

Programación Orientada a Objetos 2:

Profundización de temas.

Resultados de Aprendizaje



Uso de 'this'

Desambiguar atributos, mejorar legibilidad y pasar instancias del objeto actual.



Constructores y Sobrecarga

Inicializar objetos de múltiples formas y reutilizar código eficientemente.



Sobrecarga de Métodos

Crear métodos adaptables con distintos parámetros para un código flexible.



Encapsulamiento y Control de Acceso

Entendimiento y uso del mismo

Entendiendo la Palabra Clave 'this'

La palabra clave **'this'** en POO es una referencia al objeto actual dentro de un método o constructor. Su uso es crucial para:

- **Desambiguación:** Distinguir entre atributos de instancia y parámetros con el mismo nombre.
- **Legibilidad:** Hacer el código más claro y fácil de entender.
- **Paso de Instancia:** Enviar el objeto actual como argumento a otros métodos.

Constructores y Sobrecarga: Flexibilidad en la Inicialización

Los constructores son métodos especiales utilizados para inicializar nuevos objetos. La sobrecarga de constructores permite definir múltiples formas de crear instancias de una clase, adaptándose a diferentes escenarios y necesidades de datos.

Inicialización

Establece el estado inicial de un objeto al momento de su creación.

Reutilización

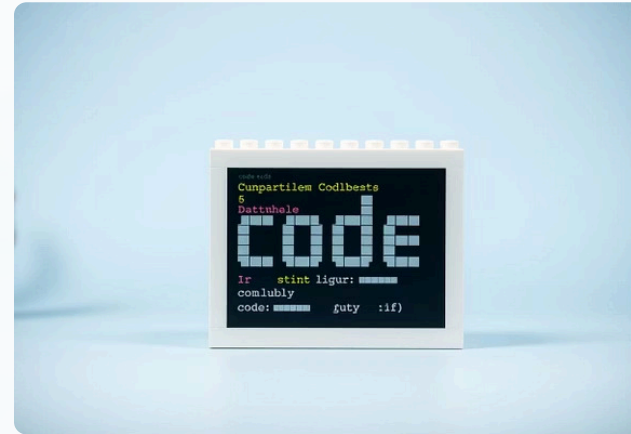
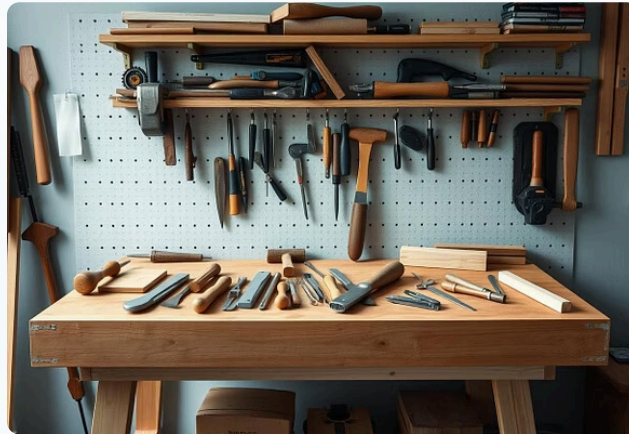
Evita código duplicado al permitir diferentes parámetros de entrada para un mismo proceso de construcción.

Flexibilidad

Ofrece al desarrollador opciones para crear objetos con diversos conjuntos de datos.

Sobrecarga de Métodos para Código Adaptable

La sobrecarga de métodos permite definir varios métodos con el mismo nombre dentro de una clase, pero con diferentes listas de parámetros. Esto mejora la modularidad y la adaptabilidad del código, permitiendo una mayor flexibilidad.



Cada método sobrecargado puede realizar una operación similar, pero en función de los argumentos proporcionados, el compilador determina cuál ejecutar, optimizando la reutilización.

Encapsulamiento y Control de Acceso



La idea es que para este punto sean capaces de utilizar modificadores de acceso (`public`, `private`, `protected`) y métodos de acceso (`getters` y `setters`) para garantizar la seguridad y consistencia de los datos en una aplicación orientada a objetos.

Los modificadores de acceso (**`public`**, **`private`**, **`protected`**) regulan esta visibilidad.

Los **`getters`** (métodos de acceso) y **`setters`** (métodos de modificación) son la forma en la que interactuamos con los datos internos del objeto.