

Unidad 4: Optimización, documentación y control de versiones

Práctica 8. Refactorizando en Eclipse y Netbeans

Criterios de Evaluación Relacionados

Con estos ejercicios prácticos se evalúan los siguientes criterios de evaluación con los resultados de aprendizaje correspondientes.

Resultados de aprendizaje (RA)	Pond. RA	Criterios de evaluación	Pond. criterio	UD
RA4	16%	4.a) Se han identificado los patrones de refactorización más usuales.	1.76%	4
		4.b) Se han elaborado las pruebas asociadas a la refactorización	1.76%	
		4.e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.	1.76%	

Técnica de evaluación	Ejecución práctica	
Instrumento de evaluación	Checklist o guía de evaluación	Puntuación
Cambia el nombre a las variables		2
Extrae constantes		2
Extrae métodos		2
Convierte variables a atributos		2
Cambia firma de un método		2
Total:		10

1) Refactorización en Eclipse

Descarga el código fuente **EDT5_08varios** de la Moodle y aplica la siguiente refactorización que puedes ver en el código, siguiendo los siguientes pasos. Haz todos los cambios en la clase **Despues.java**:

1º) Las variables **a**, **b** y **c** pasan a llamarse **operando1**, **operando 2** y **resultado**, respectivamente

Para ello, marca las variables y desde el menú contextual **Refactorizar/Redenominar**

2º) Extrae como constantes las cadenas “**Introduce un número:**”, “**suma**” y “**resta**”

Para ello, selecciona cada una de estas cadenas y desde el menú contextual **Refactorizar/Extraer constante** crea las tres constantes indicando un nombre significativo.

3º) Extrae la lectura de los operandos a un método llamado **recogeDato()**

Para ello, marca el código de la primera lectura (2 líneas) y desde el menú contextual **Refactorizar/Extraer método**

4º) Convierte las variables locales **operando1**, **operando2** y **resultado** en atributos

Para ello, marca cada una de las variables y desde el menú contextual **Refactorizar/Convertir variable local en campo**

5º) Modifica el método **recogeDato()** a **recogeNumero()** manteniendo la versión previa como obsoleta.

Para ello, se marca el nombre del método y desde el menú contextual **Refactorizar/Cambiar signatura del método**.

Modifica lo que sea necesario (nombre, visibilidad, tipo, etc...) y marca “mantener método antiguo” y “marcar como en desuso”

1) Refactorización en Netbeans

Importa a Netbeans el proyecto de Eclipse y haz lo mismo. Trabaja sobre la clase **Despues.java**

1º) Las variables **a**, **b** y **c** pasan a llamarse **operando1**, **operando 2** y **resultado**, respectivamente

Para hacerlo en Netbeans, se marca la variable a cambiar y desde el menú contextual **Reestructurar/Cambiar de nombre (Refactor/Rename)**

2º) Extra como constantes las cadenas “Introduce un número”, “suma”, “resta”.

Para ello, se marca cada una de las cadenas y desde el menú contextual **Reestructurar/Introduce/Introducir constante (Refactor/Introduce/Constant)**

3º) Extrae la lectura de los operandos a un método llamado **recogeDato()**

Para ello se marca el código de la primera lectura (2 líneas) y desde el menú contextual **Reestructurar/Introduce/Introducir método (Refactor/Introduce/Method)**.

¿Qué ha ocurrido en este caso? Fíjate bien, ¿ha cambiado todas las ocurrencias del mismo código?

4º) Convierte las variables locales **operando1**, **operando2** y **resultado** en atributos

¿Qué ocurre?

5º) Modifica **recogeDato()** por **recogeNumero()**, manteniendo la versión previa como obsoleta

¿Qué ocurre? Las opciones para modificar métodos son distintas a este propósito.