



Recibe una cálida:

¡Bienvenida!

Te estábamos esperando 😊 

➤ El Paradigma de Orientación a Objetos

Plan formativo: Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



REPASO CLASE ANTERIOR

En la clase anterior trabajamos :

- ✓ Encapsulamiento
- ✓ Modificadores de acceso

LEARNING PATHWAY

4.5

Start! 🏁

El Paradigma de
Orientación a
Objetos

Accesadores y Mutadores

Getters y Setters

Cafetera

OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



Comprender cómo aplicar los métodos de acceso y modificación de objetos.





Rompehielo 🧊

Abriendo una caja fuerte: 🙌

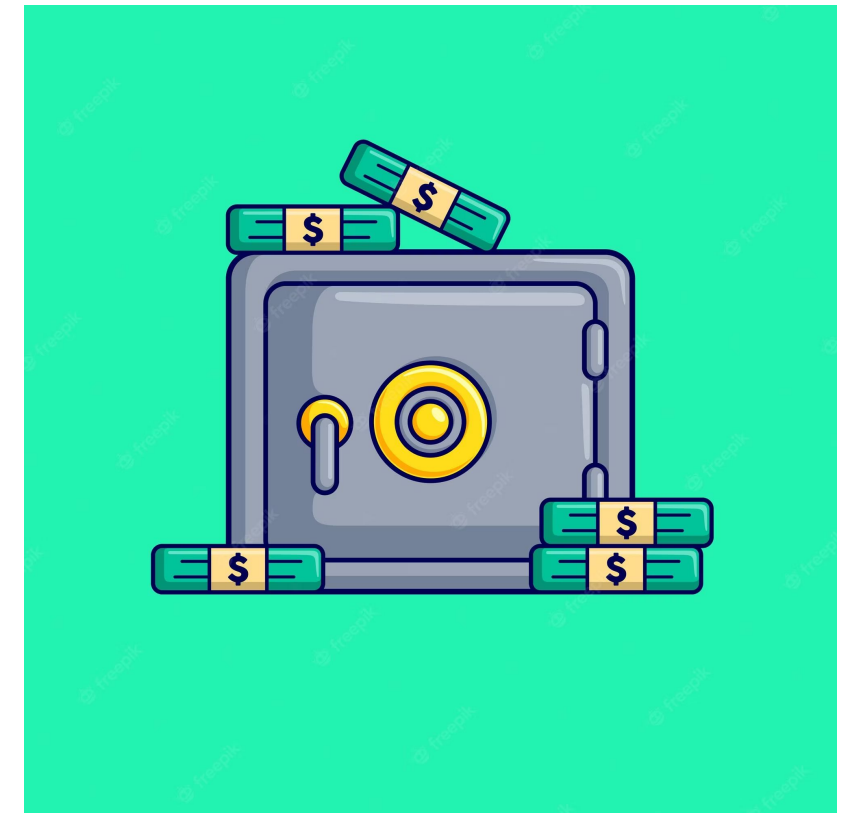
Esta caja fuerte es como una clase en POO, y dentro tiene atributos valiables como joyas, dinero, documentos importantes, etc.

Estos valores son como los atributos privados de una clase.

Para acceder a ellos se requiere una llave que abra la cerradura, y para modificar su contenido se requiere introducir la nueva información.

Respondamos en el chat:

¿Cómo abrirían la caja fuerte? ¿Cómo modificarían los valores internos?



➤ Accesadores y Mutadores



Accesadores



¿Qué son los getters?:

Los accesadores o métodos **get** se utilizan para **consultar el estado de un objeto**.

Un método get se declara **public** y a continuación se indica el tipo de dato que devuelve.



El método no recibe parámetros y en el cuerpo del método se utiliza **return** para devolver el valor correspondiente al atributo requerido.

```
public String getNombre() {  
    return nombre;  
}
```



Mutadores

¿Qué son los setters?:

Los mutadores o métodos **set** se utilizan para **modificar el estado de un objeto**.

Un método **set** se declara **public** y devuelve **void**.

La lista de **parámetros** incluye **el tipo y el valor a modificar** y en el cuerpo del método se asigna el parámetro de la declaración al atributo del objeto.

```
public void setNombre(String nombre) {  
    this.nombre= nombre;  
}
```

