Metodologia e Linguagem de Programação Orientada a Objetos

Drª. Alana Morais

Aula Passada

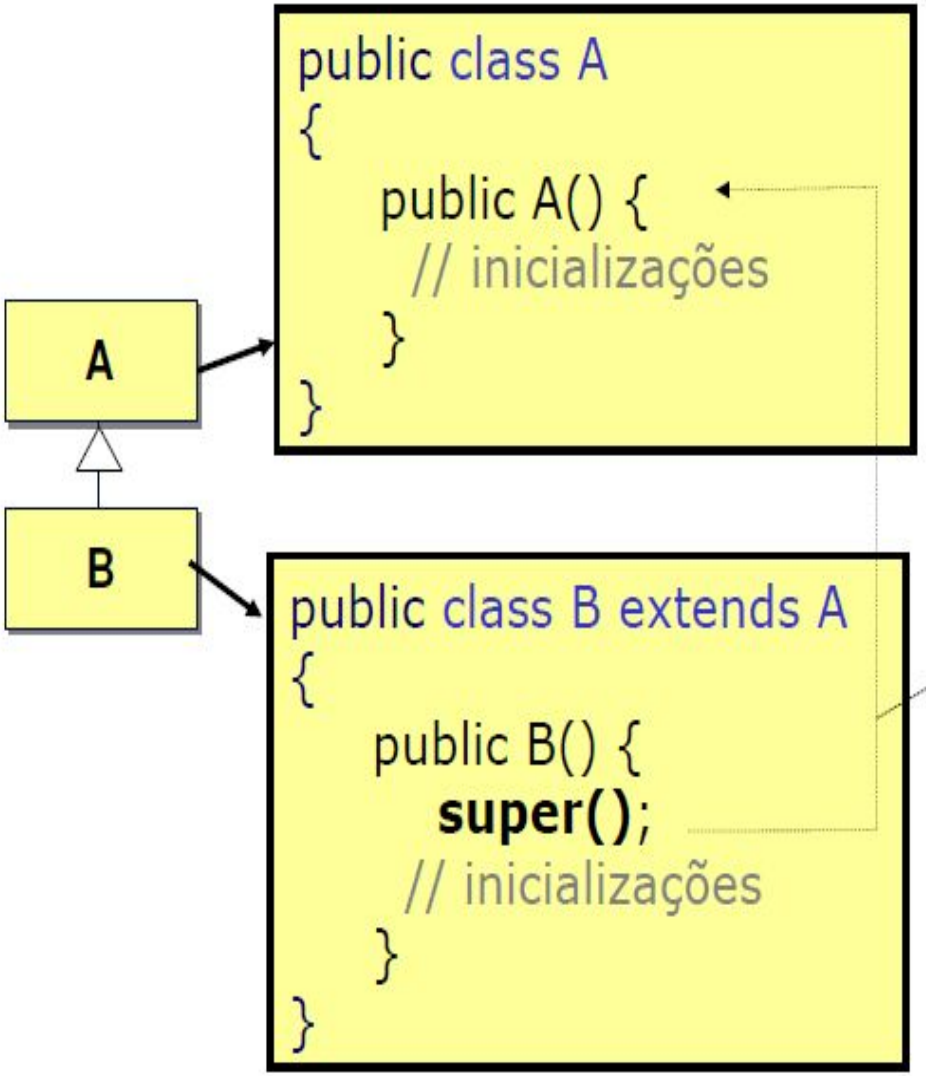
- Listas
  - ArrayList e Array
- Herança

Herança em java

- Herança é um tipo de **relacionamento** entre 2 classes onde:
  - Uma classe herda (compartilha) todas as propriedades e métodos de outra classe.
  - Subclasse herda de superclasse.

- Em Java, a palavra-chave utilizada para criar uma subclasse é **extends**.
  - A sintaxe para uso de **extends** é a seguinte:

```
[Subclasse a ser criada] extends [Superclasse existente]
```



- O construtor da superclasse não é herdado,
- Logo, a subclasse deverá chamá-lo, na **primeira linha** de seu construtor, através da palavra **super**

Herança em java

- O tipo de encapsulamento é muito importante na herança de métodos e atributos.

