

Engenharia de Software II

Diagrama de Classes

Profa. Thaiana Pereira dos Anjos Reis, Dra. Eng.
thaiana.anjos@ifsc.edu.br

Agenda

- Diagrama de Classes
- Relacionamento entre classes
- Multiplicidade
- Exercício

Diagrama de Classes

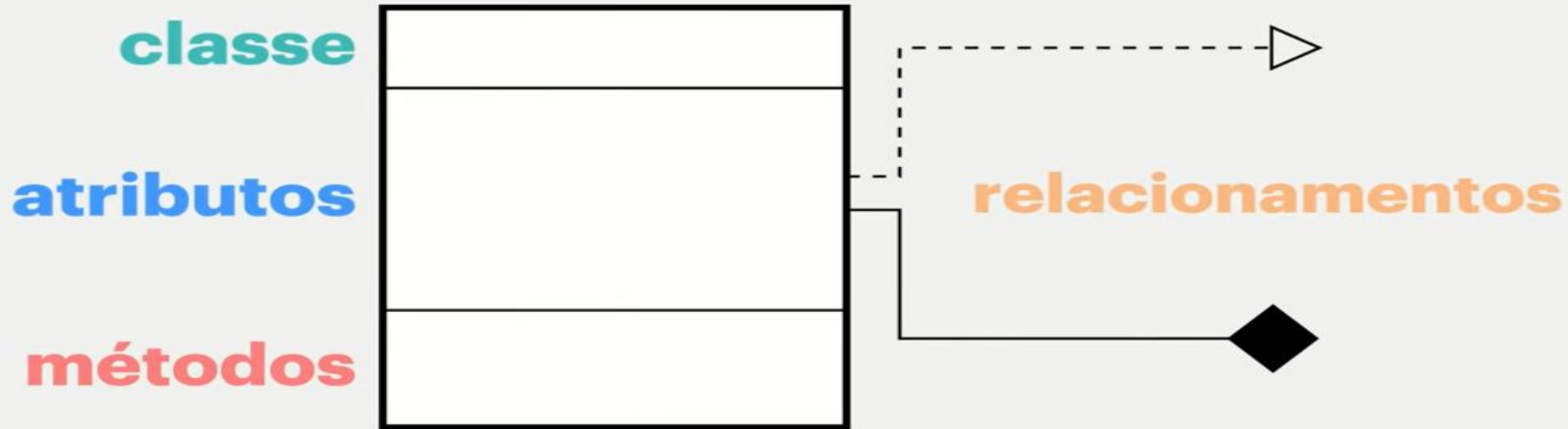
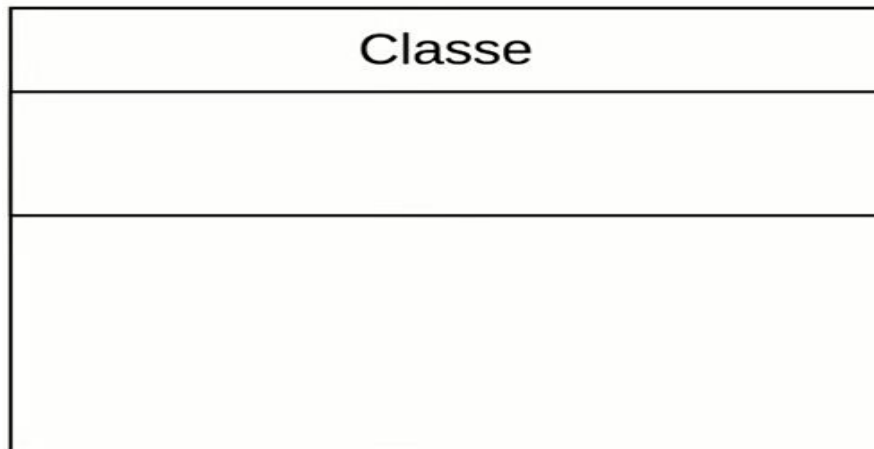


Diagrama de Classes



Boas Práticas: Nomes de Classes

O nome de uma classe deverá corresponder a um **substantivo**. A letra inicial será sempre **maiúscula**.

Caso o nome seja composto, cada palavra do nome deverá começar por uma letra maiúscula.
Exemplos: Turma, SalaDeOperacao, CaixaEletronico

Diagrama de Classes

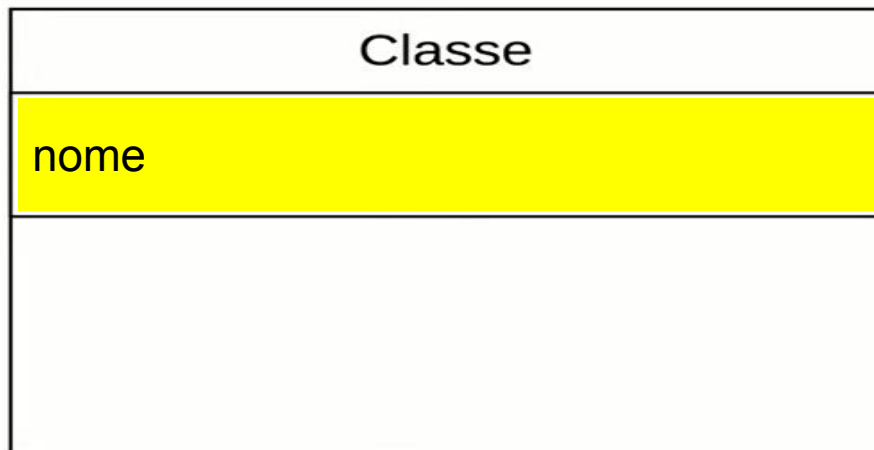
Sistema Zoológico

Animal



Diagrama de Classes

Boas Práticas: Nomes de Atributos



O nome de um atributo deverá corresponder a dados e valores que descrevem cada instância da classe. A letra inicial será sempre **minúscula**.

Caso o nome seja composto, cada palavra do nome deverá começar por uma letra maiúscula. Exemplos: nome, numeroIdentificacao, dataNascimento

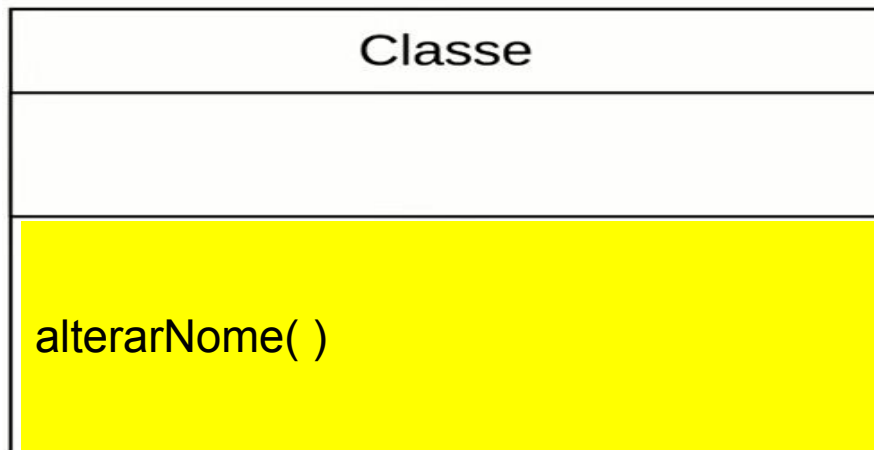
Diagrama de Classes

Sistema Zoológico

Animal
- nome: String - numIdent: Inteiro - idade: Inteiro
-



Diagrama de Classes



Boas Práticas: Nomes de Métodos

O nome de um método deverá corresponder a verbos no infinitivo. A letra inicial será sempre **minúscula**.

Caso o nome seja composto, cada palavra do nome deverá começar por uma letra maiúscula. Exemplos: nome, numeroldenficacao, dataNascimento

