

Classes Abstradas e Interfaces

Estrutura	Quando Usar (Foco)	O Que Oferece	Exemplo Clássico
Interface	Quando você precisa definir um contrato de comportamento (o "Pode fazer") que deve ser seguido por classes não relacionadas ou quando precisa de herança múltipla de tipos .	Define apenas o <i>quê</i> deve ser feito (métodos obrigatórios) e não <i>como</i> (sem implementação de estado de instância). Permite que uma classe implemente múltiplas interfaces.	<code>IImprimivel</code> , <code>IConectavel</code> , <code>ITransformavel</code> .
Classe Abstrata	Quando você precisa definir uma estrutura base (o "É um") para uma hierarquia de classes relacionadas , e precisa compartilhar código e estado (campos concretos).	Permite a definição de métodos abstratos (obrigatórios) e métodos/campos concretos (implementação compartilhada). Uma classe só pode herdar de uma única classe abstrata.	<code>Animal</code> , <code>Funcionario</code> , <code>Veiculo</code> .

Regras de Ouro

1. Herança Múltipla de Comportamento? Use Interface.

- *Exemplo:* Um `Pato` é um `Animal` (herança), mas também `IPodeNadar` e `IPodeVoar` (interfaces múltiplas).

2. Precisa Compartilhar Implementação (código e estado)? Use Classe Abstrata.

- *Exemplo:* Todo `Funcionario` precisa do campo `matricula` e do método `exibirDados()` já implementado. Apenas o cálculo do salário varia (`calcularSalario()` seria abstrato).

3. Relação de Tipos ("É um")? Use Classe Abstrata.

- *Exemplo:* Um `Caminhao` é um `Veiculo`.

4. Capacidade de Ação ("Pode fazer")? Use Interface.

- *Exemplo:* Um `Carro` pode ser `ILigavel` (Interface).