

Programación orientada a objetos

Samuel David Contreras Saavedra - 20251005085

1.1 ¿En la programación orientada a objetos que es el constructor y que función cumple?

El constructor es un método súper especial que tiene el mismo nombre que la clase y no devuelve absolutamente nada, osea que lo que hace es devolver un valor valido para inicializar la clase. Algunas de sus funciones son:

Ensamblador inicial: Su trabajo principal es asegurarse de que el objeto esté bien configurado justo después de que lo creamos con la palabra clave new..

Inicializar atributos: Lo usamos para darle valores iniciales a las variables del objeto. Por ejemplo, si creas una clase Persona, en el constructor le puedes exigir que siempre se le pase un nombre y una edad.

Garantía de existencia: Siempre que creas un objeto, se llama a un constructor, ¡sí o sí! Si no defines uno, Java pone uno por ti (el constructor vacío).

1.2 ¿En la programación orientada a objetos que es un objeto y que función cumple?

Un objeto es una instancia de una clase. Es como una cosa específica y real modelada en el código. Si la clase Teléfono es el plano, un objeto es `Mi celular` que tengo en la mano. Sus funciones son:

Representar la realidad: Su función es representar una entidad específica que tiene datos (atributos, como el color o el tamaño) y **comportamientos** (métodos, como llamar o mandar mensajes).

Almacenar estado: El objeto guarda los valores actuales (su estado: `batería=85%`, `modo="silencio"`).

Ejecutar acciones: Permite realizar comportamientos definidos en la clase (`cargar()`, `vibrar()`).

Modularidad: Ayuda a dividir el programa en unidades pequeñas y manejables.

1.3 ¿En la programación orientada a objetos que es una clase y que función cumple?

La clase es la plantilla, molde o plano para construir objetos. Es solo la *definición* de lo que el objeto podrá ser; no es una "cosa" en sí misma. Cumple funciones como:

Definir la estructura: Su función es definir la estructura y el comportamiento común que tendrán todos los objetos creados a partir de ella.

Abstracción: Nos ayuda a modelar conceptos. En la clase `CuentaBancaria` definimos que todas las cuentas tendrán un saldo (atributo) y podrán depositar() (método).

Punto de partida: Es esencial. No puedes crear un objeto sin antes haber definido su clase.

1.4 ¿Qué diferencias destacadas se aprecian entre la OOP y la programación estructurada (secuencial)?

La diferencia más básica y esencial es el enfoque es que la programación estructurada se centra en las funciones o pasos lógicos que resuelven un problema, manteniendo los datos y el código separados mientras que la programación orientada a objetos se centra en los objetos es decir une los datos con las funciones que los manipulan, creando clases que modelan el mundo real. Esta unión hace que la POO sea mucho mejor para proyectos grandes, ya que facilita la reutilización de código y ofrece mayor seguridad de datos a través del encapsulamiento, algo que es más difícil de lograr en la programación secuencial.

1.5 ¿Para qué y por qué se usan modificadores de acceso?

Los modificadores de acceso se usan principalmente para implementar el encapsulamiento, se usan para controlar la visibilidad de los elementos de una clase (atributos y métodos), determinando qué partes del código pueden acceder a ellos. son importantes por que al hacer que los datos internos sean privados, aseguramos la integridad y seguridad del objeto, ya que solo los métodos de la propia clase pueden cambiarlos, esto evita manipulaciones externas no deseadas y ayuda a mantener el código organizado, separando claramente lo que es un detalle interno de lo que es la interfaz pública que otros objetos deben usar.