Diagrama de Classes

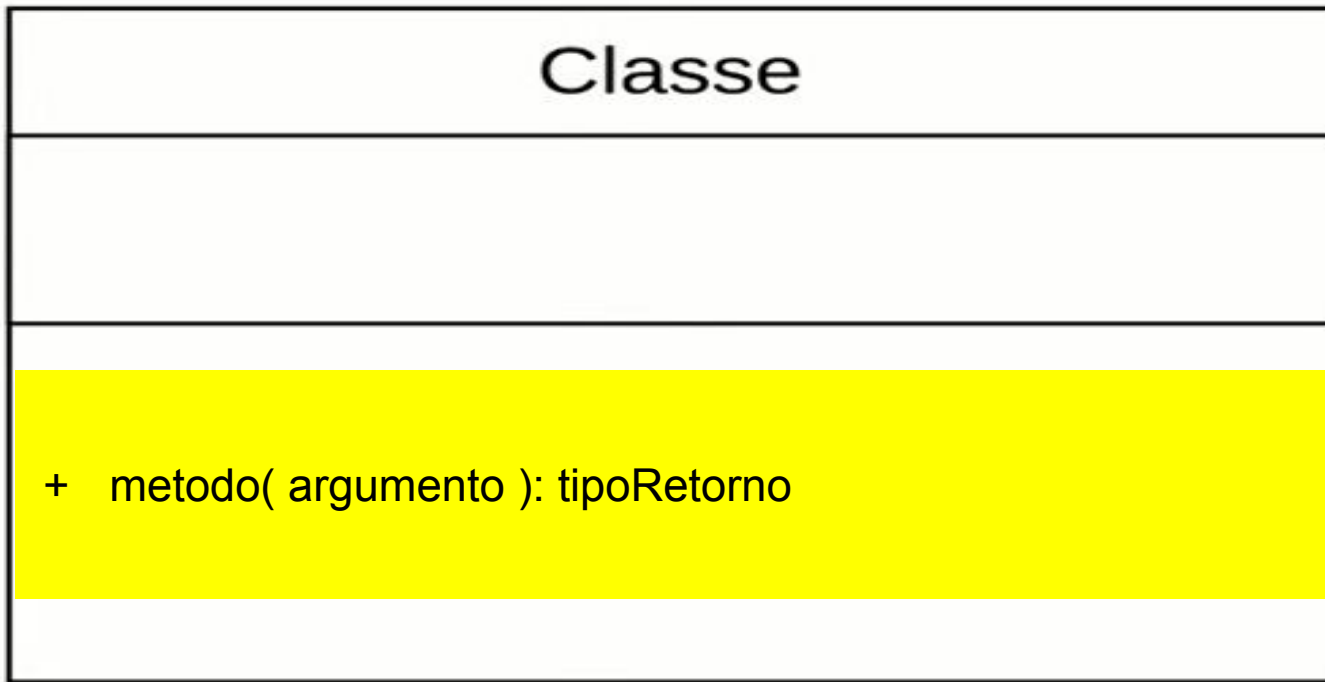


Diagrama de Classes

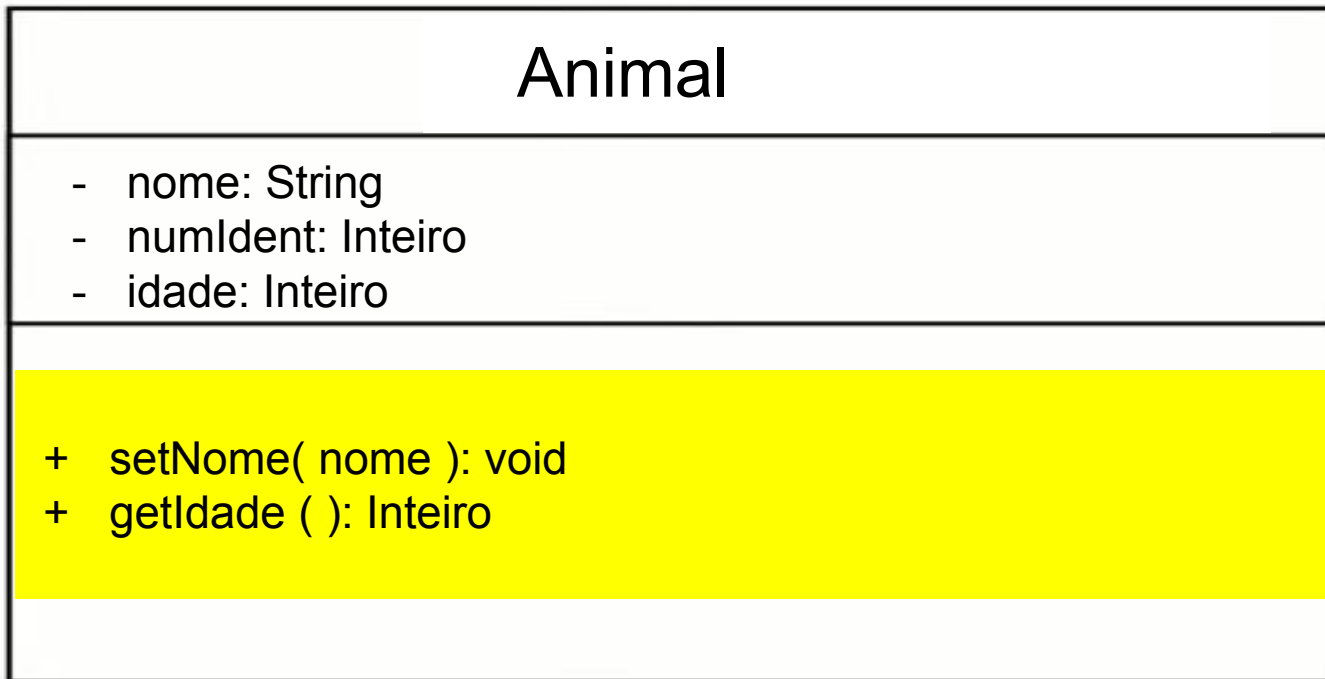


Diagrama de Classes

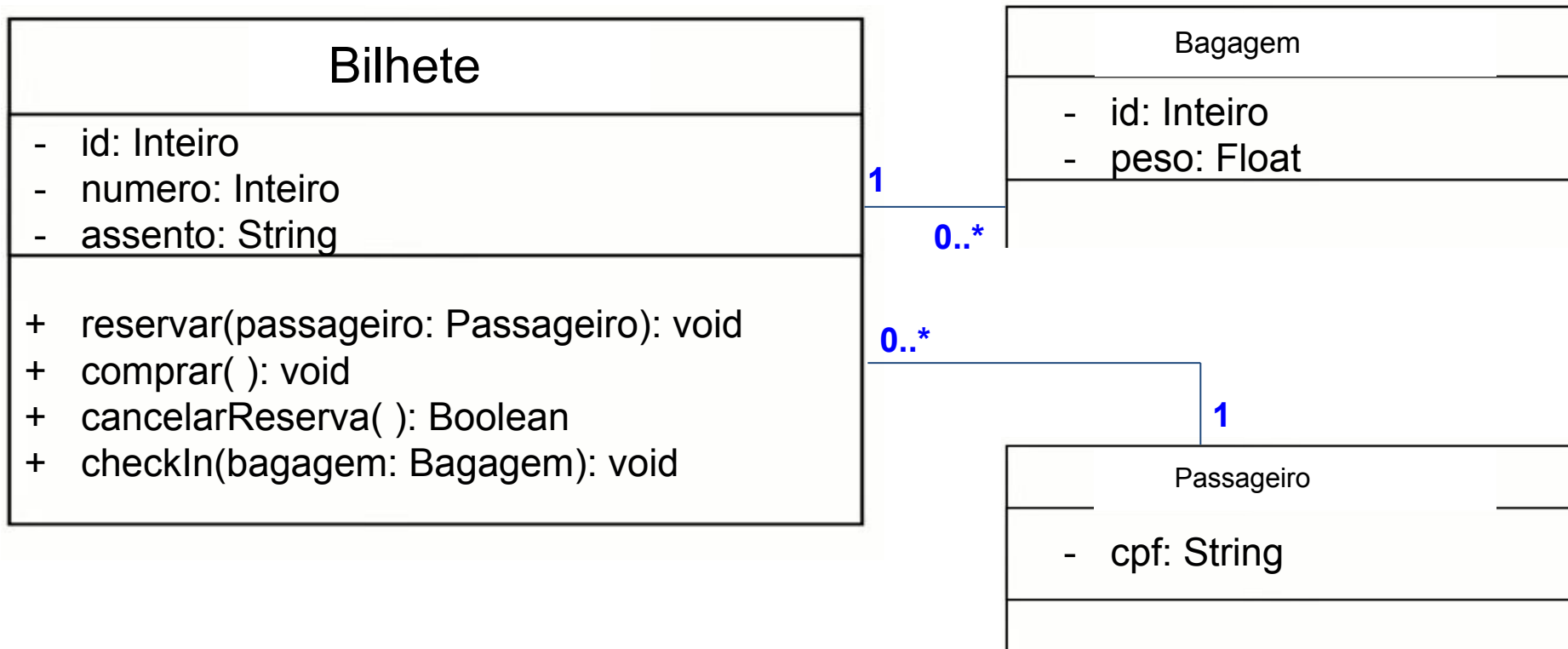


Diagrama de Classes

Sistema Zoológico

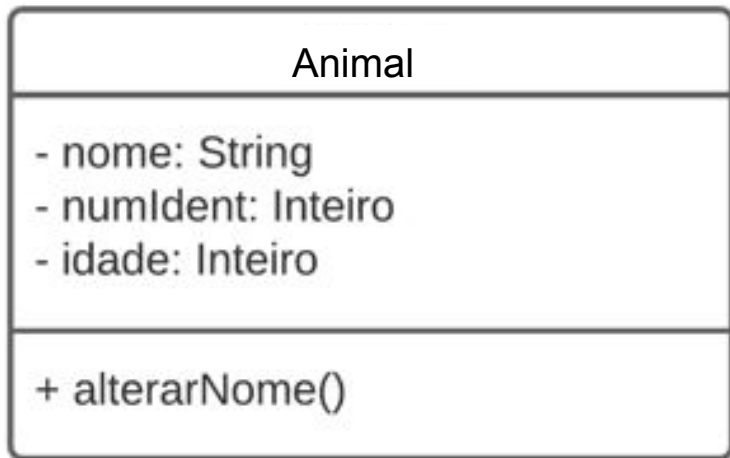
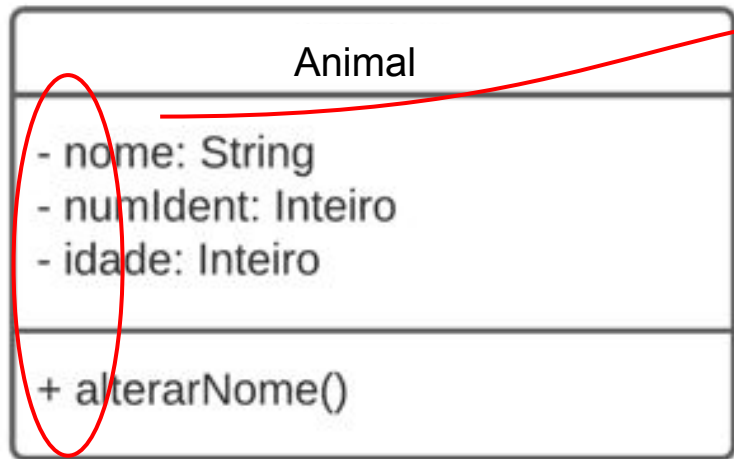


Diagrama de Classes

Visibilidade



A visibilidade de um atributo ou método define a acessibilidade desse atributo ou método.

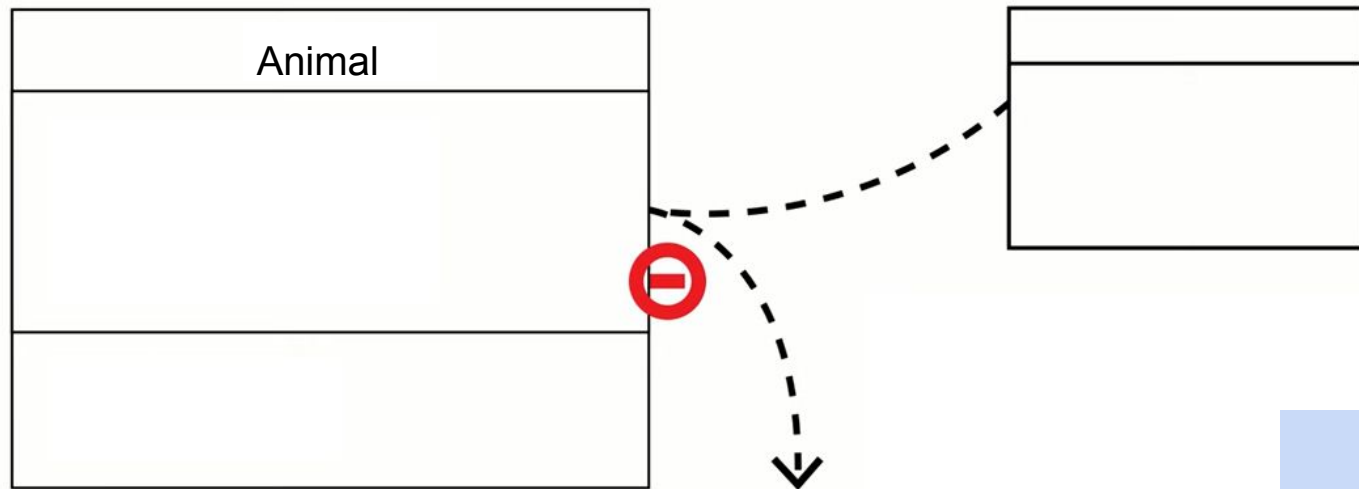
Diagrama de Classes

VISIBILIDADE

- privado
- + público
- # protegido
- ~ pacote/padrão

Diagrama de Classes

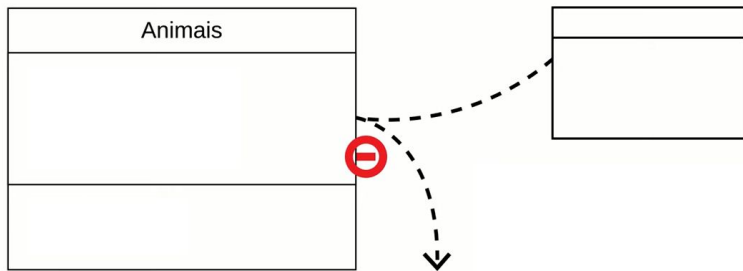
Visibilidade



- privado

Diagrama de Classes

Visibilidade

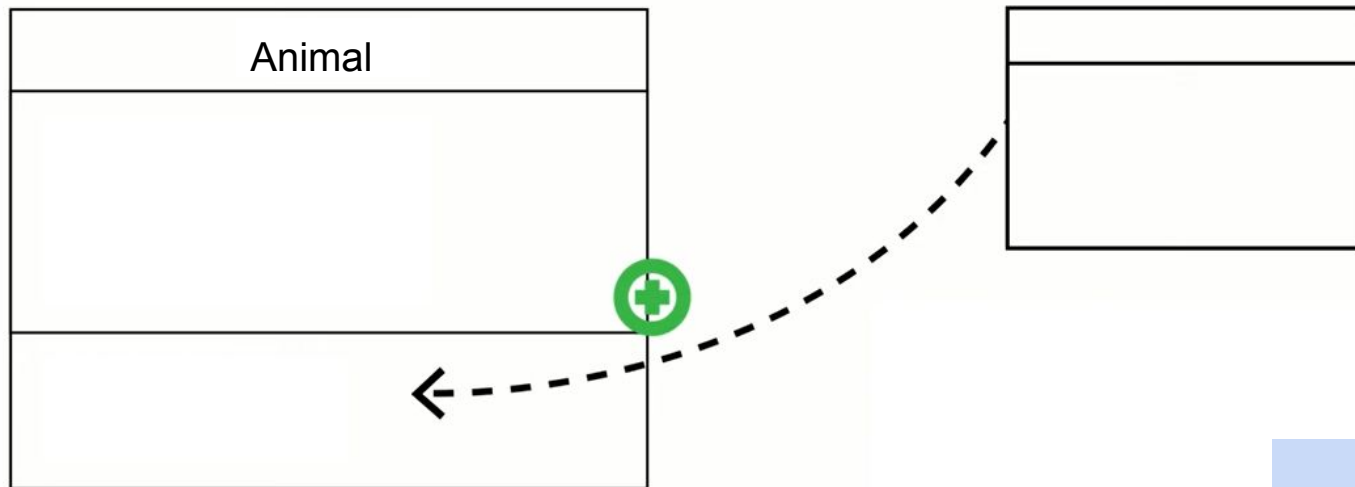


- privado

Os atributos ou métodos **não** podem ser acessados por quaisquer outras classes ou subclasses.

