JESUITAS	**DAM 31E Entornos de Desarrollo C.4.1.E.**

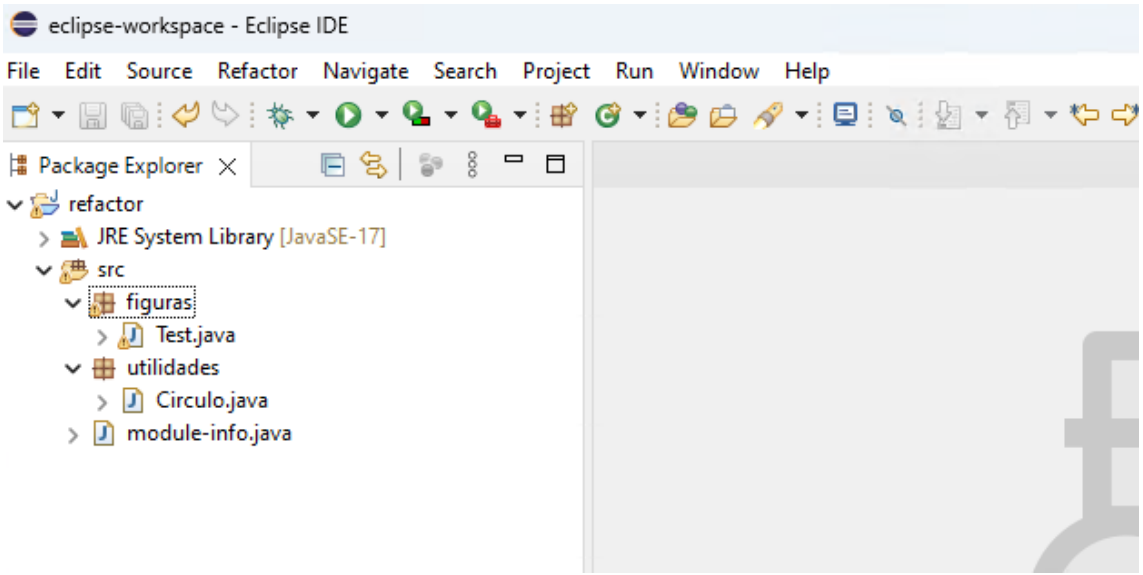
Nombre y Apellidos	Laura Villar Caballero		
Módulo	Entornos de Desarrollo		
Número de Entrega	C.4.1.E.	**Fecha**	13/01/2024

C4.1.E TAREA REFACTORIZACIÓN

Los pasos a seguir son los siguientes:

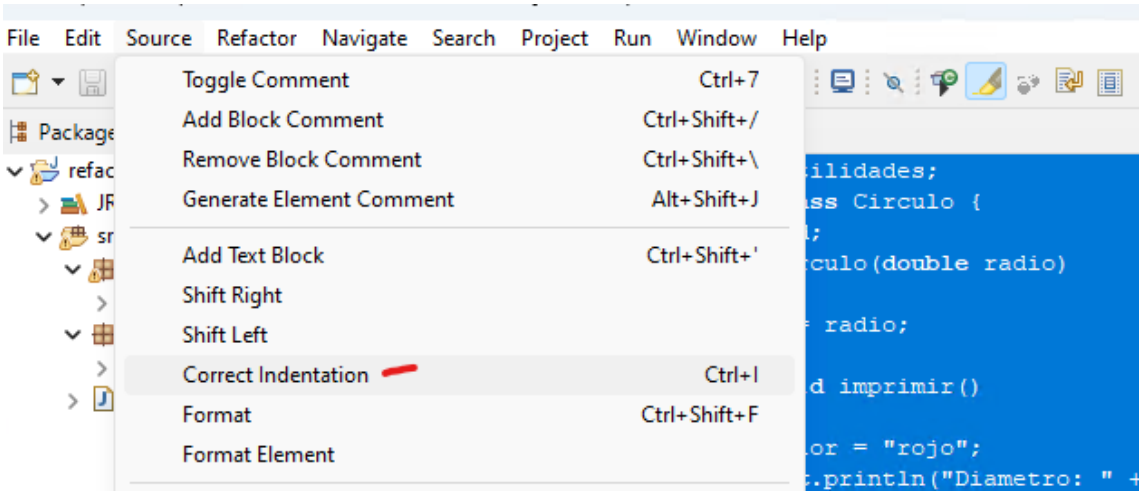
Crear el proyecto:

- Crear un proyecto Eclipse llamado “refactor”.
- Crear un paquete “utilidades” y dentro crear la clase Circulo.
- Crear un paquete “figuras” y dentro crear la clase Test.

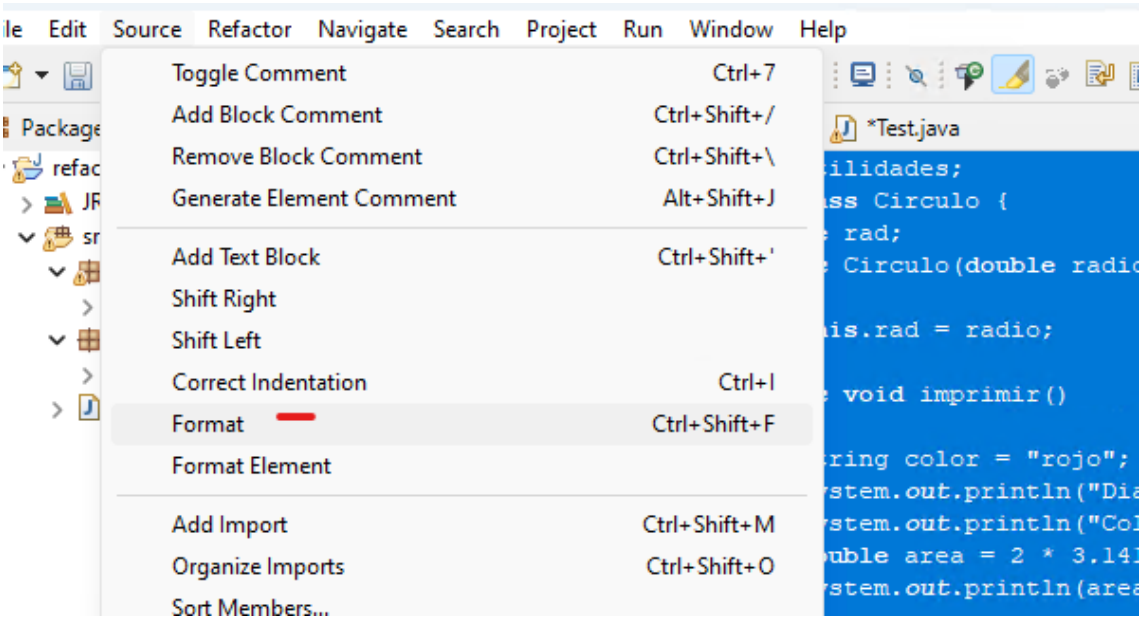


Utilizando las opciones del menú Código fuente, resuelve los siguientes pasos:

