

A yellow L-shaped line is positioned in the top-left corner of the slide.

Orientação a Objetos

A short blue horizontal line is centered below the title.

Prof. Isaac

A yellow L-shaped line is positioned in the bottom-right corner of the slide.

Introdução à Linguagem de Modelagem Unificada - UML



UML

- *Unified Modeling Language* (UML) - Linguagem de Modelagem Unificada - é uma **linguagem visual** utilizada para **modelar** sistemas computacionais **orientados a objeto**.
- Nos últimos anos, a UML consagrou-se como a linguagem-padrão de modelagem adotada pela indústria de Engenharia de Software, havendo atualmente um amplo mercado para profissionais que a dominem.

Breve Histórico

- Na década de 1980 empresas começaram a utilizar a POO para aplicativos;
- Assim, em 1990, basicamente cada empresa tinha o seu próprio processo de desenvolvimento;
- Contudo, fornecedores de softwares tinham dificuldades para atender tantos padrões diferentes.
- Logo, verificou-se a necessidade de um processo padrão de Orientação a Objetos.
- Em 1996 foram liberadas as primeiras versões da UML.
- Em 1997, a OMG* (*Object Oriented Group*) assumiu a responsabilidade pela manutenção e revisão da UML.

* organização internacional que aprova padrões abertos para aplicações orientadas a objetos

UML

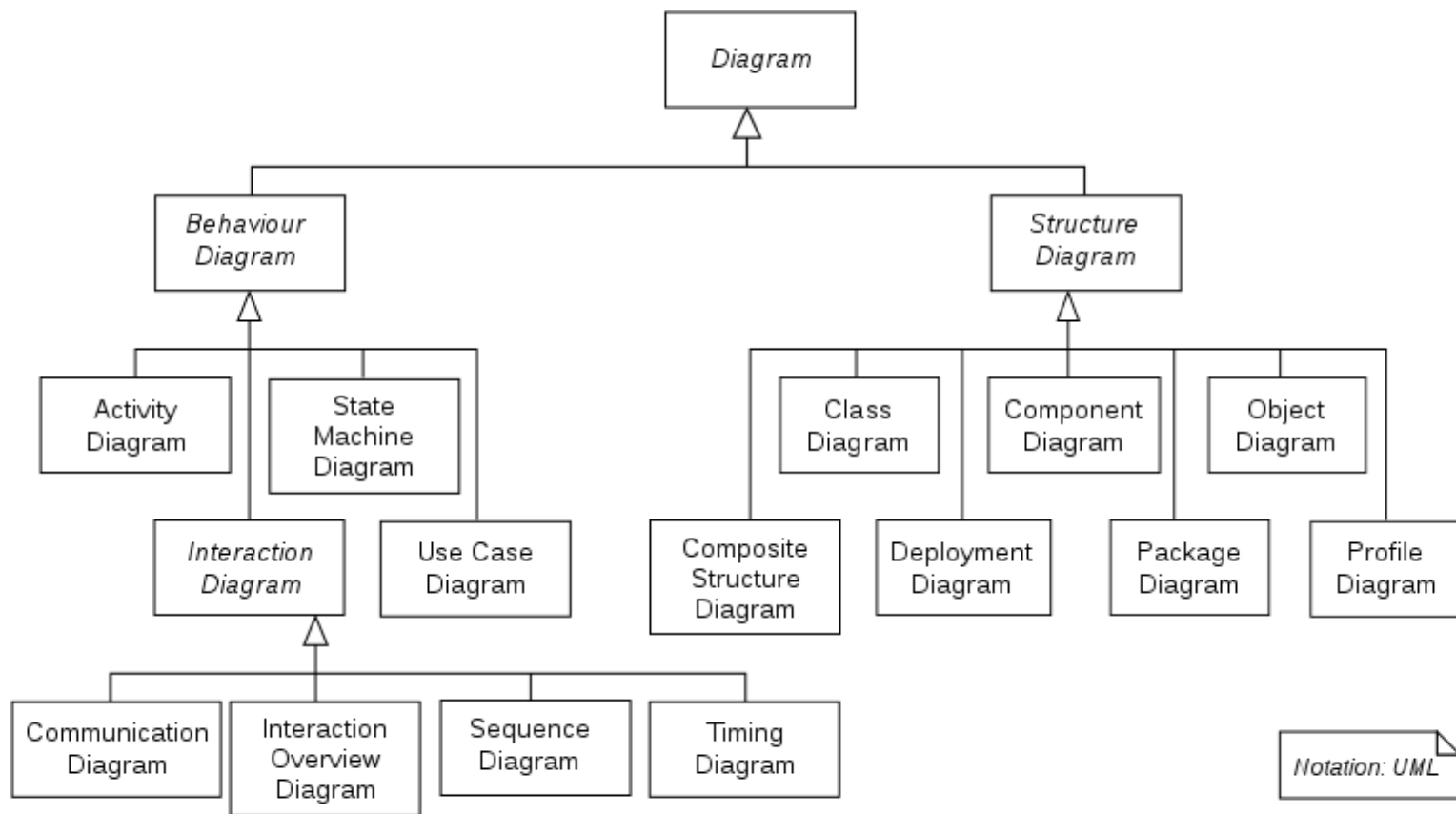
- A UML não é uma metodologia de desenvolvimento:
 - ela não diz para você como projetar seu sistema
- Mas ela auxilia a visualizar seu desenho e a comunicação entre os objetos
- Como ela é feita com base nos conceitos de OO, se encaixa perfeitamente com linguagens como Java e C++.

