Programação Orientada a Objetos

DigitalHouse > Coding School

O que é programação?

Linguagens de programação

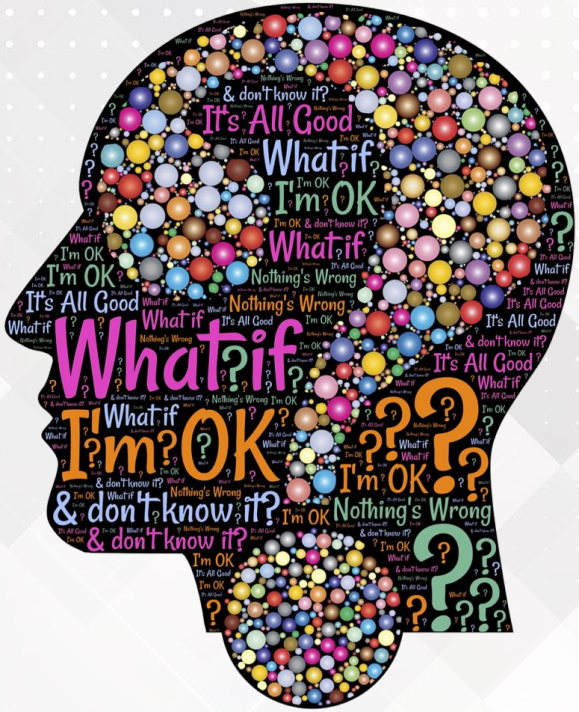
DigitalHouse > Coding School

Alto Nível

```
1 print("Hello World!")
```

Baixo Nível

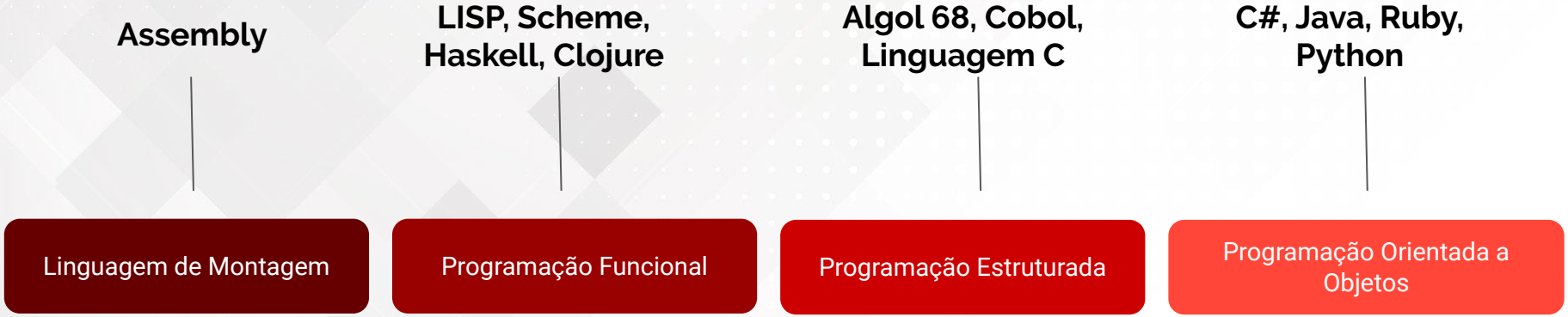
```
Assembly (x86)
1  lea si, string
2  call printf
3  hlt
4  string db "Ola mundo!", 0
5  printf PROC
6     mov AL, [SI]
7     cmp AL, 0
8     je pfend
9     mov AH, 0Eh
10    int 10h
11    inc SI
12    jmp printf
13    pfend:
14    ret
15 printf ENDP
```



Paradigma

DigitalHouse >
Coding School

Evolução dos Paradigmas



Programação Estruturada
vs
Programação Orientada a Objetos

Programação Estruturada

- Surgiu no início da década de 60 mediante a *Crise do Software*
- Características: Uso de subrotinas, Laços de repetição, condicionais e estruturas em bloco.
- Foi a base para a Orientação a Objetos

Programação
Orientada a Objetos

- Surgiu nos anos 60 atráves da linguagem Simula
- Alan Kay considerado um dos criadores do termo “Programação Orientada a Objetos”
- Objetivos: Facilitar o desenvolvimento de software e representar o mundo real.

Como programar um controle remoto?

