


# Welcome to the jungle!

**EU ESTOU USANDO  
JAVA, ESTOU PROGRAMANDO  
ORIENTADO A OBJETOS!**

**SE AS SUAS CLASSES  
NÃO REPRESENTAM ABSTRAÇÕES  
DO SISTEMA E SÃO APENAS  
REPOSITÓRIOS DE FUNÇÕES, VOCÊ  
NA VERDADE ESTÁ PROGRAMANDO  
SEGUNDO O PARADIGMA ESTRUTURADO.**





# Programação Orientada a Objetos



**Afinal, o que são Classes e Objetos?**



An abstract graphic on the left side of the slide, consisting of several overlapping, wavy, ribbon-like shapes in shades of purple and blue. These shapes flow from the top left towards the bottom left, creating a sense of movement and depth. The colors transition from a deep purple at the top to a lighter blue at the bottom.

# Diagrama de Classes



# Diagrama de Classes

Podem existir diversos diagramas de classe no modelo,  
porém

**Cada classe é única dentro de todo  
o modelo (projeto)**



# Diagrama de Classes

- Diferença entre classe e objeto?
- Que elementos que aparecem em uma classe?



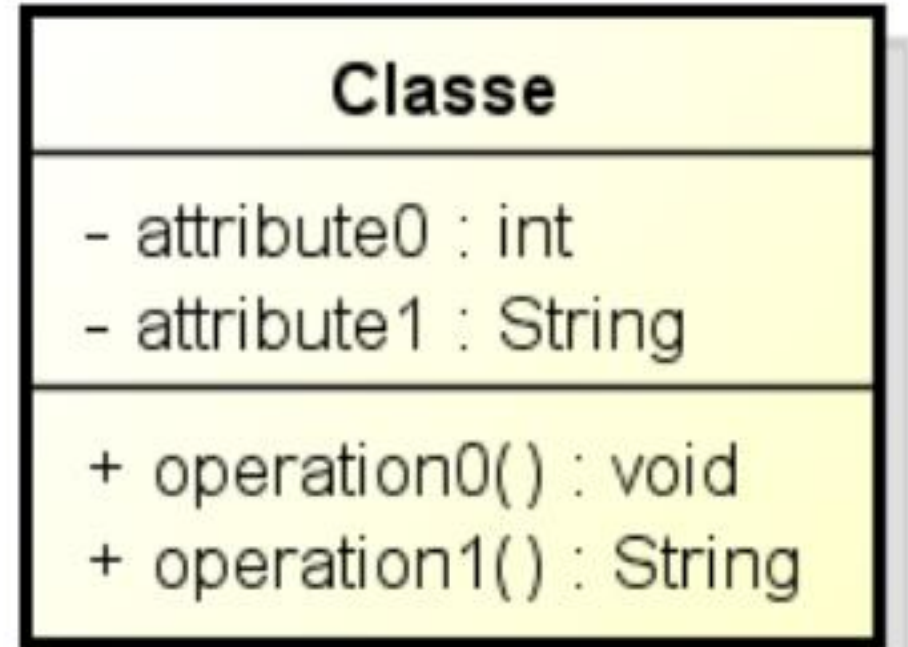
# Diagrama de Classes

- **Classe**
  - Template (modelo)
  - Definição para um recurso
    - Descreve os recursos (características, estados e comportamentos) de uma entidade e como ela pode ser utilizada
- **Objeto**
  - Instâncias de uma classe (realização)
  - “Coisas”

# Diagrama de Classes

- **Elementos**

- Compartimento de **Nome**
- Compartimento de **Atributos**
- Compartimento de **Operações**



powered by Astah





# Diagrama de Classes

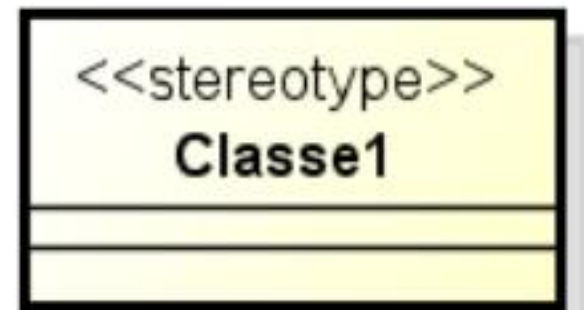
- **Compartimento de Nome**
  - **Nome da Classe**
    - Meio para reconhecer o recurso dentro do modelo
    - Sempre centralizado e estilo normal
    - Quase sempre é um substantivo simples e no singular
    - Sempre dentro de um padrão de formato de codificação
      - Nomes ruins
        - Mostra\_Cancelada
        - Mostra\_Planejada

# Diagrama de Classes

## Compartimento de Nome

- **Estereótipo**

- Identifica a função do elemento dentro do do modelo
- Não faz parte do nome da classe
- Aparece acima do nome da classe e entre os sinais << e >>
- entity, control, interface, boundary, etc



powered by Astah



# Diagrama de Classes

## Visibilidade

- Abrange atributos e operações (métodos)
- Escopo de acesso ao recurso permitido para membros da classe
  - private (-)
  - public (+)
  - package (~)
  - protected (#)



# Diagrama de Classes

## Compartimento de Atributos

- Contém as definições para todas as informações pertinentes (características e estados) que os objetos representados possuem
- O objeto pode “falar de si”
  - LocalShow: número de assentos, estado atual (aberto/fechado)



# Diagrama de Classes

## Visibilidade

- Abrange atributos e operações (métodos)
- Escopo de acesso ao recurso permitido para membros da classe
  - private (-)
  - public (+)
  - package (~)
  - protected (#)





