programación orientada a objetos

La **programación orientada a objetos (POO)** es un paradigma de programación que se basa en el uso de *objetos* para modelar entidades y conceptos del mundo real en el software.

Abstracción

- Es el proceso de simplificar un sistema complejo modelando solo los aspectos relevantes para un determinado problema. Permite enfocarse en lo esencial sin tener que preocuparse por detalles internos innecesarios.

Abstracción

```
//atributos
4 references
public String nombre {  get; set; }
5 references
public int edad {  get; set; }
4 references
public string cedula { get; set; }
```

Encapsulación

- Consiste en agrupar datos y métodos que operan sobre esos datos en una única unidad (la clase), y controlar el acceso a ellos mediante modificadores de acceso (como `public`, `private` y `protected`).

Encapsulación

```
//atributos
4 references
private String nombre {  get; set; }
5 references
private int edad {  get; set; }
4 references
private string cedula { get; set; }
```

Herencia

- Permite que una clase (la clase derivada) herede atributos y métodos de otra clase (la clase base).

Herencia

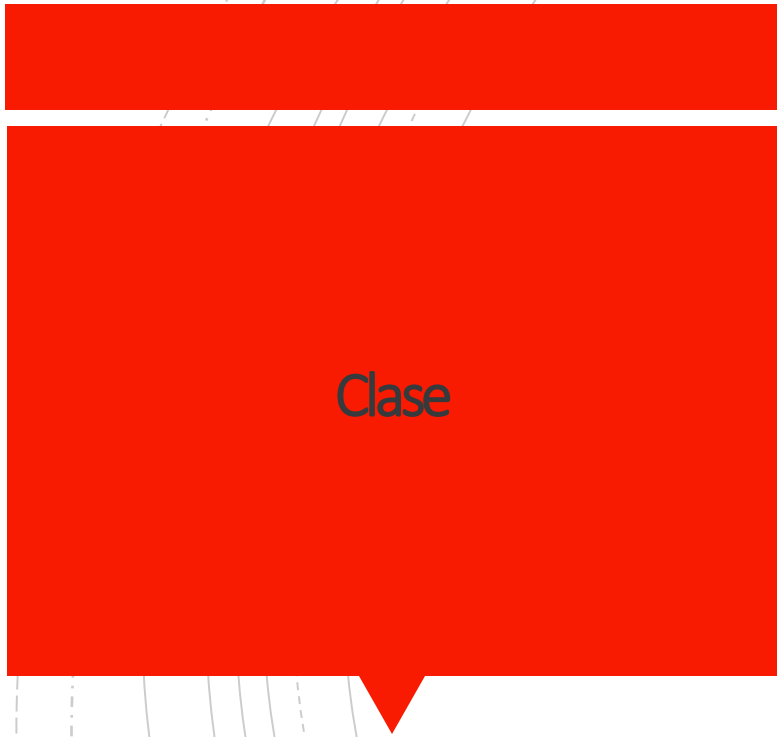
```
2 references
internal class Cuenta : Persona
{
    // Atributos de la clase
    3 references
    public Persona Titular { get; set; }
    5 references
    public double Balance { get; set; }
```

Polimorfismo

- Permite que objetos de diferentes clases puedan ser tratados de la misma forma si comparten una clase base o implementan la misma interfaz.

Polimorfismo

```
//Contructor sin parametros
0 references
public Persona() {
    nombre = "";
    edad = 0;
    cedula = "";
}
//Contructor con nombre y edad
0 references
public  Persona(String nombre, int edad) {
   this.nombre =nombre;
    this.edad = edad;
    this.cedula = "";
}
//Contructor con todos los datos como parametros
0 references
public Persona(string nombre, string cedula, int edad)
{
    this.nombre = nombre;
    this.edad = edad;
    this.cedula = cedula;
}
```



- Es una plantilla o molde que define las propiedades y métodos comunes para un conjunto de objetos.

Objeto

- Es una instancia de una clase. Por ejemplo, un objeto `miCoche` de la clase `Coche` puede tener el color "rojo", la marca "Toyota" y el modelo "Corolla".

clase abstracta

- Es una clase que está diseñada para servir como base para otras clases, pero **no se puede instanciar directamente**.

interface

- Una **interface** en C# es un contrato que define un conjunto de **métodos, propiedades, eventos o indexadores** que una clase (o estructura) debe implementar.