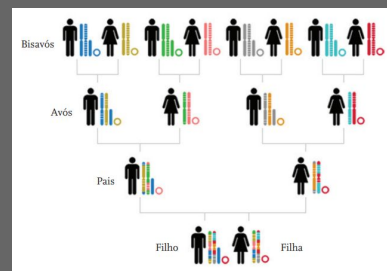


Programação de Sistemas Computacionais

Herança

emerson@paduan.pro.br

Herança



receber dos antepassados

emerson@paduan.pro.br

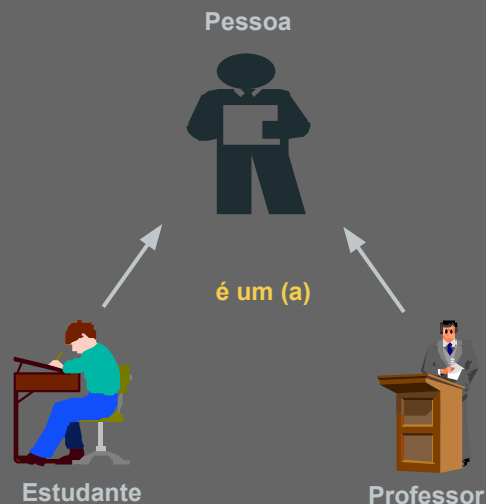
Herança

Herança é um mecanismo que permite que características comuns a diversas classes sejam derivadas de uma classe **base**, ou **superclasse**.

A herança é uma forma de reutilização de software em que novas classes são criadas a partir das classes existentes, herdando seus atributos e comportamentos e **adicionando novos** recursos que as novas classes exigem.

emerson@paduan.pro.br

Herança



emerson@paduan.pro.br

Herança

Em um relacionamento de herança:

- é necessário apenas indicar as **diferenças** entre a **subclasse** e **superclasse**
→ o reuso é automático
- geralmente é necessário redefinir métodos → um dos primeiros motivos para usar herança

emerson@paduan.pro.br

Herança

```
public class SuperClass {  
    //corpo da superclasse...  
}
```

```
public class SubClass extends SuperClass {  
    //corpo da subclasse...  
}
```

extends - indica que a criação de uma nova classe que **herda** de uma classe existente

emerson@paduan.pro.br

super

Palavra-chave **super** refere-se a uma superclasse.

- ✓ indica a chamada ao construtor da superclasse
- ✓ se a superclasse não contiver o construtor padrão e o construtor da subclasse não chamar nenhum outro construtor da superclasse explicitamente, o compilador java vai informar um erro.
- ✓ também é utilizada para invocar métodos da **superclasse** dentro da **subclasse**

emerson@paduan.pro.br

Polimorfismo

Uma das regras fundamentais da herança:

um método definido em uma *subclasse* com o mesmo nome e mesma lista de parâmetros que um método em uma de suas classes antecessoras **oculta** (sobrescreve) o método da classe ancestral a partir da subclasse.

emerson@paduan.pro.br

Polimorfismo

Existem dois tipos de polimorfismo: sobrecarga (overload) e sobreposição (override).

Sobrecarga de métodos (Overload):

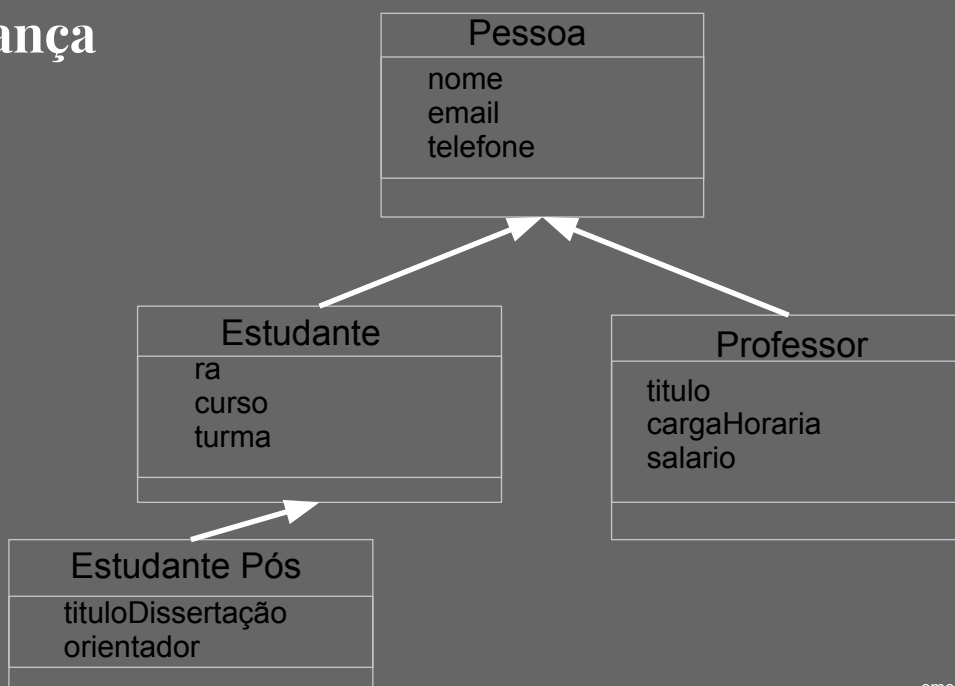
Consiste em criar variações de um mesmo método, ou seja, a criação de dois ou mais métodos com nomes iguais e parâmetros diferentes em uma classe.

Sobreposição de métodos (Override):

Um método definido em uma subclasse com o mesmo nome e mesma lista de parâmetros que um método em uma de suas classes antecessoras **oculta** o método da classe ancestral a partir da subclasse.

emerson@paduan.pro.br

Herança



emerson@paduan.pro.br

Let's Code

Vamos criar classes para testar esses conceitos.



emerson@paduan.pro.br

Exercícios

emerson@paduan.pro.br

Exercício

- Construir uma classe para representar um funcionário com os seguintes dados: nome, horas trabalhadas e valor pago por hora trabalhada. Implementar um método para calcular e retornar o salário final de um funcionário e um método imprimir.
- Criar uma subclasse para representar um funcionário sênior. A diferença entre eles é que um funcionário sênior recebe um bônus a cada 10 horas trabalhadas. Sobrescrever os métodos calcularSalario() e imprimir().
- Criar uma classe AppFuncionario para instanciar objetos da classe Funcionario e Senior e realizar chamada a seus métodos.