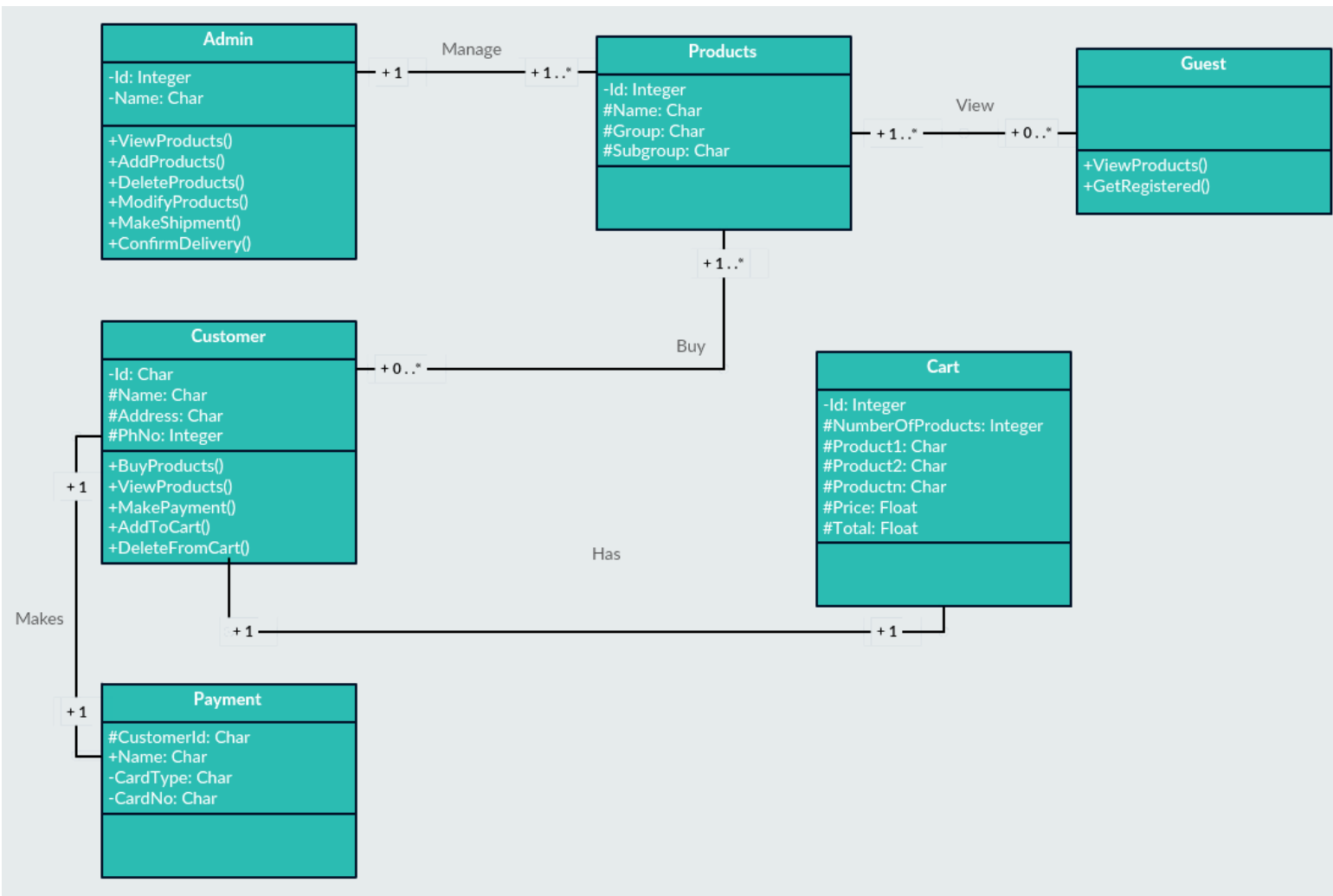


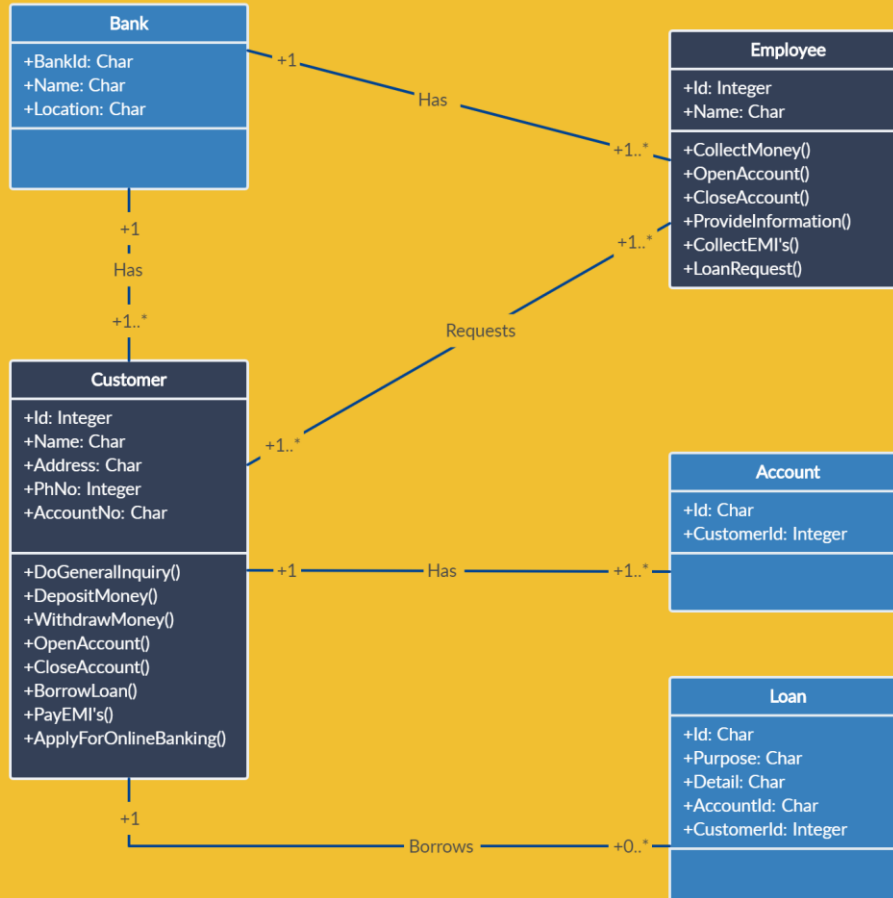
Parte 1

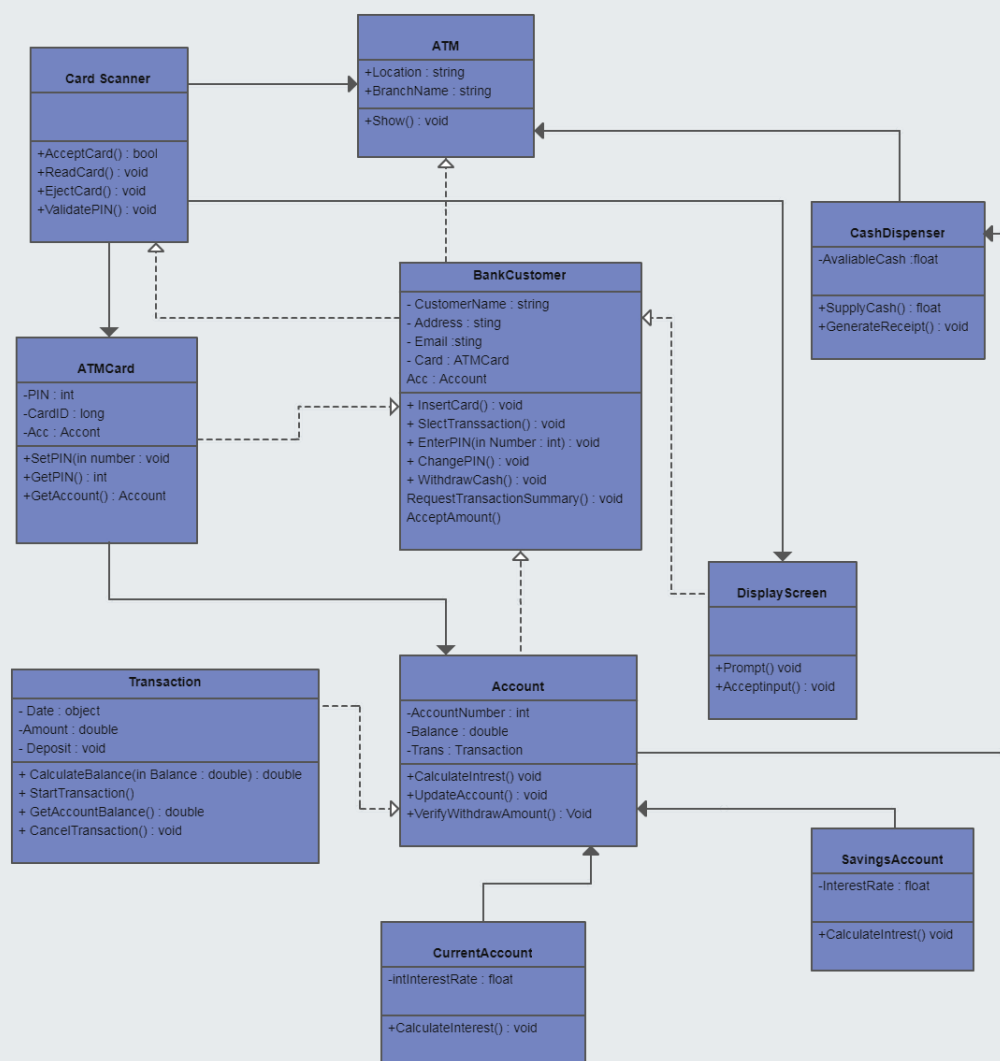
Diagrama de Classes

Professora Arianne Sarmento Torcate









Programação Orientada a Objetos

■ É um paradigma surgido nos anos 1970



Por definição, paradigma é um modelo/padrão a ser seguido. Ou seja, no contexto de programação é um modelo que indica a forma como a linguagem deve operar para possibilitar a resolução de problemas.