UML

- Versão atual: 2.5.1 (*Desde Dez/2017*)
- Conforme a **OMG***, a **UML** possui 15 tipos de diagramas, divididos em duas grandes categorias:
 - Estruturais (7 diagramas) e
 - Comportamentais (8 diagramas)

**<https://www.omg.org/spec/UML/About-UML/>*

** organização internacional que aprova padrões abertos para aplicações orientadas a objetos*



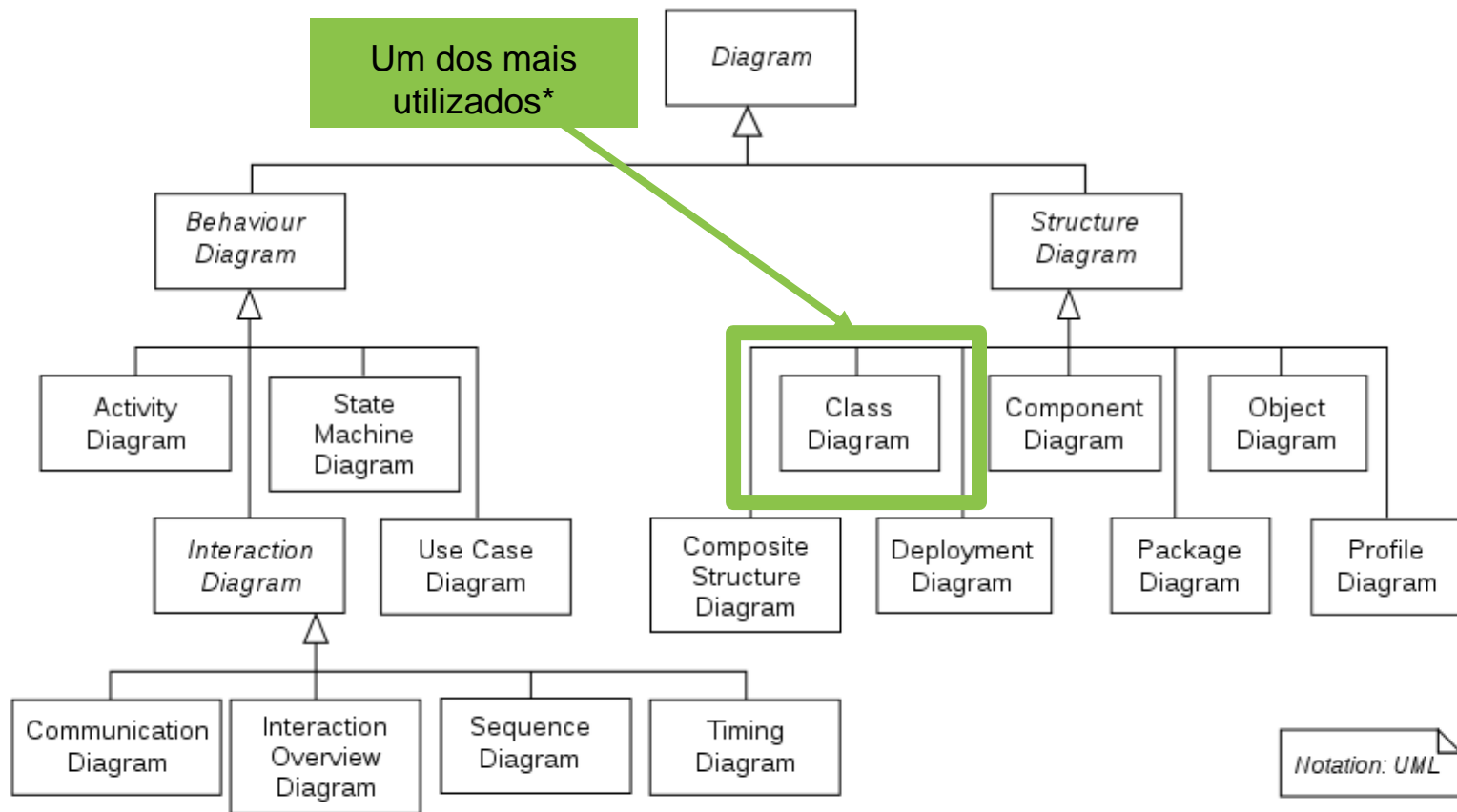


Diagrama de Classes

- Serve de apoio para a maioria dos demais diagramas
- Define a estrutura das classes utilizadas pelo sistema:
 - Define os atributos e métodos que cada classe tem
 - Estabelece como as classes se relacionam e trocam informações entre si