Diagrama de Classes

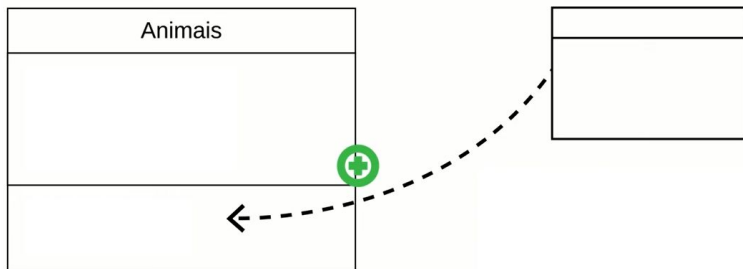
Visibilidade



+ público

Diagrama de Classes

Visibilidade

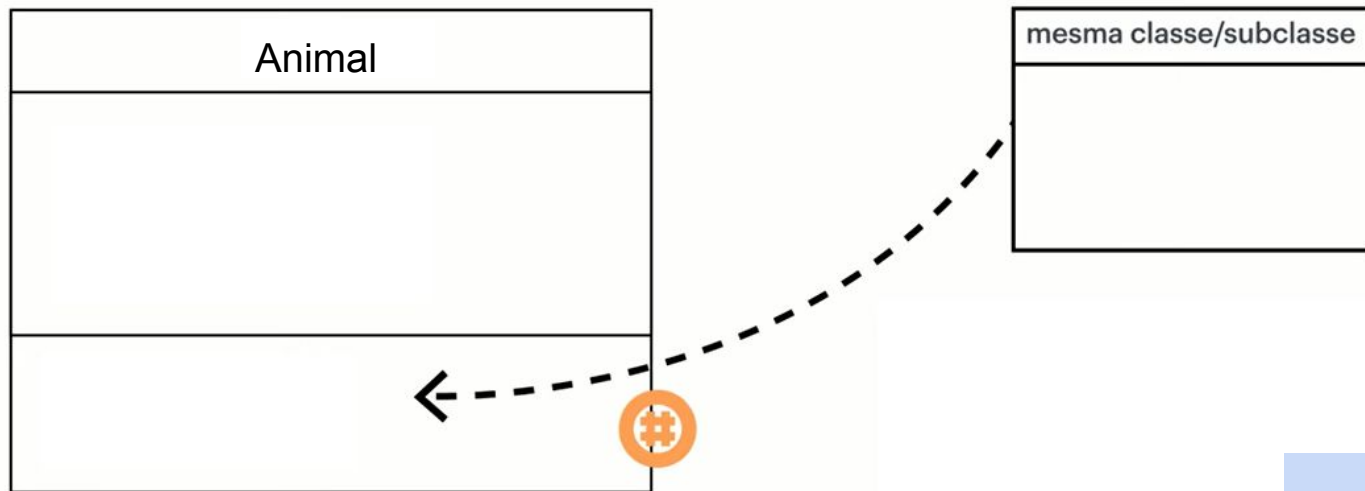


+ público

Os atributos ou métodos podem ser acessados por qualquer outra classe.

Diagrama de Classes

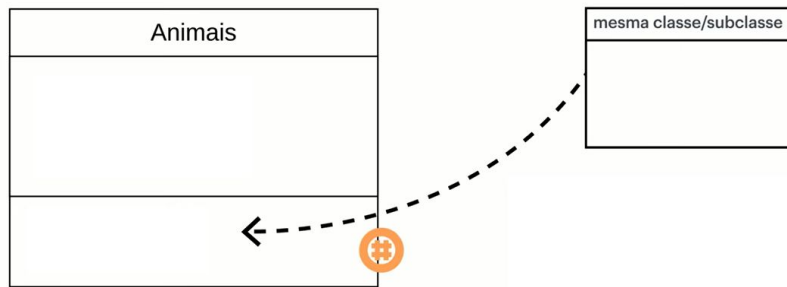
Visibilidade



protegido

Diagrama de Classes

Visibilidade



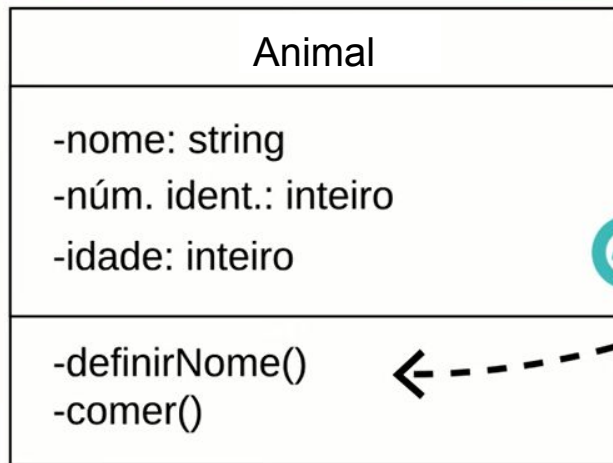
protegido

Os atributos ou métodos podem ser acessados somente por sua classe ou subclasses.

Diagrama de Classes

Visibilidade

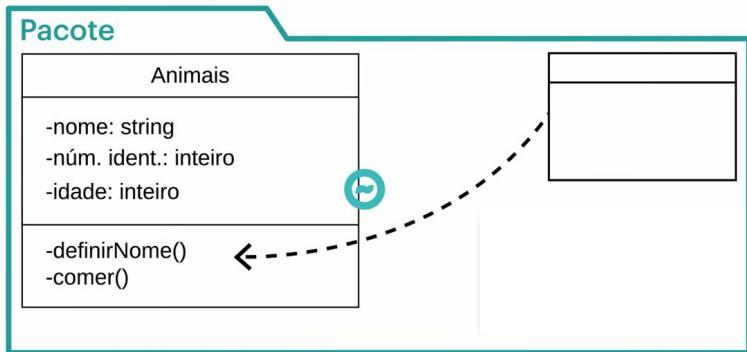
Pacote



~ pacote

Diagrama de Classes

Visibilidade



~ pacote

Os atributos ou métodos podem ser acessados por quaisquer classes ou subclasses desde que estejam dentro do mesmo pacote.

Diagrama de Classes

Normalmente,
os **atributos** são:

VISIBILIDADE

- privado
- + público
- # protegido
- ~ pacote/padrão

Normalmente,
os **métodos** são:

VISIBILIDADE

- privado
- + público
- # protegido
- ~ pacote/padrão

Diagrama de Classes

Exemplo

Cachorro
<ul style="list-style-type: none">- nome: String- raca: String- peso: Float
+ definirPorteCachorro()



nome: Bob

raca: Pug

peso: 10



nome: Mike

raca: SRD

peso: 20

Diagrama de Classes

Exemplo

Funcionario
<ul style="list-style-type: none">- nome: String- telefone: String- departamento: String
<ul style="list-style-type: none">+ atualizarTelefone()



nome: Arthur

telefone: 483344-9888

departamento: Financeiro

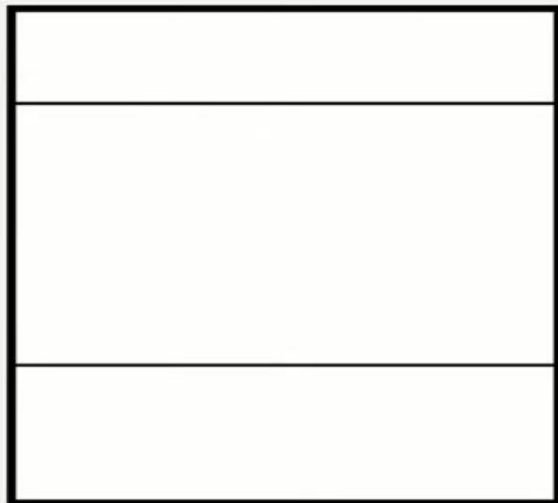
Diagrama de Classes

RELACIONAMENTOS

classe

atributos

métodos



relacionamentos



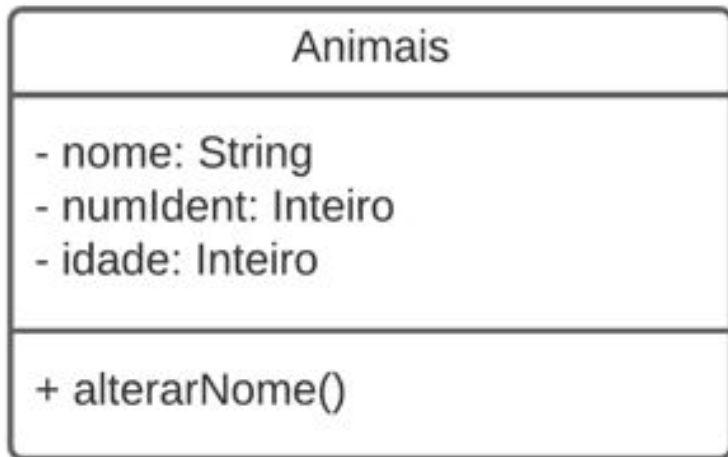
Diagrama de Classes

Herança



Diagrama de Classes

Sistema Zoológico



Sistema Zoológico

Animais
- nome: String - numIdent: Inteiro - idade: Inteiro
+ alterarNome()



Elefante
- nome: String - numIdent: Inteiro - idade: Inteiro
+ alterarNome()



Onça
- nome: String - numIdent: Inteiro - idade: Inteiro
+ alterarNome()



Panda
- nome: String - numIdent: Inteiro - idade: Inteiro
+ alterarNome()

Sistema Zoológico

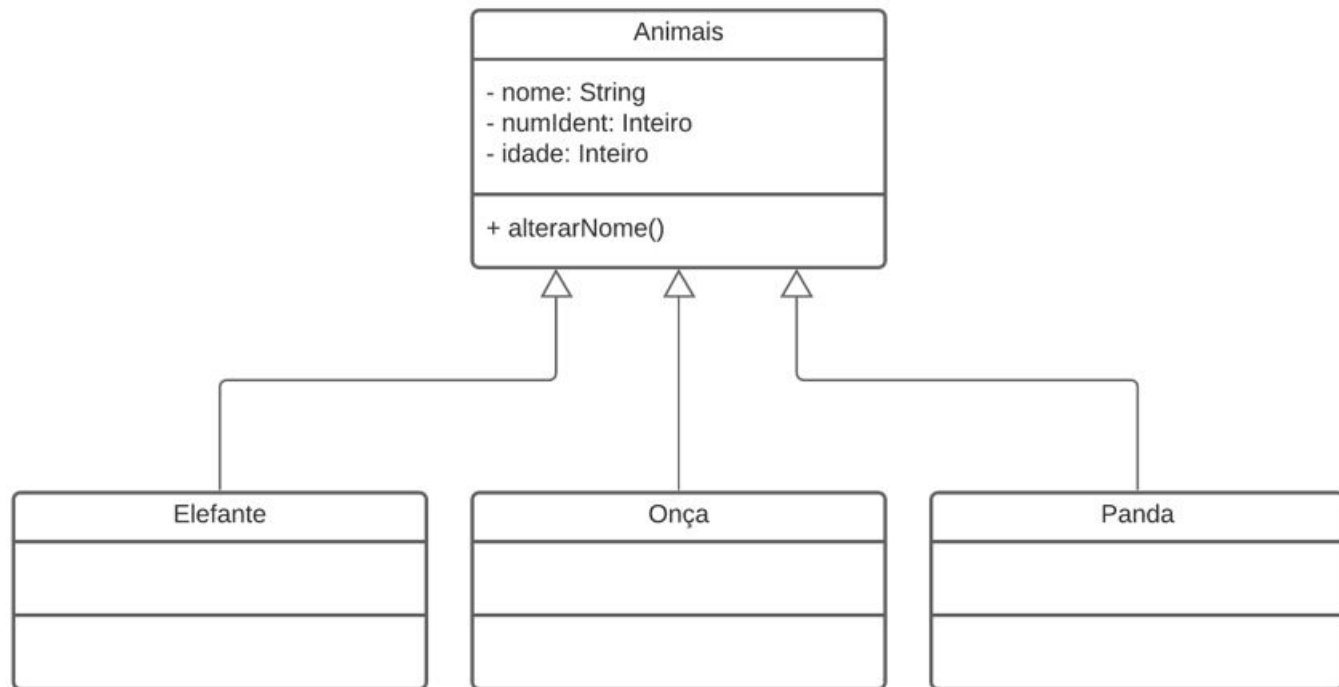
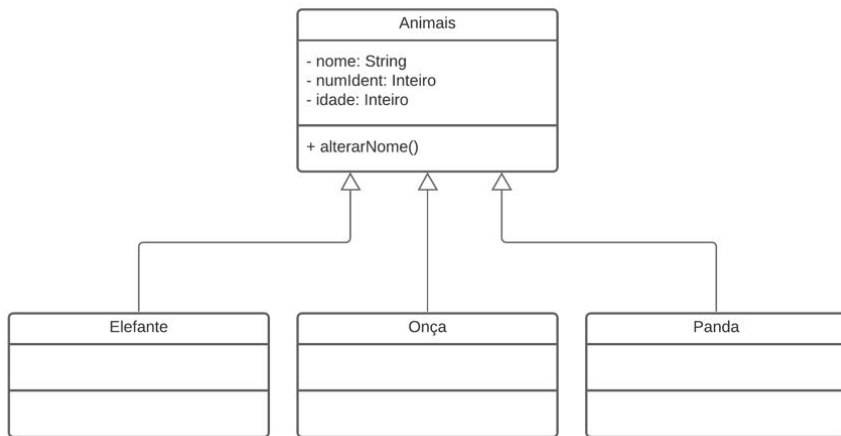