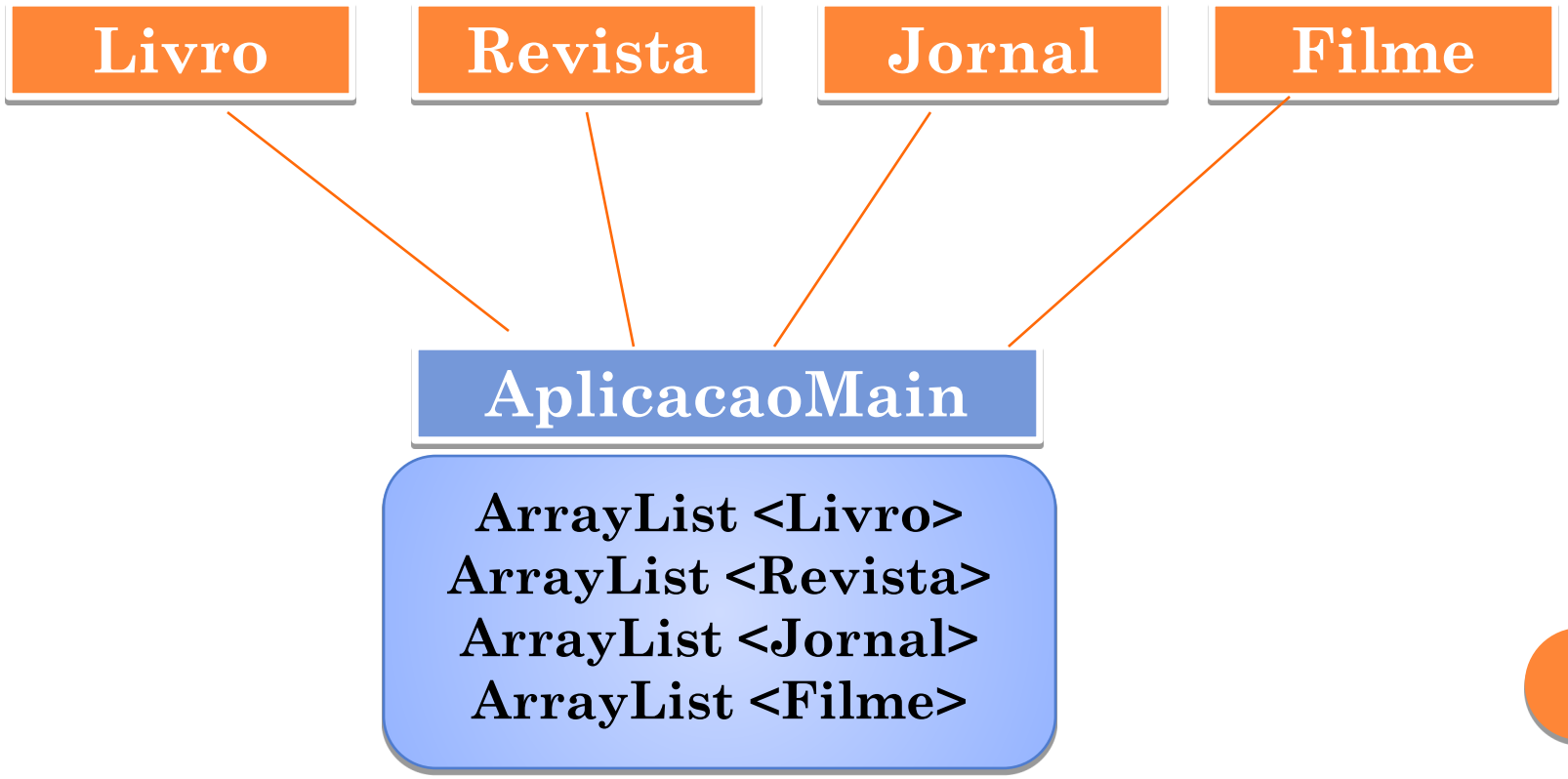
- Como saber que seu problema pode resolvido com herança.
  - Muitos autores recomendam usar o teste do "**É - UM**"

Teste do É-UM

- Você deve perguntar:
  - X **É UM** Y ?
    - Onde X e Y são classes.
    - Se a resposta for **positiva** e fizer sentido a pergunta provavelmente a **herança poderá ser usada**.

