**Exercício padrão de projeto Decorator em TypeScript**

// Interface para representar um sanduíche

interface Sanduiche {

getDescricao(): string;

getCusto(): number;

}

// Implementação básica de um sanduíche

class SanduicheBase implements Sanduiche {

getDescricao(): string {

return 'Sanduíche';

}

getCusto(): number {

return 0;

}

}

// Decorador abstrato

abstract class Decorador implements Sanduiche {

protected sanduicheDecorado: Sanduiche;

constructor(sanduiche: Sanduiche) {

this.sanduicheDecorado = sanduiche;

}

abstract getDescricao(): string;

abstract getCusto(): number;

}

// Implementação concreta de um sanduíche de frango assado

class FrangoAssado extends SanduicheBase {

getDescricao(): string {

return 'Frango Assado';

}

getCusto(): number {

return 4.5;

}

}

// Implementação concreta do decorador Pepperoni

class Pepperoni extends Decorador {

constructor(sanduiche: Sanduiche) {

super(sanduiche);

}

getDescricao(): string {

return this.sanduicheDecorado.getDescricao() + ', Pepperoni';

}

getCusto(): number {

return this.sanduicheDecorado.getCusto() + 0.99;

}

}

// Implementação concreta do decorador QueijoMussarelaRalado

class QueijoMussarelaRalado extends Decorador {

constructor(sanduiche: Sanduiche) {

super(sanduiche);

}

getDescricao(): string {

return this.sanduicheDecorado.getDescricao() + ', Queijo Mussarela Ralado';

}

getCusto(): number {

return this.sanduicheDecorado.getCusto() + 2.0;

}

}

// Exemplo de uso

let sanduiche: Sanduiche = new FrangoAssado();

sanduiche = new Pepperoni(sanduiche);

sanduiche = new QueijoMussarelaRalado(sanduiche);

console.log(`Sanduíche ${sanduiche.getDescricao()} custa R$${sanduiche.getCusto().toFixed(2)}`);