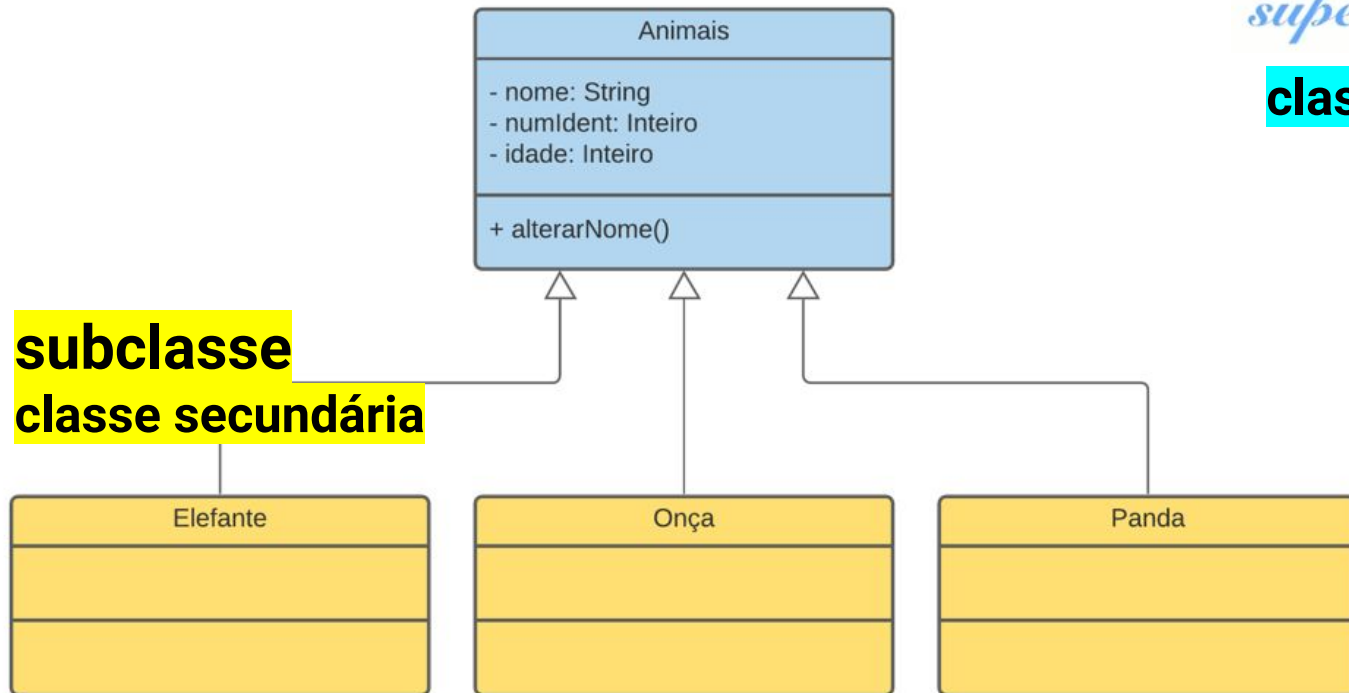
Diagrama de Classes

Herança →



A Herança possibilita que as classes compartilhem seus **atributos e métodos** entre si.

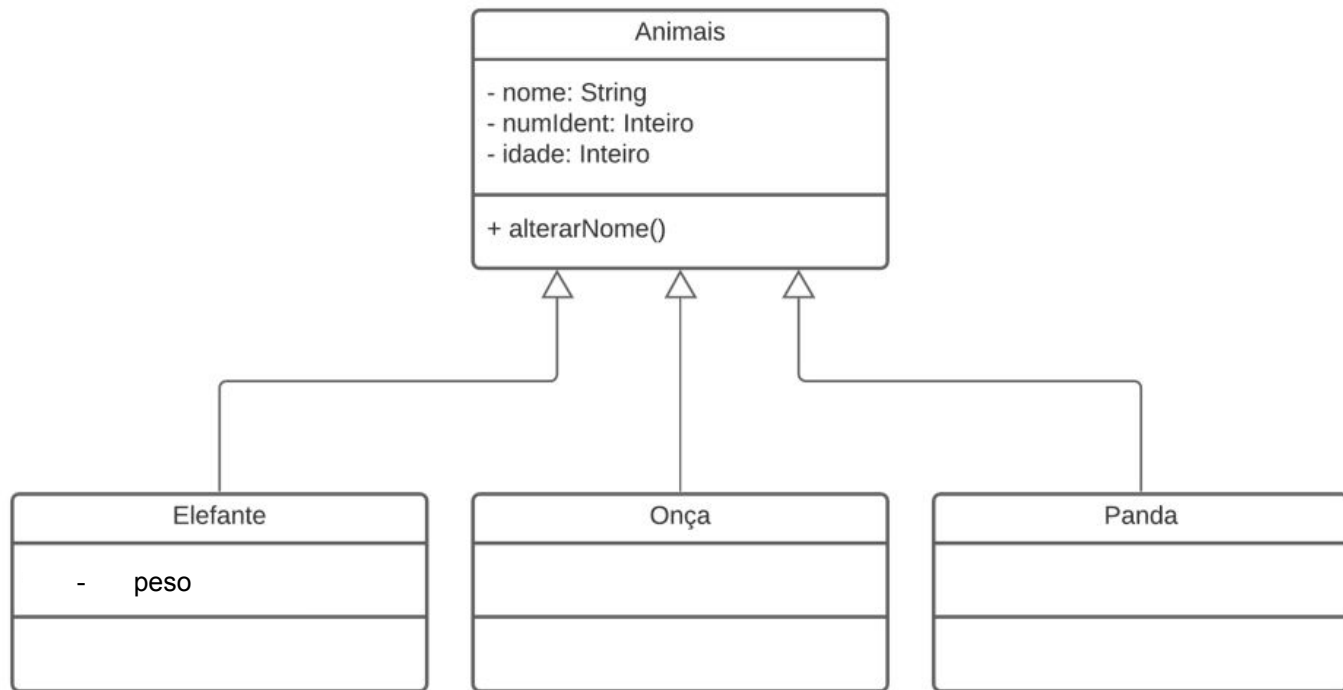
subclasse
classe secundária



superclasse 

classe primária

Sistema Zoológico



Outro Exemplo

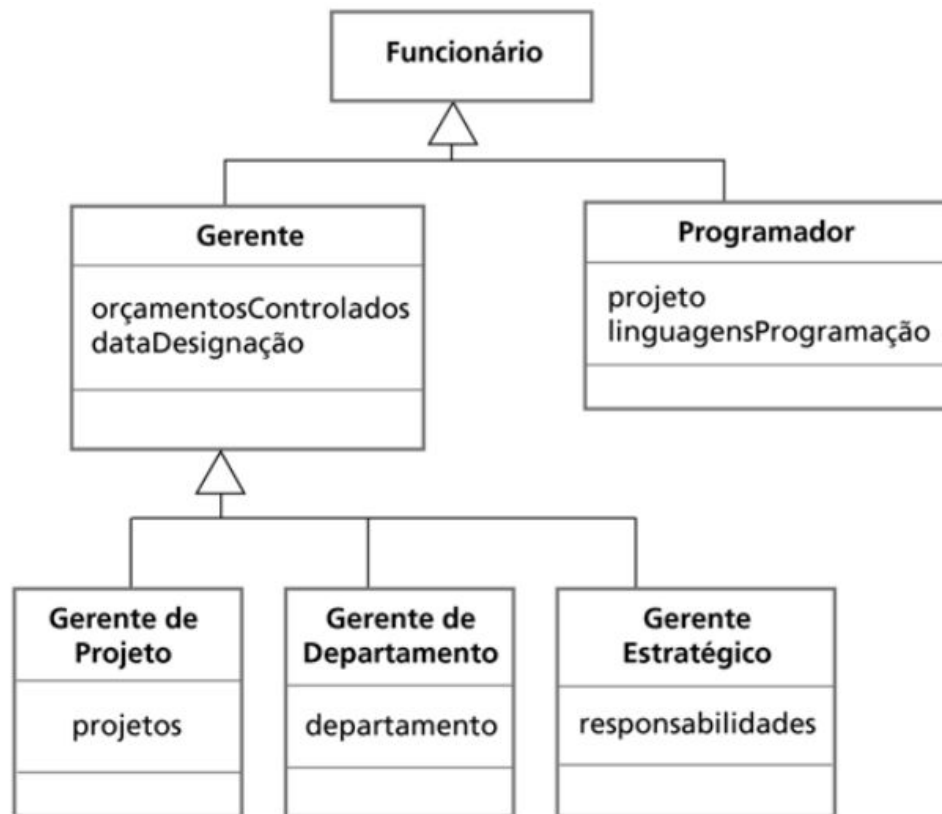
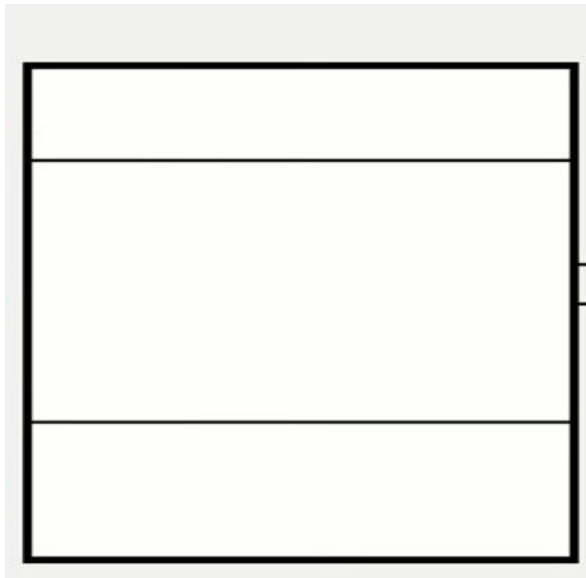


Diagrama de Classes

CLASSE ABSTRATA

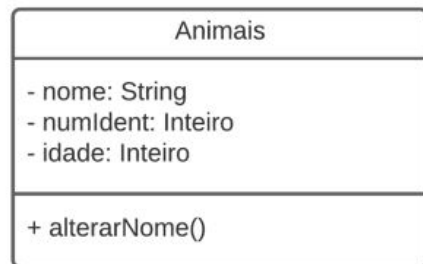


As classes abstratas são as que **não permitem realizar qualquer tipo de instância.**

São classes feitas especialmente para serem modelos para suas classes derivadas.

