Programação Orientada a Objetos

Aulas nº X: Polimorfismo

Prof. Luis Gustavo Araujo

2018

Objetivo

Compreender os conceitos e aplicações de Polimorfismo.

POO

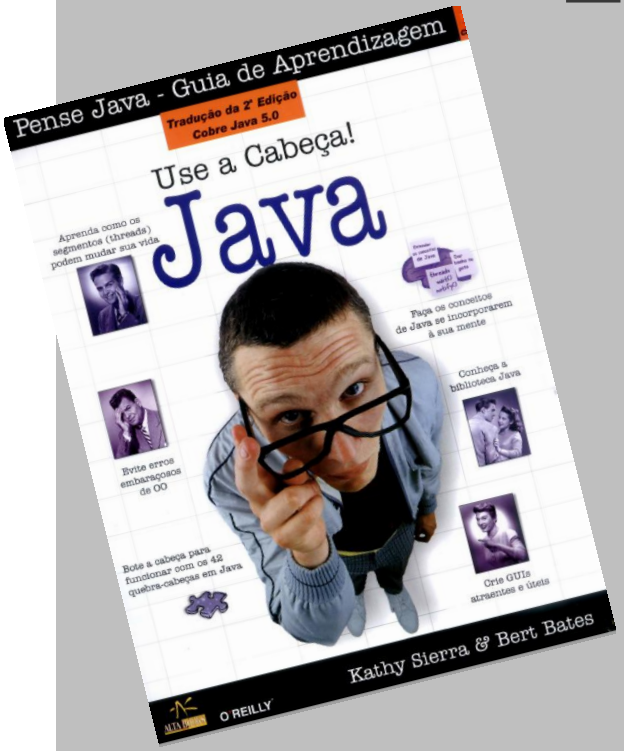
Polimorfismos

Java

Polimorfismo

O conceito de polimorfismo está relacionado intimamente à *Herança*.Ocorre quando a referência é do tipo da superclasse e o objeto é do tipo da subclasse.

Polimorfismo



Quando você declara uma variável de referência, qualquer objeto que passar no teste É-UM quanto ao tipo declarado para ela poderá ser atribuído a essa referência.

Seirra & Bates

Use a Cabeça! Java

Polimorfismo (*teste é um*)

```
1. Funcionario f1 = new Funcionario();
2.
3. Funcionario f2 = new Gerente();
4.
5. Gerente g1 = new Funcionario();
```

Funcionário é Funcionário?

Gerente é Funcionário?

Funcionário é Gerente

Polimorfismo (*teste é um*)

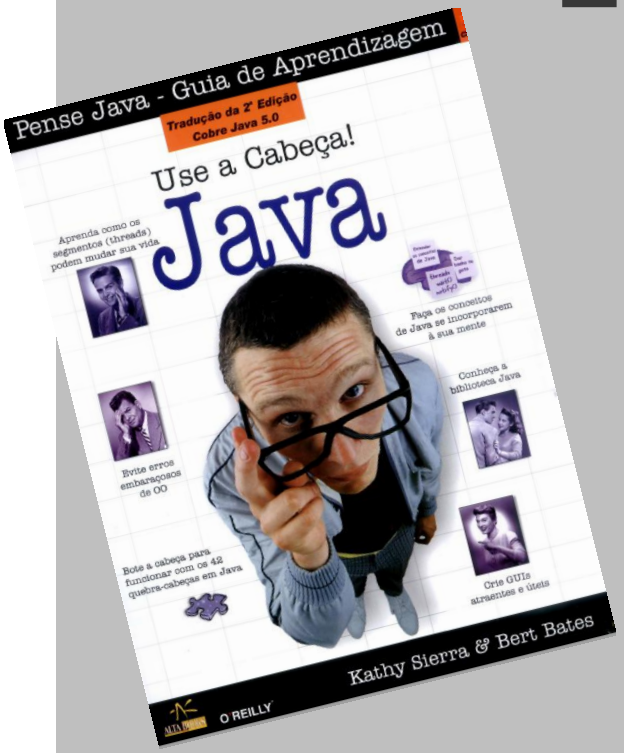
```
1. Funcionario f1 = new Funcionario();
2.
3. Funcionario f2 = new Gerente();
4.
5. Gerente g1 = new Funcionario();
```

Funcionário é Funcionário?

Gerente é Funcionário?

Funcionário é Gerente

Polimorfismo



> *Em outras palavras, qualquer coisa que estender o tipo declarado para a variável de referência poderá ser atribuído a ela.*

Seirra & Bates

Use a Cabeça! Java

Polimorfismo (*estende?*)

```
1. Funcionario f1 = new Funcionario();
2.
3. Funcionario f2 = new Gerente();
4.
5. Gerente g1 = new Funcionario();
```

Funcionário estende Funcionário?

Gerente estende Funcionário?