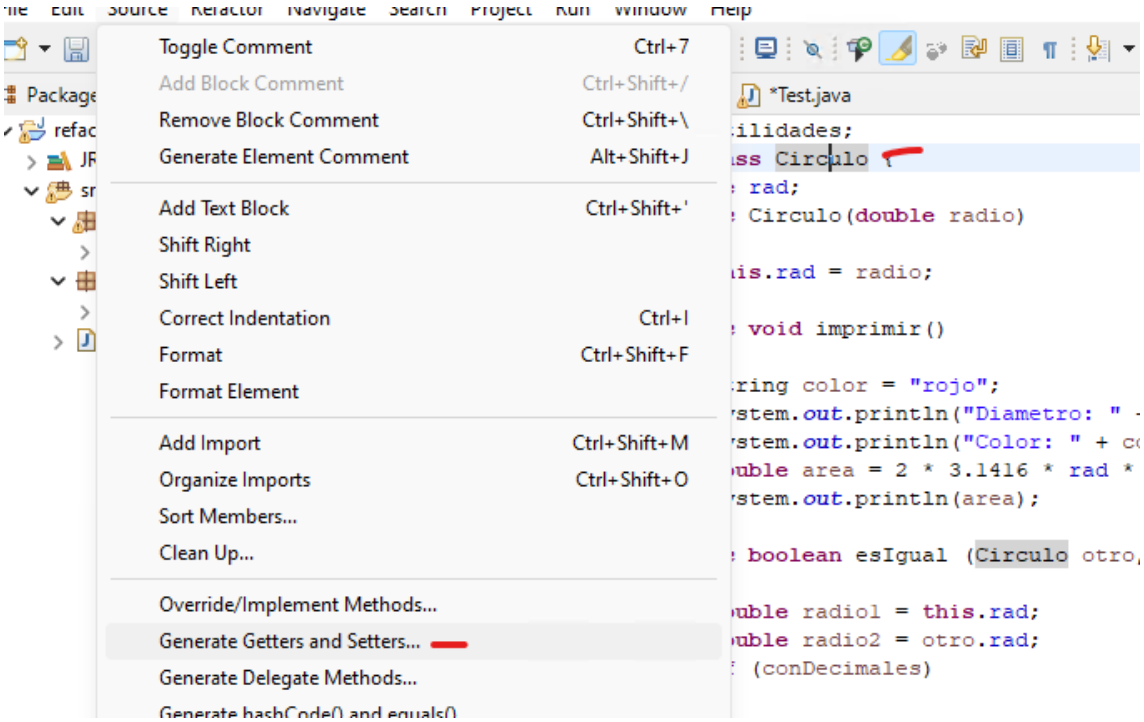
- Corregir la tabulación del código.

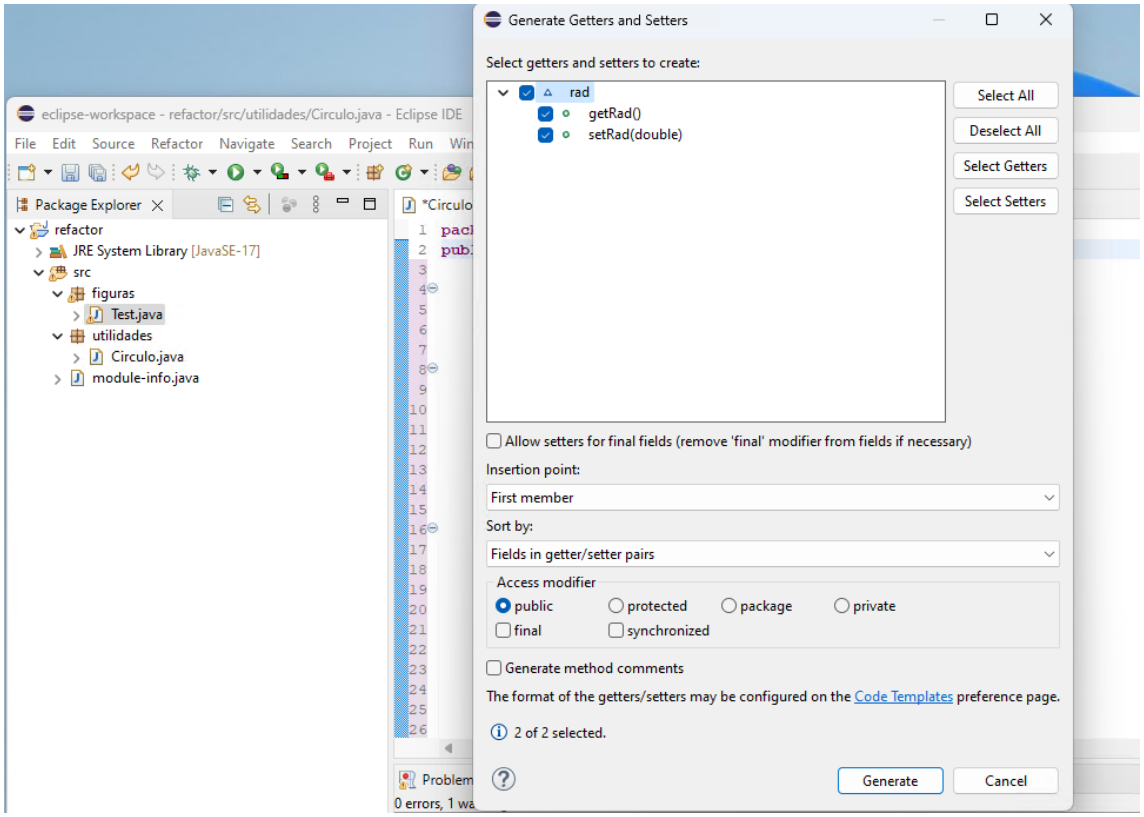


- Dar formato al código.



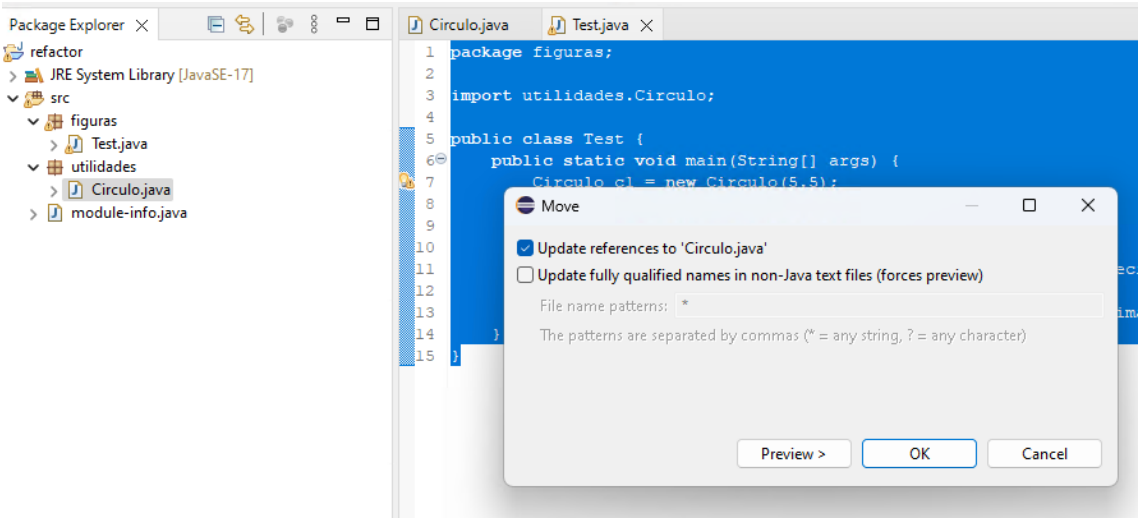
- Generar métodos get y set para la clase Circulo.