Sistema Zoológico

classe abstrata

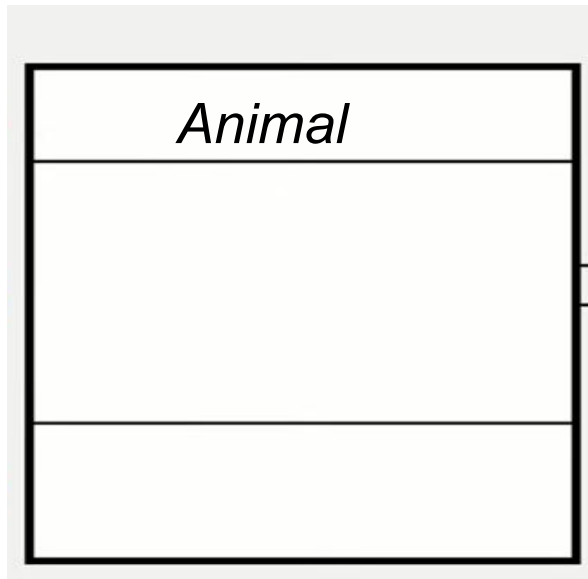


classes concretas



Diagrama de Classes

CLASSE ABSTRATA



Para sinalizar que a classe é abstrata, colocamos o seu nome em itálico ou entre os símbolos << >>

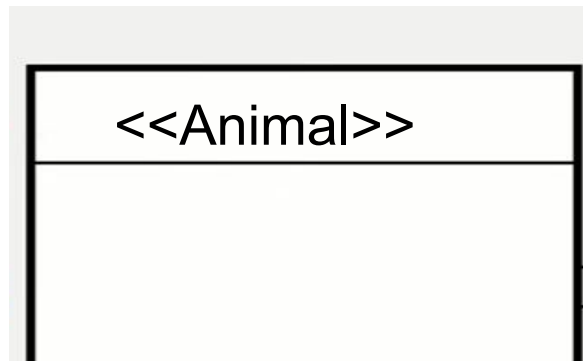
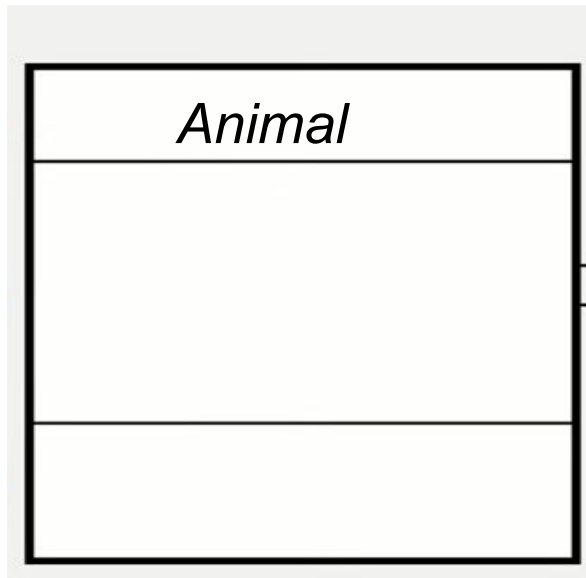


Diagrama de Classes

CLASSE ABSTRATA



As classes derivadas, via de regra, deverão **sobrescrever os métodos** para realizar a implementação dos mesmos.



Sobrescrita de método



Sistema financeiro para um banco

Diagrama de Classes

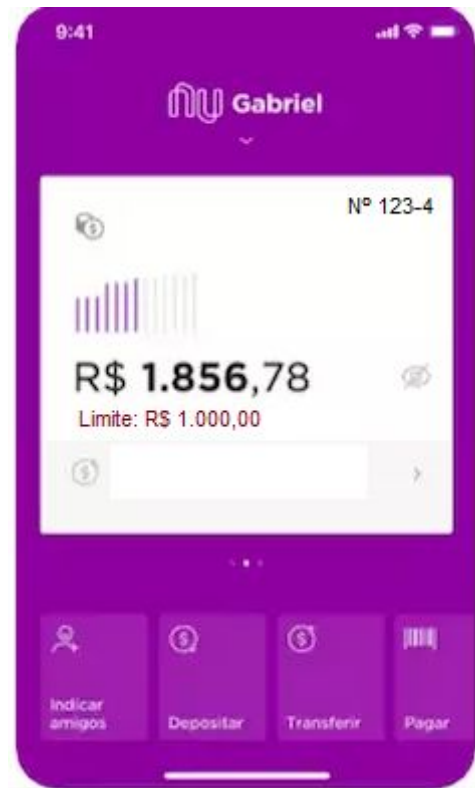
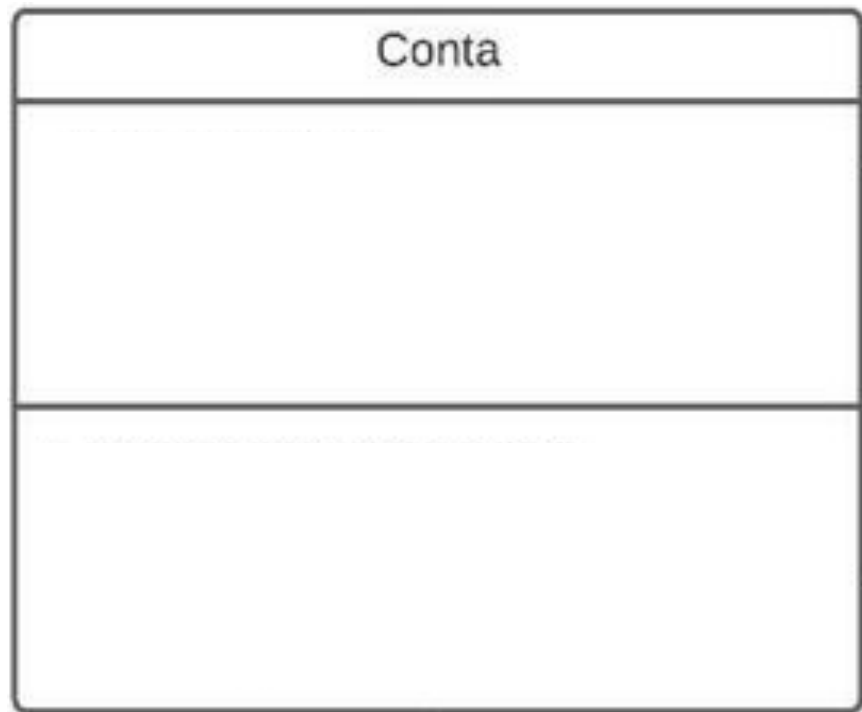


Diagrama de Classes

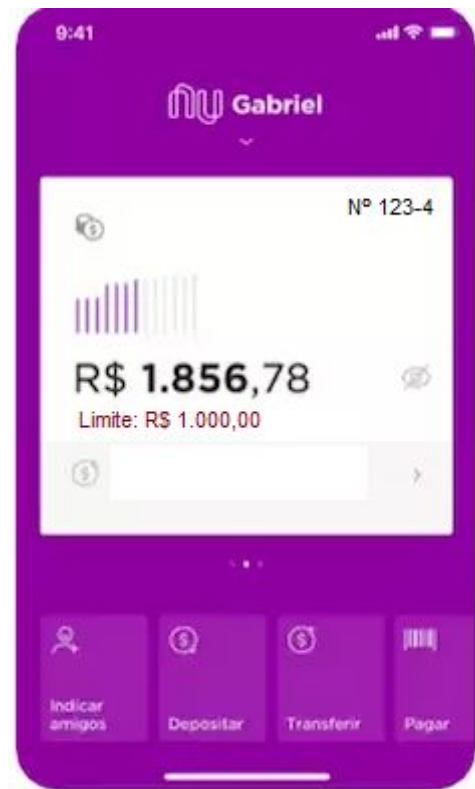
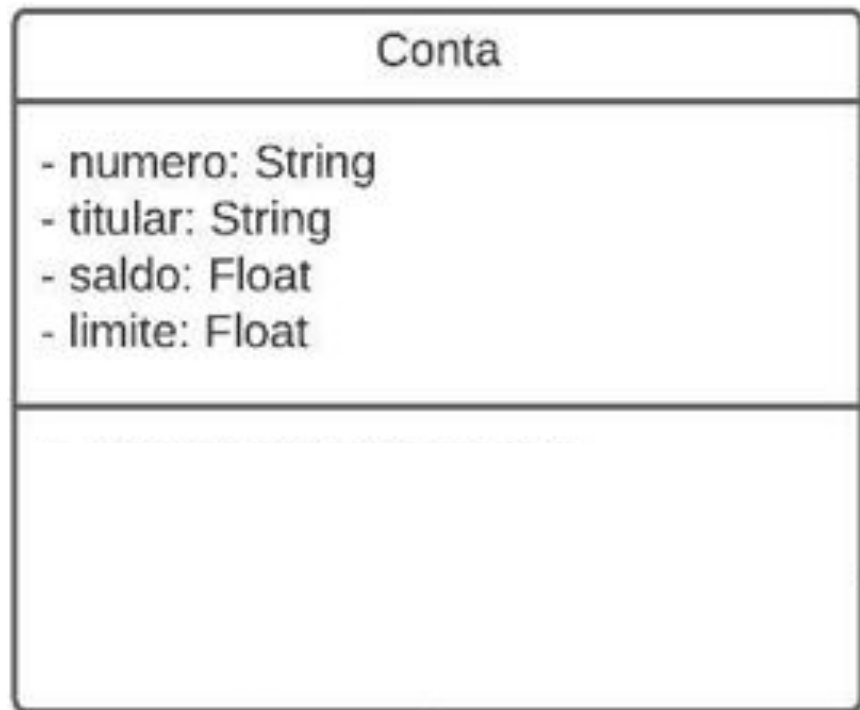


Diagrama de Classes

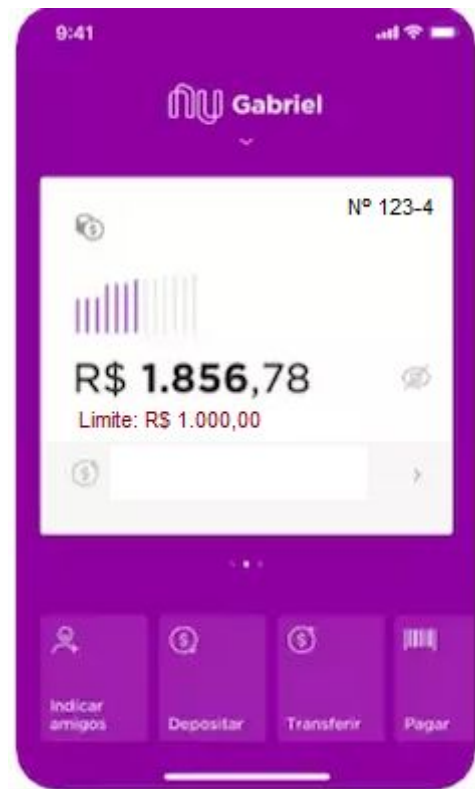
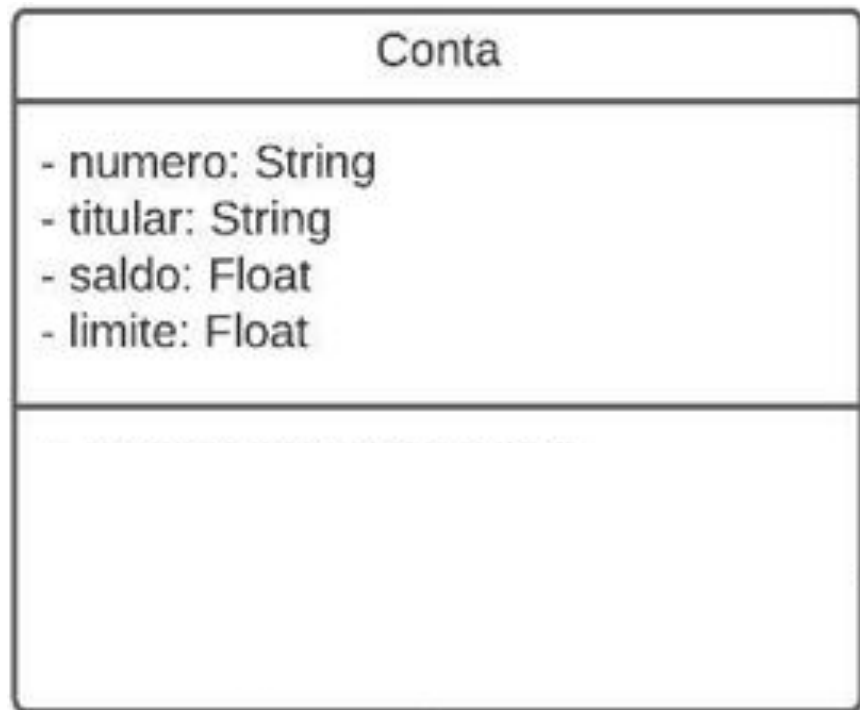


