

TEMA PROJETO: Cadeia de Restaurantes (Bob's ou Subway)

Discente: João Batista Araújo de Lima Filho.

```
public class Pedido {
```

```
// I - Atributos de no minimo três tipos distintos.
```

```
String cliente;
```

```
Double valorPedido;
```

```
int numItensPedidos;
```

```
boolean fazerPedido;
```

“Esses são alguns atributos para um pedido de uma rede de restaurantes Bob's ou Subway. Como o nome do cliente, valor do pedido, número de itens do pedido e fazer pedido.”

```
// II - Defina pelo menos DOIS métodos SEM RETORNO que atualizem os  
//valores dos atributos de forma ENCAPSULADA.
```

```
void numeroltensPedidos() {  
    numItensPedidos++;
```

```
}
```

```
void fazerPedido() {  
    valorPedido = 0d;  
    fazerPedido = true;  
}
```

“Esses são dois métodos sem retorno e que atualizam os valores dos atributos de forma encapsulada. O método numeroltensPedidos adiciona um item a cada a cada vez que um produto é adicionado ao pedido. O método fazerPedido inicia o valor do pedido com 0 e seta o atributo fazerPedido com true.”

```
// III - Defina UM método COM RETORNO mas SEM PARÂMETROS.
```

```
double addRefri() {  
    numeroltensPedidos();  
    return valorPedido += 3.50;  
}
```

“Esse método adiciona um refrigerante ao pedido, faz a chamada do método numeroltensPedidos que adiciona um item ao pedido, e retorna o atributo valorPedido atualizado com a adição do valor do refrigerante.”

```
// IV - Defina UM método COM RETORNO e COM PARÂMETROS.
```

```
String nomeCliente(String nome) {  
  
    return cliente = nome;  
}
```

“Esse método recebe o nome do cliente como parâmetro, e retorna o nome no atributo cliente.”

```
// V - Defina UM método SEM RETORNO e COM PARÂMETROS.
```

```
void addLanche(String lanche) {
```

```

switch (lanche) {

    case "Hamburguer":
        valorPedido += 9.99;
        numeroltemsPedidos();
        break;
    case "Sundae":
        valorPedido += 7.99;
        numeroltemsPedidos();
        break;
    case "Batata":
        valorPedido += 3.99;
        numeroltemsPedidos();
        break;
    case "Casquinha":
        valorPedido += 4.49;
        numeroltemsPedidos();
        break;
    default:
        System.out.println("Lanche em falta!");
}
}

```

“Nesse método é feito a adição dos lanches ao pedido, passando o nome do lanche como parâmetro e a partir do Switch Case, verifica-se qual é o lanche e fazer a atualização do valorPedido adicionando o valor do lanche e chama o método numeroltemsPedidos para atualizar o número de itens no pedido. Caso tente adicionar um lanche que não está nas condições, será apresentado a seguinte mensagem “Lanche em falta!”.

// VIII - Para gerar uma String com todos os valores dos atributos da classe, defina o método //toString().

```

@Override
public String toString() {
    return
        "Nome cliente: " + cliente +
        ", Valor do pedido: R$" + valorPedido +
        ", Número de itens do pedido: " + numltemsPedidos +
        ", Fazer pedido: " + fazerPedido ;
}

```

“Método toString para mostrar os valores dos atributos da classe Pedido.”

```

public static void main (String[]args){

```

// VI - Crie pelo menos QUATRO instâncias da sua classe dentro do método //main().