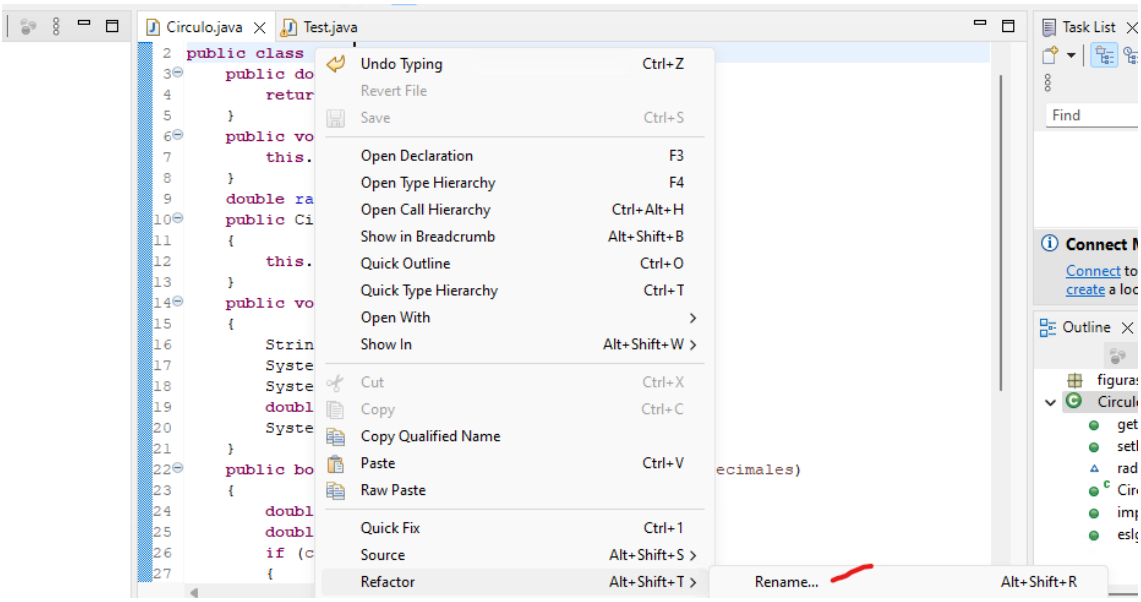


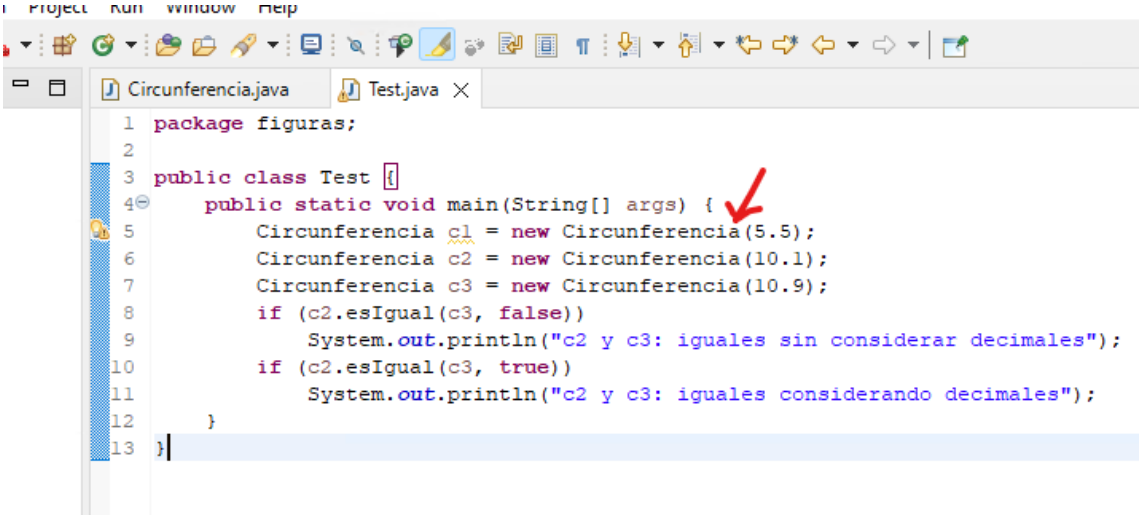
Utilizando las opciones del menú Refactorizar, resuelve los siguientes pasos:

- Mover la clase Circulo al paquete figuras.

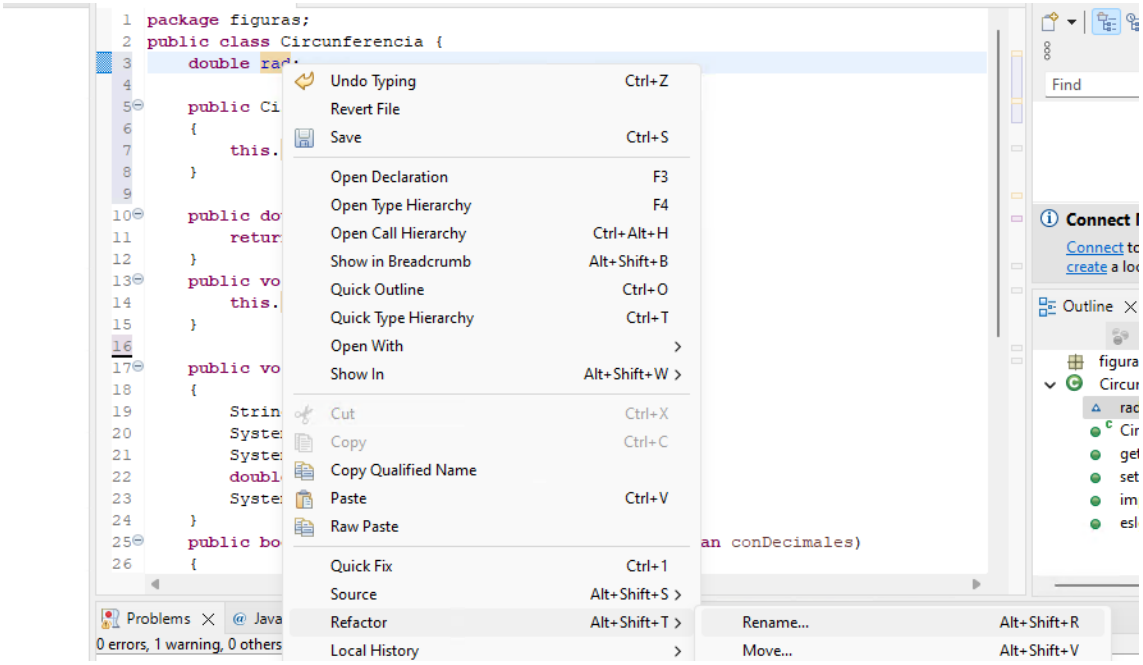


- Renombrar la clase Circulo por Circunferencia. Observar si el cambio afecta a otras clases (en este caso Test).



