

Principios Básicos de POO

Alumno: Jhojan Felipe Sánchez Zapata

Docente: Alejandro Rodas Vásquez

Universidad Tecnológica de Pereira

1 de octubre de 2024

Dentro del Paradigma Orientado a Objetos existen términos y conceptos fundamentales. Usted debe de investigar los siguientes:

Pregunta 1. ¿Qué es o qué significa el término Abstracción?

Solución. La abstracción es un principio de la programación orientada a objetos que consiste en simplificar y ocultar los detalles complejos de la implementación de un sistema o componente. Se enfoca en mostrar solo los aspectos esenciales y relevantes del objeto o proceso, permitiendo que los usuarios interactúen con una interfaz más simple sin preocuparse por cómo funciona internamente.

Pregunta 2. ¿Qué es Encapsulamiento?

Solución. El encapsulamiento es el principio que sugiere que los datos y los métodos que operan sobre esos datos deben estar agrupados en una misma entidad (una clase, por ejemplo). Además, se ocultan los detalles de implementación del mundo exterior, lo que significa que se controla el acceso a los datos a través de métodos específicos, generalmente mediante modificadores de acceso como `private`, `protected` y `public`.

Pregunta 3. ¿Qué es una Clase?

Solución. Una clase es una plantilla o molde que define las propiedades (atributos) y comportamientos (métodos) que los objetos creados a partir de ella tendrán. Es una estructura que agrupa datos y funciones relacionadas. Por ejemplo, una clase `Coche` podría tener atributos como `color` y `modelo` y métodos como `acelerar()` o `frenar()`.

Pregunta 4. ¿Qué es un Objeto?

Solución. Un objeto es una instancia de una clase. Es la representación concreta de una clase en la memoria. Cada objeto puede tener sus propios valores para los atributos definidos en la clase, pero compartirá la estructura y comportamiento que la clase define. Por ejemplo, si `Coche` es una clase, un objeto de esta clase podría ser `miCoche`, con valores específicos como `color rojo` y `modelo "2020"`.

Pregunta 5. ¿De qué se trata el “Principio de responsabilidad única”?

Solución. El principio de responsabilidad única establece que cada clase debe tener una única responsabilidad o razón para cambiar. Es decir, cada clase debe encargarse de una única tarea o funcionalidad en el sistema. Esto mejora la mantenibilidad y la cohesión del código, haciendo que sea más sencillo identificar cambios y asegurar que cada clase se especialice en un aspecto específico del sistema.