DigitalHouse > Coding School

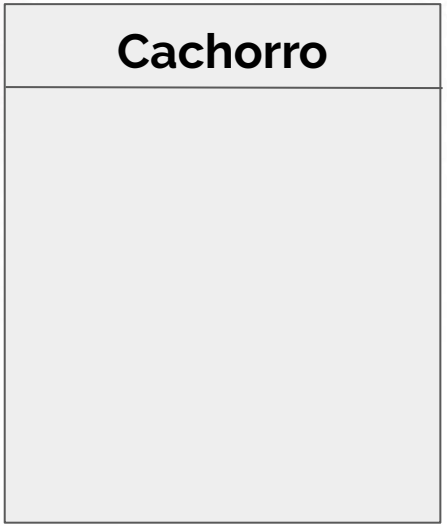
DigitalHouse > Coding School

Cachorro

Classes

“As classes são modelos de um objeto, possuindo características e comportamentos.”

Classes - Exemplo



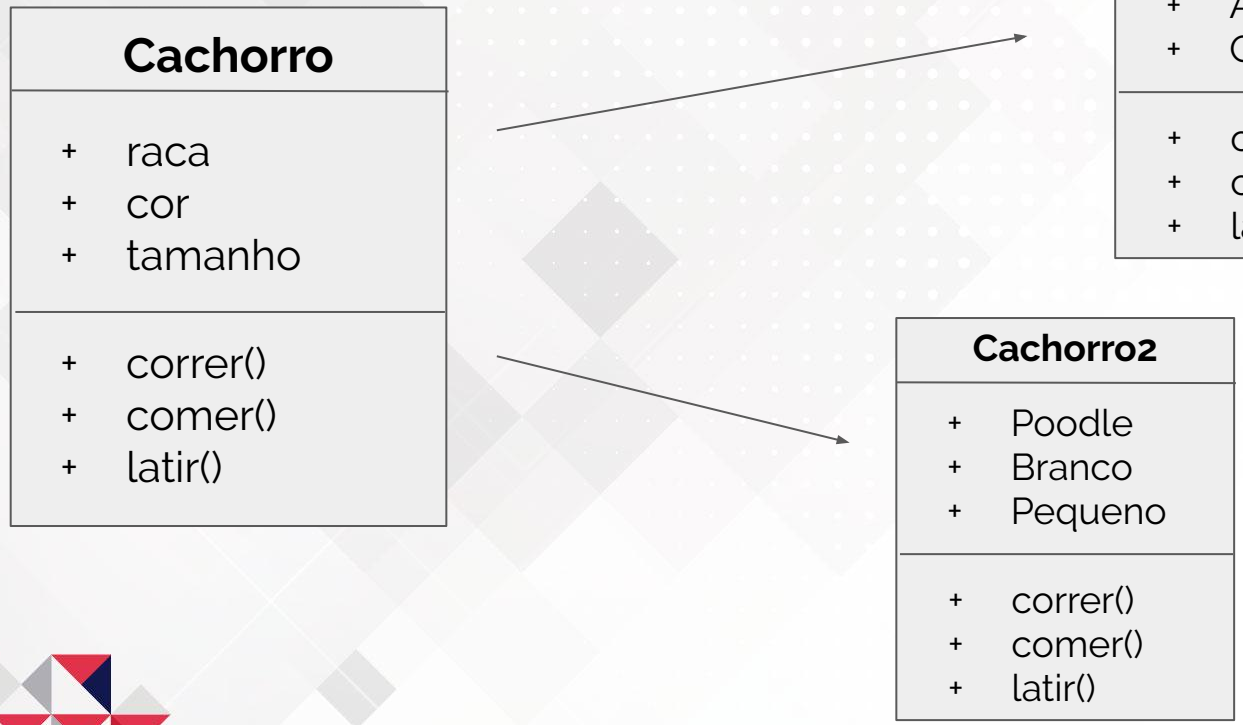
Classes - Exemplo