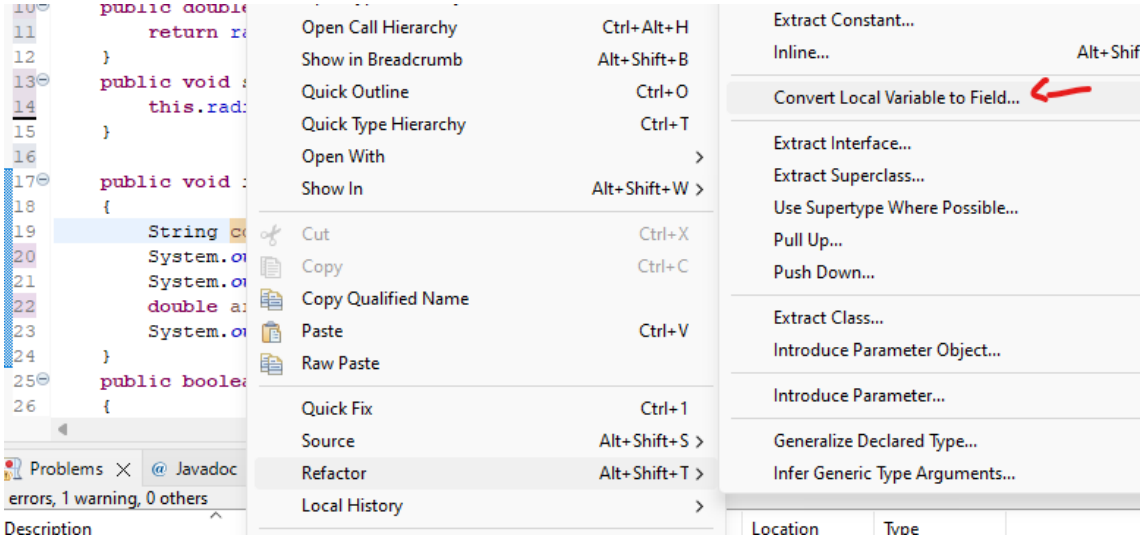


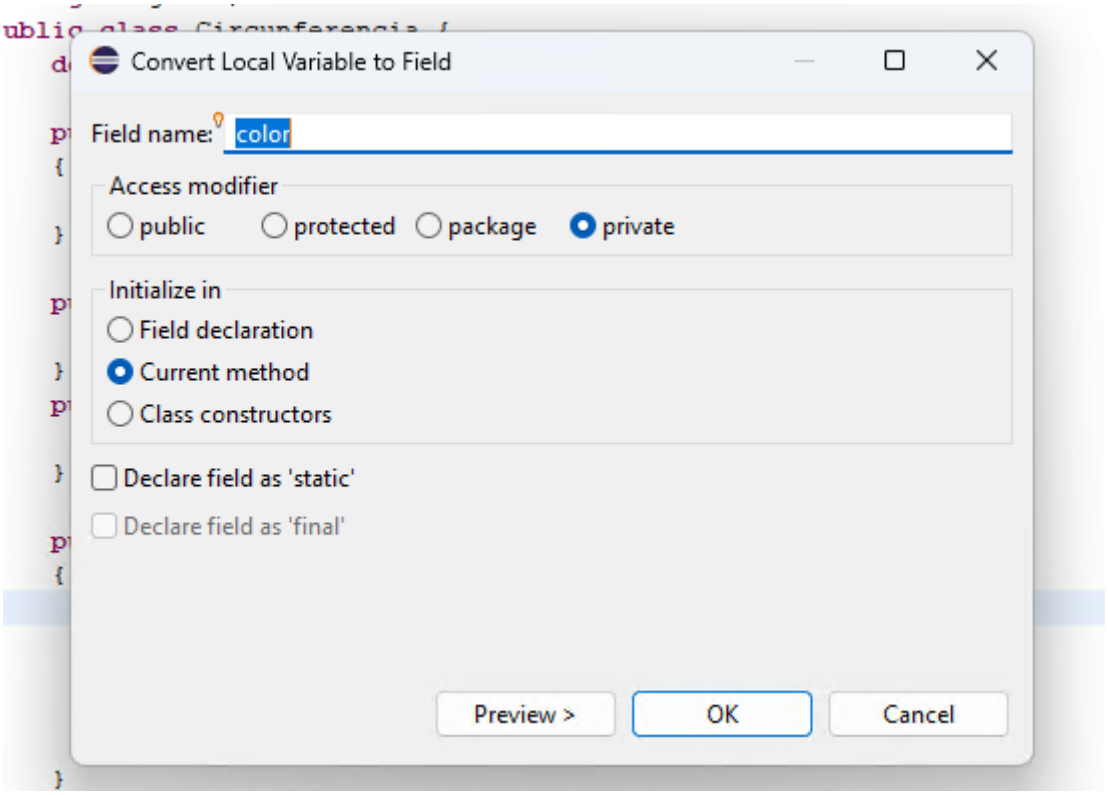
- Renombrar el atributo "rad" por "radio". ¿Cómo afecta al método get?.