Diagrama de Classes

Dividido em 3 partes

A exibição dos parâmetros dos métodos é opcional

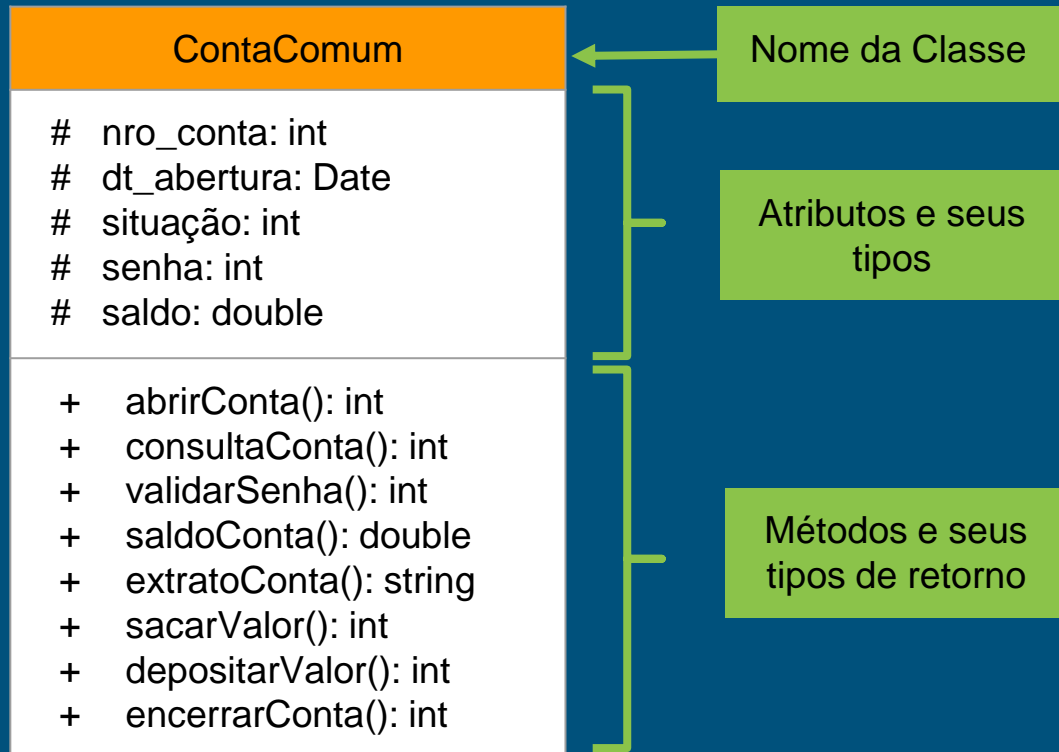


Diagrama de Classes

Métodos com
parâmetros e seus tipos
de retorno

ContaComum

```
# nro_conta: int
# dt_abertura: Date
# situação: int
# senha: int
# saldo: double

+ abrirConta(int): int
+ consultaConta(int): int
+ validarSenha(int): int
+ saldoConta(): double
+ extratoConta(Date): string
+ sacarValor(double): int
+ depositarValor(int, double): int
+ encerrarConta(): int
```

