**🔷 1. Clase**

Una **clase** es una **plantilla o modelo** que define las propiedades (atributos) y comportamientos (métodos) que tendrán los objetos creados a partir de ella.

Ejemplo: Una clase Coche podría tener atributos como color y velocidad, y métodos como acelerar() o frenar().

**🔶 2. Objeto**

Un **objeto** es una **instancia de una clase**. Es una entidad concreta que tiene valores específicos en sus atributos y puede ejecutar métodos definidos por su clase.

Ejemplo: Si Coche es la clase, un objeto podría ser un coche rojo que va a 50 km/h.

**🟢 3. Método**

Un **método** es una **función definida dentro de una clase** que describe un comportamiento del objeto.

Ejemplo: mover(), calcularArea() o saludar().

**🟠 4. Instancia**

Una **instancia** es **cuando se crea un objeto** de una clase. El objeto es la instancia concreta de esa plantilla.

Ejemplo:

cpp

CopiarEditar

Circulo c(5.0); // c es una instancia de la clase Circulo

**🔵 5. Abstracción**

La **abstracción** es el proceso de **ocultar los detalles complejos** y mostrar solo lo esencial del objeto.

Ejemplo: Cuando usas un método abrirPuerta(), no necesitas saber cómo funciona el mecanismo interno.

**🟣 6. Encapsulamiento**

El **encapsulamiento** consiste en **proteger los datos** de una clase restringiendo el acceso directo desde fuera, usando modificadores como private, protected y public.

Ejemplo: Acceder a un atributo solo a través de un método get o set.

**🔻 7. Constructor**

Un **constructor** es un **método especial** que se ejecuta automáticamente cuando se crea un objeto. Sirve para inicializar los atributos.

Ejemplo:

cpp

CopiarEditar

class Circulo {

public:

Circulo(double r) {

radio = r;

}

};

**🟡 8. Herencia**

La **herencia** permite que una clase (llamada **subclase** o **hija**) **herede atributos y métodos de otra clase** (llamada **superclase** o **padre**).

Ejemplo: Una clase Perro puede heredar de Animal.

**🔺 9. Polimorfismo**

El **polimorfismo** permite que **una misma función** tenga **comportamientos diferentes dependiendo del tipo de objeto**.

Ejemplo: El método dibujar() puede comportarse diferente si lo llama un Circulo o un Rectángulo.

¿Te gustaría ver estos conceptos aplicados en un ejemplo concreto con código en C++ o en otro lenguaje?

Principio del formulario