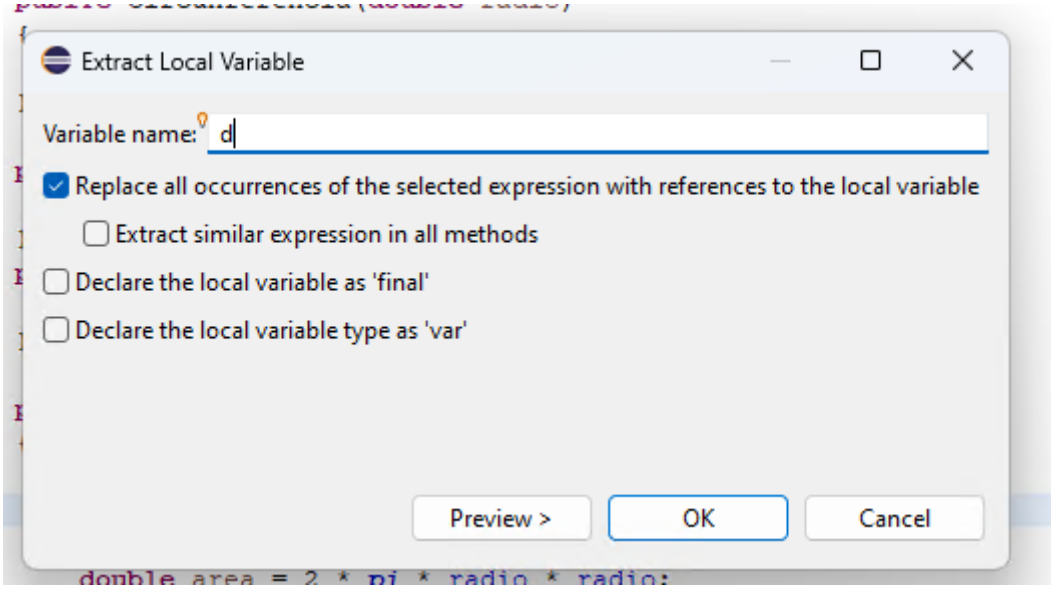
```java
public class Circunferencia {
    double radio;

    public Circunferencia(double radio)
    {
        this.radio = radio;
    }

    public double getRadio() {
        return radio;
    }
    public void setRadio(double rad) {
        this.radio = rad;
    }

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

- Convertir la variable local "color" del método imprimir en un atributo, inicializando su valor en el mismo método imprimir.

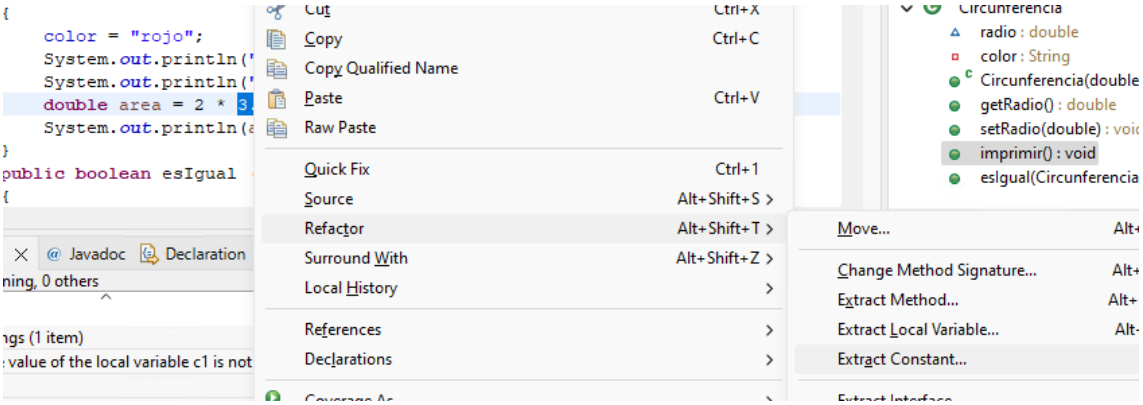


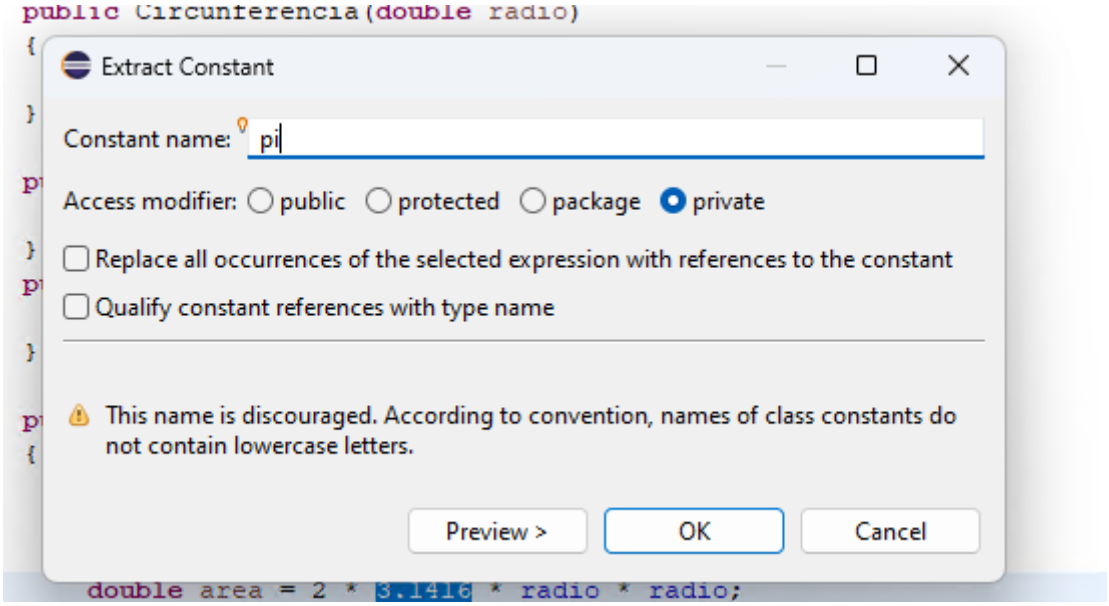


- En imprimir, en lugar de calcular y escribir el diámetro directamente en el println, extraer a una variable local "d" e imprimir dicha variable.

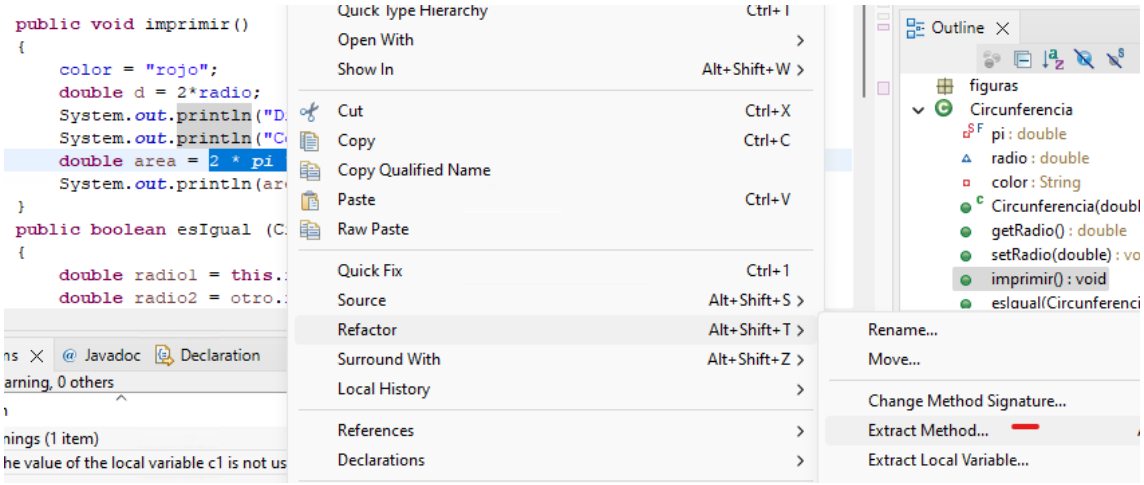


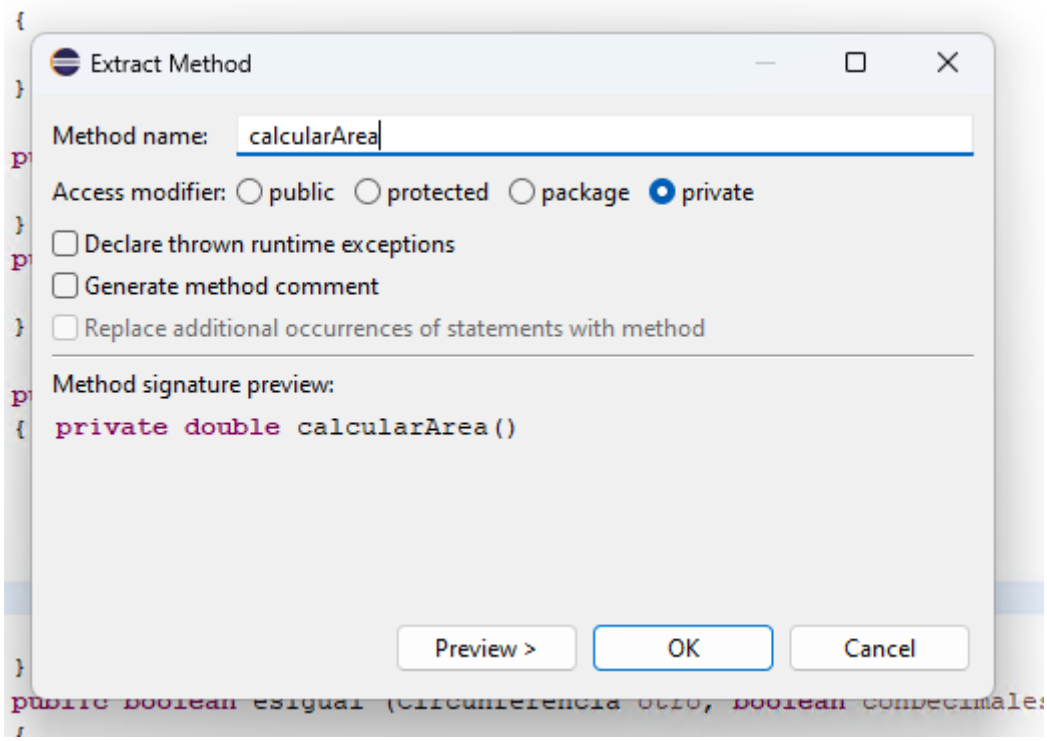
- Hacer que 3.1416 sea una constante llamada PI.



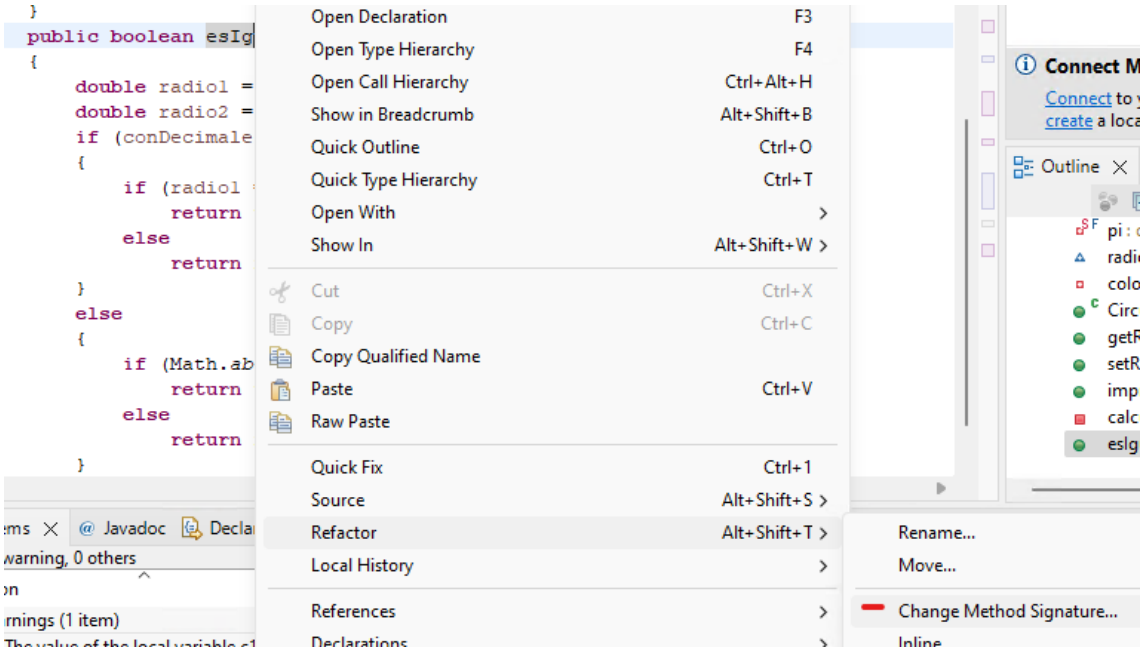