Diagrama de Classes

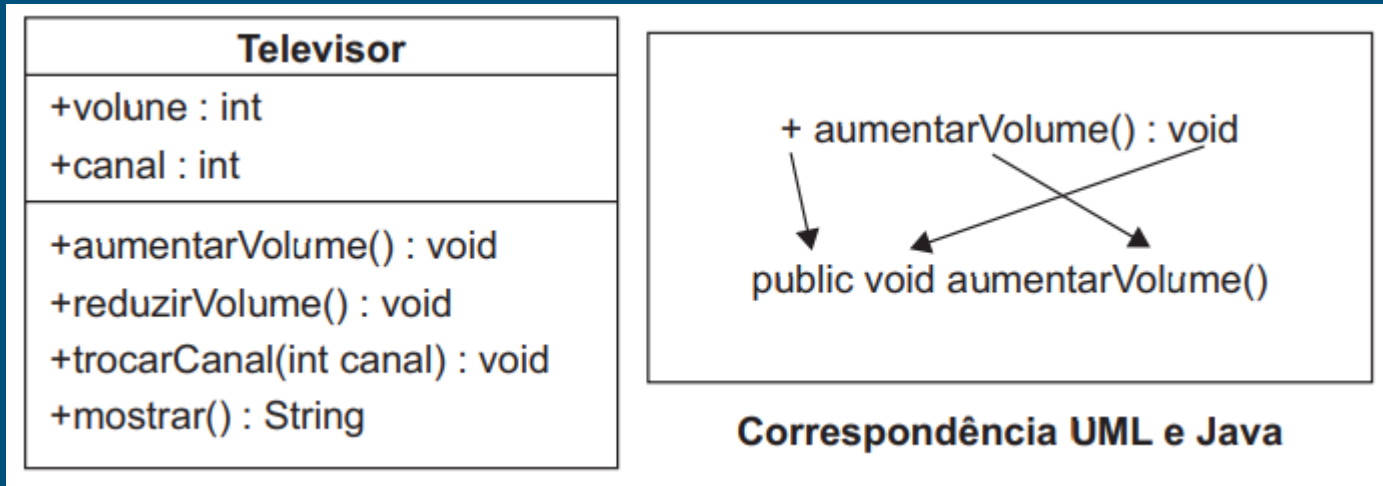


Diagrama de Classes - Visibilidade

- Indica o nível de acessibilidade
- Basicamente 3 modos de visibilidade:
 - Privada: - (menos)
 - Pública: + (mais)
 - Protegida: # (sustenido)

Diagrama de Classes

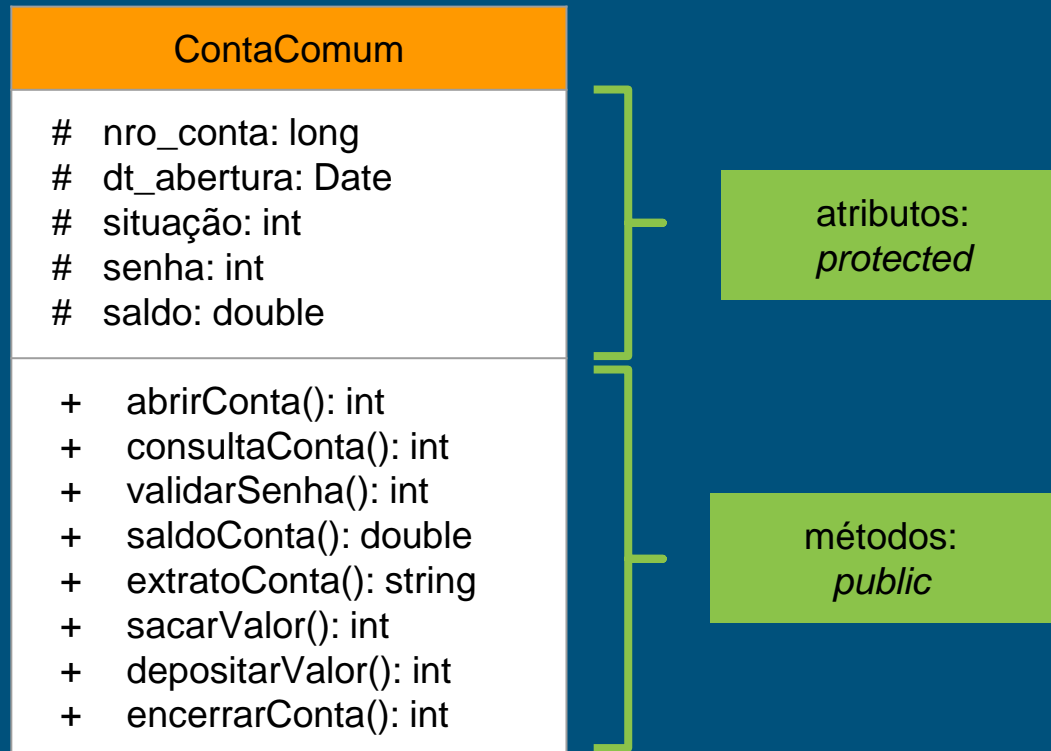


Diagrama de Classes

Métodos com
parâmetros e seus tipos
de retorno

ContaComum

```
# nro_conta: int
# dt_abertura: Date
# situação: int
# senha: int
# saldo: double

+ abrirConta(int): int
+ consultaConta(int): int
+ validarSenha(int): int
+ saldoConta(): double
+ extratoConta(Date): string
+ sacarValor(double): int
+ depositarValor(int, double): int
+ encerrarConta(): int
```