

Programación Orientada a Objetos (POO)

La **programación orientada a objetos** es un paradigma de programación que basa en los “objetos” para representar elementos en el programa a desarrollar. Dichos objetos son entidades que pueden almacenar información y realizar acciones.

Conceptos fundamentales:

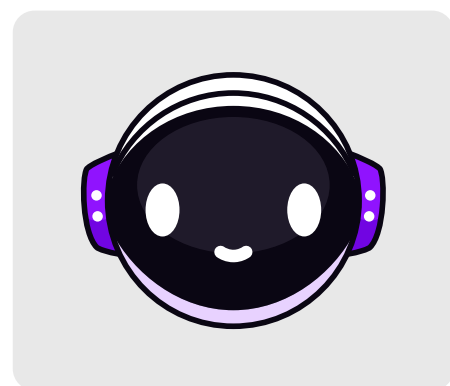
CLASE



Ej.: mascota corporativa.

Una clase es una plantilla que define las propiedades y métodos de los objetos que se crean a partir de ella.

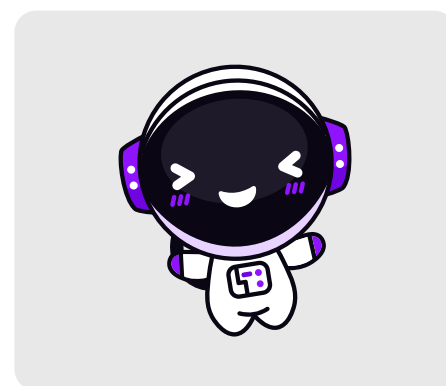
OBJETO



Ej.: mascota de DevTalles.

Es la instancia de la clase, este tendrá un conjunto de propiedades y comportamiento (métodos).

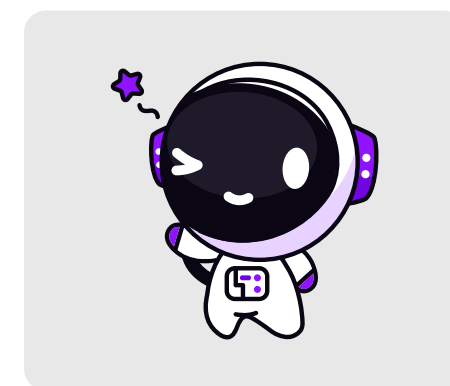
PROPIEDADES



Ej.: Devi, amigable, divertido.

Son contenedores de datos asociados a un objeto que pueden ser visibles externamente y modificados por métodos.

MÉTODOS



Ej.: comunicar, entretener.

Son funciones definidas dentro de una clase.

CONSTRUCTOR



Ej.: se encargará de inicializar los atributos de una nueva mascota.

Es un método especial que se invoca al crear un objeto y permite inicializar valores en la instancia de la clase, pudiendo incluir parámetros si es necesario.

Conceptos fundamentales:

- **Abstracción:** Permite definir las características de cada objeto por medio de las clases, para que de esta manera los objetos puedan interactuar sin que sea necesario conocer los detalles de su funcionamiento interno.
- **Encapsulamiento:** Nos permite proteger y ocultar información así como el comportamiento de los objetos. Esto permite evitar que se altere el
- **Herencia:** Permite que la funcionalidad de otra clase existente sea reutilizada, de tal manera que una clase hija extiende la funcionalidad de una clase padre.
- **Polimorfismo:** Es la capacidad de que un objeto de tener un determinado comportamiento dependiendo de su contexto. Por ejemplo, una clase hija puede implementar o sobrescribir un método definido en una clase padre.



Links de referencia

1



Cursos relacionados



Más recursos

Síguenos:  /Fernando_Her85  /DevTalles  /DevTalles

www.devtalles.com

