ABSTRAÇÃO / ENCAPSULAMENTO / HERANÇA / POLIMORFISMO

CONTINUAÇÃO DA AULA DE POO

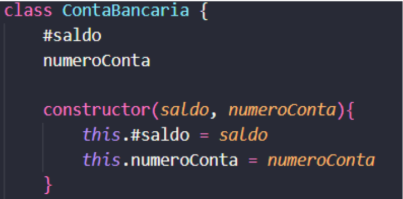
**ABSTRAÇÃO**

Ideia de ocultar os detalhes complexos e mostrar apenas o que é relevante para o usuário  
EX: Você tem um carro, mas não precisa saber dos detalhes sobre mecanismos de funcionalidades dele específicos para conseguir dirigi-lo

**ENACAPSULAMENTO**  
Significa proteger os dados internos de um objeto permitindo que eles sejam acessados e modificados apenas por métodos da própria classe, e outros métodos conhecidos como getters e setters.

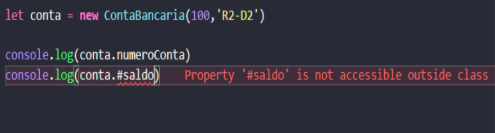
**Encapsulamento Atributo Privado**

Para privar algo no JavaScript é preciso colocar um # antes do nome do atributo, assim ele será um atributo privado.

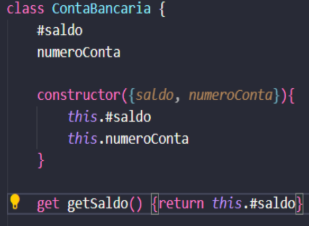


**Acessando atributo privado**

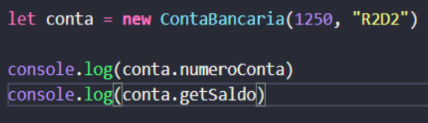
Abaixo estamos estanciando nossa classe criando um novo objeto e tentar acessar o atributo privado, onde um erro será exibido, assim para acessar temos que definir um getter para ele



**Getter**É um método especial usado para acessar (ler) o valor de um atributo de forma controlada, geralmente usado para retornar um valor, com ele é possível realizar cálculos ou transformar os dados antes de retorná-los



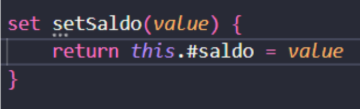
Acessando o atributo via Getter, vamos acessa-lo como se fosse um atributo



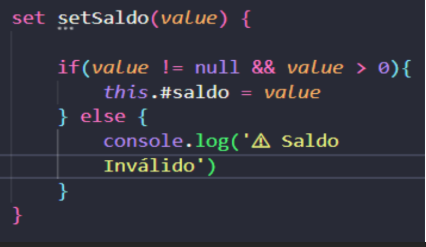
O getter não altera valores, quem faz essa função é o Setter

SETTER

Setter é um método especial para alterar ou validar o valor de um atributo de forma controlada, permite armazenar ou transformar os dados antes de armazená-los garantindo que apenas os valores corretos sejam atribuídos ao atributo

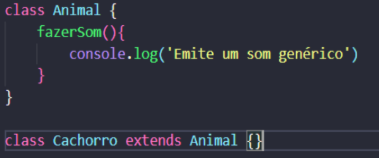


**Adicionando validações**Vamos implementar as validações no setter da nossa classe para validar se o valor inserido pelo setter é válido.



**HERANÇA**

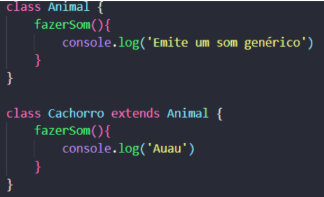
A classe animal contém um método “fazerSom”, a classe Cachorro precisa utilizar esse método para reaproveitar o método que já existe basta a classe Cachorro EXTENDER a classe animal, assim tendo o método “fazerSom”



Observe que mesmo a classe Cachorro estando vazia, ela consegue acessar os métodos e atributos da classe pai(Classe Animal)

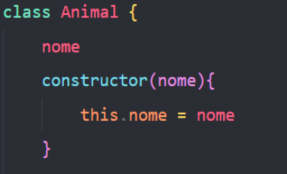
Além de conseguir reutilizar os métodos já prontos da classe pai, também é possível alterar esses métodos de acordo com que a classe pai precisa

Para isso bastar criar uma nova função com o mesmo nome da classe



Super (Classe pai)

O super é uma palavra reservada que é usada dentro de classes filhas para chamar métodos ou construtores pai

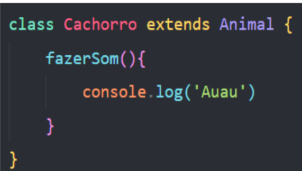


POLIMORFISMO

Significa “muitas formas”   
Ele permite que o objeto seja tratado como um tipo mais genérico, mas ainda assim execute comportamentos específicos de sua classe concreta   
Ele permite que diferentes classes implementam um mesmo método de maneira distintas.

OVERRIDE

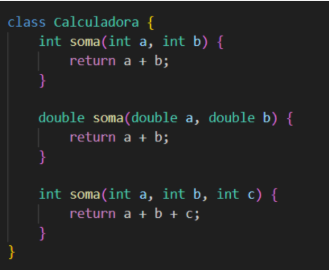
* Sobrescrita: Quando uma classe filha reimplementa um método de sua classe pai, ou seja, ela substitui o comportamento herdado



Criando acima um método com o mesmo nome para que o polimorfismo seja aplicado

OVERLOAD  
Polimorfismo de sobrecarga acontece quando um mesmo método tem várias versões diferenciadas pelo número ou tipo de parâmetros.

Segue o exemplo abaixo em JAVA:



Segue o exemplo abaixo em JAVASCRIPT:

