



Programação Orientada a Objetos

Paradigmas da Orientação a Objetos Herança

Prof. Ivan Pagnoncelli

# Orientada a Objetos Aula 5 – Tema 1

Herança

Programação





Como o próprio nome já diz, a herança é uma relação de especialização entre duas classes.

Temos, nesse caso, sempre duas classes, a classe base, que será herdada, e a classe derivada, que herda.



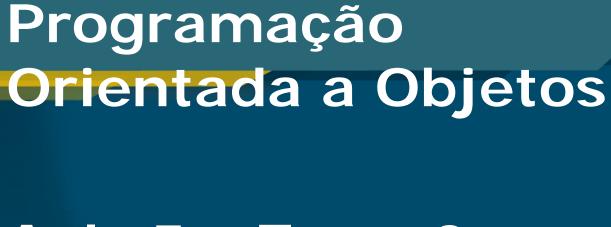
É uma maneira de compartilhamento de atributos e métodos entre os objetos. Por exemplo, temos uma classe chamada *Pessoa*, que terá os atributos nome e idade. Essa classe pode ser herdada para a criação de uma nova classe, chamada Funcionario, que terá um atributo chamado cargo.



Fazendo esta relação, não precisamos criar novamente na classe *Funcionario* os atributos *nome* e *idade*, visto que os mesmo já foram herdados da classe base, que é *Pessoa*.



A POO permite que criemos uma classe derivada de várias classes. Isso é chamado de herança múltipla. As linguagens que implementam POO mais atuais não suportam herança múltipla.





Aula 5 – Tema 2 A Herança no Java



Para criarmos uma classe derivada no Java, utilizamos a palavra reservada *extends* após no nome da classe, como abaixo:

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



No Java, a classe derivada só tem acesso aos atributos da classe base que foram declarados como protected ou public. Todas as classes do Java que fazem parte do JDK, ou seja, que fazem parte da biblioteca padrão do Java, herdam da classe Object.



O Java não permite herança múltipla. Classe abstratas no Java não podem ser instanciadas, apenas herdadas. Suas classes derivadas podem ser instanciadas normalmente.



A implementação de uma interface no Java pode ser considerado herança.

Você pode implementar várias interfaces em uma mesma classe no Java.





Aula 5 – Tema 3 Na Prática



Vamos fazer exemplos práticos, baseados no Java, de Herança.

# Tema 5 - SÍNTESE



Mostramos, nesta aula, o que é a Herança e como fazemos para utilizar este paradigma, utilizando a linguagem Java.