Diagrama de Classes

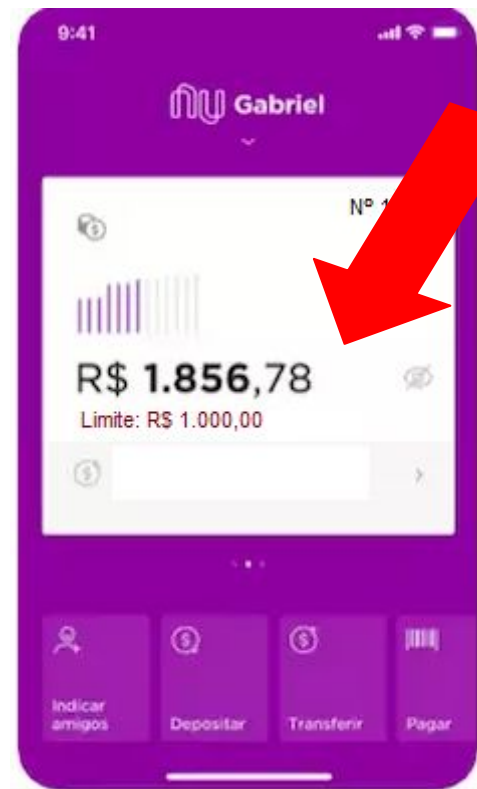
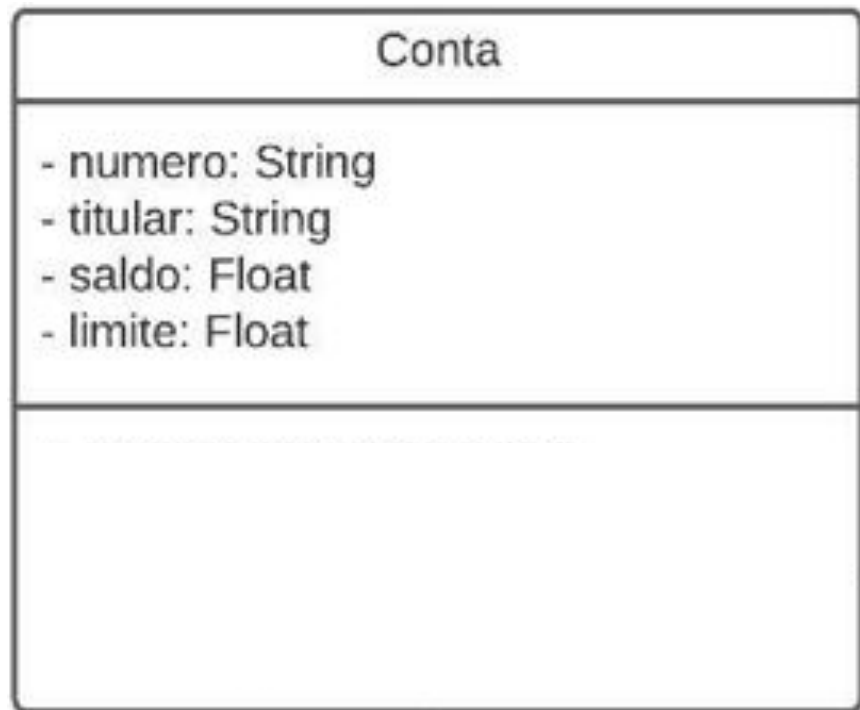
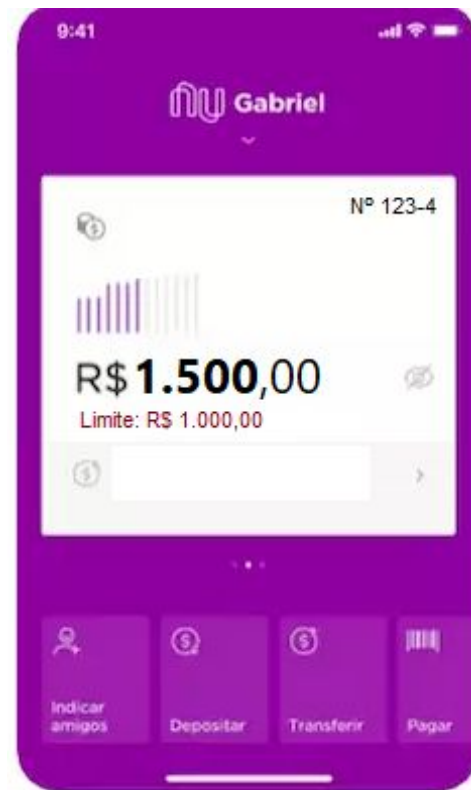


Diagrama de Classes



RELACIONAMENTOS ENTRE CLASSES

Diagrama de Classes

TIPOS DE ASSOCIAÇÃO

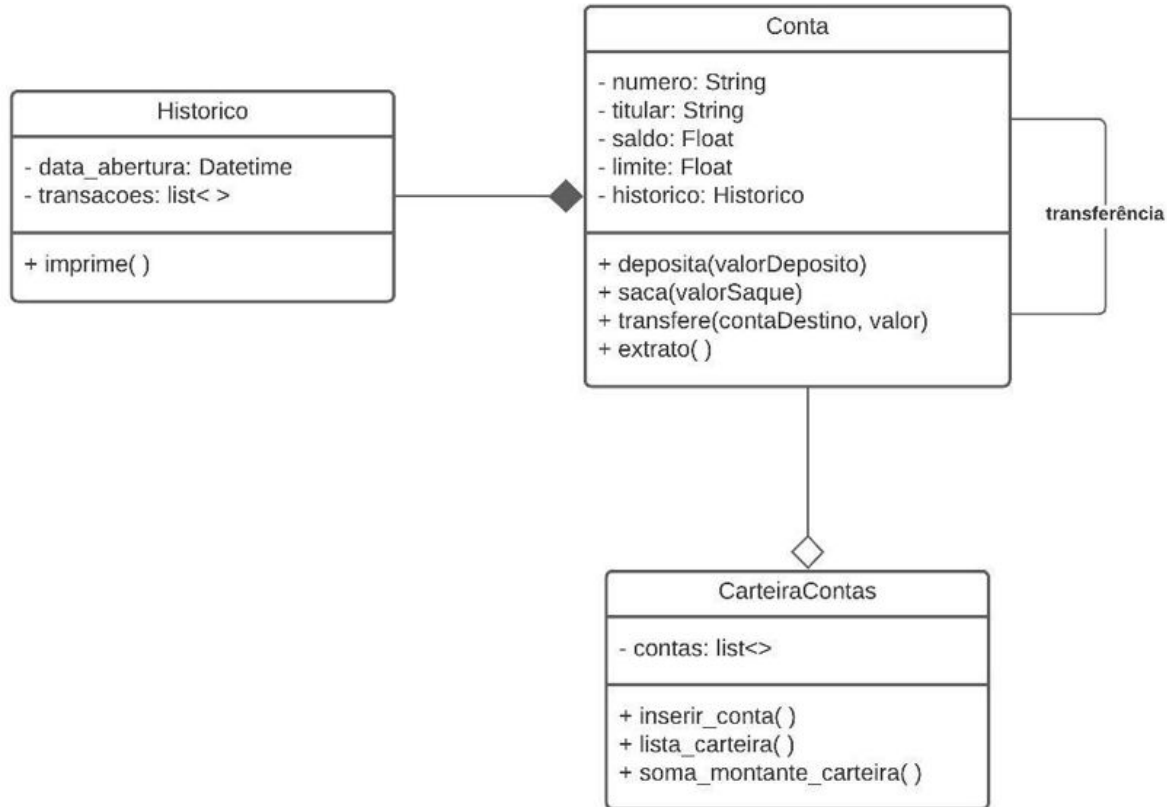


Diagrama de Classes

ASSOCIAÇÃO SIMPLES

Os objetos de uma classe precisam obter informações de objetos de outra classe.

Não existe dependência entre as classes.



Diagrama de Classes

ASSOCIAÇÃO SIMPLES

Os objetos de uma classe precisam obter informações de objetos de outra classe (ou da mesma classe).

Não existe dependência entre as classes.



Objeto da classe Conta acessa / tem conhecimento / utiliza objeto da classe Conta também. Isso acontece por meio da operação de transferência.

Diagrama de Classes

AGREGAÇÃO

**Carteira de
Contas**

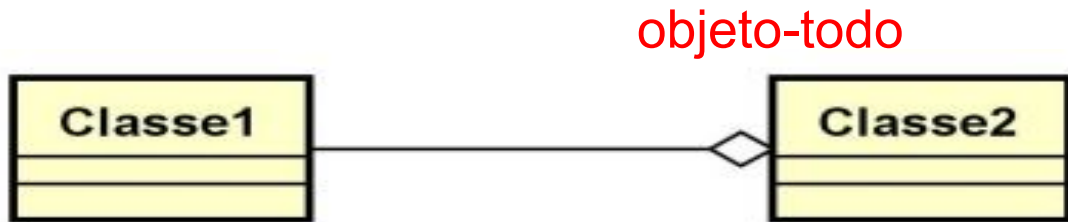


Diagrama de Classes

AGREGAÇÃO

É um tipo de associação e especifica um TODO e suas PARTES.

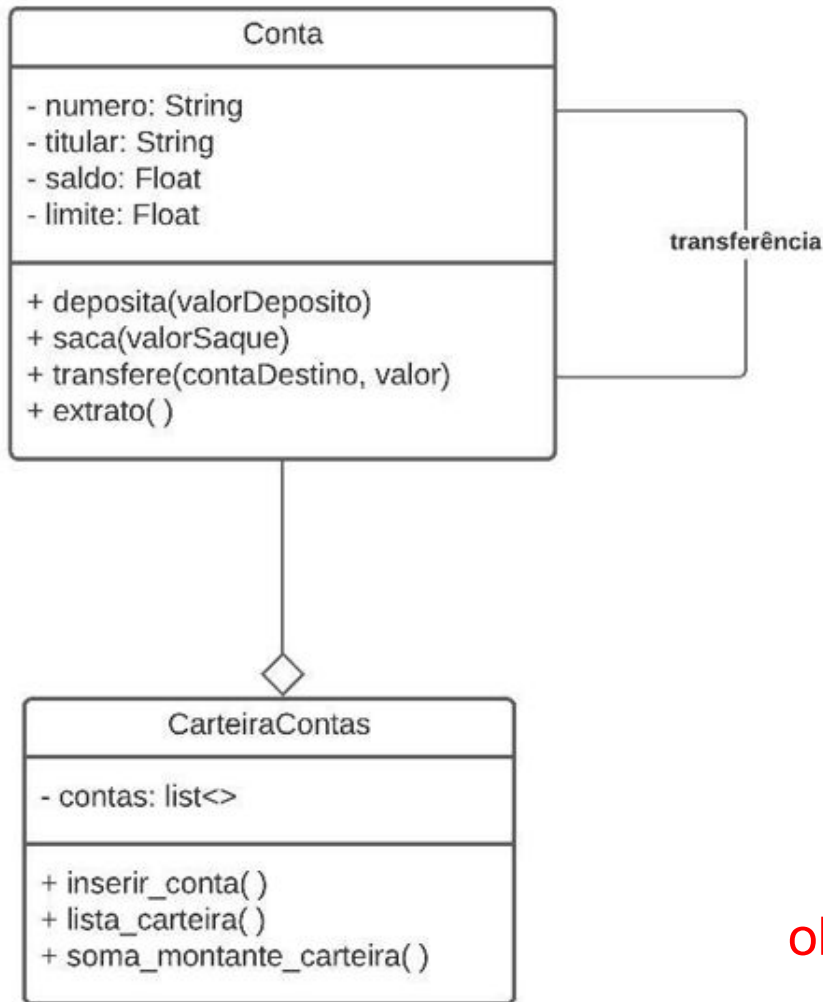
Aqui, a PARTE pode existir fora do TODO.



Diag

É um tipo de associação e especifica um TODO e suas PARTES.