Exemplo

- Livraria:



Detalhando As classes

Livro

- Atributos:
  - Nome
  - Preço
  - Código
  - Editora
  - Autor
- Métodos:
  - listar( )
  - listarAutor( )
  - atualizarEditora( )

Detalhando As classes

Revista

- Atributos:
  - Nome
  - Preço
  - Código
  - Editora
  - Edição
  - Temática
- Métodos:
  - listar( )
  - atualizarEditora( )

Detalhando As classes

Jornal

- Atributos:
  - Nome
  - Preço
  - Código
  - Edição
  - Ano
- Métodos:
  - listar( )

Detalhando As classes

Filme

- Atributos:
  - Nome
  - Código
  - Preço
  - Ano
  - Diretor
  - Gênero
- Métodos:
  - listar( )
  - listarDiretor( )
  - atualizarGênero( )

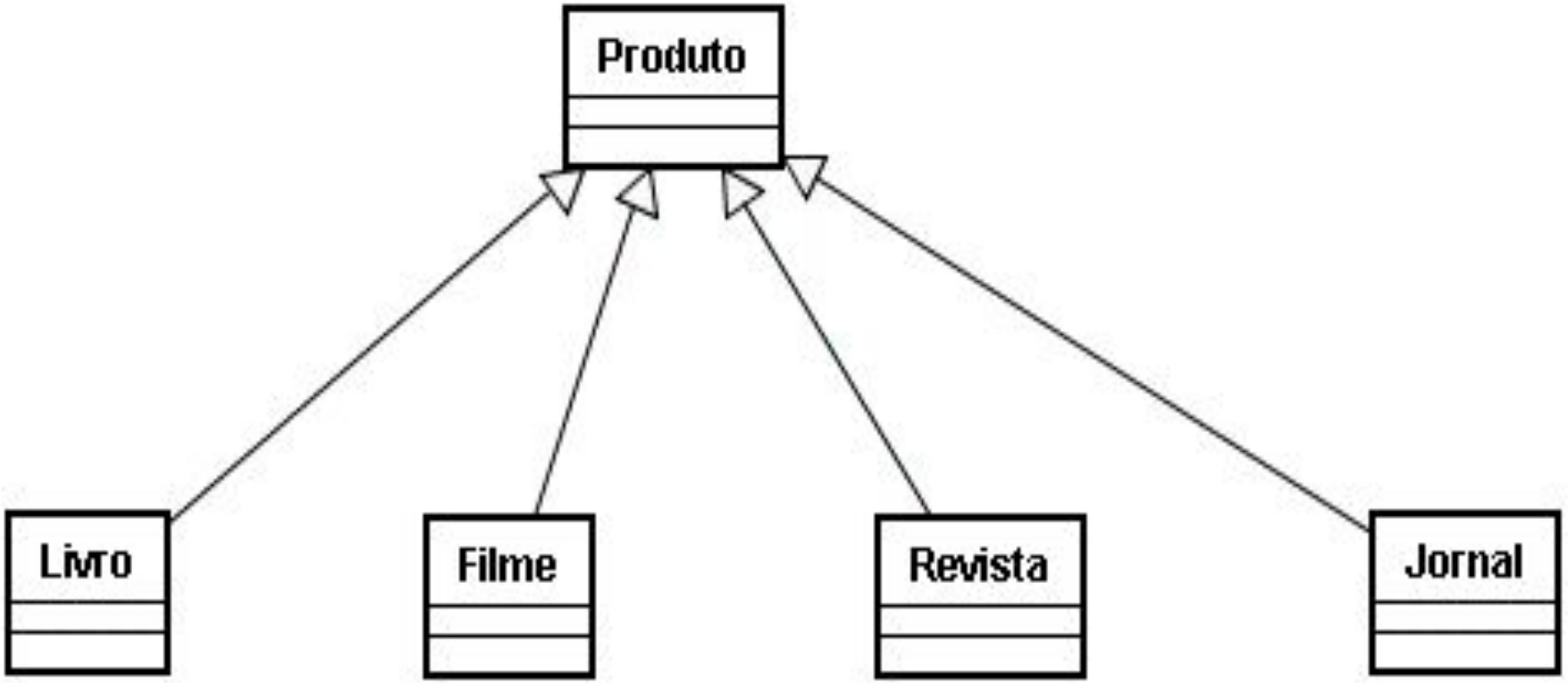
Teste "É UM"

- Teste do É-UM ??
  - Livro É UM Produto?
    - Sim
  - Revista É UM Produto?
    - Sim
  - Jornal É UM Produto?
    - Sim
  - Filme É UM Produto?
    - Sim