Características = Atributos

Comportamentos = Métodos

Objetos

Objetos - Exemplo

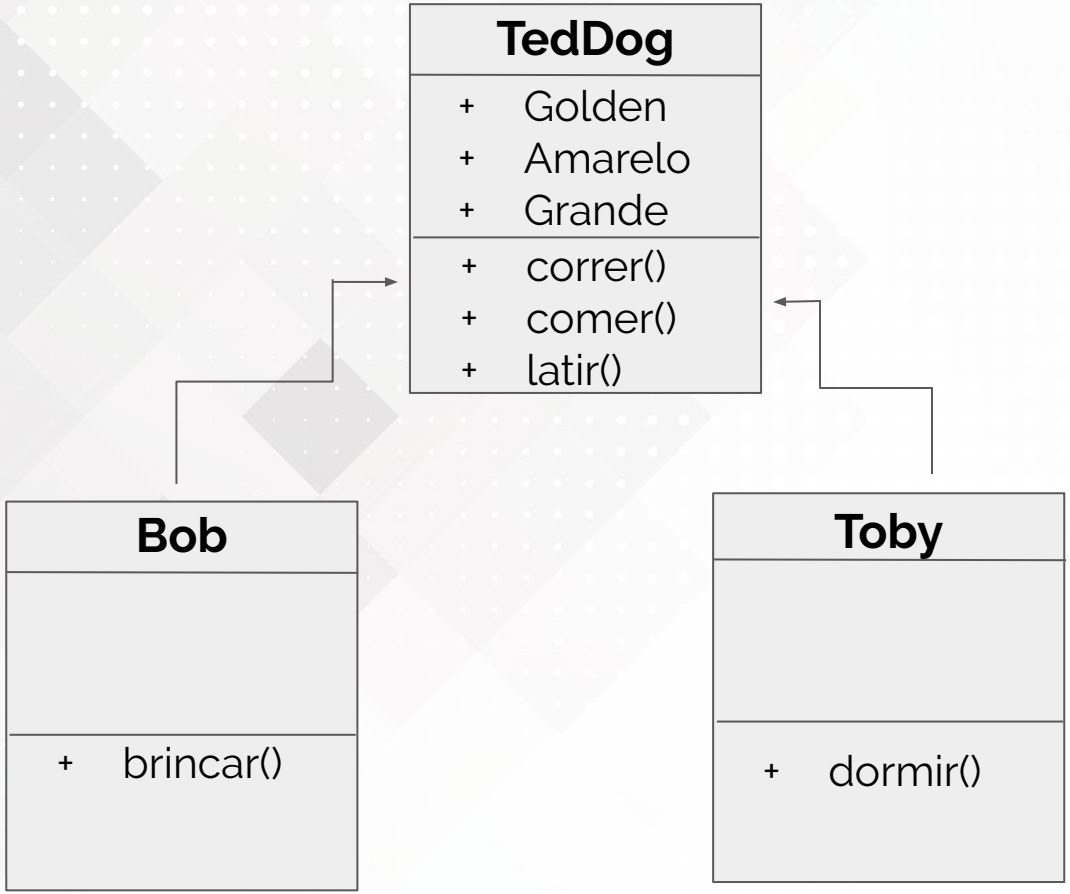


DigitalHouse > Coding School

Herança

DigitalHouse >
Coding School

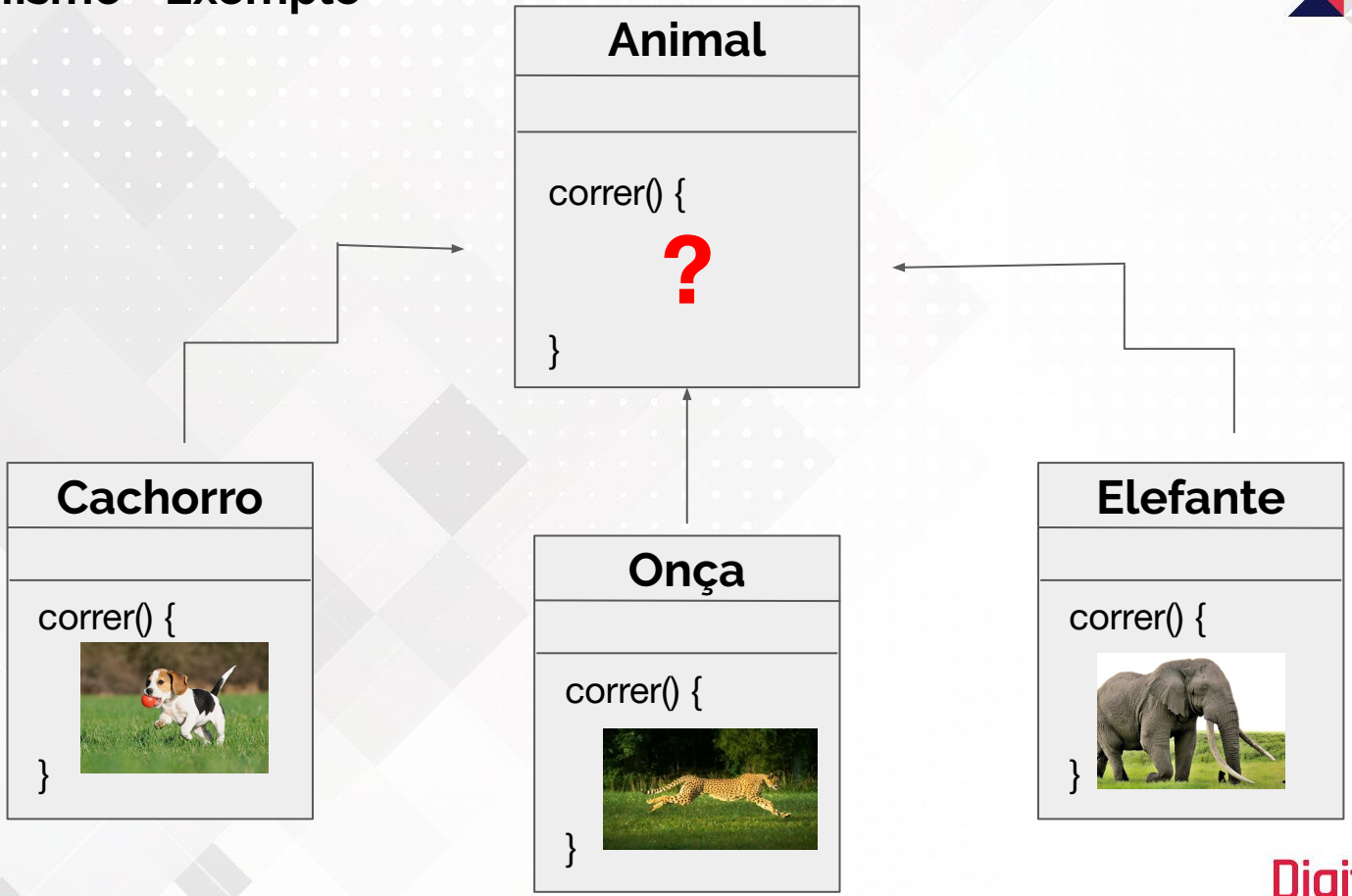
Herança - Exemplo



Polimorfismo

