

Evaluación Práctica N° 4 Refactorización

Teniendo la siguiente clase tendrás que utilizar la herramienta "refactorizacion :

Nombre del proyecto: RefactorizacionEjemplo.

Nombre de la clase: CalculadoraInteres

```
import java.util.Scanner;
public class CalculadoraInteres {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        // Entrada de datos
        System.out.println("Bienvenido a la Calculadora de
Interés");
        System.out.println("Ingrese el monto inicial: ");
        double p = sc.nextDouble();

        System.out.println("Ingrese la tasa de interés (en decimal,
ejemplo 0.05 para 5%): ");
        double r = sc.nextDouble();

        System.out.println("Ingrese el tiempo en años: ");
        int t = sc.nextInt();

        // Cálculo de interés simple
        double simple = p * r * t;
        System.out.println("El interés simple es: " + simple);

        // Cálculo de interés compuesto
        double compuesto = p * Math.pow((1 + r), t);
        System.out.println("El interés compuesto es: " +
compuesto);

        // Conversión a monedas
        System.out.println("Conversiones de moneda:");
```

```

System.out.println("Euros: " + (p * 0.85));
System.out.println("Libras: " + (p * 0.75));
System.out.println("Yenes: " + (p * 110.0));

// Repetir para mostrar resultados
System.out.println("Interés simple nuevamente: " + p * r *
t); // Código duplicado
System.out.println("Interés compuesto nuevamente: " + p *
Math.pow((1 + r), t));
    sc.close();
}
}

```

Tendrás que refactorizar lo siguiente

REFACTORIZAR	Propósito	Ventaja	Solución
--------------	-----------	---------	----------

Extracción de Métodos	Modularizar el código	Mayor claridad y reutilización de funciones.	Crear métodos como calcularInteresSimple, calcularInteresCompuesto, y mostrarConversionMonedas.
------------------------------	-----------------------	--	---

Renombrado (RENAME)	Hacer los nombres más descriptivos	Facilita la lectura y el mantenimiento.	Renombrar variables ambiguas como p, r, y t a nombres descriptivos: montoInicial, tasaInteres, tiempoAnios.
----------------------------	------------------------------------	---	---

Eliminación de Código Duplicado	Reducir redundancias	Menor riesgo de errores y cambios más fáciles.	Elimina las repeticiones de cálculos de interés simple y compuesto al usar métodos centralizados.
--	----------------------	--	---

Reemplazo de Números Mágicos Extract Constant	Uso de constantes en lugar de valores fijos	Mejora la comprensión y facilita modificaciones.	Los valores como 0.85, 0.75, y 110.0 serán reemplazados por constantes descriptivas como TASA_EUROS.
--	---	--	--

Realiza un pequeño informe bien presentado por cada petición que responda a lo siguiente:

Por cada cambio realizado, responde:

1. ¿Qué cambio se realizó?
2. ¿Por qué se realizó?
3. ¿Cómo mejora el código en términos de mantenimiento y legibilidad?

Entrega:

- Código refactorizado (.java).
- Informe de refactorización (PDF).