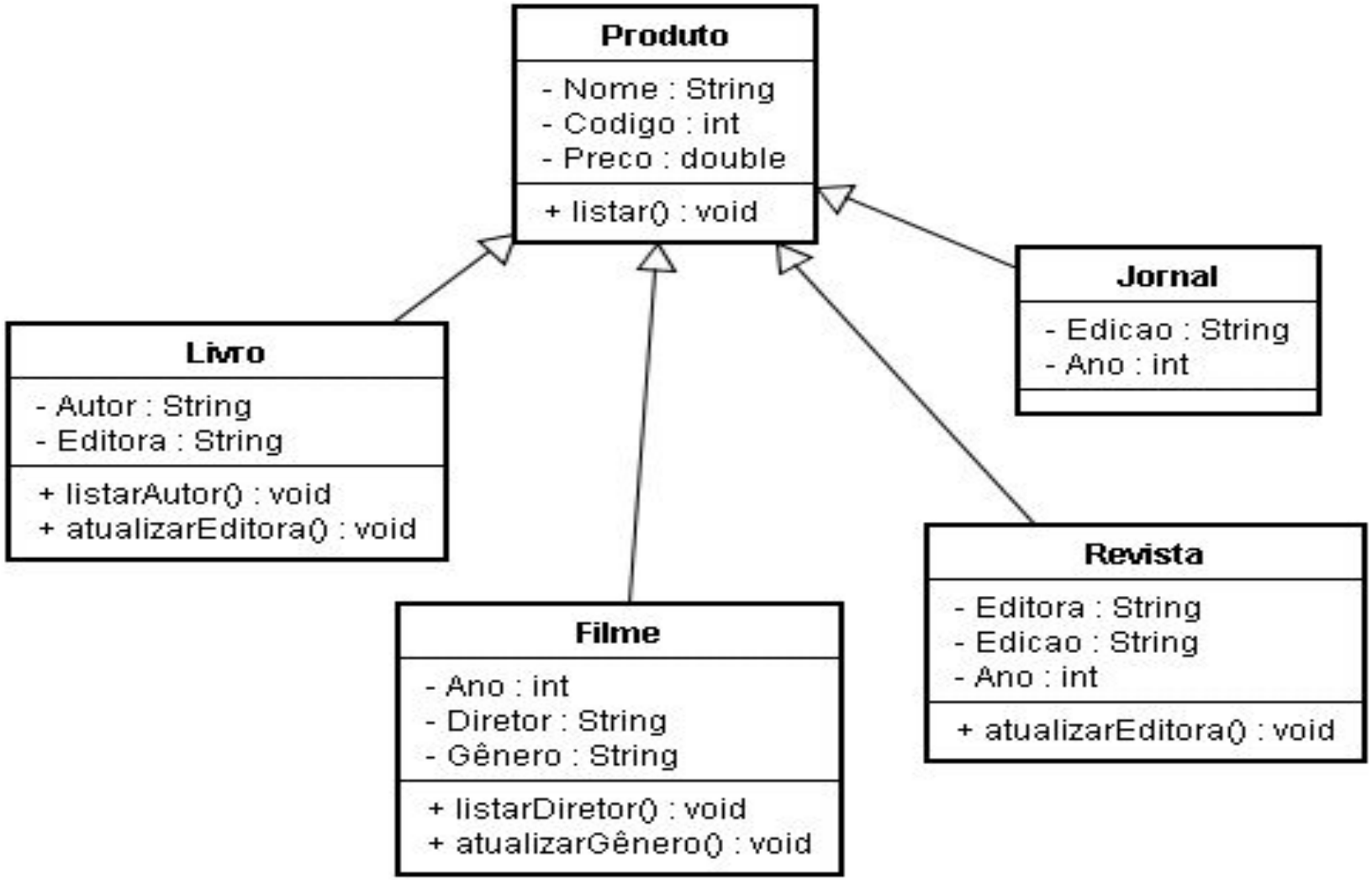
E o quer dizer isso?
Eu posso usar
HERANÇA??

Passou no teste do É UM.
Poderemos usar Herança.
Mas diagrame para ter certeza.