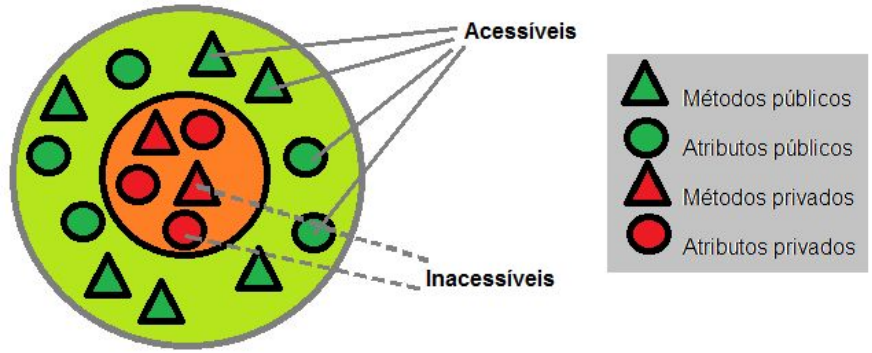
“Os mesmos atributos e métodos podem ser utilizados em objetos distintos, porém, com implementações lógicas diferentes.”

Polimorfismo - Exemplo



Encapsulamento

“É uma forma de proteger parte dos dados independente do restante do sistema.”