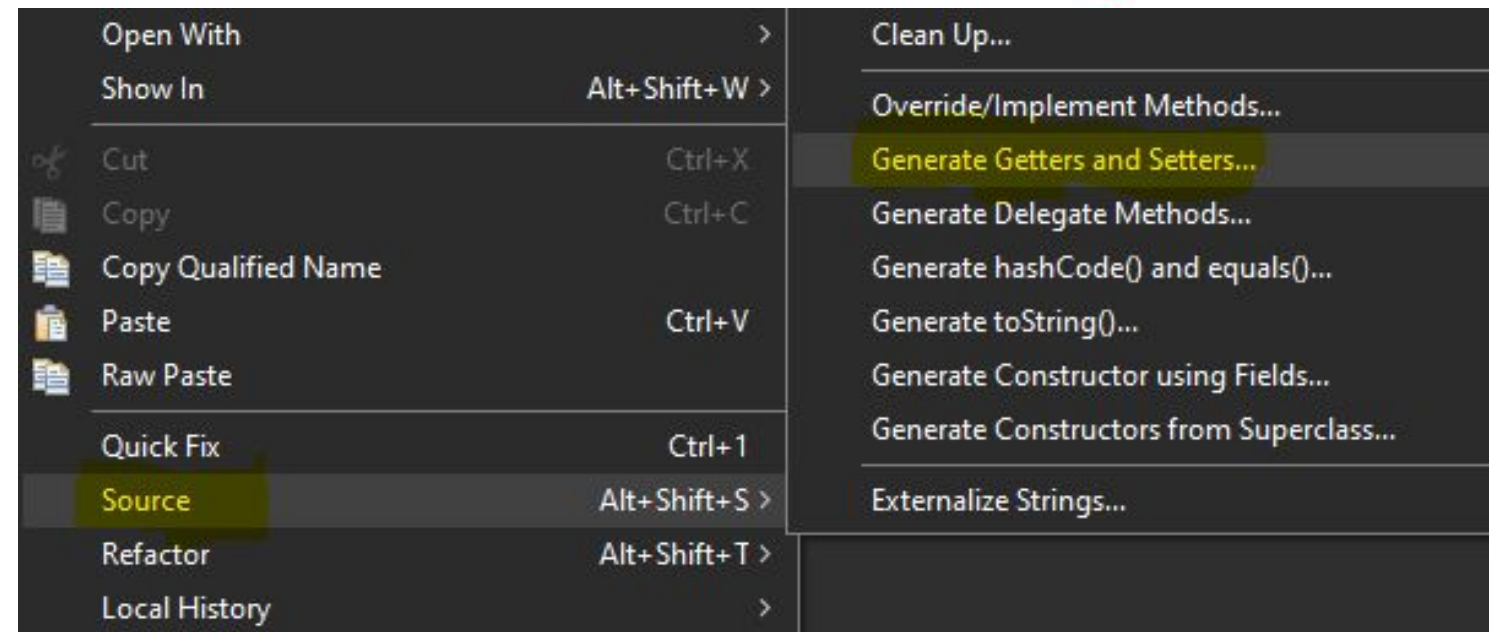




Accesadores y Mutadores

Los getters y setters, junto con los modificadores de acceso, permiten establecer un nivel adecuado de **control y seguridad en el acceso y modificación de los atributos** de una clase. Ayudan a mantener la coherencia y consistencia de los datos, además de proporcionar una interfaz más clara y segura para interactuar con los objetos de una clase.

Eclipse permite generar los get y set de cada atributo de manera automatizada haciendo **clic derecho** sobre el editor de texto, seleccionando **Source** -> **Generate Getters and Setters**.



Evaluación Integradora ✨

¿Listos para un nuevo desafío? En esta clase comenzamos a construir nuestro...

Trabajo Integrador del Módulo 💪

Iremos completándolo progresivamente clase a clase.



LIVE CODING

Ejemplo en vivo

Manipulando atributos:

En este ejercicio vamos a manipular objetos y atributos para analizar su comportamiento mediante los métodos de consulta y modificación.

- 1. En la clase llamada Cuenta (bancaria) que posee como mínimo los atributos: numeroCuenta , el nombre del cliente y el saldo actual.*

LIVE CODING

Ejemplo en vivo

Las operaciones asociadas a dicha clase son:

- 1. **Constructor** por defecto y constructor con saldo, número de cuenta y nombre.*
- 2. Agregar los métodos **getters** y **setters** correspondientes*
- 3. Método para **crear** un objeto Cuenta, pidiéndole los datos al usuario.*

Luego, vamos a realizar pruebas para acceder y modificar los datos de una cuenta y ver así el comportamiento de los métodos accesadores y mutadores





Ejercicio N° 1

Cafetera



Cafetera

¡Vamos a programar una cafetera! 🙌

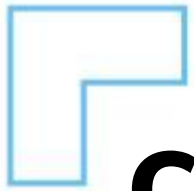
Es importante tener el control de la creación de objetos en nuestro programa, así como supervisar y asegurarnos de siempre estar trabajando con el mismo objeto. Es por esto que vamos a practicar la manipulación de atributos en el siguiente ejercicio:



Consigna: 📝

Desarrolle una clase **Cafetera** con los atributos **capacidadMaxima** (la cantidad máxima de café que puede contener la cafetera) y **cantidadActual** (la cantidad actual de café que hay en la cafetera).

Tiempo 🕒: 40 minutos



Cafetera

Métodos solicitados:



1. **Constructor predeterminado** o vacío.
2. **Constructor** con la capacidad máxima y la cantidad actual.
3. Métodos **getters** y **setters**.
4. Método **llenarCafetera()**: su función es hacer que la cantidad actual sea igual a la capacidad máxima.
5. Método **servirTaza(int)**: se pide el tamaño de una taza vacía, el método recibe el tamaño de la taza y simula la acción de servir la taza con la capacidad indicada. Si la cantidad actual de café “no alcanza” para llenar la taza, se sirve lo que reste en la máquina. El método le informará al usuario si se llenó o no la taza, y en caso de no haberse llenado informará en cuanto quedó la taza.
6. Método **vaciarCafetera()**: define la cantidad de café actual en cero.
7. Método **agregarCafe(int)**: se le pide al usuario una cantidad de café, el método lo recibe y se añade a la cafetera la cantidad de café indicada.



○

¿Alguna consulta?

+



RESUMEN

¿Qué logramos en esta clase?

- ✓ **Reconocer la importancia del encapsulamiento de información a través de la implementación de métodos accedadores y mutadores (get y set)**



#WorkingTime

Continuemos ejercitando

¡Antes de cerrar la clase! Te invitamos a: 🙌 🙌 🙌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
 - a. Material 1 (Foro)
 - b. *Lectura Módulo 4, Lección 5: página 8*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

¡Muchas Gracias!

Nos vemos en la próxima clase 🙌