■ A Programação Orientada a Objeto (POO) tem como base o conceito de objeto. Ou seja, um sistema orientado a objeto é um conjunto de objetos que representam os conceitos do mundo real, interagindo computacionalmente com as mesmas características e comportamentos reais.

- ❖ Nós entendemos o mundo como um todo composto por vários objetos que interagem uns com os outros;
- ❖ Da mesma maneira, a Orientação a Objetos consiste em considerar os sistemas computacionais não como uma coleção estruturada de processos, mas sim como uma coleção de objetos que interagem entre si;
- ❖ A proposta da orientação a Objetos é representar o mais fielmente possível as situações do mundo real nos sistemas computacionais.

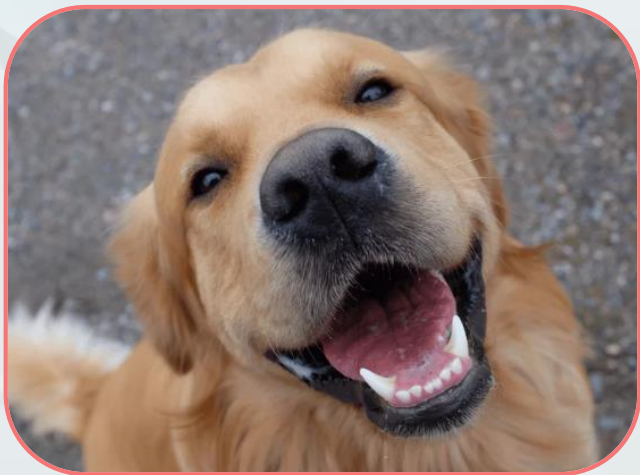


POO



Usamos o termo **objeto** para representar um determinado elemento do mundo real

Por exemplo, vamos considerar um cachorro como nosso “objeto” de estudo



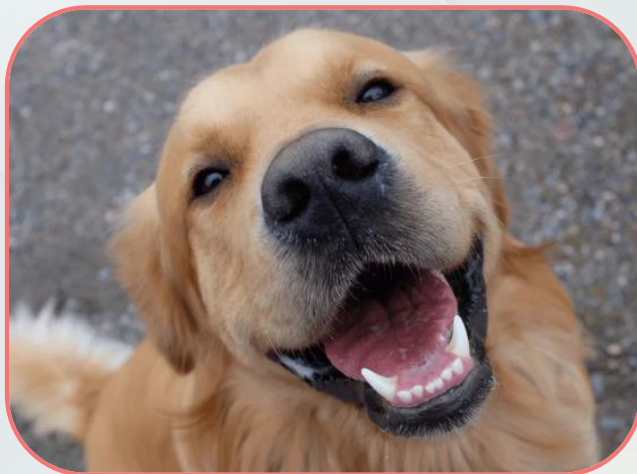
Analisando este objeto (o cachorro), podemos deduzir que o mesmo possui algumas características, como:

