
PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA - ARA0075

Aula 02

PROFESSOR: EDIBERTO MARIANO

programacaoedi@gmail.com



Estácio

PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

TEMA

INTRODUÇÃO À PROGRAMAÇÃO EM OO JAVA



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

OBJETIVOS

Explorar os conceitos de **polimorfismo** e **herança**, apresentando exemplos práticos e de implementação simples, de forma que os alunos tenham uma base sólida desses dois fundamentos da orientação a objetos.



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Herança

O paradigma orientado a objetos possui alguns pilares importantes, um deles é a herança que é um mecanismo utilizado para reaproveitamento de propriedades e métodos de classes existentes.

Ao estender uma classe (**superclasse**), a classe filha (**subclasse**) tem todas as propriedades e métodos da classe base e pode-se adicionar ou modificar métodos que se aplicam apenas a esta classe filha. Então, a classe filha "**herda**" métodos e propriedades da classe "mãe".

Herança, em Java, nada mais é do que criar classes usando outras classes já existentes. Quando e como saber que é hora de usar herança em Java - **Relação 'é um'**.

HERANÇA MÚLTIPLA: uma sub-classe herda características de mais de uma super-classe.

Algumas linguagens orientadas a objetos **não** suportam herança múltipla, por exemplo a linguagem Java.



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Herança

GENERALIZAÇÃO

Determinando uma “**Generalização**”, já que **Tecnico** é um **tipo** de **Empregado**. Isso significa dizer que o **Tecnico** tem, **além de matrícula, nome, uma terceira informação, o título**, ou seja, podemos dizer que a classe **Tecnico herda os atributos da classe Empregado, e define seus próprios atributos.**



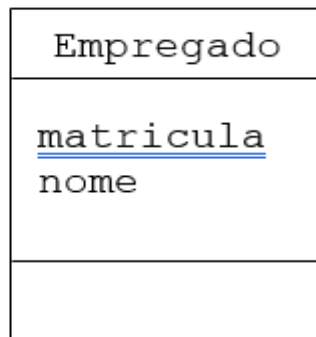
PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Herança

GENERALIZAÇÃO

Superclasse



```
public class Empregado //Classe construtora Empregado
{
    String matricula; //Se private, não poderá ser instanciada por
                    //outra classe

    String nome;}

