

Taller POO

1. Abstracción

Proceso para descartar toda información que no sea importante en un contexto particular también, captar características y funcionalidades de un objeto representados por atributos y métodos. Por ejemplo: en una persona las características el nombre, apellidos, edad, dirección y funcionalidades hablar, comer, dormir, trabajar.

2. Encapsulamiento

Proceso para almacenar datos dentro de una estructura ocultando la implementación del objeto, existen tres modificadores: cerrado, protegido, abierto. Por ejemplo: una persona tiene como atributos nombre, fecha de nacimiento y edad y como método registrar persona y calcular edad

3. Herencia

Mecanismo que permite derivar una clase a otra clase (clases que serán hijos y clases que serán padres). Por ejemplo: los mamíferos tienen unas características, como meses de gestación, pechos de las hembras y funcionalidades como dar luz, amamantar, cuando se realice la clase perro tenemos que implementar estos métodos y atributos

4. Polimorfismo

Permite que los objetos de cada clase respondan a un mensaje de distintas maneras. Por ejemplo: existe una clase animal y subclases de perro y gato y ambas tienen un método llamado sonido que devuelve el sonido que hace cada animal

5. Clases y objetos

Los objetos se crean a partir de diversas clases, sirven de plantillas para crear o personalizar objetos, Los objetos tienen atributos que muestran las características y propiedades de esta, Las clases se utilizan para definir la estructura y comportamiento de esta actuando como plantilla o modelo a partir de los objetos o características un ejemplo: Definir la clase coche y que tenga atributos como "marca" "modelo" "color" entre otras cosas

6. Métodos y atributos

La programación orientada a objetos consiste en la creación de código modular y reutilizable mediante el uso de entidades u objetos que representen los datos a manipular. De este modo, cada objeto tendrá un determinado estado (activo, inactivo, etc.), un cierto comportamiento (ejecutará un método o función) y estará diferenciado del resto de los de su especie por medio de su identidad en forma de variables o atributos. A la entidad o modelo a partir del cual se instancian los diferentes objetos del mismo tipo se le denomina clase.

7. Modularidad

El modularidad es, en programación modular y más específicamente en programación orientada a objetos, la propiedad que permite subdividir una aplicación en partes más pequeñas (llamadas módulos), cada una de las cuales debe ser tan independiente como sea posible de la aplicación en sí y de las partes restantes.

8. Reusabilidad

Significa que una vez este creado, escrito y depurado una clase, se puede hacer una disposición de otros programadores, haciendo que sirva de biblioteca para demás idiomas de programación