**Aquí se puso bien jodida la IA y solo da Tareas cuyo objetivo es ver los errores y el por que de los mismos cuando a lo largo del Tutorial Indico Explicita que va a pasar  
  
Tutorial Interactivo: final en Clases, Métodos y Atributos en Java**

En este tutorial, aprenderás cómo se utiliza la palabra clave **final** en Java y cómo afecta a las clases, métodos y atributos. Trabajaremos paso a paso con conceptos, ejercicios interactivos y preguntas que te permitirán aplicar lo aprendido de manera práctica.

**1. Concepto de final en Java**

En Java, la palabra clave **final** se utiliza para declarar clases, métodos y atributos que no pueden ser modificados una vez definidos. Dependiendo de dónde se use, final tiene efectos diferentes:

* **En clases**: Impide que la clase sea heredada.
* **En métodos**: Evita que los métodos sean sobreescritos por las subclases.
* **En atributos**: Hace que los valores sean constantes, es decir, no se pueden cambiar después de asignarse.

**2. final en Clases**

**Definición:**

Cuando declaras una clase como **final**, esta clase **no puede ser heredada**. Esto significa que no podrás crear subclases a partir de ella. Se utiliza cuando quieres evitar que otras clases extiendan la funcionalidad de tu clase.

**Ejemplo:**

java

Copiar código

final class Vehiculo {

public void conducir() {

System.out.println("Conduciendo el vehículo...");

}

}

En este ejemplo, la clase **Vehiculo** es final, lo que impide que cualquier otra clase pueda extender (heredar) de ella.

**Tarea 1:**

Intenta crear una subclase de la clase **Vehiculo** en el siguiente código y observa el error que genera.

java

Copiar código

class Coche extends Vehiculo {

public void acelerar() {

System.out.println("El coche está acelerando.");

}

}

**Pregunta 1**: ¿Qué error te muestra el compilador cuando intentas heredar de una clase final?

**3. final en Métodos**

**Definición:**

Un método declarado como **final** no puede ser sobreescrito en una subclase. Esto se usa para evitar que se modifique la implementación de un método específico cuando se hereda una clase.

**Ejemplo:**

java

Copiar código

class Animal {

public final void dormir() {

System.out.println("El animal está durmiendo.");

}

}

class Perro extends Animal {

// Intentar sobrescribir este método dará un error

// public void dormir() {

// System.out.println("El perro está durmiendo.");

// }

}

En este ejemplo:

* El método **dormir** en la clase Animal está marcado como final, lo que impide que la subclase Perro lo sobrescriba.

**Tarea 2:**

Intenta sobrescribir el método **dormir** en la clase Perro y observa el error que te muestra.

**4. final en Atributos**

**Definición:**

Cuando un atributo es declarado como **final**, se convierte en una **constante**. Su valor solo puede ser asignado una vez y no puede ser modificado posteriormente.

**Ejemplo:**

java

Copiar código

class Circulo {

final double PI = 3.14159;

public void mostrarConstante() {

System.out.println("El valor de PI es: " + PI);

}

}

En este ejemplo:

* El atributo **PI** es final, lo que significa que su valor no puede cambiar en ninguna parte del código.

**Tarea 3:**

Escribe una clase llamada **Rectangulo** con los atributos final **ancho** y **alto**. Luego, intenta cambiar el valor de uno de los atributos después de haber sido inicializado.

**5. Usos Comunes del Modificador final**

* **Protección de Clases y Métodos**: Utilizar final en una clase o método es útil cuando deseas **proteger la implementación** de la herencia o sobrescritura.
* **Definición de Constantes**: El uso más común de final en atributos es para definir **constantes**, especialmente en el caso de valores fijos como matemáticos (PI, e), configuraciones, o límites en el programa.

**Resumen de final**

| **Lugar donde se usa final** | **Efecto** |
| --- | --- |
| **Clase** | La clase no puede ser heredada por ninguna subclase. |
| **Método** | El método no puede ser sobrescrito en subclases. |
| **Atributo** | El valor del atributo no puede cambiarse una vez asignado. |

**Ejercicio Completo**

**Objetivo:**

Crea una clase llamada **Banco** con el atributo final **NOMBRE\_BANCO**, un método final llamado **mostrarInfo** que imprima información sobre el banco, y otra clase llamada **Sucursal** que intente heredar de Banco y sobrescribir el método mostrarInfo.

**Pasos:**

1. Declara la clase Banco con el atributo final **NOMBRE\_BANCO**.
2. Declara el método final mostrarInfo.
3. Intenta crear la subclase Sucursal que intente modificar este método y observa el error que genera.

**Preguntas y Respuestas**

**Pregunta 1**: ¿Qué error te muestra el compilador cuando intentas heredar de una clase final?  
**Respuesta**: El compilador muestra un error que indica que no se puede heredar de una clase marcada como final. Esto protege a la clase de ser extendida por otras.

**Pregunta 2**: ¿Qué error te muestra cuando intentas sobrescribir un método final en una subclase?  
**Respuesta**: El compilador muestra un error que indica que no se puede sobrescribir un método declarado como final. Esto asegura que el comportamiento del método no sea modificado por subclases.

**Conclusión**

Con el uso de la palabra clave **final** puedes proteger el comportamiento de tu código evitando que se modifiquen clases, métodos o atributos que deberían permanecer inalterables. Esta es una herramienta poderosa para asegurarte de que ciertos elementos de tu programa se comporten de manera predecible y constante.