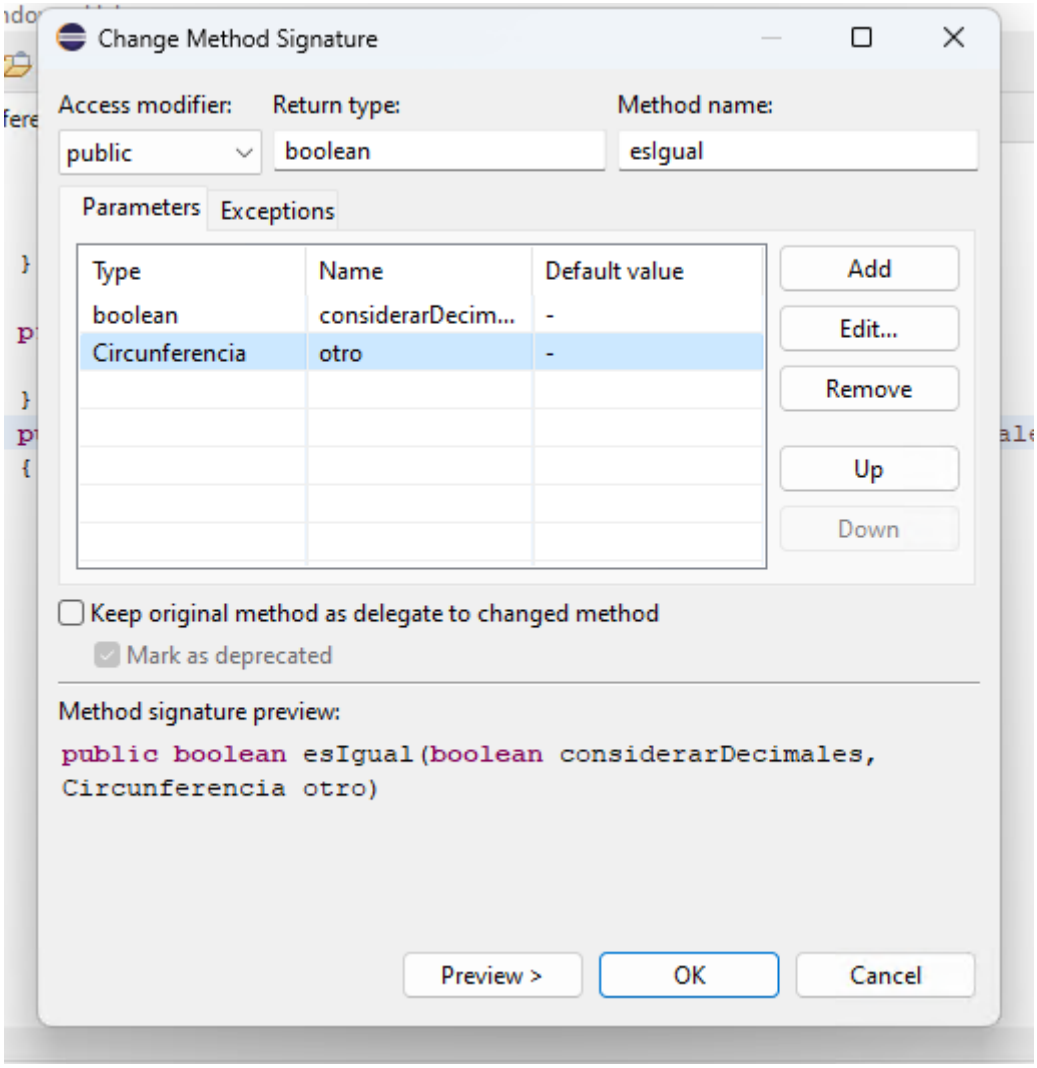
- Extraer el cálculo del área a un método llamado calcularArea. No recibirá parámetros y devolverá un double.





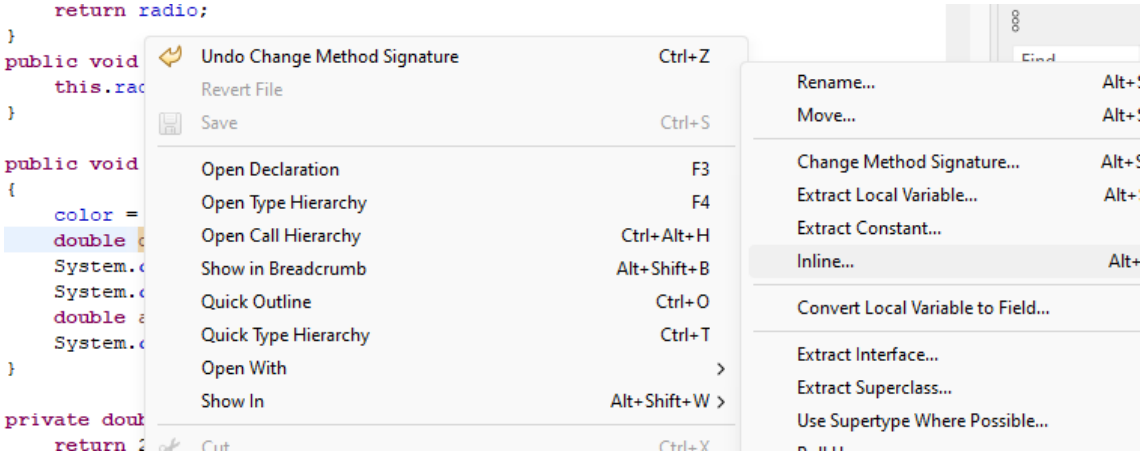
- Cambiar la firma o cabecera del método esIgual, invirtiendo el orden de los parámetros y cambiando el nombre de conDecimales por considerarDecimales. ¿Cómo afecta el cambio a la clase Test, en la que se usaba este método?.


public boolean esIg
{
double radio1 =
double radio2 =
if (conDecimale
{
if (radiol
return
else
return
}
else
{
if (Math.ab
return
else
return
}
Open Declaration F3
Open Type Hierarchy F4
Open Call Hierarchy Ctrl+Alt+H
Show in Breadcrumb Alt+Shift+B
Quick Outline Ctrl+O
Quick Type Hierarchy Ctrl+T
Open With
Show In Alt+Shift+W
Cut Ctrl+X
Copy Ctrl+C
Copy Qualified Name
Paste Ctrl+V
Raw Paste
Quick Fix Ctrl+1
Source Alt+Shift+S
Refactor Alt+Shift+T
Local History
References
Declarations
Rename...
