

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

Discente: João Batista Araújo de Lima Filho.

Classe Sanduiche

```
import java.lang.reflect.Array;
import java.util.ArrayList;
import java.util.Scanner;

public class Sanduiche {

    // I- Crie uma classe Java relacionada ao seu tema de projeto. A classe deve ter, pelo menos, três
    //atributos, sendo um deles do tipo String.

    String nomeSanduiche;
    String tamanho;
    int numeroDeCarnes;
    boolean comparaString;
    Double preco;

    // a) Defina dois construtores distintos para esta classe.
    Sanduiche(String nomeSanduiche) {
        this.nomeSanduiche = nomeSanduiche;
        this.tamanho = "Medio";
        this.precoSanduiche(nomeSanduiche);
    }

    Sanduiche(String nomeSanduiche, String tamanho) {
        this.nomeSanduiche = nomeSanduiche;
        this.tamanho = tamanho;
        this.precoSanduiche(nomeSanduiche);
    }

    void numCarnes(int numeroDeCarnes) {
        this.numeroDeCarnes = numeroDeCarnes;
        if (numeroDeCarnes > 1) {
            this.preco += (numeroDeCarnes - 1) * 3.99;
        }
    }

    // b) Compare o atributo String de cada par de
    //instâncias para ver se têm o mesmo valor e mostre a resposta na saída do console. Faça um
    //método com retorno booleano que recebe como entrada dois objetos deste tipo. O
    //método retorna true se os objetos possuem os mesmos valores nos seus atributos, caso
    //contrário retorna false.

    boolean isComparaString(Sanduiche sanduiche1, Sanduiche sanduiche2) {

        if (sanduiche1.nomeSanduiche.equals(sanduiche2.nomeSanduiche)){
            return this.comparaString = true;
        }
        return this.comparaString = false;
    }
}
```

```

void precoSanduiche(String nomeSanduiche) {

    switch (nomeSanduiche) {

        case "Frango Tomato Artesanal":
            if (tamanho.equals("Pequeno")) {
                this.preco = 13.99;
            }
            else if (tamanho.equals("Medio")) {
                this.preco = 14.99;
            }
            else {
                this.preco = 15.99;
            }
            break;

        case "Cheddar Australiano":
            if (tamanho.equals("Pequeno")){
                this.preco = 10.99;
            }
            else if (tamanho.equals("Medio")) {
                this.preco = 11.99;
            }
            else {
                this.preco = 12.99;
            }
            break;

        case "Crispy Bacon":
            if (tamanho.equals("Pequeno")) {
                this.preco = 11.99;
            }
            else if (tamanho.equals("Medio")) {
                this.preco = 12.99;
            }
            else {
                this.preco = 13.99;
            }
            break;

        default:
            this.preco = 9.99;

    }
}

```

ATIVIDADE 5

// 1 - Crie um método.
 // a) Deve ser um método estático.
 // b) Este método deve utilizar passagem de parâmetros por referência (tipo //referenciado).
 // c) Mostre a alteração dos valores dos atributos do parâmetro (objeto) dentro do //método persiste mesmo depois que a execução do método é encerrada. Exiba o código e //as saídas do programa no relatório.

```

public static void nomeSanduiche(Sanduiche sanduiche) {
    sanduiche.nomeSanduiche = "Bob's Frango Artesanal";
    System.out.println("Dentro do metodo: "+sanduiche.nomeSanduiche);
}

```

“Esse método estático utiliza passagem de parâmetro por referência, recebendo um Sanduiche e mudando o seu nome. Desta forma a alteração do nome do sanduiche persiste mesmo depois da execução do método ser encerrada.”

```
// 2 - Crie um método.  
// a) Utilize sobrecarga de métodos. Crie outro método com o mesmo nome do  
// método da questão anterior porém com parâmetros diferentes.  
// b) Este método deve utilizar passagem de parâmetros por valor (tipo primitivo).  
// c) Mostre que a alteração do valor o parâmetro dentro do método não persiste  
// após o final da execução do método. Exiba o código e as saídas do programa no  
// relatório.
```

```
public static void nomeSanduiche(String nome) {
    nome = "Big Bob Frango";
    System.out.println("Dentro do metodo: "+nome);
}
```

“Esse metodo estatico foi criado usando sobrecarga de metodo. Ele recebe um tipo primitivo String com o nome do sanduiche. Modifica o atributo dentro do metodo, mas ele não persiste após a finalização da execução do metodo”

```
@Override
public String toString() {
    return "Sanduiche encontrado: " + nomeSanduiche +
        ", tamanho: " + tamanho +
        ", preco: R$" + preco;
}
```

```
public static void main(String[] args) {
```

```
System.out.println("==*==*==*==*==*==*==* METODO COM PASSAGEM POR  
REFERENCIA ==*==*==*==*==*==*==*\n");
```

“Aqui é usado o metodo com passagem de parametro por referencia. E mostra o antes o durante e o depois”

```
Sanduche sanduche = new Sanduche("Costela Artesanal");
System.out.println("Antes: "+sanduche.nomeSanduche);
Sanduche.nomeSanduche(sanduche);
System.out.println("Depois: "+sanduche.nomeSanduche);
```