```

```
public class Tecnico extends Empregado // Classe construtora
// Tecnico

```

Subclasse



```
{
    String titulo;
}

```

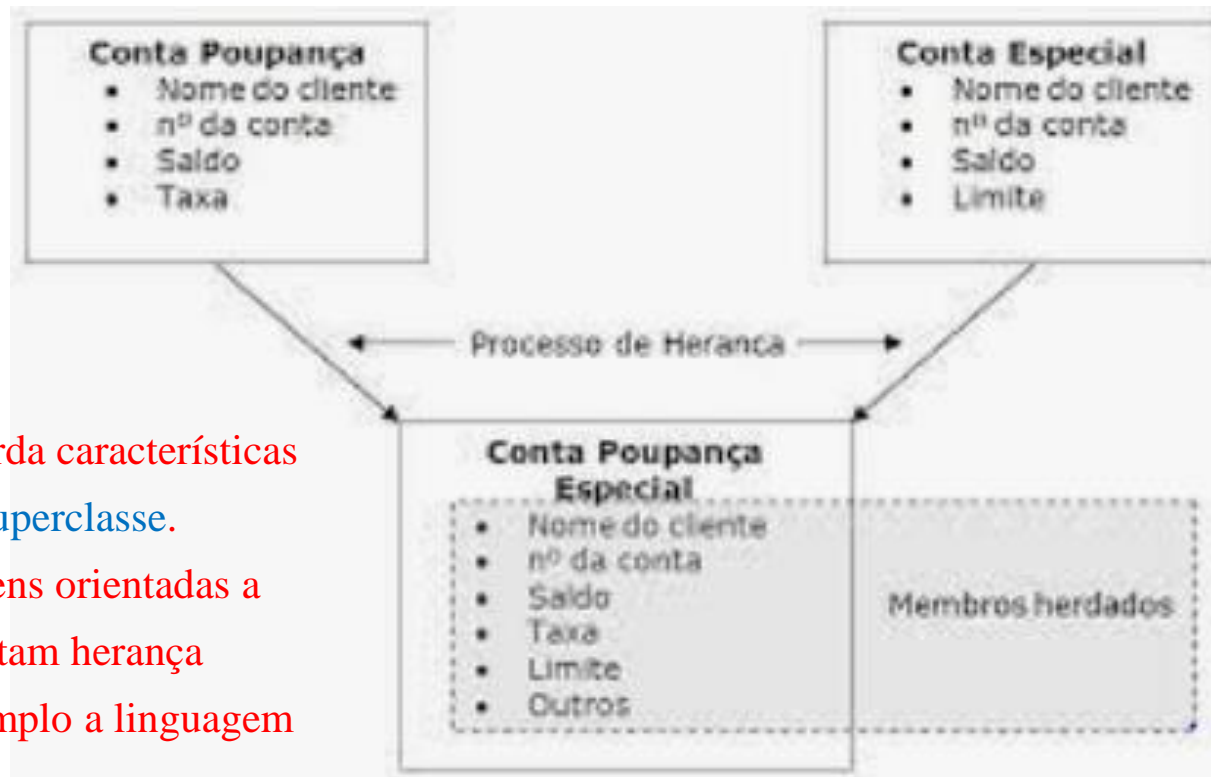


PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Herança - **MÚLTIPLA**

SuperClasse



SubClasse

uma subclasse herda características de mais de uma superclasse.

Algumas linguagens orientadas a objetos não suportam herança múltipla, por exemplo a linguagem Java.

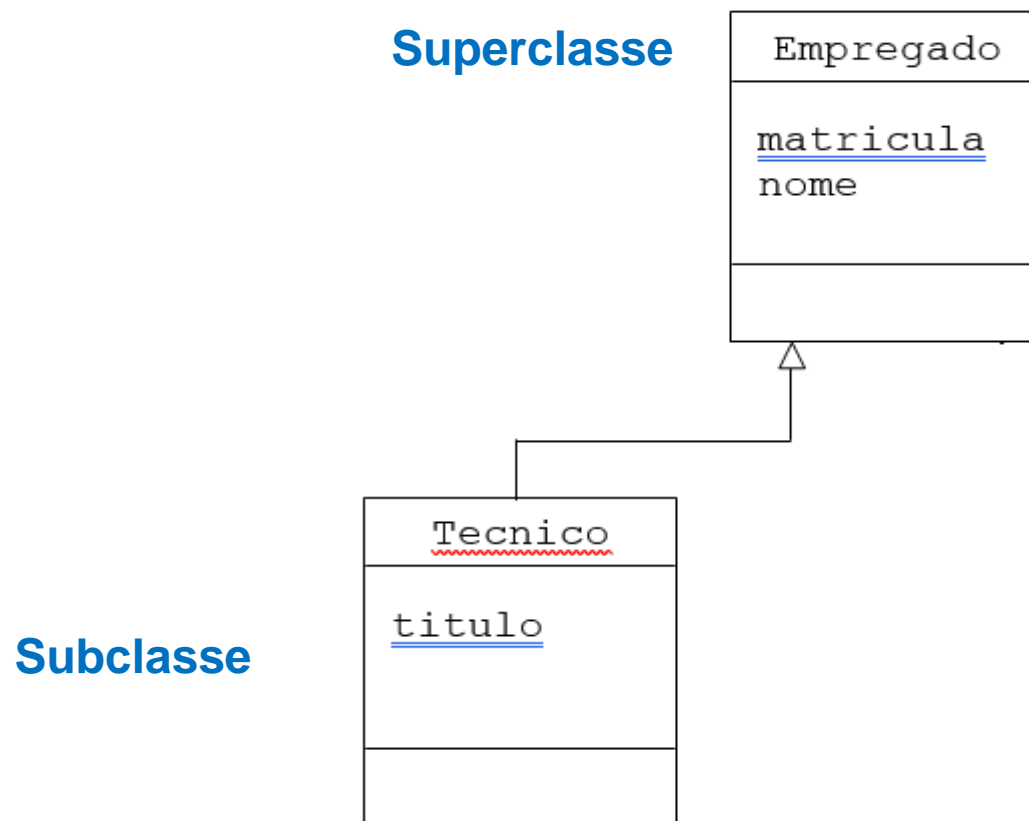


PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Herança – **PRÁTICA**

GENERALIZAÇÃO





PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE **HERANÇA** E POLIMORFISMO: O BÁSICO

```
package Herança00;  
public class Empregado {  
    Integer matricula; //obs se private String nome, NÃO PODERÁ SER INSTANCIADA POR OUTRA CLASSE  
    String nome;      //obs se private String nome, NÃO PODERÁ SER INSTANCIADA POR OUTRA CLASSE  
}
```

```
package Herança00;  
public class Tecnico extends Empregado {  
    String titulo;  
}
```