Diagramando o exemplo



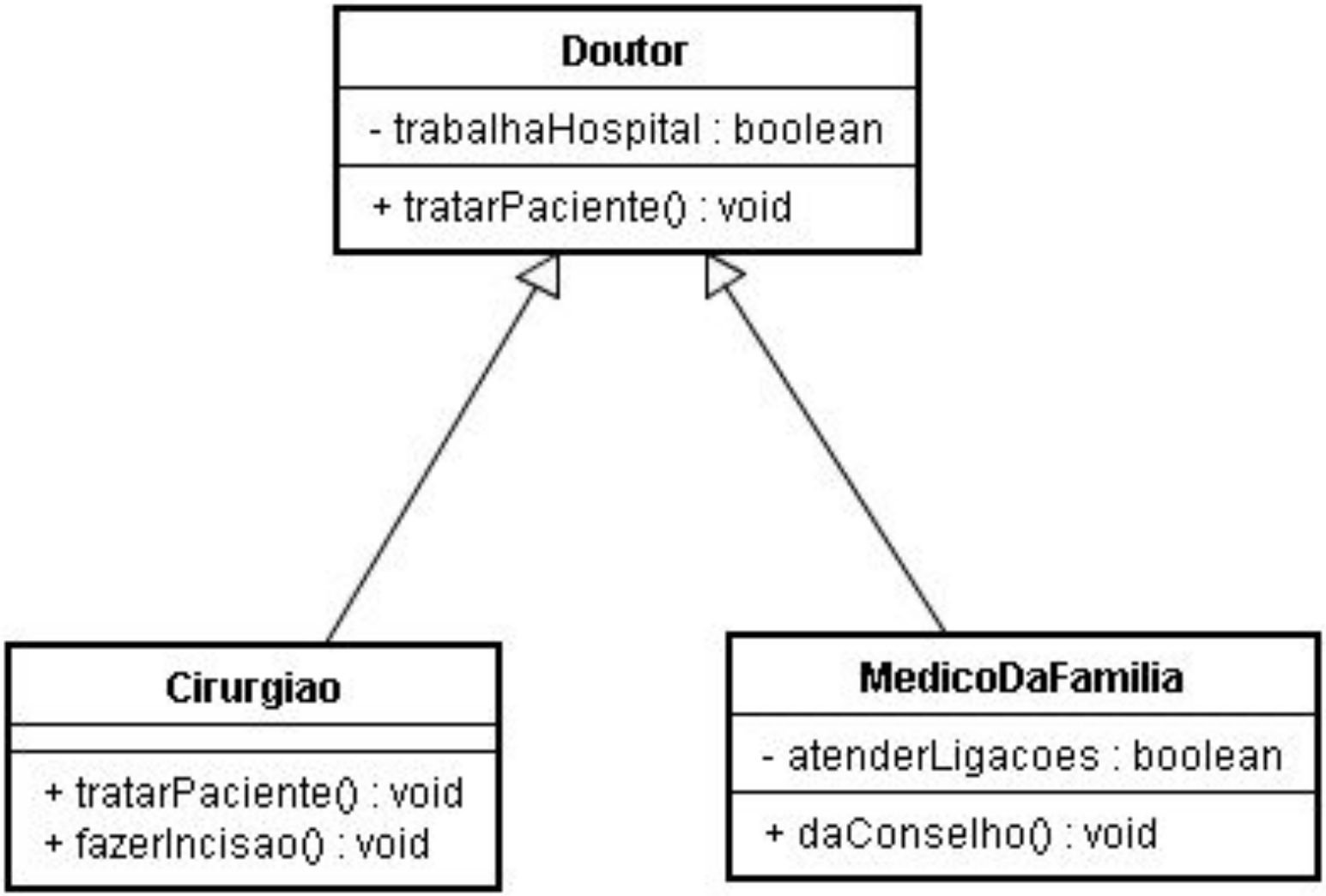
Diagramando o exemplo



Questões

- Superclasse?
- Subclasse?
- E se por acaso o método listar( ) não for o suficiente para classe Livro??
- E se eu precisar usar o método listar( ) da superclasse ? (Considerando que o método foi sobreposto)

Exercício Sala



Exercício Sala

- Quantas variáveis de instância há na classe Cirurgião?
- Quantas variáveis de instância há na classe MedicoDaFamilia?
- Quantos métodos há na classe Doutor?
- Quantos métodos há na classe MedicoDaFamilia?
- Um objeto da classe MedicoDaFamilia pode usar o método tratarPaciente( )?

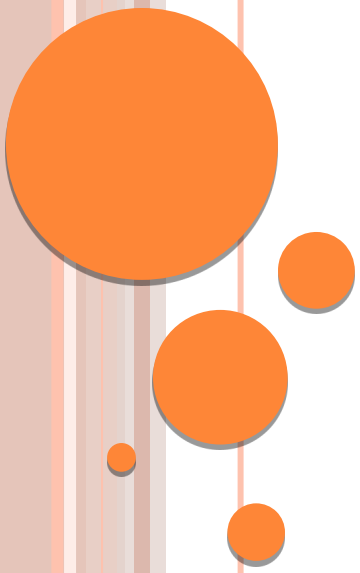
É possível fazer várias heranças em uma mesma classe em Java?

