Consulta fundamentos de la programación

Juan Martín Betancur Uribe

1. Principios de la programación orientada a objetos

La programación orientada a objetos es un paradigma de programación que se basa en la creación de objetos que contienen tanto datos como métodos. Los principios fundamentales de la programación orientada a objetos son:

* Abstracción: es el proceso de identificar las características esenciales de un objeto y de ignorar las características no esenciales. En la programación orientada a objetos, se crea una clase para representar un objeto y se definen las características esenciales de ese objeto en esa clase.
* Herencia: es la capacidad de una clase para heredar los atributos y métodos de otra clase. En la programación orientada a objetos, se puede crear una clase derivada que hereda los atributos y métodos de una clase base.
* Polimorfismo: es la capacidad de un objeto para tomar muchas formas diferentes. En la programación orientada a objetos, se puede definir un método en una clase base y luego reemplazarlo en una clase derivada para que el método se comporte de manera diferente.

1. Comparativo entre las funciones en programación y la programación orientada a objetos

* Estructura del código: Las funciones son unidades de código independientes que se utilizan para realizar una tarea específica y se llaman desde otro lugar del código. En cambio, en la programación orientada a objetos, se crean clases que contienen tanto datos como métodos relacionados con un objeto específico.
* Reutilización de código: Las funciones son muy útiles para la reutilización de código, ya que una función que realiza una tarea específica se puede llamar desde diferentes partes del código. En la programación orientada a objetos, se puede reutilizar el código a través de la herencia y la composición, ya que las clases pueden heredar los métodos y atributos de otras clases y también pueden contener objetos de otras clases.
* Encapsulamiento: En la programación orientada a objetos, el encapsulamiento es una técnica para ocultar los detalles de implementación de un objeto y exponer solo una interfaz pública. En las funciones, no se aplica el encapsulamiento ya que toda la lógica y los detalles de implementación de la función se encuentran en la función misma.
* Polimorfismo: El polimorfismo es la capacidad de un objeto para tomar muchas formas diferentes. En la programación orientada a objetos, el polimorfismo se logra mediante la herencia y el uso de interfaces, lo que permite que los objetos se comporten de diferentes maneras según su tipo. En las funciones, no existe el polimorfismo ya que cada función realiza una tarea específica y no cambia su comportamiento según su entorno.
* En resumen, las funciones y la programación orientada a objetos son enfoques diferentes para el diseño y la estructura del código en programación. La programación orientada a objetos es más adecuada para proyectos más grandes y complejos, ya que permite una mejor organización del código y la reutilización de código a través de la herencia y la composición. Las funciones son útiles para tareas específicas y simples que se pueden reutilizar en diferentes partes del código.