



Linguagem de Programação Orientada a Objetos

(EAGS SIN 2020)

AULA 1.2 - Abstração

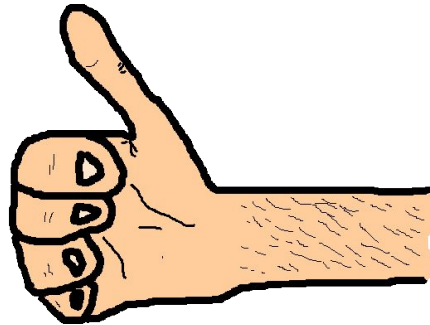
Apresentado por 2S SIN NETTO e 2S SIN MOURA



Se liga !

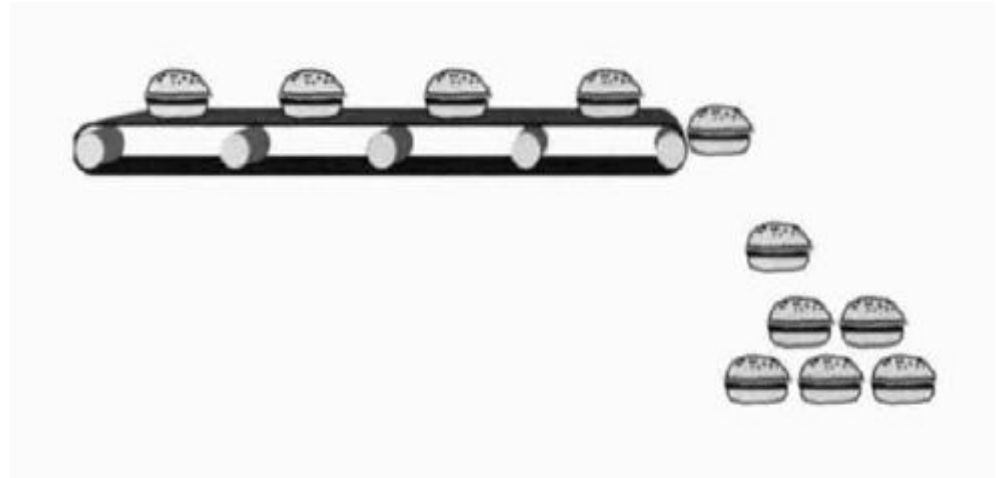
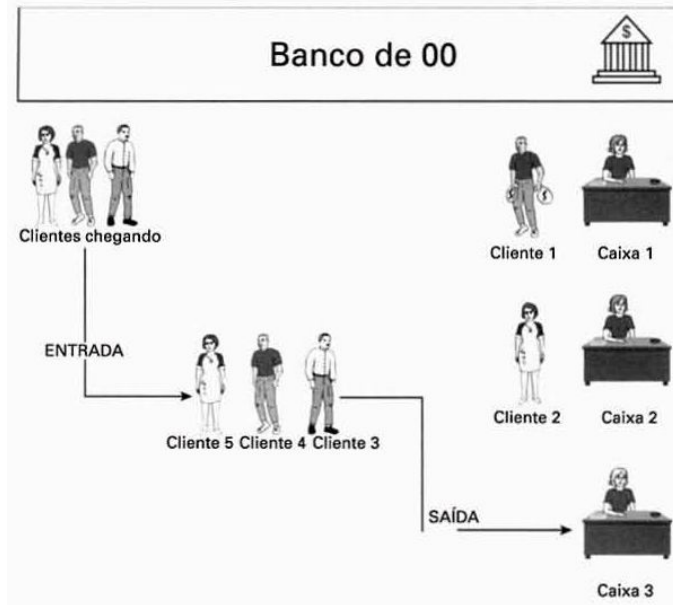
Você vai aprender:

1. O que é abstração e domínio de negócio
2. O que é classe e objeto
3. O que é atributo e método



Abstração

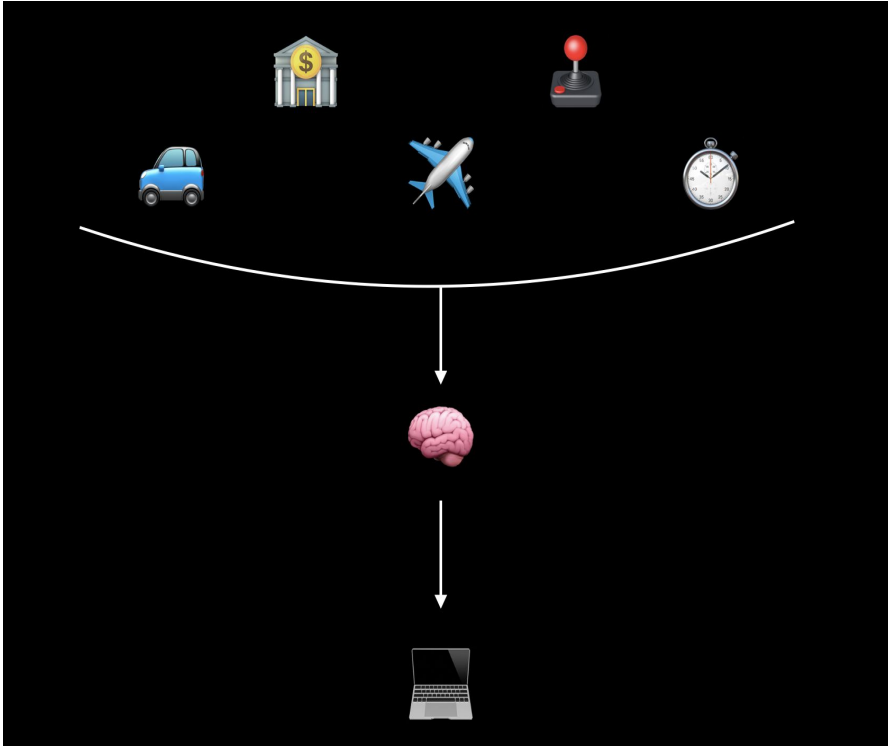
O que há de comum nessas duas imagens?



O exercício da abstração

A POO é a forma de se programar mais próximo da realidade. A abstração é programar a partir do comportamento comum definido pelas coisas no mundo real.

- **Comportamento genérico:** É o comportamento comum das coisas que podem ser extraídos do mundo real e programados para serem utilizados em qualquer domínio de negócio.
- **Domínio** é o espaço onde reside o problema. Uma esteira pode ser utilizada para levar hambúrgueres de um lado para outro ou para levar carros em uma linha de montagem.
- **Foco** orienta o programador a desenvolver o que é comum sem se preocupar com o domínio. Em uma fila, devemos nos atentar para o comportamento (FIFO) sem se preocupar se é uma fila de hambúrgueres ou de carros.



Classificação de objetos



A biologia classifica todos os cães, gatos, elefantes e seres humanos como mamíferos. Características compartilhadas dão a essas criaturas separadas um senso de comunidade. No mundo do software, as classes agrupam objetos relacionados da mesma maneira.

- **Classe:** Define todas as características comuns a um tipo de objeto. Especificamente as classes definem todos os atributos e comportamentos expostos pelo objeto.
- **Objeto** é o que se cria a partir de uma classe. É a materialização da classe que ganha as características e comportamentos definidos em seu interior
- **Atributos** são características de um objeto. Altura de uma pessoa e cor de um carro são exemplos de atributo
- **Métodos** representam o comportamento dos objetos e também são definidos no interior da classe

PRÁTICA

1. CRIAR UMA CLASSE DE NOME ALUNO
2. CRIAR ATRIBUTOS EM FORMA DE VARIÁVEL
3. POPULAR OS VALORES DAS VARIÁVEIS
4. ALTERAR VARIÁVEIS PARA ATRIBUTOS DE CLASSE
5. POPULAR OS VALORES DOS ATRIBUTOS DENTRO DA CLASSE

Abstração e o princípio da responsabilidade única



Abstração é o processo de simplificar um problema difícil.