DigitalHouse > Coding School

UML

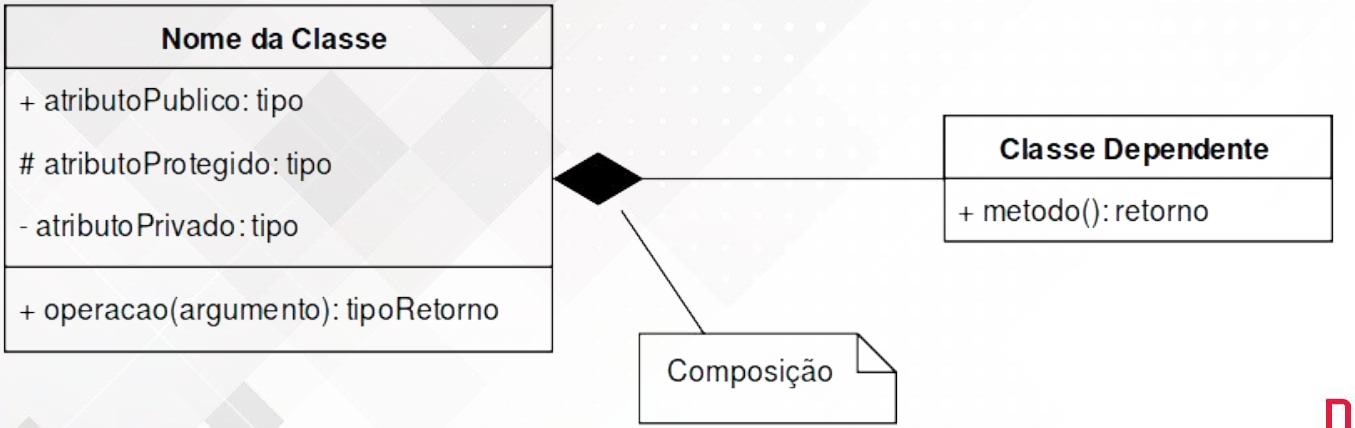
Unified Modeling Language

DigitalHouse > Coding School

Diagrama de Classes

Tipo: int, String, boolean, **OBJETO**

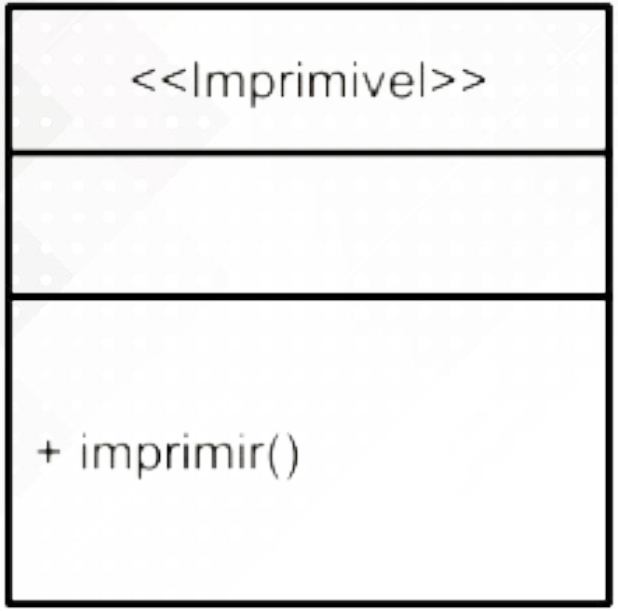
Retorno: o resultado que devolve a função.



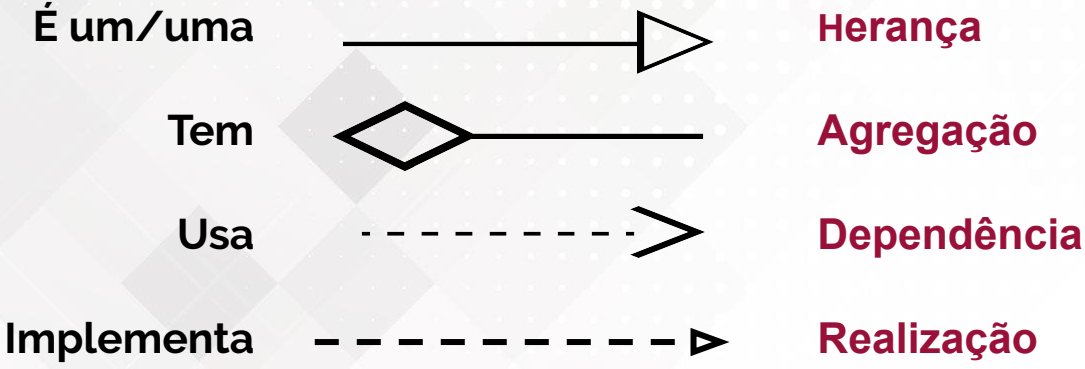
DigitalHouse > Coding School

Interfaces

As interfaces são padrões definidos através de contratos ou especificações.



Relacionamentos



Relacionamento – é um/uma

É uma

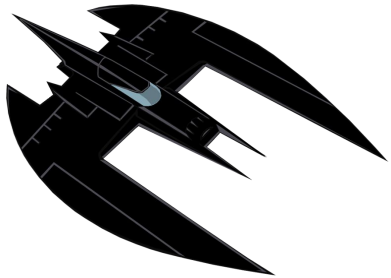
Relacionamento – Tem

Tem

Relacionamento – usa

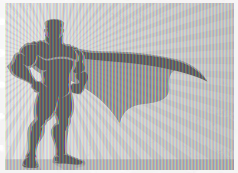
usa

Veículo



DigitalHouse > Coding School

Relacionamento – implementação



Implementa Implementa

DigitalHouse > Coding School

Exercícios

DigitalHouse > Coding School