Funcionário estende Gerente

Polimorfismo (*estende?*)

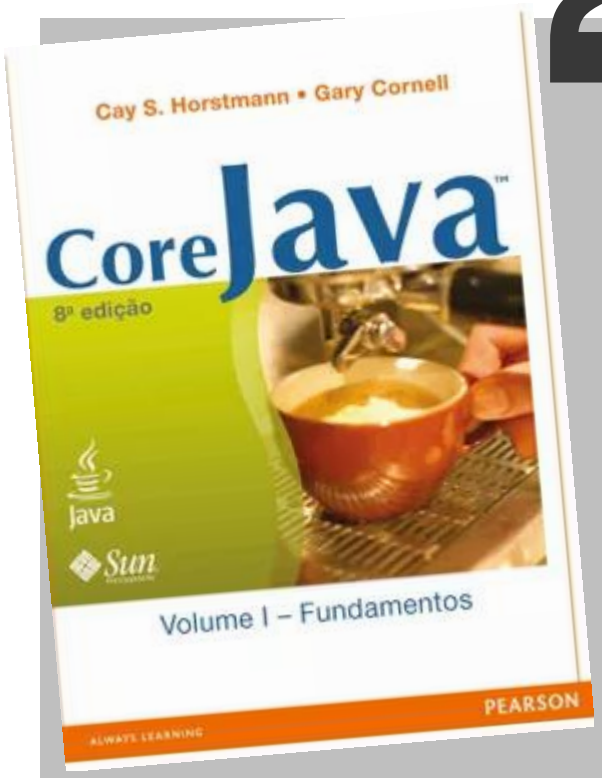
```
1. Funcionario f1 = new Funcionario();
2.
3. Funcionario f2 = new Gerente();
4.
5. Gerente g1 = new Funcionario();
```

Funcionario estende Funcionário?

Gerente estende Funcionário?

Funcionário estende Gerente

Polimorfismo



Esses principio afirma que você pode usar um objeto de subclasse sempre que o programa espera um objeto de superclasse.

Hortmann & Cornell

Core Java

Polimorfismo

```
1. Funcionario f1 =  new  Gerente(“Luis”, 2000, 500);
2. f1.getSalario(); //retorna salario+bonus
3.
4. Funcionario f2 =  new  Funcionario(“Ana”, 1500);
5. f2.getSalario(); //retorna salario
```

Polimorfismo

```
1.  public class  Funcionario {
2.   String nome
3.   double salario;
4.
5.   public Funcionario(String nome, double salario){
6.       this.nome = nome;
7.       this.salario = salario;
8.    }
9.
10.   public double getSalario(){
11.     return this.salario + this.bonus;
12.   }
13. }
```

Polimorfismo

```
1.  public class Gerente extends  Funcionario {
2.   double bonus;
3.
4.   public Gerente(String nome, double salario){
5.      super(nome, salario);
6.      this.bonus = 0;
7.    }
8.    @Overrride
9.    public double getSalario(){
10.     return this.salario + this.bonus;
11.   }
12.
13.   public double getBonus(){
14.       return this.bonus;
15.   }
16. }
```

Polimorfismo

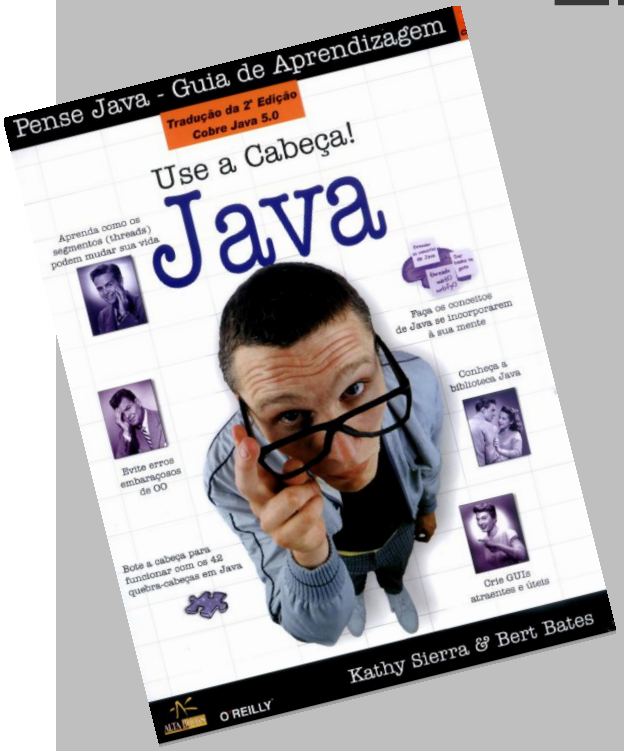
```
1. Funcionario f1 =  new  Gerente(“Luis”, 2000, 500);
2. f1.getSalario(); //retorna salario+bonus
3. f1.getBouns(); //ERRO
4.
5. Funcionario f2 =  new  Funcionario(“Ana”, 1500);
6. f2.getSalario(); //retorna salario
```

Polimorfismo (exemplo)

```
1.  List<Funcionario> arrayFunc =  new  ArrayList<Funcionario>();
2.  arrayFunc.add(new  Gerente("Luis", 2000, 500));
3.  arrayFunc.add(new  Funcionario("Ana", 1500));
4.
5.  for (Funcionario a : arrayFunc) {
6.       System.out.println( a.getNome() );
7.  }
```

```
arrayFunc.add(new  Setor("Teste de Software"); //ERRO Setor não é
    um Funcionario
```

Polimorfismo



> *Por que podemos ter certeza que o polimorfismo funcionará dessa maneira? Porque é sempre seguro supor que qualquer tipo de subclasse terá os métodos que achamos estar chamando no tipo da superclasse?*

Seirra & Bates

Use a Cabeça! Java

Referências Técnicas

Sierra, Kathy, and Bert Bates. **Use a cabeça!: java**. Alta Books, 2007.

HORSTMANN, S.; CORNELL, G.; **Core JAVA**. 1. vol. São Paulo: Pearson Education, 2010.

TURNI, R. **Desbravando Java e Orientação a Objetos**: um guia para o iniciante da linguagem. São Paulo: Casa do Código, 2016.

Programação Orientada a Objetos

Aulas nº X: Polimorfismo

Prof. Luis Gustavo Araujo

2018