCilindro = new Cilindro(base, 10.0f);
        System.out.println(miCilindro);
        miCilindro.trasladar(2.0f, 2.0f);
        System.out.println(miCilindro);
        System.out.println(
            new Cilindro(new Circulo(new Punto(3.0f, 5.0f), 4.0f),
            10.0f));
    }
}
```