Aqui, a PARTE pode existir fora do TODO.



objeto-todo

Diagrama de Classes

COMPOSIÇÃO

**Histórico de
Transferências**

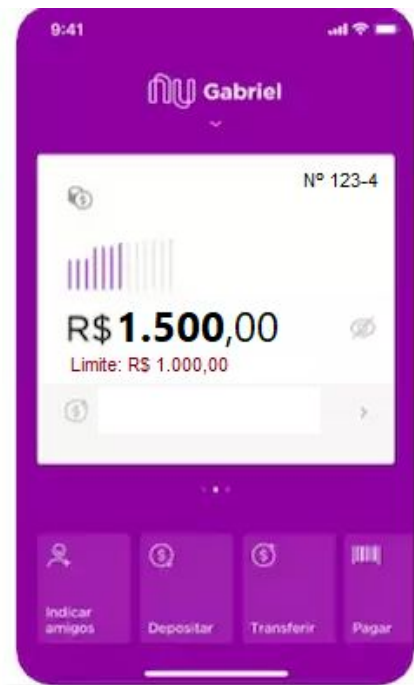
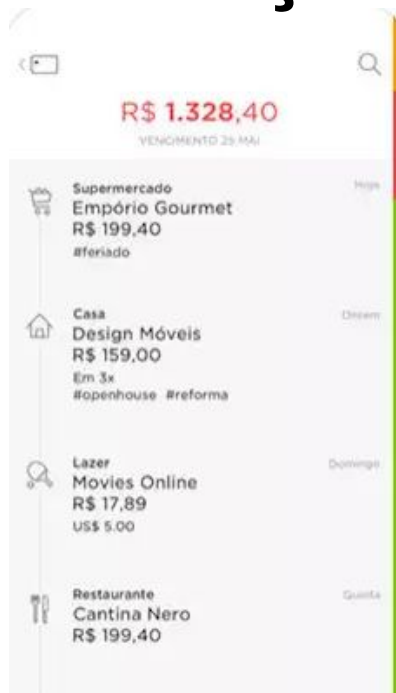


Diagrama de Classes

COMPOSIÇÃO

É um tipo de associação e especifica um TODO e suas PARTES.

Aqui, a PARTE **NÃO** consegue existir fora do TODO

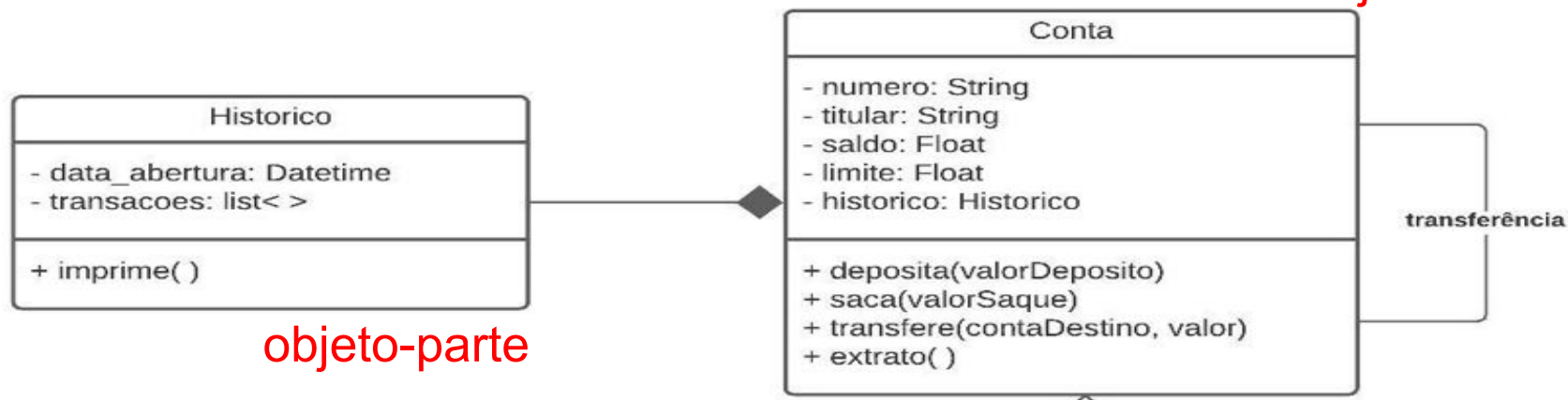


Diagrama de Classes

COMPOSIÇÃO

Se a **Conta** for excluída? O que acontece com o Histórico?

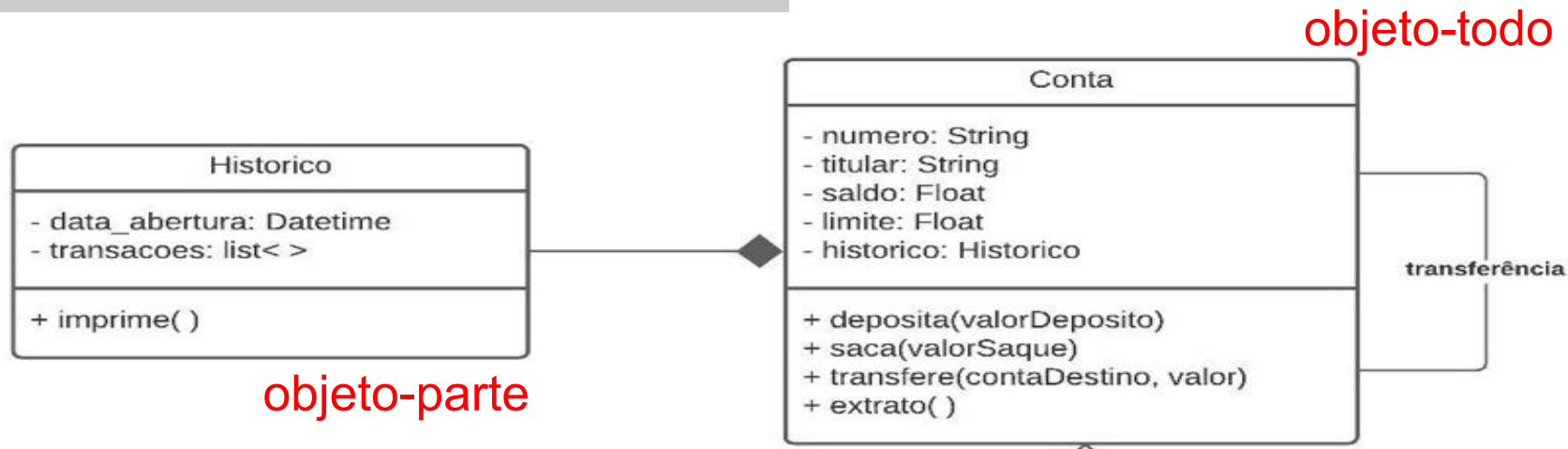


Diagrama de Classes

COMPOSIÇÃO

**Histórico de
Transferências**

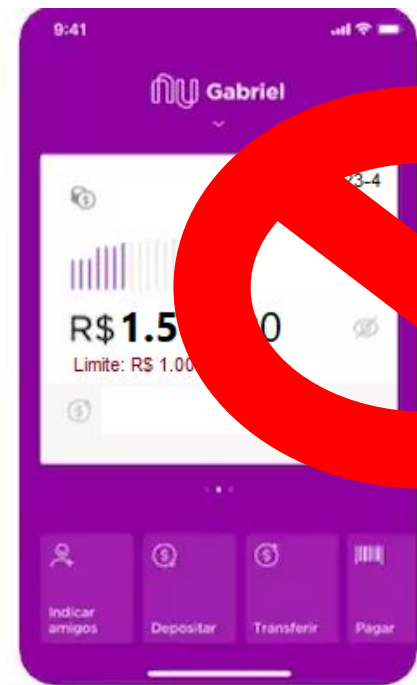
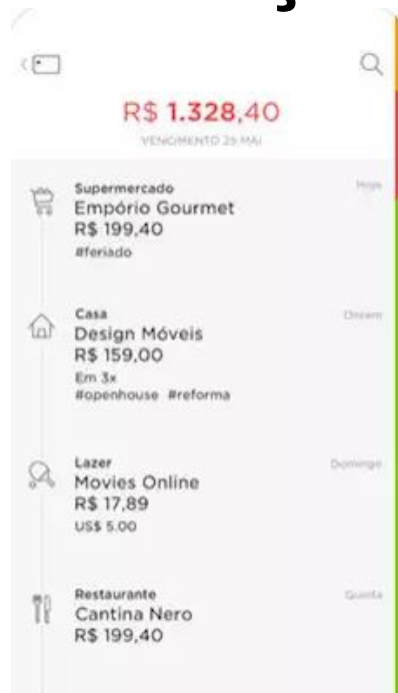


Diagrama de Classes

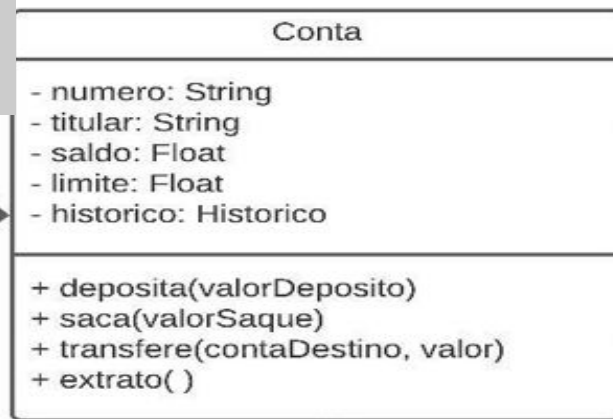
COMPOSIÇÃO

Se a **Conta** for excluída? O que acontece com o Histórico?

Quando o objeto-todo é excluído, os objetos partes também são!



objeto-parte



objeto-todo

transferência

Diagrama de Classes

TIPOS DE ASSOCIAÇÃO

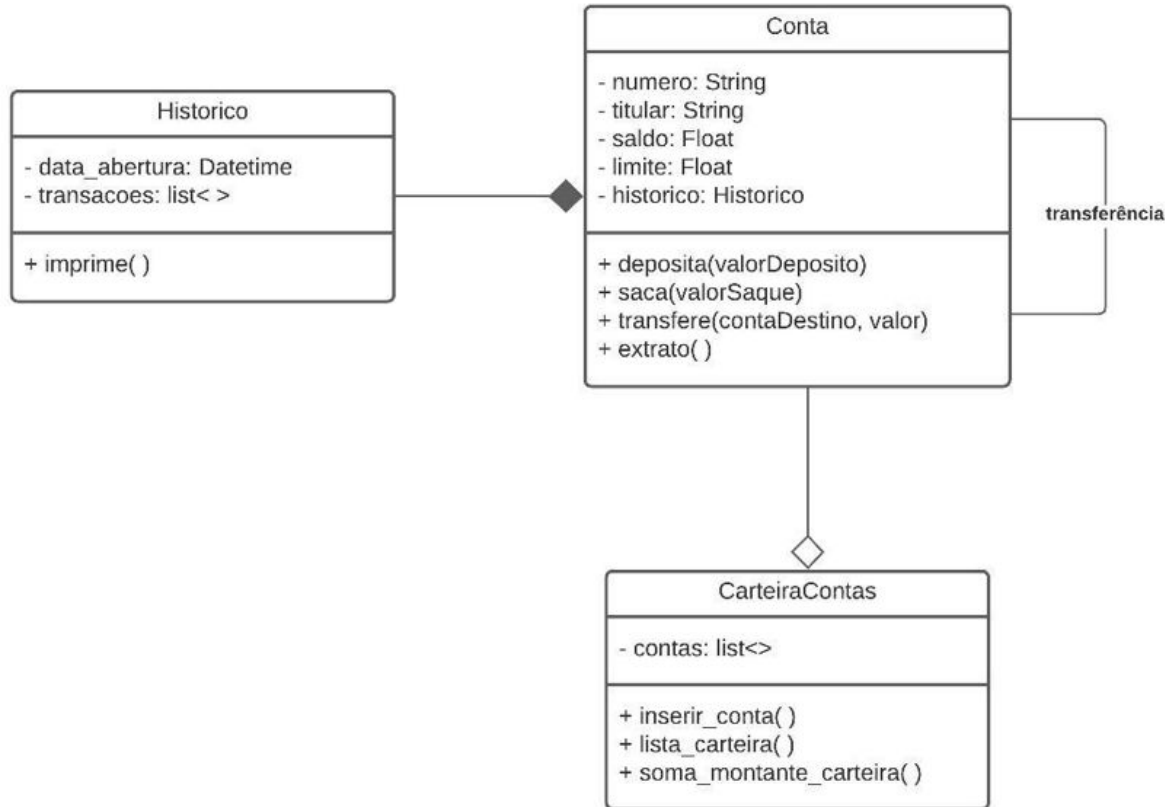


Diagrama de Classes

MULTIPLICIDADE

Descreve o relacionamento entre as classes e permite definir **restrições numéricas** nesses relacionamentos.