- Nome, Idade, Comprimento de pelos, cor dos olhos, peso, etc

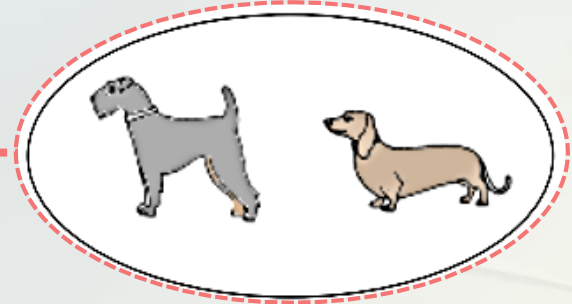
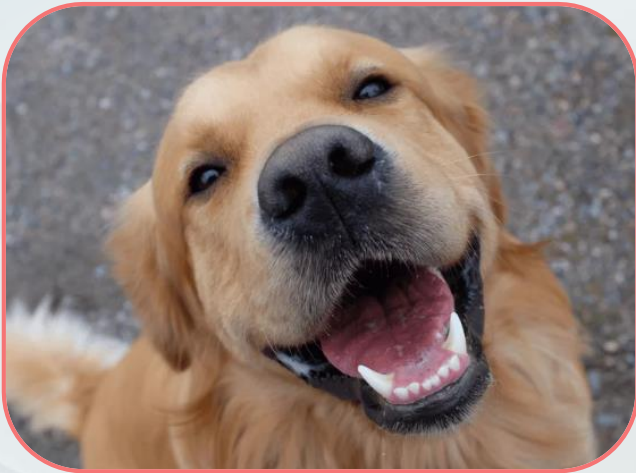
As características que descrevem um objeto são chamadas de **atributos**

Além do conjunto de características que descrevem o cachorro, podemos também identificar um conjunto de ações que ele é capaz de executar (Exemplo: Latir, Comer, Andar, etc)

As ações que um objeto pode executar são chamadas de **métodos**

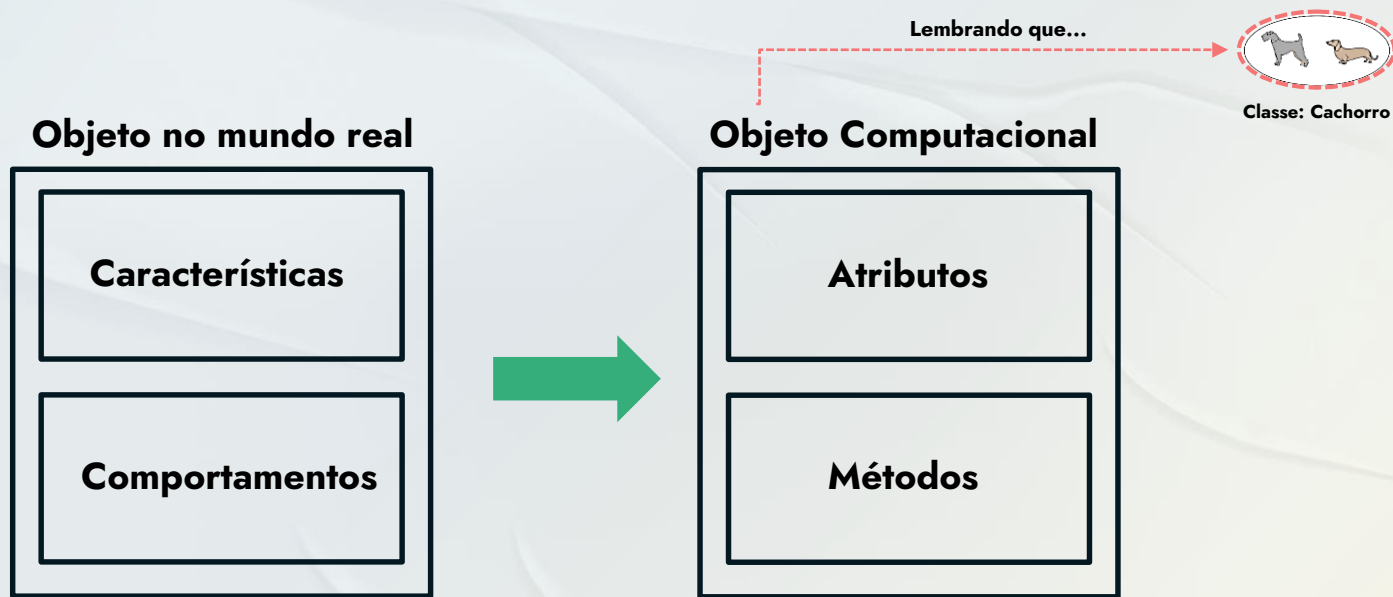


Uma **classe** representa um conjunto de objetos que possuem características e comportamentos comuns



Classe: Cachorro

Um objeto é a representação computacional de um elemento ou processo do mundo real. Cada objeto possui um conjunto de características e comportamentos.