Herança Múltipla em Java

- É o conceito de herança de duas ou mais classes
- Java não tem suporte
- A linguagem Java possui apenas herança simples (uma classe possui no máximo uma classe pai)
  - Mas permite que uma classe implemente várias interfaces. (Veremos na próxima aula)

Exercício

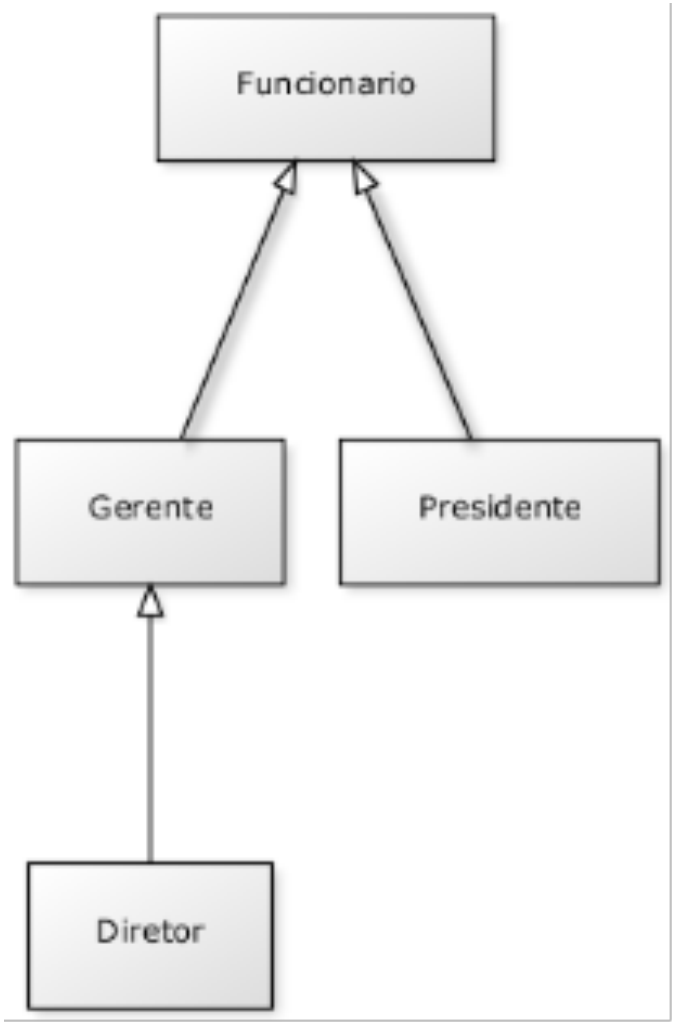
- Considere um mercado onde são comercializados dois tipos de produtos:
  - Produtos sem necessidade especial de conservação ou validade (classe Produto)
  - Produtos que exigem uma determinada temperatura de conservação e validade (classe ProdutoCons)
- Observações:
  - Um produto de interesse para um participante é composto por:
    - identificação
    - nome
    - código do produto que este quer comprar (ou vender)
    - quantidade
    - preço



Classes Abstratas

O que seria isso só pelo nome?

Problema Inicial



- Atributos de Funcionário:
  - nome, cpf e salario;
- Não há bonificação em Funcionário
- A bonificação deve ser dada por meio do método: public double getBonificacao()

