

Características de la programación orientada a objetos (POO).

La programación orientada a objetos es un paradigma de programación basado en la creación de objetos para modelar entidades del mundo real. Los objetos tienen propiedades que los definen y comportamientos que definen cómo interactúan con otros objetos. Características clave de la programación orientada a objetos:

1. Abstracción:

Definición: La abstracción le permite centrarse en los aspectos importantes de un objeto, ignorando detalles menores. Parece crear un modelo simplificado del objeto real.

Ejemplo: Un coche real tiene un motor complejo con muchas piezas, pero cuando lo conduces, necesitas saber que tiene acelerador, frenos y volante. La abstracción te permite ignorar los detalles del motor y concentrarte en lo que es importante al conducir.

2. Encapsulación:

Definición: La encapsulación combina los datos (propiedades) y el comportamiento (métodos) de un objeto en una sola entidad. Oculta los detalles internos de un objeto y solo expone su interfaz pública para interactuar con él.

Ejemplo: una cápsula de medicamento tiene un recubrimiento (atributo) que controla el ingrediente activo (atributo) y la liberación (método). El usuario no necesita saber cómo funciona la cápsula, sólo cómo tomarla.

3. Herencia:

Definición: La herencia permite que un nuevo objeto (subclase) herede las propiedades y el comportamiento de un objeto existente (clase principal). Esto fomenta la reutilización del código y facilita la creación de jerarquías de clases.

Ejemplo: En la jerarquía de la clase Animal, la clase Carnívoros heredaría las características de la clase Animal (respirar y tener sangre caliente) y podría agregar sus propias características (por ejemplo, amamantar a sus crías).

4. Polimorfismo:

Definición: El polimorfismo permite que objetos de diferentes clases respondan de manera diferente al mismo mensaje. Esto significa que una misma operación puede tener diferentes operaciones según el tipo de objeto.

Ejemplo: En un videojuego, la clase Personaje puede tener diferentes subgrupos, como Héroe y Villano. Ambos pueden tener un método de "ataque", pero la ejecución exacta de ese ataque varía de una subclase a otra.

5. Clases y Objetos:

Clases: Las clases son patrones que definen las propiedades y el comportamiento de un tipo de objeto. Actúan como restos incorporados de objetos individuales.

Objetos: Los objetos son instancias individuales de una clase. Tienen propiedades y métodos definidos en la clase y pueden interactuar con otros objetos.

Ejemplo: los estudiantes definen atributos y comportamientos de los estudiantes, como clase, nombre, edad y especialización. Un objeto de clase representará a un estudiante específico con valores únicos para cada atributo.

6. Métodos y atributos:

Métodos: Los métodos son acciones que un objeto puede realizar. Están definidos en una clase y llamados por otros objetos.

Atributos: Los atributos son propiedades o características de un objeto. Representan el estado del objeto y pueden ser modificados por sus métodos.

Ejemplo: un objeto Estudiante puede tener métodos como "inscribirse", "aprobar el curso" y "obtener promedio". Los adjetivos pueden ser "nombre", "edad", "carrera" y "carrera".

7. Modularidad:

Definición: La modularidad divide un programa en módulos independientes y autónomos, cada uno con responsabilidades específicas. Esto mejora la organización del código, facilita el mantenimiento y fomenta la reutilización.

Ejemplo: una aplicación de gestión de biblioteca puede tener módulos para gestionar libros, autores, préstamos y usuarios. Cada módulo tendrá sus propias responsabilidades y estará conectado con otros a través de interfaces definidas.

8. Reusabilidad:

Definición: La reutilización le permite utilizar código que ya ha sido escrito en diferentes partes de un programa o incluso en diferentes programas. Esto ahorra tiempo y esfuerzo de desarrollo y reduce la duplicación de código.

Ejemplo: las clases y métodos bien diseñados se pueden reutilizar en diferentes partes de un programa o en programas completamente diferentes. Por ejemplo, varios programas que necesitan este tipo de funcionalidad pueden utilizar una clase de conexión de base de datos.