**COMPREENDENDO ORIENTAÇÃO A OBJETOS**

Apresentação do curso

Paradigmas e Pilares

DECLARATIVE

LOGIC FUNCTIONAL DATABASE

DATAFLOW

IMPERATIVE

PROCEDURAL/OBJECT ORIENTED/PARALLELPROCESSING

PARADIGMS

IMPERATIVO FOCA COMO VAMOS RESOLVER O PROBLEMA;

DECLARATIVO FOCA NO QUE VAMOS FAZER;

Na orientação a objetos, os programas são “objetos” que possuem uma série de propriedades.

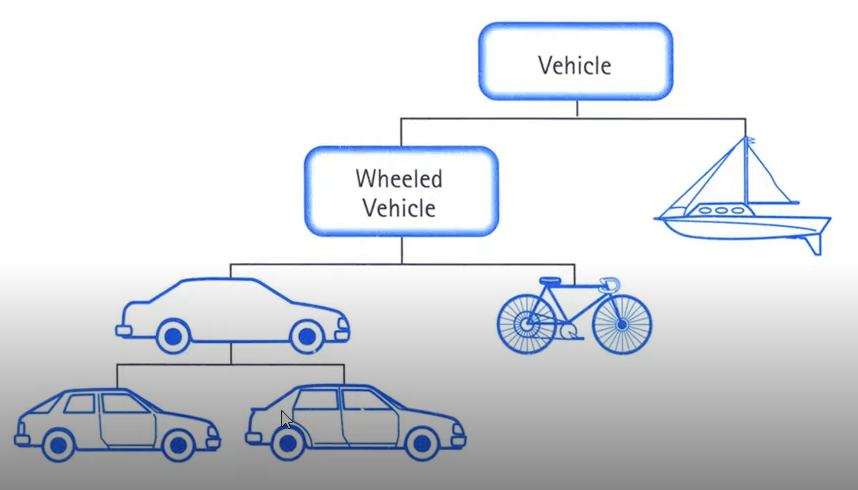
Pilares:

* Herança

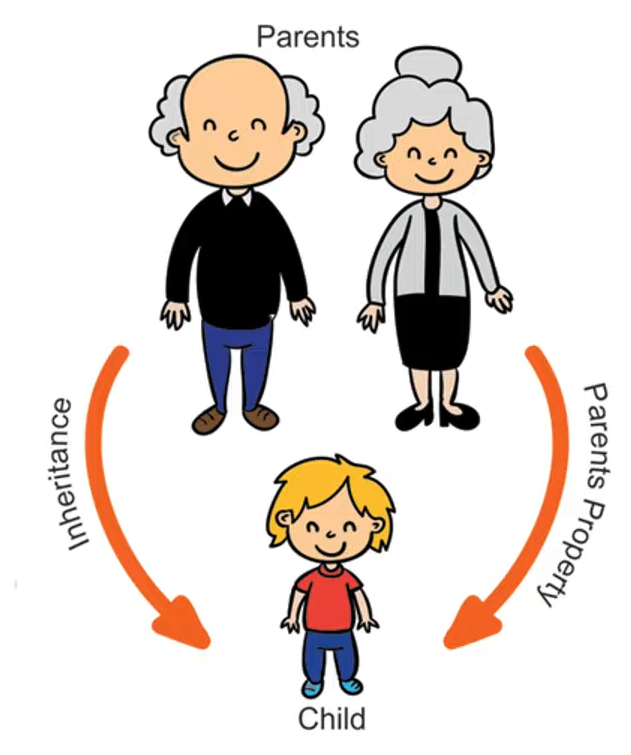


* Polimorfismo
* Encapsulamento
* Abstração

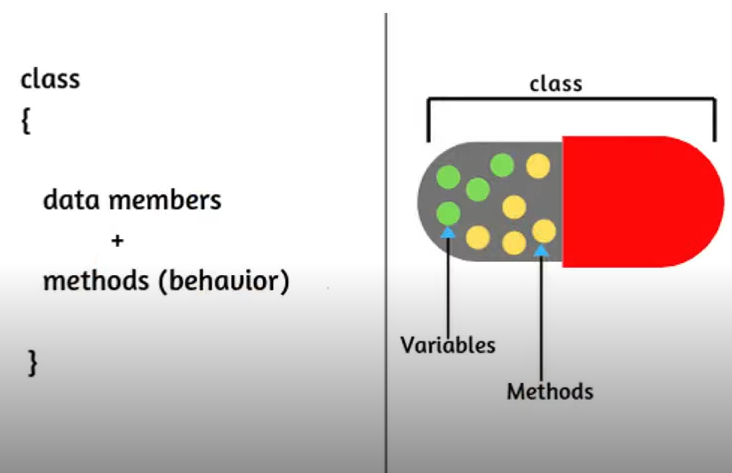
Abstração – Processo mental que consiste em isolar um aspecto determinado de um estado relativamente complexo, a fim de simplificar a sua avaliação, classificação ou para permitir a comunicação do mesmo.