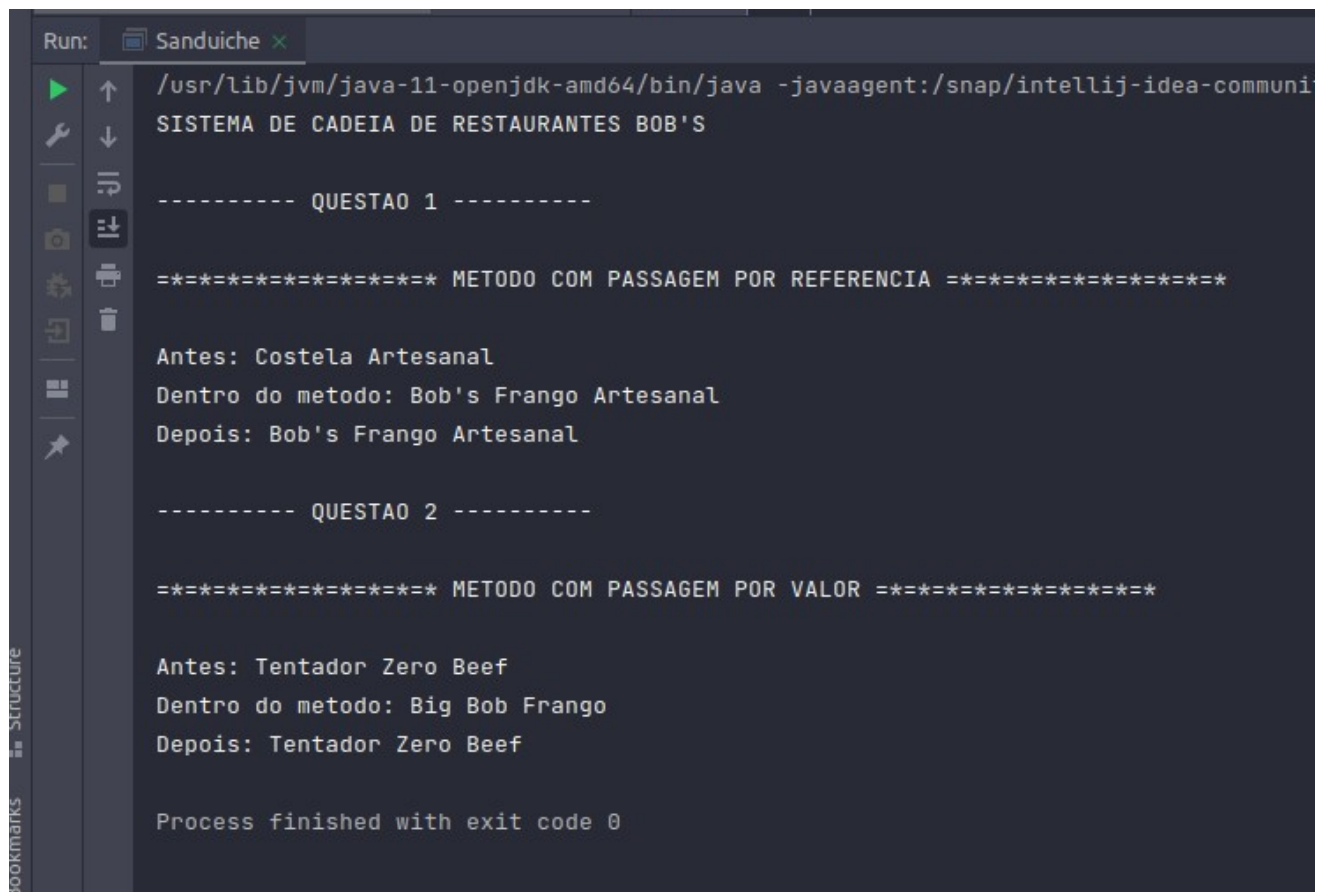
```
System.out.println("\n==*==*==*==*==*==* METODO COM PASSAGEM POR VALOR  
==*==*==*==*==*==*\n");
```

“Aqui é usado o metodo com passagem de parametro por valor. E mostra o antes o durante e o depois”

```
Sanduche sanduche2 = new Sanduche("Tentador Zero Beef");
System.out.println("Antes: "+sanduche2.nomeSanduche);
Sanduche.nomeSanduche(sanduche2.nomeSanduche);
System.out.println("Depois: "+sanduche2.nomeSanduche);
}
```

}

IMAGENS DAS SAÍDAS NO CONSOLE



The image shows a screenshot of an IDE's console window. The title bar of the console tab is labeled 'Run: Sanduiche x'. The console output is as follows:

```
/usr/lib/jvm/java-11-openjdk-amd64/bin/java -javaagent:/snap/intellij-idea-communi
SISTEMA DE CADEIA DE RESTAURANTES BOB'S

----- QUESTAO 1 -----

***** METODO COM PASSAGEM POR REFERENCIA *****

Antes: Costela Artesanal
Dentro do metodo: Bob's Frango Artesanal
Depois: Bob's Frango Artesanal

----- QUESTAO 2 -----

***** METODO COM PASSAGEM POR VALOR *****

Antes: Tentador Zero Beef
Dentro do metodo: Big Bob Frango
Depois: Tentador Zero Beef

Process finished with exit code 0
```

The IDE interface includes a sidebar on the left with icons for Run, Debug, Structure, and other tools. The 'Structure' tab is currently selected in the sidebar.