“Foram criadas quatro instâncias da classe Pedido. Que foram elas pedido1, pedido2, pedido3 e pedido4.”

```

System.out.println("      PEDIDOS CADEIA DE RESTAURANTES BOB'S OU SUBWAY");

System.out.println("\n***** PEDIDO 1 *****\n");

```

```

Pedido pedido1 = new Pedido(); // VII - Faça pelo menos SEIS chamadas dos seus métodos.
System.out.println(pedido1.toString()); // VIII - Exiba os valores dos seus objetos antes e depois da
chamada de cada método.
pedido1.fazerPedido();

```

```

System.out.println(pedido1.toString());
pedido1.nomeCliente("Maria");
System.out.println(pedido1.toString());
pedido1.addLanche("Sundae");
System.out.println(pedido1.toString());
pedido1.addLanche("Hamburguer");
System.out.println(pedido1.toString());
pedido1.addRefri();
System.out.println(pedido1.toString());
pedido1.addRefri();
System.out.println(pedido1.toString());
pedido1.addLanche("Fas");
System.out.println(pedido1.toString());

System.out.println("\n***** PEDIDO 2 *****\n");

Pedido pedido2 = new Pedido(); // VII - Faça pelo menos SEIS chamadas dos seus métodos.
System.out.println(pedido2.toString()); // VIII - Exiba os valores dos seus objetos antes e depois da
chamada de cada método.
pedido2.fazerPedido();
System.out.println(pedido2.toString());
pedido2.nomeCliente("João");
System.out.println(pedido2.toString());
pedido2.addLanche("Hamburguer");
System.out.println(pedido2.toString());
pedido2.addLanche("Batata");
System.out.println(pedido2.toString());
pedido2.addLanche("Casquinha");
System.out.println(pedido2.toString());
pedido2.addRefri();
System.out.println(pedido2.toString());
pedido2.addRefri();
System.out.println(pedido2.toString());

System.out.println("\n***** PEDIDO 3 *****\n");

Pedido pedido3 = new Pedido(); // VII - Faça pelo menos SEIS chamadas dos seus métodos.
System.out.println(pedido3.toString()); // VIII - Exiba os valores dos seus objetos antes e depois da
chamada de cada método.
pedido3.fazerPedido();
System.out.println(pedido3.toString());
pedido3.nomeCliente("Karla");
System.out.println(pedido3.toString());
pedido3.addLanche("Hamburguer");
System.out.println(pedido3.toString());
pedido3.addLanche("Hamburguer");
System.out.println(pedido3.toString());
pedido3.addLanche("Casquinha");
System.out.println(pedido3.toString());
pedido3.addLanche("Batata");
System.out.println(pedido3.toString());
pedido3.addRefri();
System.out.println(pedido3.toString());
pedido3.addRefri();
System.out.println(pedido3.toString());

System.out.println("\n***** PEDIDO 4 *****\n");

Pedido pedido4 = new Pedido(); // VII - Faça pelo menos SEIS chamadas dos seus métodos.
System.out.println(pedido4.toString()); // VIII - Exiba os valores dos seus objetos antes e depois da
chamada de cada método.

```

```

pedido4.fazerPedido();
System.out.println(pedido4.toString());
pedido4.nomeCliente("Lucas");
System.out.println(pedido4.toString());
pedido4.addLanche("Hamburguer");
System.out.println(pedido4.toString());
pedido4.addLanche("Sundae");
System.out.println(pedido4.toString());
pedido4.addLanche("Casquinha");
System.out.println(pedido4.toString());
pedido4.addLanche("Casquinha");
System.out.println(pedido4.toString());
pedido4.addRefri();
System.out.println(pedido4.toString());
}
}

```

“Aqui em cada novo pedido foram feitas pelo menos 6 chamadas dos métodos da classe Pedido. E também em cada Pedido antes e depois de cada chamada de método foi exibido os valores dos objetos usando o método toString.”