Herança – O objeto filho herda propriedades e métodos do objeto pai.



Encapsulamento – Cada classe tem propriedades e métodos independentes do restante do código



Polimorfismo – Objetos podem herdar a mesma classe do pai, mas se comportarem de forma diferente quando invocamos seus métodos



**ORIENTAÇÃO A OBJETOS EM JAVASCRIPT (OOJS)**

Protótipos e Classes

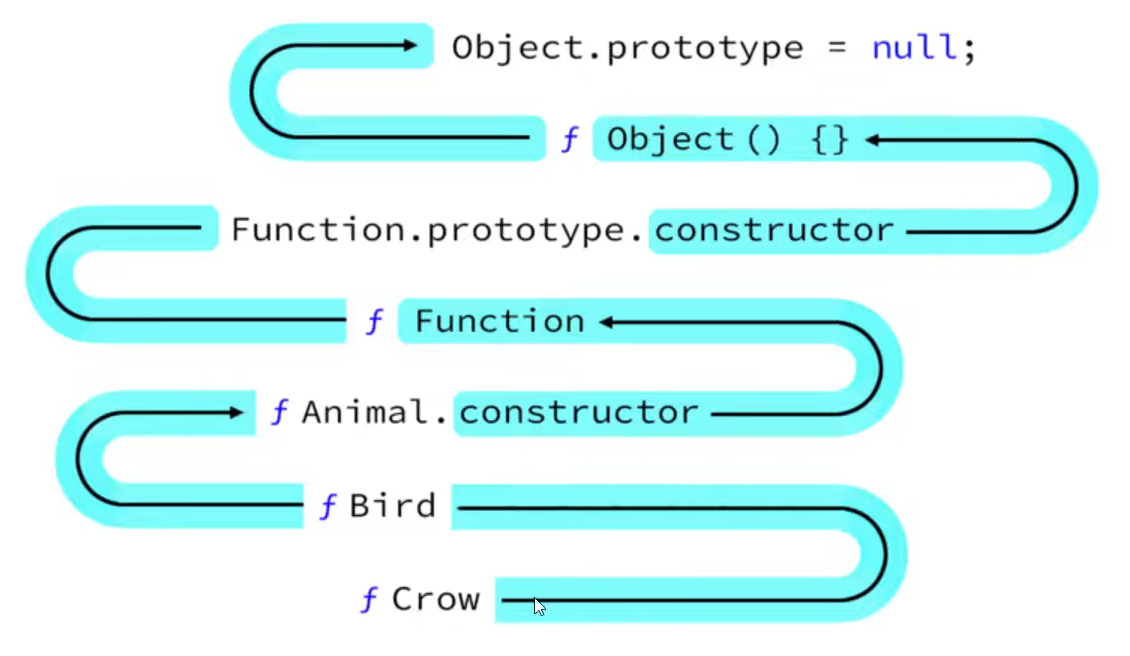
**PROTÓTIPOS**

Todos os objetos Javascript herdam propriedades e métodos de um prototype.

O objeto object.prototype está no topo desta cadeia.



Cadeia de protótipos (prototype chain)

