

Polimorfismo

Polimorfismo é o princípio pelo qual duas ou mais classes derivadas de uma mesma superclasse podem invocar métodos que têm a mesma identificação (assinatura), mas comportamentos distintos, especializados para cada classe derivada, usando para tanto uma referência a um objeto do tipo da superclasse.

```
public class Produto {  
  
    ...  
  
    public void calcularFrete(int km) {  
        this.frete = km * 1.5;  
    }  
    public void calcularFrete(char regioao) {  
        switch(regiao) {  
            case 'I': System.out.println("Interior");  
                this.frete = 100;  
                break;  
            case 'M': System.out.println("Metropolitana");  
                this.frete = 0;  
                break;  
            case 'L': System.out.println("Litoral");  
                this.frete = 20;  
                break;  
            default: System.out.println("Informe: (I) Interior (M)  
Metropolitana (L) Litoral");  
                break;  
        }  
    }  
}
```

```
public class Principal {  
  
    public static void main(String[] args) {  
  
        Produto prod = new Produto();  
        prod.setDescricao("Lote de livros");  
        prod.calcularFrete(30);  
        System.out.println(prod.getDescricao() + " serão entregues a um  
custo de R$" + prod.getFrete());  
  
        Produto prod2 = new Produto();  

```

```
        prod2.setDescricao("Livros de programação");
        prod2.calcularFrete('L');
        System.out.println(prod2.getDescricao() + " serão entregue a um
custo de R$" + prod2.getFrete());

    }

}
```

Exemplo 1

Produto2.java

```
public class Produto2 {
    private int codigo;
    private double preco;
    private String descricao;
    public int getCodigo() {
        return codigo;
    }
    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
    public double getPreco() {
        return preco;
    }
    public void setPreco(double preco) {
        this.preco = preco;
    }
    public String getDescricao() {
        return descricao;
    }
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
    public double calcularFrete(int km) {
        double valor = km * 1.05;
        return valor;
    }
}
```

Filme2.java

```
public class Filme2 extends Produto2 {  
  
}
```

Livro2.java

```
public class Livro2 extends Produto2{  
  
}
```

Para instanciar as classes Filme2.java e Livro2.java podemos criar os respectivos objetos:

```
Filme2 filme = new Filme2();  
Livro2 livro = new Livro2();
```

ou

```
Produto2 filme = new Filme2();  
Produto2 livro = new Livro2();
```

CarrinhoCompras2.java

```
import java.util.ArrayList;  
import java.util.List;  
  
import javax.swing.JOptionPane;  
  
public class CarrinhoCompras2 {  
  
    List<Produto2> listaProd = new ArrayList<Produto2>();  
  
    public void adicionar(Produto2 prod) {  
        listaProd.add(prod);  
    }  
  
}
```

TesteCompras2.java

```
public class TesteCompras2 {  
    public static void main(String[] args) {  
        Produto2 filme = new Filme2();  
        Produto2 livro = new Livro2();  
  
        CarrinhoCompras2 meuCarrinho = new CarrinhoCompras2();  
        meuCarrinho.adicionar(filme);  
        meuCarrinho.adicionar(livro);  
    }  
}
```

Exemplo 2

Produto3.java

```
import java.text.DecimalFormat;  
import java.text.NumberFormat;  
  
import javax.swing.JOptionPane;  
public class Produto3 {  
    private int codigo;  
    private double preco;  
    private String descricao;  
    public int getCodigo() {  
        return codigo;  
    }  
    public void setCodigo(int codigo) {  
        this.codigo = codigo;  
    }  
    public double getPreco() {  
        return preco;  
    }  
    public void setPreco(double preco) {  
        this.preco = preco;  
    }  
    public String getDescricao() {  
        return descricao;  
    }  
    public void setDescricao(String descricao) {  
        this.descricao = descricao;  
    }  
}
```

```

        public double calcularFrete(int km) {
            double valor = km * 1.05;
            return valor;
        }
        public void inserirDados() {
            this.codigo =
Integer.parseInt(JOptionPane.showInputDialog("Digite código: "));
            this.descricao = JOptionPane.showInputDialog("Digite código: ");
            this.preco =
Double.parseDouble(JOptionPane.showInputDialog("Digite código: "));
        }

        @Override
        public String toString() {
            NumberFormat formatar = DecimalFormat.getCurrencyInstance();
            String texto = "Este produto é um " + this.descricao + " possui o
código " + this.codigo + " e custa " + formatar.format(this.preco);
            return texto;
        }
    }
}

```

CarrinhoCompras3.java

```

import java.util.ArrayList;
import java.util.List;

import javax.swing.JOptionPane;

public class CarrinhoCompras3 {
    List<Produto3> listaProd = new ArrayList<Produto3>();

    public void adicionar(Produto3 prod) {
        int km = Integer.parseInt(JOptionPane.showInputDialog("Quantos
quilometros para entrega?"));
        JOptionPane.showMessageDialog(null, "Frete para o produto " +
prod.getDescricao() + " é " + prod.calcularFrete(km));
        listaProd.add(prod);
    }

    public List<Produto3> gerarLista(){
        return this.listaProd;
    }
}

```

Desafio

Elabore uma classe de testes chamada TesteProduto1 (com um método main()). Solicite para o usuário digitar a descrição do objeto do tipo Produto e a distância de quilômetros para entrega. Então calcule o preço do frete e exiba-o com o nome do produto.

Produto2.java

```
public class Produto2 {
    private int codigo;
    private double preco;
    private String descricao;
    private double frete;

    public int getCodigo() {
        return codigo;
    }
    public void setCodigo(int codigo) {
        this.codigo = codigo;
    }
    public double getPreco() {
        return preco;
    }
    public void setPreco(double preco) {
        this.preco = preco;
    }
    public String getDescricao() {
        return descricao;
    }
    public void setDescricao(String descricao) {
        this.descricao = descricao;
    }
    public void calcularFrete(int km) {
        this.frete = km * 1.5;
    }
    public double getFrete() {
        return frete;
    }
    public void setFrete(double frete) {
        this.frete = frete;
    }
}
```

TesteProduto1.java

```
import javax.swing.JOptionPane;

public class TesteProduto1 {
    public static void main(String[] args) {
        Produto2 prod = new Produto2();
        prod.setDescricao(JOptionPane.showInputDialog("Digite o nome do produto:"));
        int km = Integer.parseInt(JOptionPane.showInputDialog("Digite quantos quilômetros de distância:"));
        prod.calcularFrete(km);
        System.out.println(prod.getDescricao() + " serão entregues a um custo de R$" +
                           prod.getFrete());
    }
}
```