

## Atividade Adapter

Utilizando o conteúdo apresentado sobre o padrão Adapter no seminário, crie as classes necessárias para o funcionamento do código.

### 1. Adapter para Conversão de Temperatura

// Interface desejada

```
interface ConversorTemperatura {  
    double celsiusParaFahrenheit(double celsius);  
    double fahrenheitParaCelsius(double fahrenheit);  
}
```

// Classe existente, mas que só converte Celsius para Fahrenheit

```
class ConversorCelsius {  
    public double converterParaFahrenheit(double celsius) {  
        return (celsius * 9/5) + 32;  
    }  
}
```

//RESPOSTA

```
class AdaptadorTemperatura implements ConversorTemperatura {  
    private ConversorCelsius conversorCelsius;
```

```
    public AdaptadorTemperatura() {  
        this.conversorCelsius = new ConversorCelsius();  
    }
```

@Override

```
    public double celsiusParaFahrenheit(double celsius) {  
        return conversorCelsius.converterParaFahrenheit(celsius);  
    }
```

@Override

```
    public double fahrenheitParaCelsius(double fahrenheit) {
```

```
// Implementação inversa
return (fahrenheit - 32) * 5/9;
}
}
```

## 2. Adapter para Sistema de Pagamento

```
// interface alvo
interface ProcessadorPagamento {
    void processarPagamento(double valor);
}

// Sistema antigo
class SistemaPagamentoLegado {
    public void efetuarPagamento(String valor) {
        System.out.println("Pagamento legado processado: R$" + valor);
    }
}

// Teste de funcionamento
public class TestePagamento {
    public static void main(String[] args) {
        System.out.println("=== Teste do sistema ===");

        // Usando o adapter
        ProcessadorPagamento processador = new AdaptadorPagamento();

        // Testes
        System.out.println("\n1. Pagamento de R$ 100,00:");
        processador.processarPagamento(100.00);

        System.out.println("\n2. Pagamento de R$ 55,75:");
        processador.processarPagamento(55.75);

        System.out.println("\n3. Pagamento de R$ 1234,56:");
        processador.processarPagamento(1234.56);
    }
}
```

```
// Teste com o sistema legado
```

```
SistemaPagamentoLegado legado = new SistemaPagamentoLegado();  
System.out.println("Sistema legado direto:");  
legado.efetuarPagamento("100,00");
```

```
}
```

```
}
```

```
//Resposta
```

```
class AdaptadorPagamento implements ProcessadorPagamento {  
    private SistemaPagamentoLegado sistemaLegado;
```

```
    public AdaptadorPagamento() {  
        this.sistemaLegado = new SistemaPagamentoLegado();  
    }
```

```
@Override
```

```
    public void processarPagamento(double valor) {  
        String valorString = String.format("%.2f", valor);  
        sistemaLegado.efetuarPagamento(valorString);
```

```
    }
```

```
}
```