SAÍDA EM AMBIENTE GRÁFICO

DADOS DO TÉCNICO

ⓘ Digite a matrícula do Técnico:

1172

OK Cancel

Input

? Digite a matrícula do Técnico:

1172

OK Cancel

DADOS DO TÉCNICO

Matricula-----: 1172
Nome-----: Ediberto Silva
Titulo-----: Analista de Sistemas

OK



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Polimorfismo

Reutilização do mesmo nome para um conceito correlato em tipos diferentes. O sistema escolherá qual o método que deve ser utilizado por meio de:

Sobrecarga (Overloading) → comparação feita em **tempo de compilação** que pergunta “Qual método possui os tipos de argumentos correspondentes?” e escolhe o método que será usado. Recurso usado no caso **de métodos de mesmo nome, definidos em uma mesma classe, com argumentos diferentes;**

Sobreposição (Overriding) → comparação feita em **tempo de execução** que pergunta “Qual método foi definido para este tipo de objeto? Existe uma redefinição deste método na classe do objeto ou deve ser usado o método definido na superclasse?”.

Override nada mais é do que a substituição de métodos, variáveis.



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Polimorfismo - Sobrecarga (Overloading)

```
class OverloadTeste {  
    void test() {  
        System.out.println("Nenhum Parâmetro");  
    }  
    //método com um parâmetro tipo inteiro  
    void test(int a) {  
        System.out.println("a: " +a);  
    }  
    //método com dois parâmetros tipo inteiro  
    void test(int a, int b) {  
        System.out.println("a e b: "+a+" "+b);  
    }  
    // método com um parâmetro tipo double  
    double test(double a) {  
        System.out.println("double a: " + a);  
        return a*a;  
    }  
}
```



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Polimorfismo - Sobreposição (Overriding)

```
class Pensamento { //SUPERCLASSE

    void mensagem() {

        System.out.println("mensagem da Superclasse pensamento...");

    }

}

class Conselho extends Pensamento {

    @Override // @Override anotação em Java 5 é opcional, mas útil.

    public void mensagem() {

        System.out.println("Mensagem da Subclasse Conselho ...");

    }

}
```



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E **POLIMORFISMO**: O BÁSICO

Como sugestão, o professor pode apresentar a seguinte situação: Crie uma classe chamada **Forma2D** que possui os atributos **dimensao1** e **dimensao2**; também possui o método **área**. Crie duas classes que herdam as características e atributos da classe **Forma2D**: classe **Triângulo** e **Quadrado**. Os métodos de cálculo de área devem ser adaptados e executados para demonstrar o conceito de polimorfismo.



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

```
Start Page x Polimorfismo01.java x Forma2D.java x Triangulo.java x Quadrado.java x Herança.java x Telefone.java x TelSemFio.java x Herança00.java x
Source History [Icons]
1 package polimorfismo01;
2 import javax.swing.JOptionPane;
3 public class Polimorfismo01 {
4     public static void main(String[] args) {
5         Forma2D f2d = new Forma2D();
6         String dim1 = JOptionPane.showInputDialog("Primeira dimensão :");
7         f2d.dim1 = Float.parseFloat(dim1);
8         String dim2 = JOptionPane.showInputDialog("Segunda dimensão :");
9         f2d.dim2 = Float.parseFloat(dim2);
10        f2d.area(f2d.dim1, f2d.dim2);
11        Triangulo triang = new Triangulo();
12        triang.area(f2d.dim1, f2d.dim2);
13        Quadrado quad = new Quadrado();
14        quad.area(f2d.dim1, f2d.dim2);
15    }
16 }
```



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E **POLIMORFISMO**: O BÁSICO

EXECUÇÃO

Input

Primeira dimensão :

8

OK Cancel

Input

Segunda dimensão :

6

OK Cancel

DADOS DA SUPERCLASSE - Prof. EDIBERTO

Primeira dimensão : 8.0

Segunda dimensão : 6.0

OK

ÁREA DO TRIÂNGULO - Prof. EDIBERTO

Area : 24.0

OK

ÁREA DO QUADRADO - Prof. EDIBERTO

Area : 48.0

OK



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Polimorfismo - Overloading

Em Java é possível definir dois ou mais métodos de dentro da mesma classe que compartilham o mesmo nome, desde que as respectivas declarações de parâmetros são diferentes. Quando for este o caso, os métodos são ditos ser sobrecarregado, e o processo é referido como método de sobrecarga.