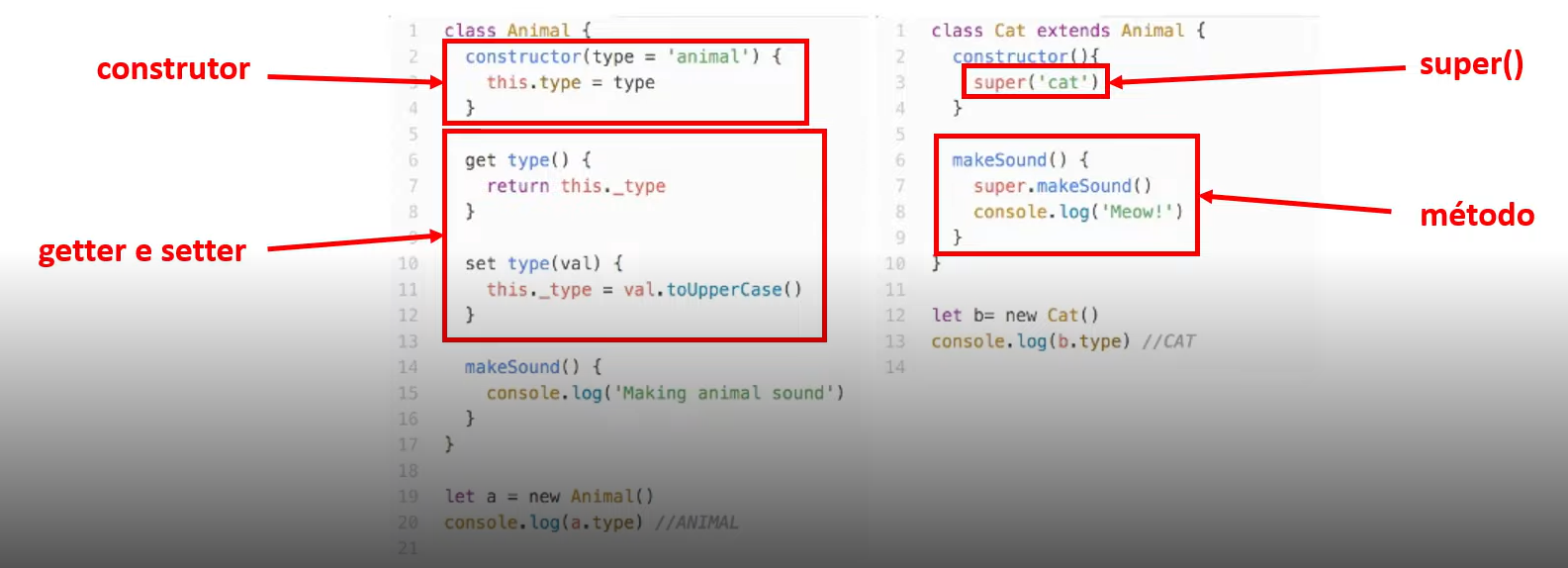


**CLASSES**

Syntatic sugar: uma sintaxe feita para facilitar a escrita



Javascript não possui classes nativamente. Todas as classes são objetos e a herança se dá por protótipos.



Atividade prática

**Atividade: Conta Bancária**

Nesta atividade, vamos testar os conceitos de Orientação a Objetos simulando a criação de diversos tipos de contas bancárias e operações disponíveis em cada uma.

1. Crie a classe ContaBancaria, que possui os parâmetros agencia, numero, tipo e saldo;
2. Dentro de ContaBancaria, construa o getter e o setter de saldo;
3. Dentro de ContaBancaria, crie os métodos sacar e depositar;
4. Crie uma classe-filha chamada ContaCorrente que herda todos esses parâmetros e ainda possua o parâmetro cartaoCredito;
5. Ainda em ContaCorrente, construa o getter e o setter de cartaoCredito;
6. Ainda em ContaCorrente, faça com que o tipo seja 'conta corrente' por padrão;
7. Crie uma classe-filha chamada ContaPoupanca que herda todos os parâmetros de ContaBancaria;
8. Crie uma classe-filha chamada ContaUniversitaria que herda todos os parâmetros de ContaBancaria;
9. Faça com que o método saque de ContaUniversitaria apenas seja capaz de sacar valores **menores que 500 reais**.

##### Quais são os quatro pilares da Orientação a Objetos?

Abstração, Herança, Encapsulamento e Polimorfismo

##### Qual a melhor definição de Herança?

Classes descendentes herdam características de suas classes ascendentes

##### Qual a melhor definição de Polimorfismo?

O conceito de que classes mais complexas herdem métodos de classes mais abstratas e os executem a sua maneira

##### Qual a melhor definição de Abstração?

Diminuir complexidade

##### Caso um parâmetro tenha getters e setters, por convenção, como ele deve ser nomeado?

Com o sinal\_como prefixo (ex.\_saldo)

##### Sobre o método super() é correto afirmar que:

Ele passa os argumentos enviados para a classe ascendente

##### Qual a melhor definição de Encapsulamento?

Propriedades e métodos de uma classe dizem respeito apenas a ela

##### Dadas as afirmações abaixo I. O método super() pode ser utilizado em classes que herdam propriedades de outras II. Getters e setters são utilizados como modificadores de um valor III. Getters e setters ferem o conceito de encapsulamento São verdadeiras

I e II

##### Dadas as afirmações abaixo I. A sintaxe de Classes em Javascript é apenas um “açúcar sintético”. II. Javascript utiliza Herança Prototipada III. Funções dentro de um objeto são chamadas de “métodos” São verdadeiras

Todas as alternativas

##### Sobre protótipos, é correto afirmar que:

São responsáveis pela herança