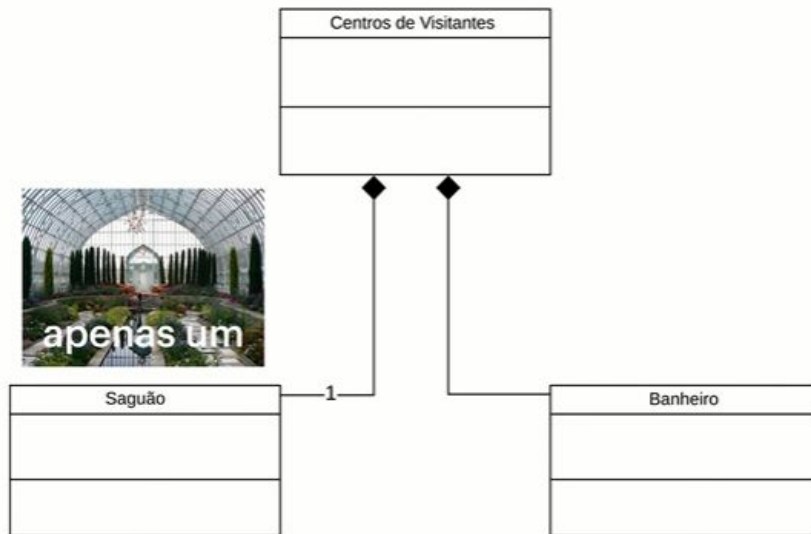


Diagrama de Classes

MULTIPLICIDADE

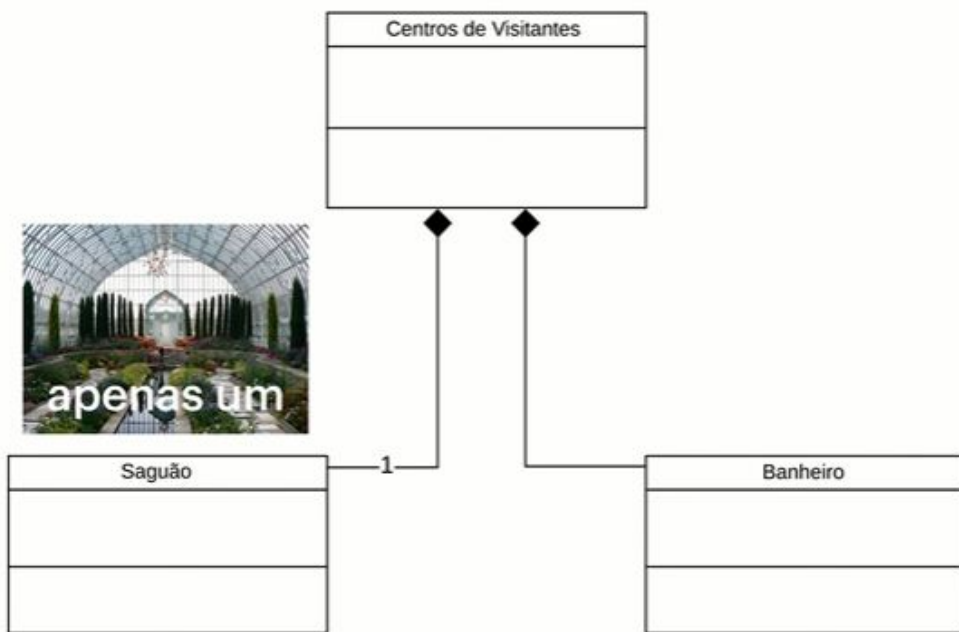


Diagrama de Classes

MULTIPLICIDADE

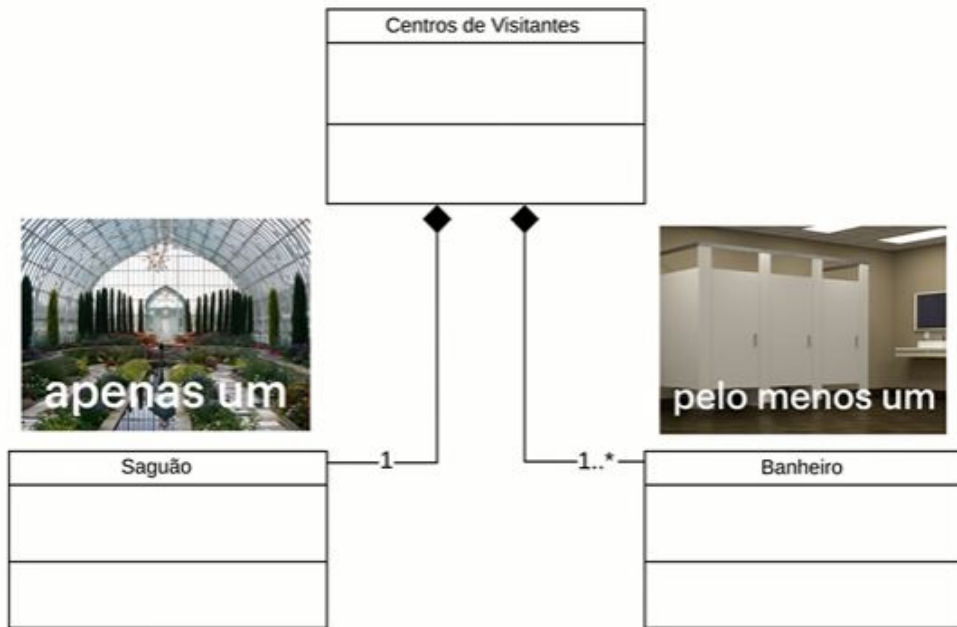
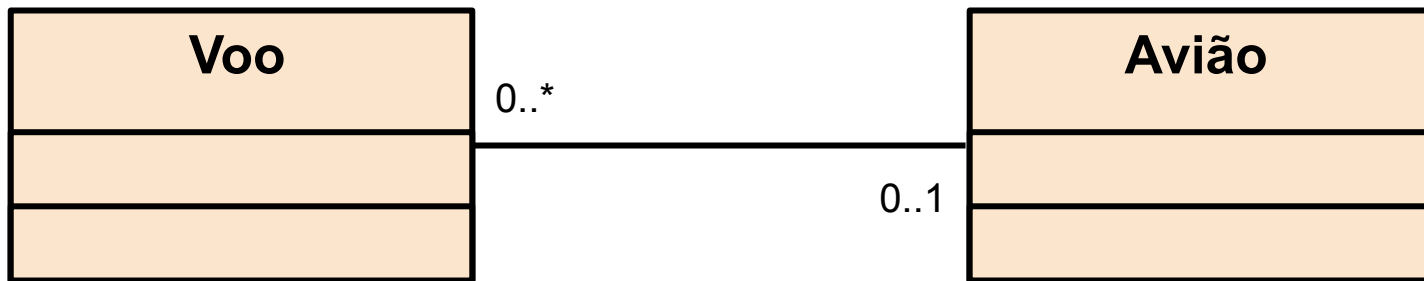


Diagrama de Classes

MULTIPLICIDADE

Associação Simples.

Ex: um voo é cadastrado para um avião



Leitura do diagrama:

Um Voo pode ter 0 ou 1 avião associado.

Um avião pode estar associado a nenhum ou vários voos.

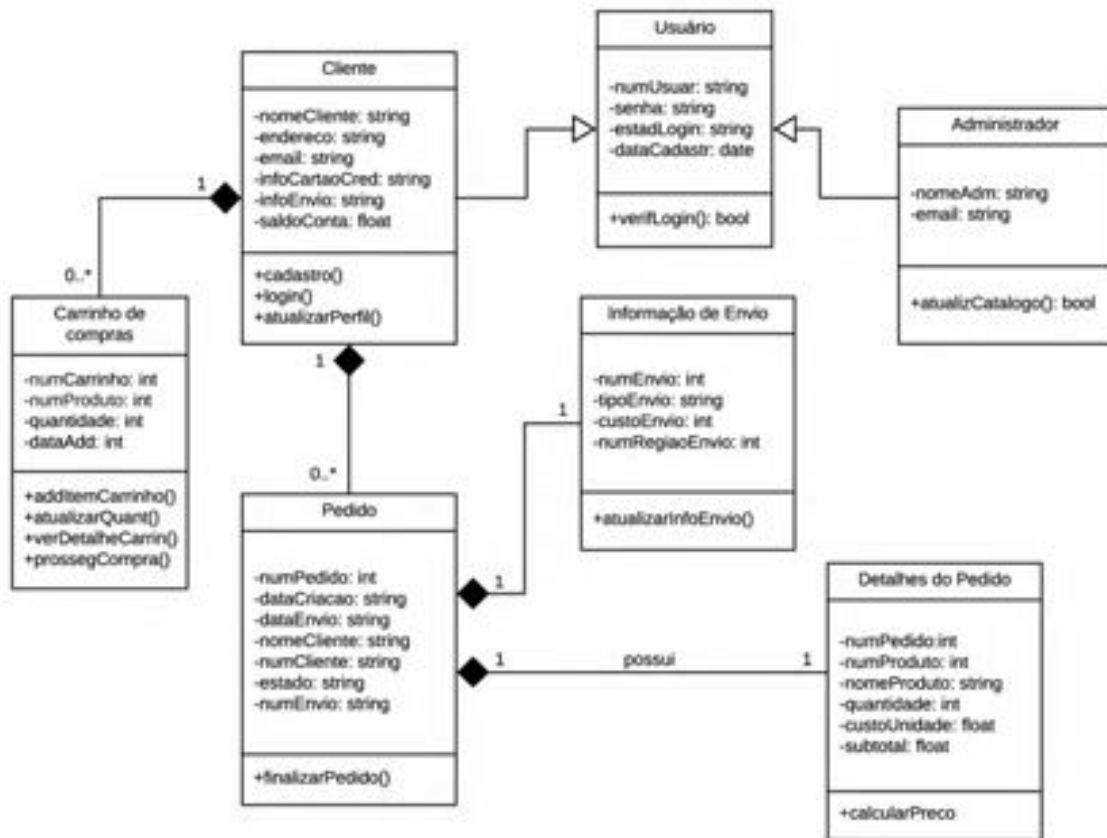
Diagrama de Classes

MULTIPLICIDADE

0..1	No máximo um. Indica que os objetos da classe associada não precisam obrigatoriamente estar relacionados.
1..1	Um e somente um. Indica que apenas um objeto da classe se relaciona com os objetos da outra classe.
0..*	Muitos. Indica que podem haver muitos objetos da classe envolvidos no relacionamento
1..*	Um ou muitos. Indica que há pelo menos um objeto envolvido no relacionamento.
3..5	Valores específicos.

Exemplo real de um sistema

Carrinho de compras online

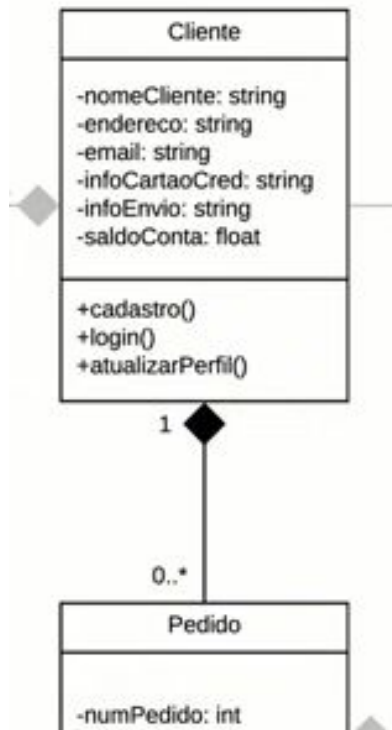


Exemplo

- Verificar no diagrama:
 - Cliente e Administrador herdam de Usuário
 - Administrador pode atualizar o catálogo, já o cliente não
 - Composição: **Carrinho de compras** e **Pedido** são partes de um **Cliente**, ou seja, se Cliente foi excluído, seu carrinho e seus pedidos também são excluídos.
 - Da mesma forma com **Pedido**, se ele for excluído, **Informação de envio** e **Detalhes do Pedido** também são excluídos

Exemplo

- Verificar no diagrama:
 - Multiplicidade: o Cliente pode se cadastrar e nunca ter feito um pedido (0) ou pode ser um cliente assíduo e fazer diversos pedidos (*)
 - E por outro lado, um pedido pode pertencer somente a 1 Cliente



Exercício

Exercício

- Cada estudante deverá analisar o código do seu projeto realizado na UC Programação Orientada a Objetos e criar o Diagrama de Classes do seu projeto.
 - Utilize o software Lucidchart ou outro similar;
 - Desenhe as classes (atributos e métodos)
 - Nos métodos, especifique os argumentos e o retorno
 - Insira os relacionamentos
 - Insira as multiplicidades

Dúvidas?

Profa. Thaiana Pereira dos Anjos Reis, Dra. Eng.

thaiana.anjos@ifsc.edu.br