IMAGENS DA SAÍDA NO TERMINAL

```

PEDIDOS CADEIA DE RESTAURANTES BOB'S OU SUBWAY

***** PEDIDO 1 *****

Nome cliente: null, Valor do pedido: R$null, Número de itens do pedido: 0, Fazer pedido: false
Nome cliente: null, Valor do pedido: R$0.0, Número de itens do pedido: 0, Fazer pedido: true
Nome cliente: Maria, Valor do pedido: R$0.0, Número de itens do pedido: 0, Fazer pedido: true
Nome cliente: Maria, Valor do pedido: R$7.99, Número de itens do pedido: 1, Fazer pedido: true
Nome cliente: Maria, Valor do pedido: R$17.98, Número de itens do pedido: 2, Fazer pedido: true
Nome cliente: Maria, Valor do pedido: R$21.48, Número de itens do pedido: 3, Fazer pedido: true
Nome cliente: Maria, Valor do pedido: R$24.98, Número de itens do pedido: 4, Fazer pedido: true
Lanche em falta!
Nome cliente: Maria, Valor do pedido: R$24.98, Número de itens do pedido: 4, Fazer pedido: true

***** PEDIDO 2 *****

Nome cliente: null, Valor do pedido: R$null, Número de itens do pedido: 0, Fazer pedido: false
Nome cliente: null, Valor do pedido: R$0.0, Número de itens do pedido: 0, Fazer pedido: true
Nome cliente: João, Valor do pedido: R$0.0, Número de itens do pedido: 0, Fazer pedido: true
Nome cliente: João, Valor do pedido: R$9.99, Número de itens do pedido: 1, Fazer pedido: true
Nome cliente: João, Valor do pedido: R$13.98, Número de itens do pedido: 2, Fazer pedido: true
Nome cliente: João, Valor do pedido: R$18.47, Número de itens do pedido: 3, Fazer pedido: true
Nome cliente: João, Valor do pedido: R$21.97, Número de itens do pedido: 4, Fazer pedido: true
Nome cliente: João, Valor do pedido: R$25.47, Número de itens do pedido: 5, Fazer pedido: true

***** PEDIDO 3 *****

Nome cliente: null, Valor do pedido: R$null, Número de itens do pedido: 0, Fazer pedido: false
Nome cliente: null, Valor do pedido: R$0.0, Número de itens do pedido: 0, Fazer pedido: true
Nome cliente: Karla, Valor do pedido: R$0.0, Número de itens do pedido: 0, Fazer pedido: true
Nome cliente: Karla, Valor do pedido: R$9.99, Número de itens do pedido: 1, Fazer pedido: true
Nome cliente: Karla, Valor do pedido: R$19.98, Número de itens do pedido: 2, Fazer pedido: true
Nome cliente: Karla, Valor do pedido: R$24.47, Número de itens do pedido: 3, Fazer pedido: true
Nome cliente: Karla, Valor do pedido: R$28.46, Número de itens do pedido: 4, Fazer pedido: true
Nome cliente: Karla, Valor do pedido: R$31.96, Número de itens do pedido: 5, Fazer pedido: true
Nome cliente: Karla, Valor do pedido: R$35.46, Número de itens do pedido: 6, Fazer pedido: true

```

```
***** PEDIDO 4 *****
```

```
Nome cliente: null, Valor do pedido: R$null, Número de itens do pedido: 0, Fazer pedido: false  
Nome cliente: null, Valor do pedido: R$0.0, Número de itens do pedido: 0, Fazer pedido: true  
Nome cliente: Lucas, Valor do pedido: R$0.0, Número de itens do pedido: 0, Fazer pedido: true  
Nome cliente: Lucas, Valor do pedido: R$9.99, Número de itens do pedido: 1, Fazer pedido: true  
Nome cliente: Lucas, Valor do pedido: R$17.98, Número de itens do pedido: 2, Fazer pedido: true  
Nome cliente: Lucas, Valor do pedido: R$22.47, Número de itens do pedido: 3, Fazer pedido: true  
Nome cliente: Lucas, Valor do pedido: R$26.96, Número de itens do pedido: 4, Fazer pedido: true  
Nome cliente: Lucas, Valor do pedido: R$30.46, Número de itens do pedido: 5, Fazer pedido: true
```

```
Process finished with exit code 0
```

Version Control Run TODO Problems Terminal Build

All files are up-to-date (5 minutes ago)