

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

Discente: João Batista Araújo de Lima Filho.

Questão 2 Classe CarnesInsuficienteException

```
package Atividade12;

// 2- Pelo menos, 2 Exceções

    “Essa é uma exceção que trata o erro de carnes insuficientes”

public class CarnesInsuficienteException extends Exception {
    private int numeroCarnes;
    public CarnesInsuficienteException(int numeroCarnes) {
        super("Número insuficiente de carnes!");
        this.numeroCarnes = numeroCarnes;
    }
}
```

Classe NumeroMaximoException

```
package Atividade12;

// 2- Pelo menos, 2 Exceções

    “Essa exceção trata o erro de número maximo de carnes no sanduiche”

public class NumeroMaximoException extends Exception{

    public NumeroMaximoException(String mensagem) {
        super(mensagem);
    }

}
```

Questão 3 A Classe Abstrata Lanche

```
package Atividade12;

// 3- a) Classe Abstrata.

    “Essa é a classe abstrata Lanche”

public abstract class Lanche {

    private String nome;
    private double preco;
    private String tamanho;
```

```

public Lanche(String nome, double preco, String tamanho) {
    this.nome = nome;
    this.preco = preco;
    this.tamanho = tamanho;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public double getPreco() {
    return preco;
}

public void setPreco(double preco) {
    this.preco = preco;
}

public String getTamanho() {
    return tamanho;
}

public void setTamanho(String tamanho) {
    this.tamanho = tamanho;
}
}

```

Questões 3 B e 4 B

Classe Sanduiche que extends Lanche e implements IPagamento, Item, Serializeble

```

package Atividade12;

import java.io.Serializable;

// 3- b) Pelo menos duas subclasses.

    "Essa é classe Sanduiche é uma subclasse de Lanche"

// 4- b) Herança múltipla.

    "Aqui temos uma Herança múltipla"

public class Sanduiche extends Lanche implements IPagamento, Item, Serializable {

    private String nome;
    private String tamanho;
    private double preco;
    private int numeroDeCarnes;
    private String pontoDaCarne;

    int id;
}

```

```
public Sanduiche(String nome, String tamanho, double preco) {
    super(nome, preco, tamanho);
}

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

void pontoCarne(String ponto) {
    this.pontoDaCarne = ponto;
}

public String getNome() {
    return nome;
}

public void setNome(String nome) {
    this.nome = nome;
}

public String getTamanho() {
    return tamanho;
}

public void setTamanho(String tamanho) {
    this.tamanho = tamanho;
}

public double getPreco() {
    return preco;
}

public void setPreco(double preco) {
    this.preco = preco;
}

@Override
public String toString() {
    return "Sanduiche: "+nome +", Tamanho: "+ tamanho+", Preço: "+preco;
}

@Override
public int getID() {
    return id;
}

@Override
public void setID(int id) {
    this.id = id;
}
}
```

```

@Override
public void formaPagamento(String forma) {

}

@Override
public void valorTotal() {

}
}

```

Classe Sanduiche2 que extends Lanche e implements Item, Serializable

```

package Atividade12;

import java.io.Serializable;

// 3- b) Pelo menos duas subclasses.

    “Essa classe Sanduiche2 é outra subclasse da classe Lanche”

public class Sanduiche2 extends Lanche implements Item, Serializable {

    int id;
    Sanduiche sanduiche;

    public Sanduiche2(String nome, String tamanho, double preco){
        super(nome,preco,tamanho);
    }

    @Override
    public String toString() {
        return sanduiche.toString();
    }

    @Override
    public int getID() {
        return id;
    }

    @Override
    public void setID(int id) {
        this.id = id;
    }
}

```

Questão 4 A Interface Item

```

package Atividade12;

// 4- a) Pelo menos, 2 Interfaces.

```

“Essa é uma interface de Item”

```
public interface Item {  
  
    int getID();  
  
    void setID(int id);  
  
}
```

Interface IPagamento

```
package Atividade12;  
  
// 4- a) Pelo menos, 2 Interfaces.  
  
“Essa é outra interface de pagamentos”  
  
public interface IPagamento {  
  
    void formaPagamento(String forma);  
    void valorTotal();  
  
}
```

Questão 5 Classe Main (GUI)

```
package Atividade12;  
  
import java.io.IOException;  
import java.util.Scanner;  
  
public class Main extends RepositorioSanduiche{  
  
    // 5- Atualize a GUI para incluir as novas funções do sistema. Demonstre o  
    // funcionamento do sistema através de prints da tela de saída (console).  
  
    “Essa é a GUI do sistema, que mostra o funcionamento da mesma”  
  
    static final String opcoes = "Digite qual operacao voce deseja fazer: \n1 - Cadastrar  
Sanduiche. \n2 - Mostrar Sanduiches. " +  
        "\n3 - Sair.\n";  
  
    public static void main(String[] args) {  
  
        Scanner sc = new Scanner(System.in);  
        boolean continuar = true;  
  
        RepositorioSanduiche repositorioSanduiche = null;  
        String arquivoRepositorio = "sanduiches.txt";  
        try {
```

```

        repositorioSanduiche =
RepositorioSanduiche.carregarRepositorio(arquivoRepositorio);
    } catch (IOException e) {
        System.out.println("Nao foi possivel carregar o repositorio.");
        repositorioSanduiche = new RepositorioSanduiche();
    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    }
}

while (continuar) {
    System.out.println(opcoes);
    int leitura = sc.nextInt();

    if (leitura == 1) {
        System.out.println("Digite o nome do Sanduiche: ");
        String a = sc.nextLine();
        String nome = sc.nextLine();

        System.out.println("Digite o tamanho do Sanduiche: ");
        String tamanho = sc.next();

        System.out.println("Digite o preco do Sanduiche: ");
        double preco = sc.nextDouble();

        Item sanduiche = new Sanduiche2(nome, tamanho, preco);
        repositorioSanduiche.cadastraSanduiche(sanduiche);
        // salvando repositorio no arquivo
        try {
            repositorioSanduiche.salvarRepositorio(arquivoRepositorio);
        } catch (IOException e) {
            e.printStackTrace();
            System.out.println("Nao foi possivel salvar o arquivo.");
        }

    } else if (leitura == 2) {
        for (int i = 0; i < repositorioSanduiche.ultimoID(); i++) {
            Item a = repositorioSanduiche.recuperar(i);
            System.out.println(a);
        }
    } else if (leitura == 3) {
        continuar = false;
    }

}

}

}

```