Classes abstratas

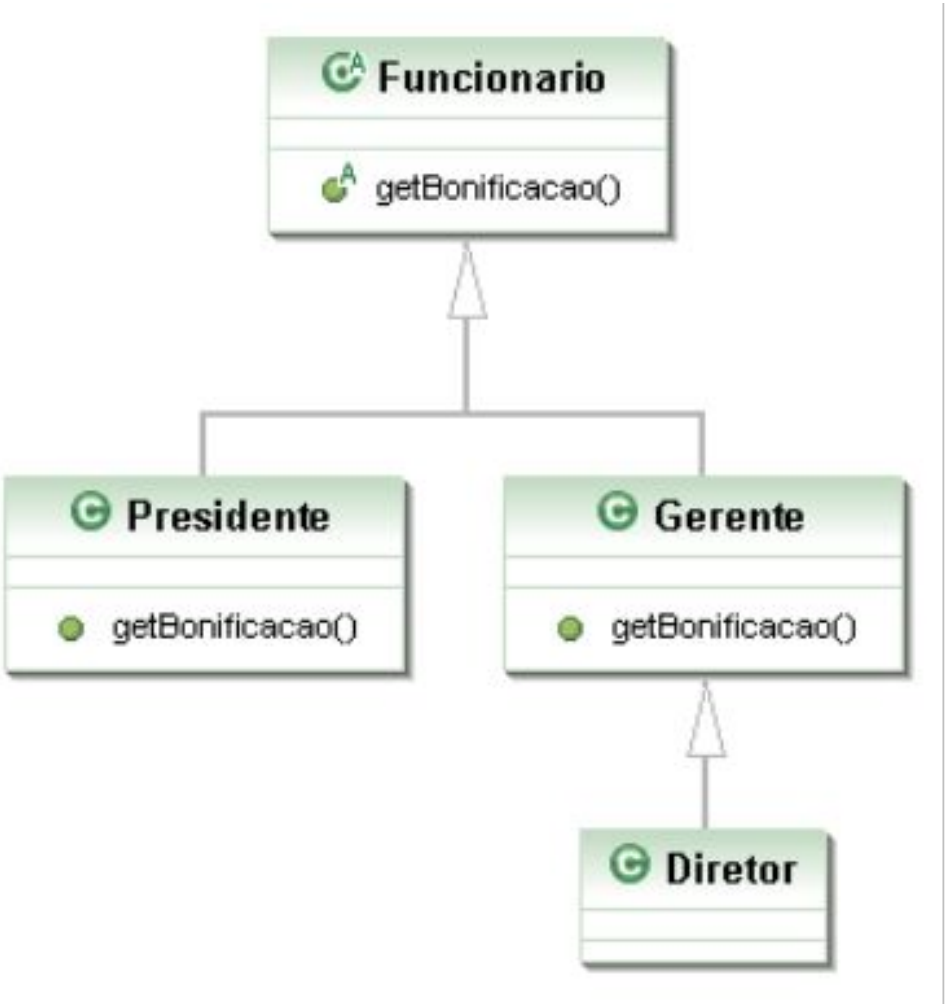
- Relacionam com os princípios de HERANÇA
- Servem como “modelo” para outras classes que dela herdem, não podendo ser instanciada por si só.
- Deve ser formada por pelo menos um método abstrato
- Sintaxe
  - public abstract class ClasseAbstrata{…}