Quando começamos a resolver um problema, não nos preocupamos com cada detalhe. Em vez disso, simplificamos, tratando apenas dos detalhes pertinentes a uma solução.

Vantagens da abstração:

- **Resolução de problemas mais facilmente:** Quando o domínio de negócio passa a não ser mais importante o foco está no comportamento correto genérico, a entendimento do problema e a implementação da solução se torna mais clara e simples.
- **Reutilização:** É a esteira funcionando para vários domínios de negócio. A garantia do funcionamento está na adequação da esteira ao domínio e não o domínio à esteira.

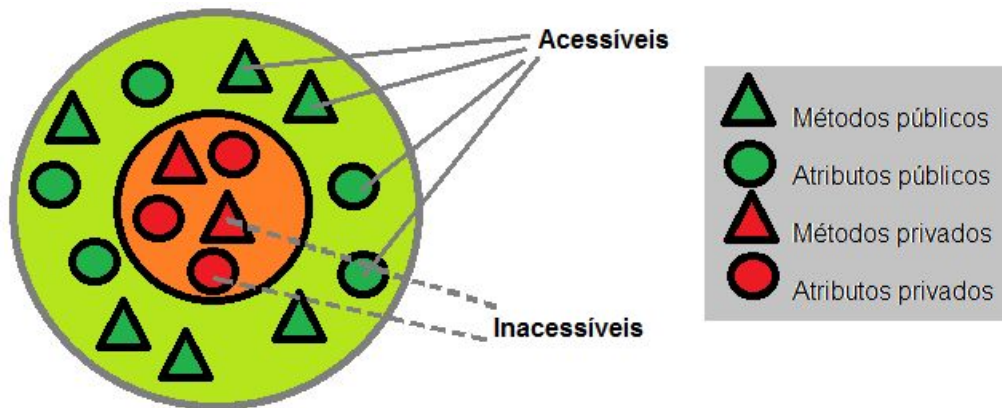
A classificação do objeto deve haver apenas um motivo para existir. Em outras palavras, uma classe deve existir para criar um objeto que tenha características e comportamentos específicos a sua natureza.

A abstração eficaz

Muitas das vezes, ao tentarmos resolver um problema nos preocupamos em desenvolvermos a solução que irá resolver um problema específico, como nos orienta o Princípio da Responsabilidade Única, porém, o conceito de abstração da Orientação a Objetos nos orienta a limitar a especialização da nossa solução para que a reutilização da solução seja possível para diferentes domínios. Regras para a abstração eficaz:

- Trate do caso geral e não do caso específico.
- Ao confrontar vários problemas diferentes, procure o que for comum a todos. Tente ver um conceito e não um caso específico.
- Não esquecer de que temos um problema a resolver. A abstração é valiosa, mas não podemos descuidar do problema na esperança de escrever código abstrato.
- A abstração pode não estar prontamente aparente. A abstração pode não "saltar" à nossa frente na primeira, segunda ou terceira vez que tentarmos resolver um problema que está sujeito a ser abstraído.
- Preparemo-nos para a falha. É quase impossível escrever uma abstração que funcione em todas as situações.

Encapsulamento - A proteção da responsabilidade única



Um importante pilar da POO que garante:

- **Reuso**, pois uma vez cumprindo com o comportamento esperado, qualquer outra classe poderá fazer uso desse comportamento.
- **Reduz efeitos colaterais**, pois quando há a proteção da responsabilidade única, somente alterando o código interior de uma classe que seu comportamento da poderá ser alterado.
- **Poupa tempo**, pois não é preciso reescrever código uma vez que ele já existe encapsulado em uma classe.

Tipos de visualização:

- **Público**: Todos podem acessar aquilo que for definido como public. Classes, atributos, construtores e métodos podem ser public.
- **Privado**: A única classe capaz de acessar os atributos, construtores e métodos privados é a própria classe. Classes como conhecemos, não podem ser private, mas atributos, construtores e métodos sim.



Obrigado

Você aprendeu:

1. O que é abstração e domínio de negócio
2. O que é classe e objeto
3. O que é atributo e método

