

Características POO

Abstracción: las características específicas de un objeto, aquellas que lo distinguen de los demás tipos de objetos y que logran definir límites conceptuales respecto a quién está haciendo dicha abstracción del objeto. Una abstracción se enfoca en la visión externa de un objeto, separa el comportamiento específico de un objeto, a esta división que realiza se le conoce como la barrera de abstracción, la cuál se consigue aplicando el principio de mínimo compromiso.

Encapsulamiento: Agrupación de datos, atributos y el código que opera sobre ellos, dentro de la unidad llamada clase. Esta agrupación oculta la implementación interna de la clase permitiendo que sólo exponga los métodos públicos para interactuar con los datos de forma segura y controlada.

Beneficios:

Protección de datos: Limita el acceso a los datos sensibles, evitando modificaciones no deseadas

Simplicidad: oculta la complejidad interna, facilitando el uso de la clase

Modificaciones seguras<. Permite realizar cambios internos sin afectar el uso externo de la clase.

Herencia: Permite a una clase heredar las características de otra clase. Esto creas una jerarquía de clases donde las clases derivadas pueden especializar o extender la funcionalidad de la clase base

Beneficios:

Reutilización de código: Evita la duplicación de código, promoviendo la eficiencia y el mantenimiento del código.

Jerarquías de clases: Permite organizar las clases en relaciones de “es-un” y “es-una”-instancia-especial-de” reflejando relaciones del mundo real.

Polimorfismo:

El polimorfismo es la capacidad de un objeto de tomar diferentes formas o responder a la misma llamada de manera diferente según su tpo real. En la POO, el polimorfismo se implementa principalmente a través de:

Sobrecarga de métodos: Permite que un método tenga diferentes implementaciones en distintas clases derivadas.

Despacho dinámico: Determina la implementación del método a llamar en tiempo ejecución en función del tipo real del objeto

Beneficios:

Flexibilidad: Permite escribir código más flexible y adaptable a diferentes situaciones.

Clases y objetos:

Clase: En un modelo o plantilla que define la estructura y el comportamiento de un conjunto de objetos. Especifica los atributos y métodos que comparten los objetos de esa clase .

Objeto: Es una instancia individual de una clase. Se crea a partir de la clase y posee los atributos y métodos definidos en ella. Los objetos encapsulan datos y comportamiento, y pueden interactuar entre sí a través de sus métodos.

Métodos y atributos:

Método: Es una función asociada a una clase que define un comportamiento específico. Los métodos operan sobre los datos de la clase y pueden ser invocados desde objetos de la clase

Atributo: Es una variable asociada a una clase que representa una característica o propiedad de los objetos de esa clase. Los atributos almacenan datos que definen el estado de un objeto.

Modularidad:

La modularidad es un principio fundamental de la POO que se refiere al diseño de software en módulos independientes y bien definidos. Cada módulo encapsula una unidad funcional específica y tiene una interfaz clara para interactuar en otros modelos.

Beneficios:

Organización: Permite el código de manera clara y estructurada facilitando su comprensión y mantenimiento.

Reutilización: Promueve la reutilización de módulos en diferentes proyectos, reduciendo la duplicación del código

Reusabilidad:

La reusabilidad es la capacidad de utilizar código existente en diferentes partes de un programa o en distintos proyectos. En la POO, la reusabilidad se fomenta a través de:

Encapsulamiento: Permite crear componentes reutilizables que encapsulan datos y comportamiento.

Herencia: Facilita la creación de clases reutilizables que heredan funcionalidad de otras clases.

Modularidad: Promueve el diseño de módulos independientes y reutilizables.

Beneficios de la reusabilidad:

Productividad: Aumenta la productividad del desarrollo al reducir la necesidad de escribir código nuevo desde cero.

Mantenimiento: Simplifica el mantenimiento del código al utilizar componentes probados y confiables.