Métodos Abstratos

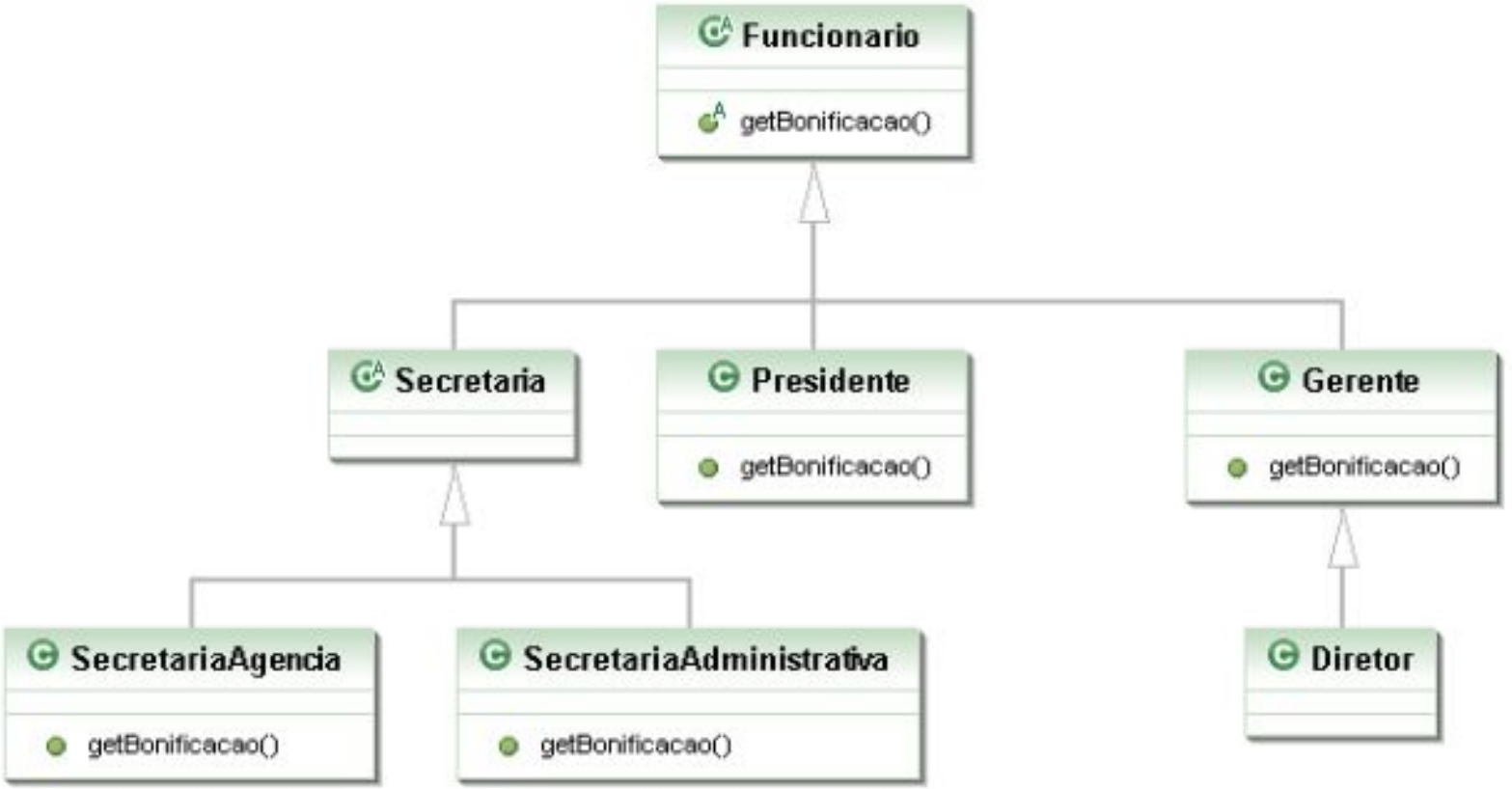
- Assinaturas de métodos
  - Encapsulamento
  - Indicação de abstração
  - Retorno
  - Nome do método
  - Parâmetros
  - ;
- Só existem dentro de classes abstratas
- Sintaxe

```
public abstract void NomeMetodoAbstrato();
```

Vamos voltar para o nosso problema?



Abstrações encadeadas



Conclusões

- Uma classe que estende uma classe normal também pode ser abstrata!
  - Ela não poderá ser instanciada, mas sua classe pai sim!
- Uma classe abstrata não precisa necessariamente ter um método abstrato.

Exercício de Sala 1

1. Escreva uma classe abstrata chamada CartaoWeb.
2. Essa classe representa todos os tipos de cartões web e conterá apenas um atributo: destinatario (tipo String).
3. Declarar o método public abstract void showMessage().
4. Crie classes filhas da classe CartaoWeb: DiaDosNamorados, Natal, Aniversario.
5. Cada uma dessas classes deve conter um método construtor que receba o nome do destinatário do cartão.
6. Cada classe também deve implementar o método showMessage(), mostrando uma mensagem ao usuário com seu nome e que seja específica para a data de comemorativa do cartão.
7. Escreva um programa e no método main crie um array de CartaoWeb. Insira instâncias dos 3 tipos de cartões neste array. Após, use um laço for para exibir as mensagens deste cartão chamando o método showMessage().
8. Teste sua classe

Exercício de Sala 2

- Implemente a hierarquia de classes ContaBancaria (superclasse), ContaCorrente (com senha, número, saldo e quantidade de transações realizadas) e ContaPoupanca (com senha, número, saldo e taxa de rendimento).
- Quando uma ContaBancaria for criada, informe a senha da conta por parâmetro.
- Na classe ContaBancaria, crie os seguintes métodos abstratos: saca(double valor) deposita(double valor) tiraExtrato()
  - nesta mesma classe, crie o método alteraSenha, que recebe uma senha por parâmetro e deve confirmar a senha anterior (via teclado), e somente se a senha anterior estiver correta a senha recebida por parâmetro deve ser atribuída.
- Implemente os métodos abstratos nas classes ContaCorrente e ContaPoupanca.
- Crie os métodos de acesso para os atributos de ContaCorrente e ContaPoupanca.
- Crie uma classe Teste.

Dúvidas?

alanamm.prof@gmail.com