Assim, métodos sobrecarregados devem diferir no tipo e / ou o número dos seus parâmetros. Enquanto os métodos sobrecarregados podem ter diferentes tipos de retorno, o tipo de retorno é insuficiente para distinguir duas versões de um método. Quando o Java encontra uma chamada a um método sobrecarregado, ele simplesmente executa a versão do método cujos parâmetros coincidir com os argumentos usados na chamada. Aqui está um exemplo simples que ilustra a sobrecarga de método:



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Polimorfismo - Overloading

```
class Sobrecarga {
    void teste()
    {
        System.out.println("Método sem argumentos");
    }
    void teste(Integer tx1) {
        System.out.println("Valor de a : "+tx1);
    }
    void teste(Double tx1) {
        System.out.println("Valor de b : "+tx1);
    }
    void teste(Integer tx1, Integer tx2) {
        System.out.println("Valor de a + a : "+(tx1+tx1));
    }
    void teste(Double tx1, Double tx2) {
        System.out.println("Valor de b + b : "+(tx1+tx2));
    }
}
```



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Polimorfismo - Overriding

Override nada mais é do que a substituição de métodos, variáveis.

Método de substituição, em programação orientada a objeto, é um recurso de linguagem que permite uma subclasse da classe ou da criança para fornecer uma implementação específica de um método que já é fornecido por uma de suas superclasses ou classes pais.



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Polimorfismo - **Overriding**

```
class Pensamento //SUPERCLASSE
{
    void mensagem()
    {
        System.out.println("mensagem da Superclasse pensamento...");
    }
}

class Conselho extends Pensamento //SUBCLASSE
{
    @ Override // @ Override anotação em Java 5 é opcional, mas útil.
    public void mensagem()
    {
        System.out.println("Mensagem da Subclasse Conselho ...");
    }
}
```



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

- Atividade verificadora de aprendizagem:

Nessa aula, o professor deve dividir a turma em **grupos** e propor para cada um deles os seguintes problemas:

1. Apresente uma situação na qual seria possível e vantajoso a utilização do mecanismo de herança. Faça a sua implementação.
2. Apresente uma situação em que o polimorfismo poderia ser aplicado. Faça a sua implementação.

A atividade deve ser desenvolvida durante a sala de aula e ao final os grupos devem apresentar as situações propostas.



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

Leitura específica

[1] Os conteúdos abordados nessa aula estão disponíveis no livro FURGERI, Sérgio. Java 8 - Ensino Didático: Desenvolvimento e Implementação de Aplicações. Capítulo 7 - Orientação a Objetos. Seções 7.1 e 7.2 Disponível em:
<https://integrada.minhabiblioteca.com.br/#/books/9788536519340/cfi/28!/4/2@100:0.00>

Aprenda+

Assista ao vídeo: - Orientação a Objetos: Herança. Disponível em:
<https://www.youtube.com/watch?v=MOXLCjL4Ik4>



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

- Exercícios

01 - Implemente um programa em Java usando **sobrecarga** de método que:

a – leia via teclado dois valores: um tipo inteiro e outro tipo real.

b – após os números informados, exiba as operações abaixo:

b.1 – valor do primeiro.

b.2 – valor do primeiro, e o valor do segundo.

b.3 – valor do segundo.

b.4 – valor do primeiro vezes três.

b.5 – valor do segundo vezes dois.



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

- Exercícios

02 - Codifique um programa em Java usando **sobrescrita** de métodos que, leia via teclado matrícula e nome de um funcionário com seus quatro dependentes.



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

- Exercícios

03 – Crie um diagrama das classes e desenvolva um sistema em java de acordo com a situação abaixo.

aplicação simples de herança na linguagem Java, como sugestão o professor pode apresentar a seguinte situação:

Uma indústria necessita de um sistema para cadastrar produtos produzidos.

Ela produz **Telefones** cujos atributos são: código, número de série, modelo, peso e dimensão.

