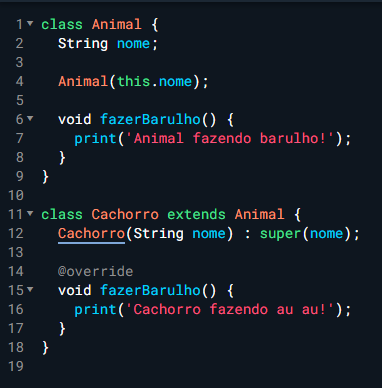
Aula 18:

Orientação a Objetos - Parte 2

Nesta aula, continuaremos nossa jornada pelo mundo da programação orientada a objetos em Dart. Vamos explorar conceitos avançados, incluindo herança, polimorfismo e encapsulamento. Ao final desta aula, você terá uma compreensão sólida desses conceitos e poderá aplicá-los em seus próprios projetos.

Herança em Dart:

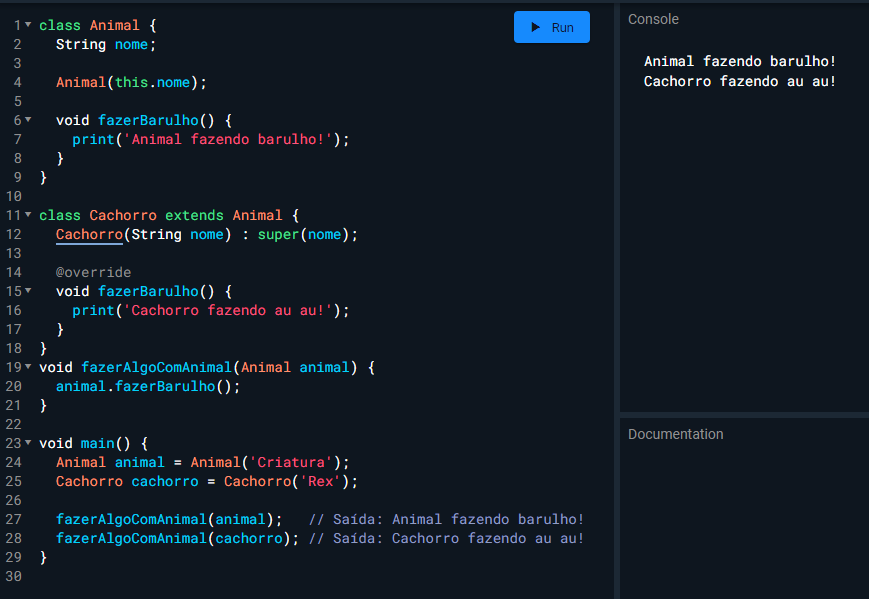
A herança é um conceito fundamental em programação orientada a objetos. Em Dart, você pode criar uma hierarquia de classes onde uma classe mais especializada herda características de uma classe mais geral. Vamos criar uma classe `Animal` e uma classe `Cachorro` que herda de `Animal`.



Neste exemplo, a classe `Cachorro` herda da classe `Animal` e substitui o método `fazerBarulho`. O uso de `@override` indica que estamos substituindo um método da classe pai.

Polimorfismo em Dart:

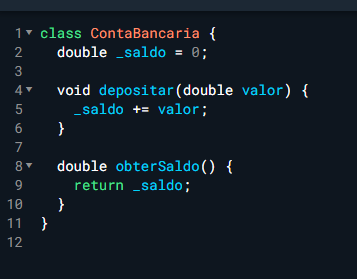
O polimorfismo permite que objetos de diferentes classes sejam tratados de maneira uniforme. Em Dart, isso é alcançado através da criação de métodos com parâmetros do tipo da classe base.



A função `fazerAlgoComAnimal` aceita qualquer objeto do tipo `Animal` ou suas subclasses. Isso ilustra o polimorfismo, onde diferentes objetos respondem de maneira única a um método comum.

Encapsulamento em Dart:

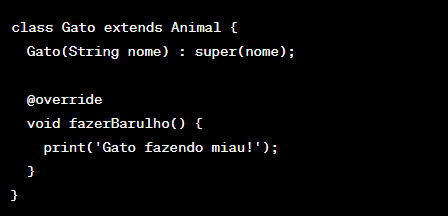
O encapsulamento envolve ocultar os detalhes internos de uma classe e fornecer uma interface para interação externa. Em Dart, isso é alcançado usando modificadores de acesso como `public`, `private` e `protected`.



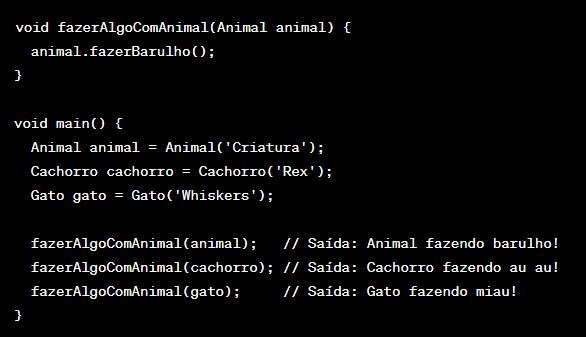
No exemplo acima, `\_saldo` é uma variável privada acessível apenas dentro da classe `ContaBancaria`. Métodos como `depositar` e `obterSaldo` fornecem uma interface controlada para interagir com essa variável privada.

Exercícios Práticos:

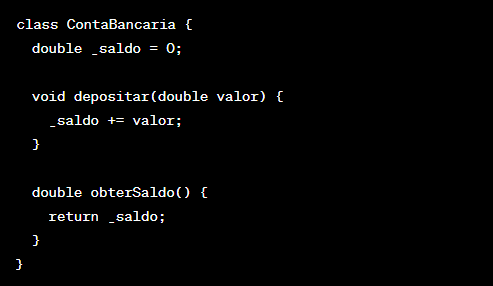
1. Crie uma classe `Gato` que herda de `Animal`. Implemente o método `fazerBarulho` para imprimir "Gato fazendo miau!".



1. Experimente o polimorfismo chamando `fazerAlgoComAnimal` com uma instância de `Gato`.



3. Aplique encapsulamento em uma classe de sua escolha, definindo variáveis privadas e fornecendo métodos públicos para interação externa.



Ao final desta aula, você terá uma compreensão sólida dos conceitos avançados de orientação a objetos em Dart. Pratique esses conceitos em seu próprio código para fortalecer seu entendimento. Na próxima aula, mergulharemos na manipulação de strings em Dart.