

Programação Orientada a Objetos

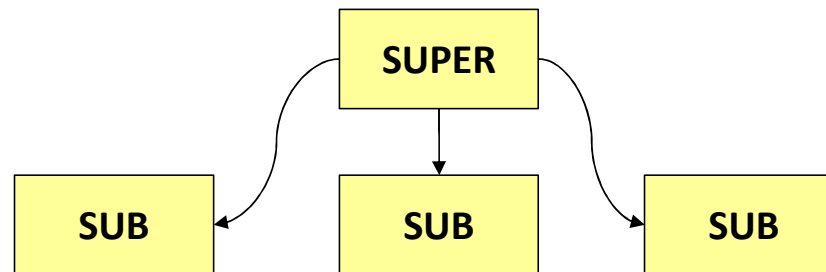
Herança

Prof. Tito Kenzo



Herança

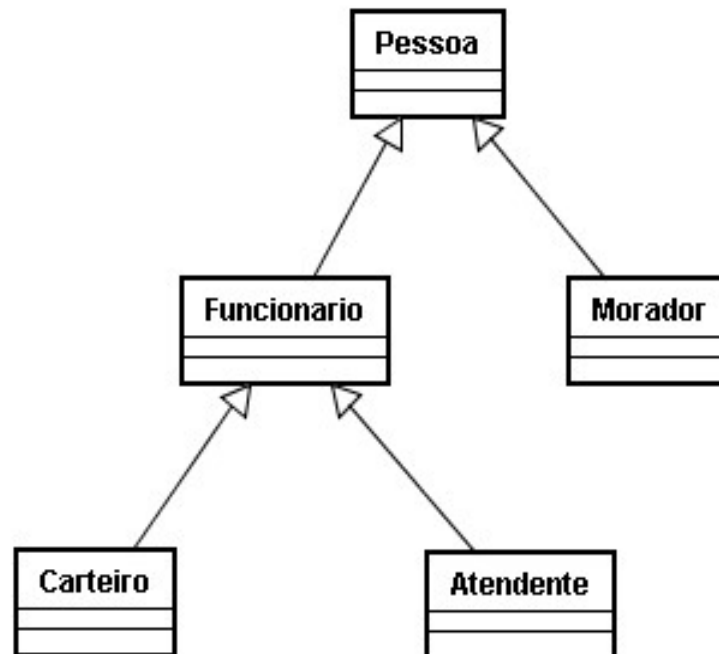
- Definição de uma classe em termos de outras;
 - Permite redefinição de métodos e variáveis;
 - A classe mais genérica é chamada de superclasse e as classes mais específicas de subclasse.
- A subclasse herda todos os atributos e métodos da superclasse.



Vantagens

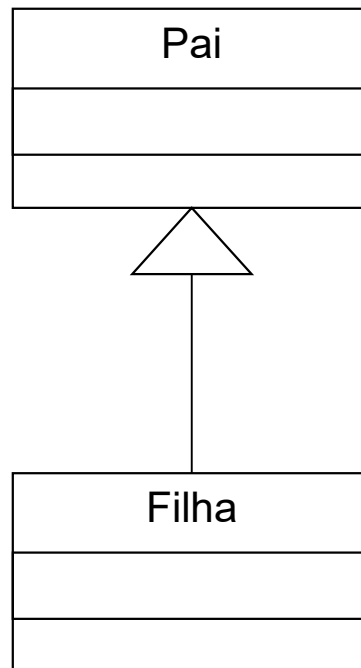
- Compartilhamento de código;
- Reuso de métodos e variáveis das superclasses para as subclasses;
- Simplifica a manutenção do código;
- Apoia a padronização do projeto;
- Codifica menos!

Exemplo



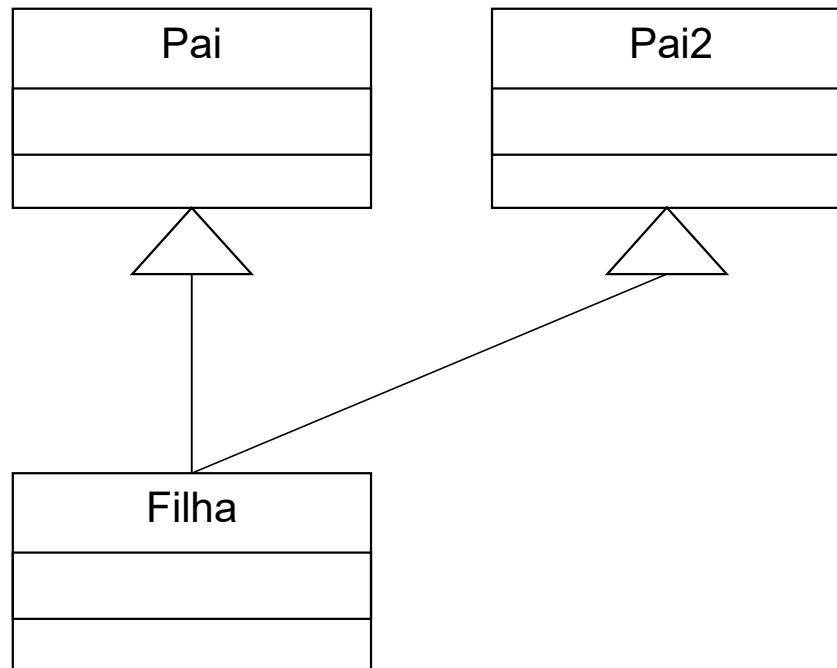
Tipos de Herança

- Herança Simples: uma subclasse herda atributos e métodos diretamente de apenas uma superclasse.



Tipos de Herança

- Herança Múltipla: uma subclasse herda atributos e métodos diretamente de mais de uma superclasse.



Como fazer em Java?

- Java implementa herança simples e não múltipla
 - Java sugere/"faz" a herança múltipla através da implementação de *interfaces*, usando a palavra reservada *implements*;
- A herança é feita através da palavra reservada **extends**;

Ex.:

```
class Pessoa {  
    ...  
}  
  
class Funcionario extends Pessoa {  
    ...  
}
```

Exemplo

```
class Fruta {  
    int gramas;  
    int cals_por_gramas;  
  
    int total_calorias(){  
        return gramas * cals_por_gramas;  
    }  
}
```

```
class Citros extends Fruta{  
    void espremer() { /* ..... */ }  
}
```


Dicas do Tio

- A forma de validar herança é usar a frase “é um”.
- Exemplo:
 - Bicicleta e avião possuem rodas, assento, capacidade de bagagem...

Bicicleta **é um** avião

Avião **é uma** bicicleta

Exercício

- Exercício
 - Definir uma hierarquia de herança para as seguintes classe: Figura, Polígono, Triângulo, Retângulo, Quadrado, Elipse e Círculo.