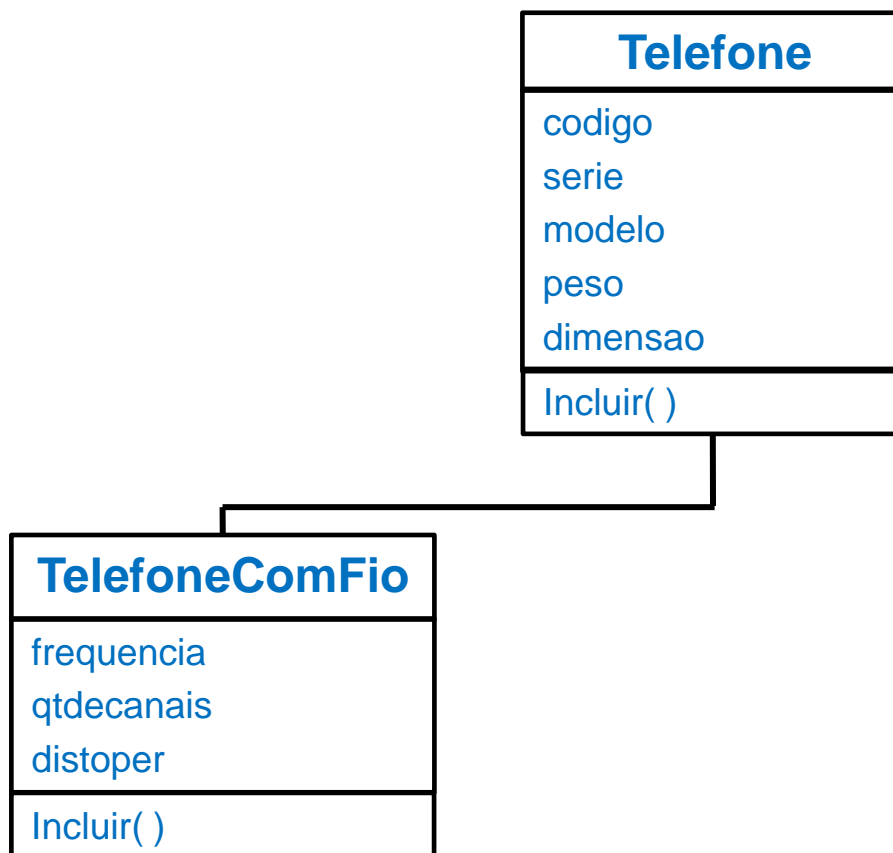
Essa mesma empresa começou a produzir telefones sem fio que compartilham as mesmas características do telefone com fio, porém possuem novas características que são: frequência, quantidade de canais e distância de operação.



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

1.2 IMPLEMENTAÇÃO DE HERANÇA E POLIMORFISMO: O BÁSICO

Herança





PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

- Atividade Autônoma Aura:

1) A empresa de desenvolvimento de software "Atlas Development" foi contratada por uma empresa de seguros para desenvolver um software na Linguagem Java que irá avaliar o perfil de dos contratantes de seguro para automóveis. A Atlas já desenvolveu sistemas parecidos para duas outras empresas. Em um dos módulos do sistema, um desenvolvedor verificou que se herdasse métodos e atributos de duas outras classes, poderia então desenvolver a classe necessária para ser utilizada em parte do módulo em questão. Nesse caso, que tipo de herança está sendo realizada e se Java suportaria esse mecanismo.

- a) Herança por prototipagem, Java implementa
- b) Herança múltipla, Java implementa
- c) Herança simples, Java implementa
- d) Herança múltipla, Java não implementa
- e) Herança por prototipagem, Java não implementa



PROGRAMAÇÃO ORIENTADA A OBJETOS EM JAVA

- Atividade Autônoma Aura:

2) Um programador está desenvolvendo um software que irá controlar aceleradores de motores de forma automática, evitando por exemplo, que o motorista acelere se houver um objeto próximo ao veículo. O software deve ser adaptado para diversos tipos de veículos como automóveis, caminhões e motos. No entanto, sabemos que esses veículos possuem características distintas, mesmo apresentando o comportamento de aceleração do motor. Nesse caso, o programador deverá implementar o método acelerar() para todos os veículos, utilizando o mesmo nome mas com implementações distintas. Que conceito de orientação a objetos o problema acima está se referindo.

- a) Herança
- b) Abstração
- c) Polimorfismo
- d) Encapsulamento
- e) Construtor