Como identificar? Observe um objeto e pergunte:

1. Quais as possíveis características desse objeto?
2. Quais comportamentos (ações) que ele pode executar?

Diagrama de classe

- » O diagrama da UML utilizado para representar o aspecto estático é o **diagrama de classes**
- » Podemos afirmar de maneira mais simples que seria um conjunto de objetos com as mesmas características, assim saberemos identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes

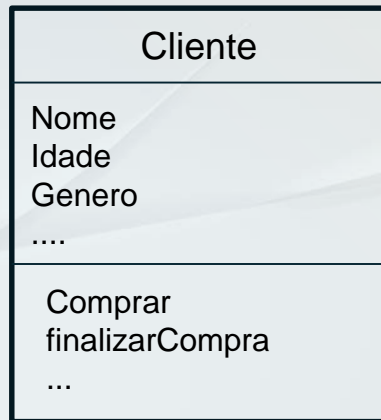
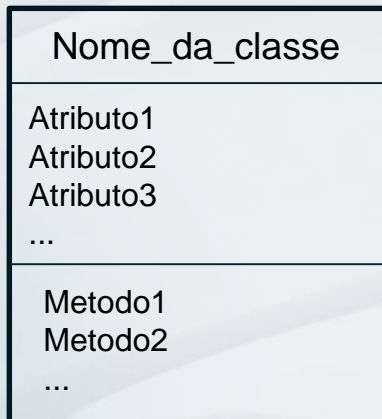
Pontos importantes:

- O diagrama de classes é um dos mais importantes e mais utilizados da UML;
- Representação das principais classes;
- Atributos e Métodos;
- Relacionamento entre as classes;
- Uma visão estática do sistema.



Classes

Na UML, uma classe é representada por um retângulo com três compartimentos, sendo para: Nome da classe (primeiro), Atributos (segundo) e Operações/métodos (terceiro).



Nome da classe



Atributos



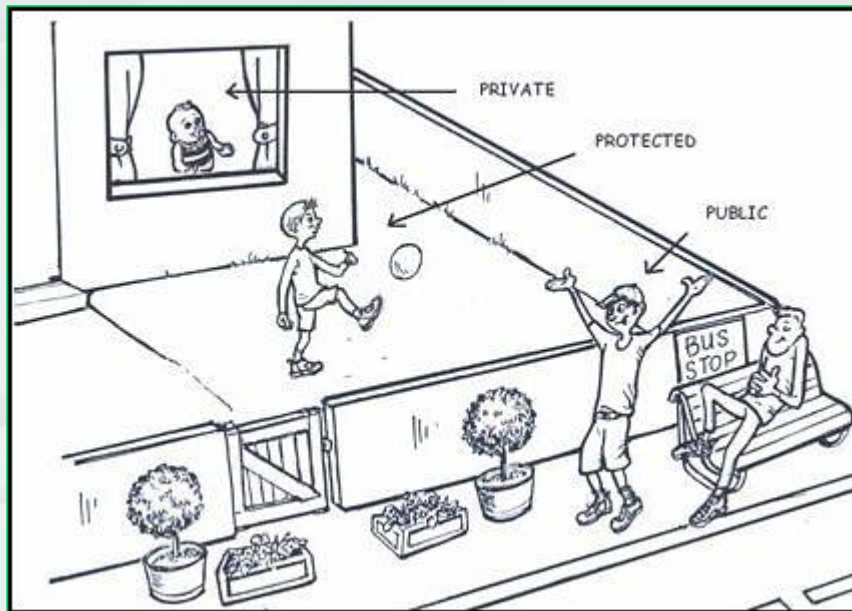
Métodos/Operações

- Atributos - **Visibilidade**

- Representam o conjunto de características (estado) dos objetos daquela classe

Visibilidade	Símbolo	Significado
Pública	+	Visível em qualquer classe de qualquer pacote
Protegida	#	Visível para classes do mesmo pacote
Privada	-	Visível somente para classe

Visibilidade	Símbolo	Significado
Pública	+	Visível em qualquer classe de qualquer pacote
Protegida	#	Visível para classes do mesmo pacote
Privada	-	Visível somente para classe



Visibilidade	Símbolo	Significado
Pública	+	Visível em qualquer classe de qualquer pacote
Protegida	#	Visível para classes do mesmo pacote
Privada	-	Visível somente para classe

Cliente
+ Nome # Idade - Genero
Comprar finalizarCompra ...