# Diagrama de Classes

## **Multiplicidade**

- Especifica a quantidade de valores que podem estar associados a um elemento do modelo
- Intervalo [0..5]
- Quantidade específica [2..2], [2]
- Intervalo sem limite [1..\*]



# Diagrama de Classes

## Compartimento de Atributos

- Cada atributo deve ser definido com o nível em que a aplicação possa garantir a integridade o atributo e do sistema como um todo
  - Visibilidade
  - Nome
  - : (dois pontos)
  - Tipo do atributo (dado)
- Valor Default (opcional)
- Valor Derivado (opcional)
- String de propriedade (outras informações sobre o atributo)

# Diagrama de Classes

## ContaCorrente

- saldo : double = 0
- correntista : Cliente[1..\*]
- dataAbertura : Date {required}
- attribute5 : int

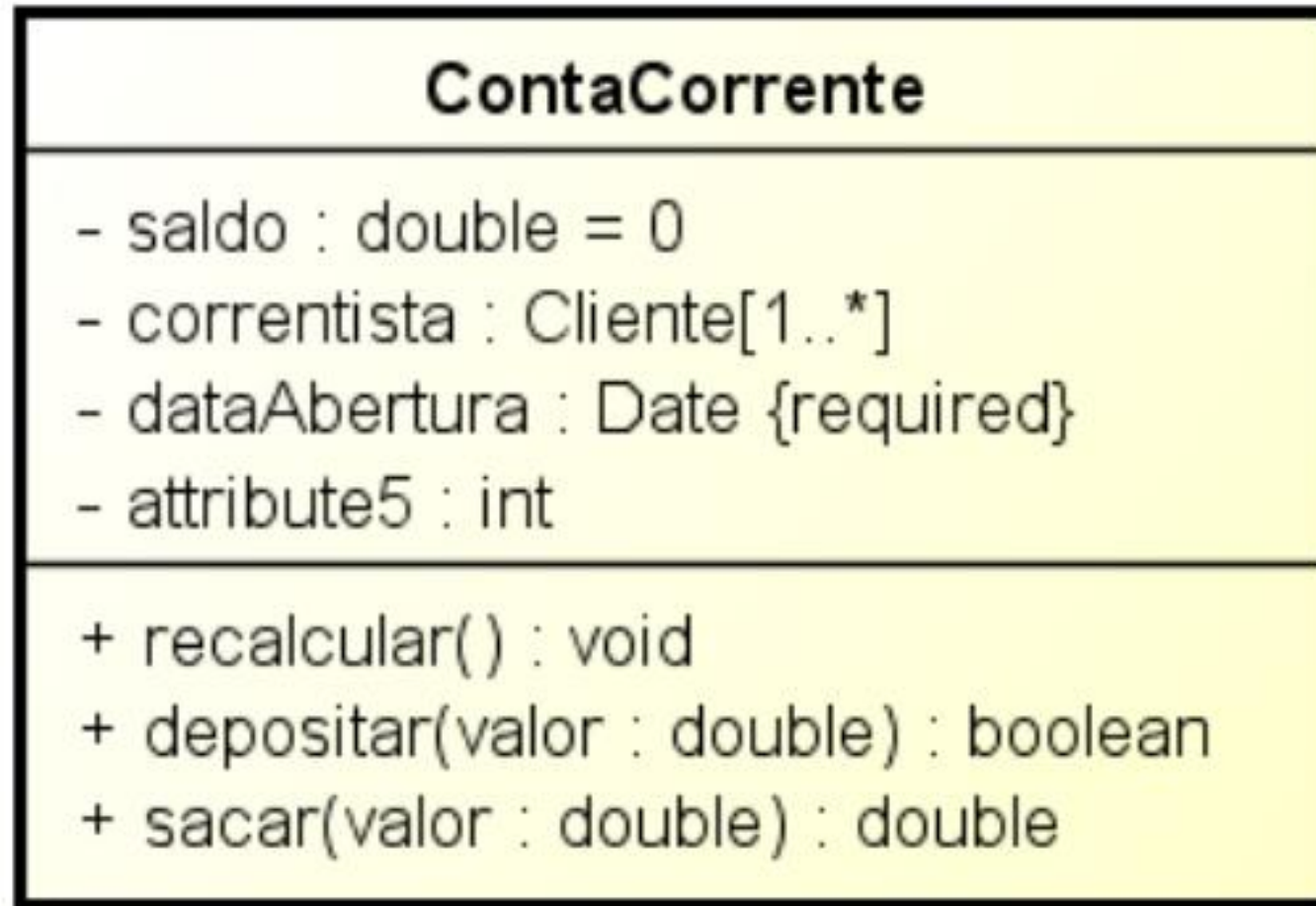


# Diagrama de Classes

## **Compartimento de Operações**

- Lista os comportamentos (métodos) da classe / objeto
- Serviços que um objeto pode oferecer
- Define a(s) informação(ões) necessária(s) para invocar o comportamento
- A operação deve ser implementado (desenvolvido)
  - Visibilidade
  - Nome
  - Parametros (nenhum, um ou muitos / primitivos ou objetos)
  - : (dois pontos)
  - Tipo do retorno (nenhum, primitivo ou objeto / um ou lista)

# Diagrama de Classes







# Diagrama de Classes

## Relacionamentos

- Explicação sobre como os recursos (classes/objetos) podem se relacionar
  - Associação
  - Generalização
  - Dependência



Por hoje é  
só  
Pessoal!