Move...
Change Method Signature...
Inline
Connect M
Connect to
create a loca
Outline
Javadoc
warning, 0 others
rnings (1 item)

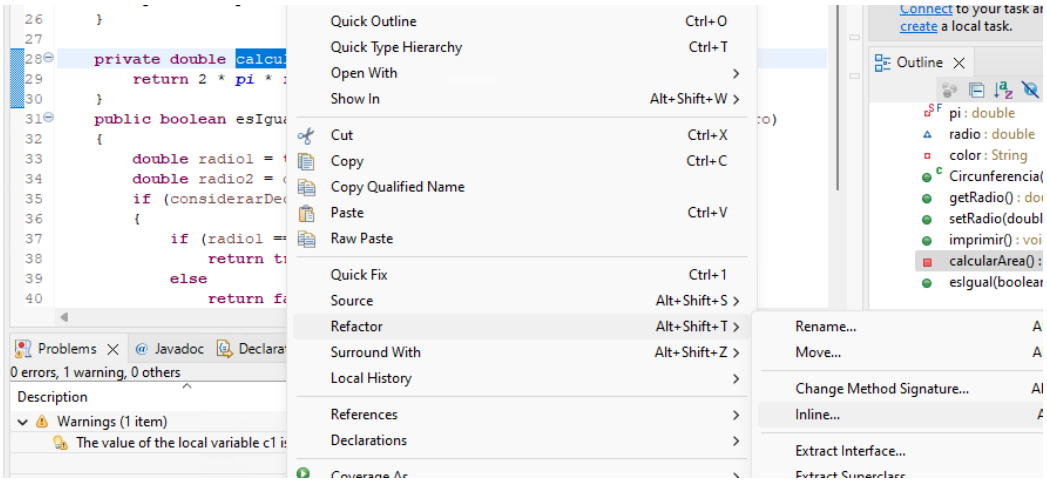

Change Method Signature
Access modifier:
Return type:
Method name:
public
boolean
eslgual
Parameters
Exceptions
Type
Name
Default value
boolean
considerarDecim...
-
Circunferencia
otro
-
Add
Edit...
Remove
Up
Down
Keep original method as delegate to changed method
Mark as deprecated
Method signature preview:
public boolean esIgual(boolean considerarDecimales,
Circunferencia otro)
Preview >
OK
Cancel

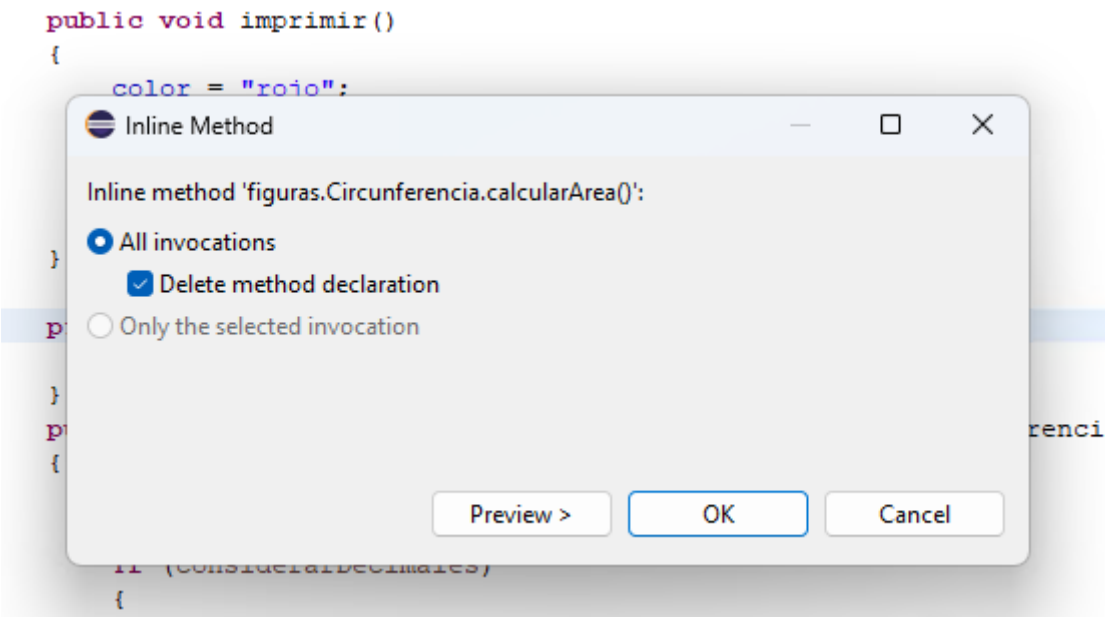
```
public class Test {
    public static void main(String[] args) {
        Circunferencia c1 = new Circunferencia(5.5);
        Circunferencia c2 = new Circunferencia(10.1);
        Circunferencia c3 = new Circunferencia(10.9);
        if (c2.esIgual(false, c3))
            System.out.println("c2 y c3: iguales sin considera
