

## Ejercicio de refactorización en Eclipse

*Todos los pasos se resuelven a través de los menús Source y Refactor*

1. Crear un proyecto Eclipse llamado “refactor”
2. Crear un paquete “utilidades” y, dentro, una clase Circulo:

```
package utilidades;
public class Circulo
{
    private double rad;

    public Circulo(double radio)
    {
        this.rad = radio;
    }

    public void imprimir()
    {
        String color = "rojo";
        System.out.println("Diámetro: " + 2*rad);
        System.out.println("Color: " + color);
        double area = 2 * 3.1416 * rad * rad;
        System.out.println(area);
    }

    public boolean esIgual (Circulo otro, boolean conDecimales)
    {
        double radiol = this.rad;
        double radio2 = otro.getRad();
        if (conDecimales)
        {
            if (radiol == radio2)
            return true;
            else
                return false;
        }
        else
        {
            if (Math.abs(radiol-radio2)<1)
                return true;
            else
                return false;
        }
    }
}
```

3. Crear un paquete “figuras” y dentro crear la clase Test con este código:

```
package figuras;
public class Test
{
    public static void main(String[] args)
    {
        Circulo c1 = new Circulo(5.5);
        Circulo c2 = new Circulo(10.1);
    }
}
```

```
Circulo c3 = new Circulo(10.9);

if (c2.esIgual(c3, false))
    System.out.println("c2 y c3 son iguales si no
tenemos en cuenta decimales");

if (c2.esIgual(c3, true))
    System.out.println("c2 y c3 son iguales incluso si tenemos en
cuenta decimales");

}
}
```

4. Mediante las utilidades de "Source":

- Genera métodos get y set para la clase Circulo
- Corrige la indentación del código
- Dale formato al código

Utilizando ahora las opciones de refactorización, resuelve los siguientes pasos:

5. Mover Circulo al paquete figuras
6. Observar la clase Test: hacer doble clic en el Warning y obedecer a su primera sugerencia para solucionar el problema automáticamente.
7. Renombrar la clase Circulo por Circunferencia. Observar si el cambio afecta a otras clases (en este caso Test).
8. Renombrar el atributo "rad" por "radio". ¿Cómo afecta al método get?
9. Convertir la variable local "color" del método imprimir en un atributo, inicializando su valor en el mismo método imprimir.
10. En imprimir, en lugar de calcular y escribir el diámetro directamente en el println, extraer a una variable local "d" e imprimir dicha variable.
11. Hacer que 3.1416 sea una constante llamada PI.
12. Extraer el cálculo del área a un método llamado calcularArea. No recibirá parámetros y devolverá un double.
13. Cambiar la firma o cabecera del método esIgual, invirtiendo el orden de los parámetros y cambiando el nombre de conDecimales por considerarDecimales. ¿Cómo afecta el cambio a la clase Test, en la que se usaba este método?
14. Ahora usaremos "inline" para deshacer algunos cambios, es decir, hacer el código más concreto. Seleccionar la variable "d" (diámetro) y hacer que su valor se use en línea, desapareciendo por tanto la variable.
15. Seleccionar la llamada al método calcularArea y hacer que su código se incorpore en la misma línea, desapareciendo la necesidad de usar el método (puedes borrar el método después).
16. Seleccionar la constante PI y hacer que su valor se incorpore a las líneas en que se usa, desapareciendo por tanto la constante.
17. Intenta hacer la misma operación con el atributo "color". ¿Es posible eliminar un atributo y utilizar su valor en línea?