- Atributos – **Tipos de dados**

- O tipo do atributo corresponde ao tipo de dado que será utilizado no código fonte

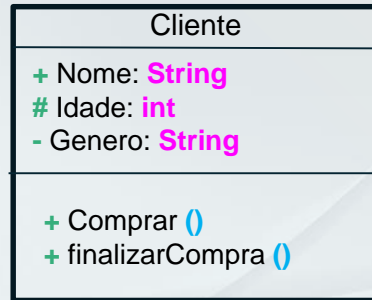
Nome	Representação	Significado
Inteiros (Integer)	int	Armazena números inteiros (Ex: idade = 25)
	long	Armazena números inteiros maiores que int (Ex: populacaoMundial = 78000000000L)
Ponto Flutuante (Floating Point)	float	Armazena números de ponto flutuante (decimais) com precisão simples (ex: altura = 1.75f)
	double	Armazena números de ponto flutuante (decimais) com precisão dupla (Ex: pi = 3.141592653589793)
Caractere (Character)	Char	Armazena um único caractere, por exemplo, 'a', '1', '\$'.
Cadeia de Caracteres	String	Armazena uma sequência de caracteres, por exemplo, "Hello, World!".
Booleano	boolean	Armazena um valor verdadeiro ou falso, representado por true ou false.
Data/Horas	Date	Armazena valores referente a data e hora

Nome	Representação	Significado
Inteiros (Integer)	int	Armazena números inteiros (Ex: idade = 25)
	long	Armazena números inteiros maiores que int (Ex: populacaoMundial = 7800000000L)
Ponto Flutuante (Floating Point)	float	Armazena números de ponto flutuante (decimais) com precisão simples (ex: altura = 1.75f)
	double	Armazena números de ponto flutuante (decimais) com precisão dupla (Ex: pi = 3.141592653589793)
Caractere (Character)	Char	Armazena um único caractere, por exemplo, 'a', '1', '\$'.
Cadeia de Caracteres	String	Armazena uma sequência de caracteres, por exemplo, "Hello, World!".
Booleano	boolean	Armazena um valor verdadeiro ou falso, representado por true ou false.
Data	Date	Armazena valores referente a data e hora

Cliente
+ Nome: String # Idade: int - Genero: String
Comprar finalizarCompra

Métodos

- Os métodos (também chamados de operações) são funções que definem o comportamento das classes
- Resumidamente, também possui visibilidade (assim como os atributos)



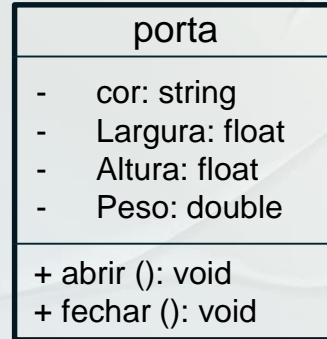
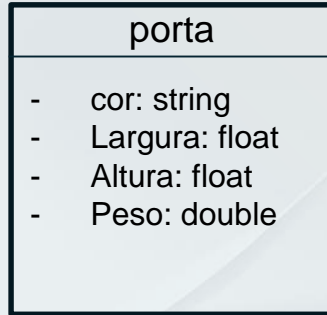
Métodos

- Resumidamente, os métodos também possuem:
- ❖ Os **parâmetros** são as entradas que um método recebe para realizar sua operação. Eles consistem de um nome e um tipo de dado, e são listados entre parênteses após o nome do método.
- ❖ O **tipo de retorno** é o tipo de dado que o método retorna após a execução. Se o método não retorna nenhum valor, o tipo de retorno é void.

Exemplo:

contaBancaria
<ul style="list-style-type: none">- numeroConta: String- Saldo: double
<ul style="list-style-type: none">+ consultarSaldo (): double+ depositar (valor: double): void+ sacar (valor: double): boolean+ transferir (valor: double, contaDestino: string): boolean

Outro exemplo:



Utilizando o software Draw.io (ou manualmente), crie uma classe chamada aluno. Lembre de adicionar corretamente pelo menos 5 atributos e 5 métodos.

Utilizando o software Draw.io (ou manualmente), crie uma classe chamada Carro. Lembre de adicionar corretamente pelo menos 5 atributos e 5 métodos.

Parte 1

Diagrama de Classes

Professora Arianne Sarmento Torcate

Dúvidas?