        if (c2.esIgual(true, c3))
            System.out.println("c2 y c3: iguales considerando
    }
}
```

- Ahora se propone usar "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.

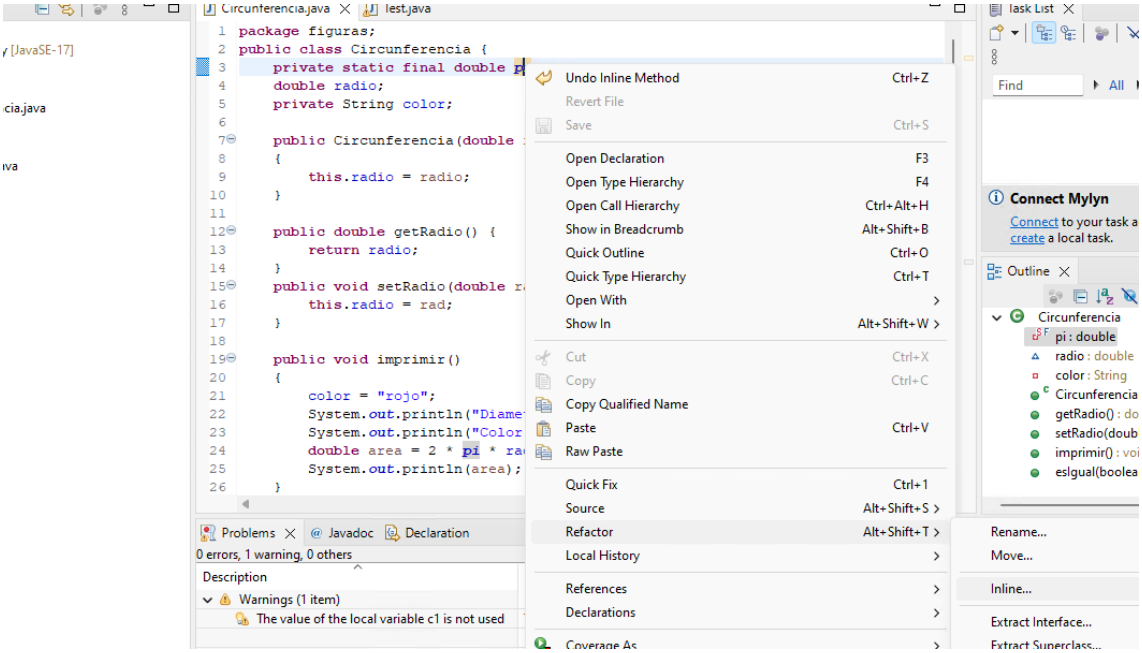


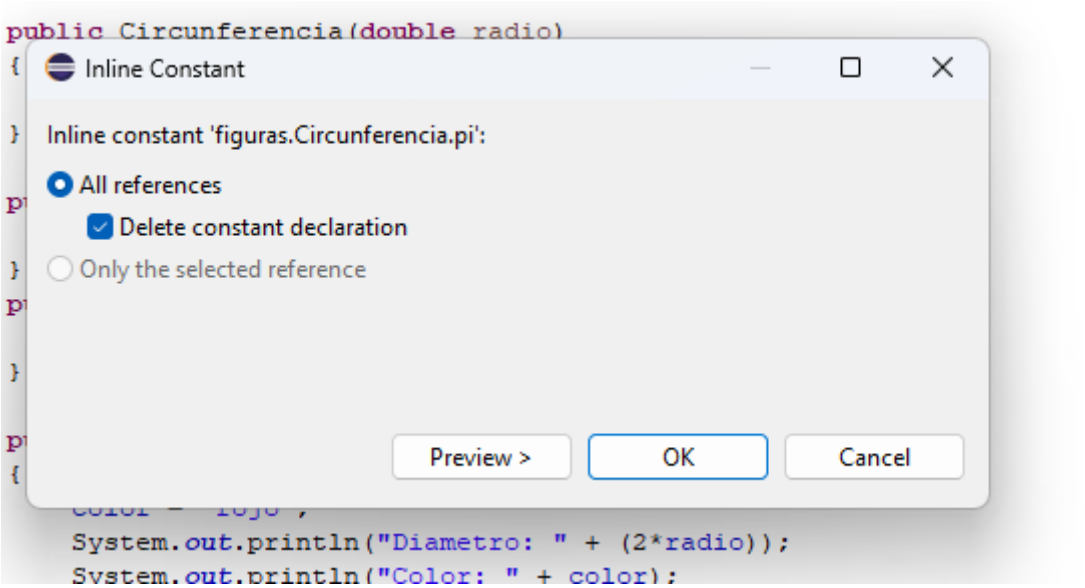
- Seleccionar la llamada al método calcularArea y hacer que su código se incorpore en la misma línea, desapareciendo la necesidad de usar el método (se puede borrar el método después).





- Seleccionar la constante PI y hacer que su valor se incorpore a las líneas en que se usa, desapareciendo por tanto la constante.




public Circunferencia(double radio)
{
}
p
}
p
}
p
{
Inline Constant
Inline constant 'figuras.Circunferencia.pi':
All references
Delete constant declaration
Only the selected reference
Preview >
OK
Cancel
System.out.println("Diametro: " + (2*radio));
System.out.println("Color: " + color);