

A Nova Estrutura Orientada a Objetos

Vamos precisar das seguintes classes:

Produto.java: Representa um único produto (você já tem uma ótima base).

Setor.java: Representa uma seção do supermercado (Limpeza, Mercearia) e conterá uma lista de produtos.

Catalogo.java: Representará o catálogo completo do supermercado, organizando todos os setores.

ItemCarrinho.java: Representará um item dentro do carrinho de compras (um produto + a quantidade desejada).

Carrinho.java: Representará o carrinho de compras do cliente, com a lista de itens e métodos para calcular o total.

SupermercadoTerminal.java: Será nossa classe principal (main), responsável pela interação com o usuário no console.

O Código Java Completo

Aqui estão as classes prontas para você. Você pode criar um novo projeto Java e adicionar cada uma delas como um novo arquivo .java.

```
public class Produto {  
    private int id;  
    private String nome;  
    private double preco;  
  
    public Produto(int id, String nome, double preco) {  
        this.id = id;  
        this.nome = nome;  
        this.preco = preco;  
    }  
  
    public int getId() {  
        return id;  
    }  
  
    public String getNome() {  
        return nome;  
    }  
  
    public double getPreco() {  
        return preco;  
    }  
  
    @Override  
    public String toString() {  
        return "ID: " + id + " | " + nome + " | R$ " + String.format("%.2f", preco);  
    }  
}
```

```
import java.util.ArrayList;

import java.util.List;


public class Setor {

    private String nome;

    private List<Produto> produtos = new ArrayList<>();


    public Setor(String nome) {

        this.nome = nome;

    }


    public void adicionarProduto(Produto produto) {

        this.produtos.add(produto);

    }


    public String getNome() {

        return nome;

    }


    public List<Produto> getProdutos() {

        return produtos;

    }


    public Produto getProdutoPorIndice(int indice) {

        if (indice >= 0 && indice < produtos.size()) {

            return produtos.get(indice);

        }

        return null;

    }

}
```

```
public class ItemCarrinho {  
  
    private Produto produto;  
  
    private int quantidade;  
  
  
    public ItemCarrinho(Produto produto, int quantidade) {  
  
        this.produto = produto;  
  
        this.quantidade = quantidade;  
  
    }  
  
  
    public Produto getProduto() {  
  
        return produto;  
  
    }  
  
  
    public int getQuantidade() {  
  
        return quantidade;  
  
    }  
  
  
    public double getSubtotal() {  
  
        return produto.getPreco() * quantidade;  
  
    }  
  
}
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
public class Carrinho {
```

```
    private List<ItemCarrinho> itens = new ArrayList<>();
```

```
    public void adicionarItem(Produto produto, int quantidade) {
```

```
        itens.add(new ItemCarrinho(produto, quantidade));
```

```
        System.out.println(quantidade + "x " + produto.getNome() + " adicionado ao carrinho.");
```

```
    }
```

```
    public double calcularTotal() {
```

```
        double total = 0;
```

```
        for (ItemCarrinho item : itens) {
```

```
            total += item.getSubtotal();
```

```
        }
```

```
        return total;
```

```
    }
```

```
    public void exibirResumo() {
```

```
        System.out.println("\n--- RESUMO DA COMPRA ---");
```

```
        if (itens.isEmpty()) {
```

```
            System.out.println("Carrinho vazio.");
```

```
            return;
```

```
        }
```

```
        for (ItemCarrinho item : itens) {
```

```
            System.out.println(item.getProduto().getNome() + "---- " + item.getQuantidade() + " X R$ " +
```

```
String.format("%.2f", item.getProduto().getPreco()) + " = R$ " + String.format("%.2f", item.getSubtotal()));
```

```
        }
```

```
        System.out.println("-----");
```

```
        System.out.println("TOTAL DA COMPRA = R$ " + String.format("%.2f", calcularTotal()));
```

```
        System.out.println("-----");
```

```
    }
```

```
}
```

```
import java.util.ArrayList;

import java.util.List;

public class Catalogo {

    private List<Setor> setores = new ArrayList<>();

    public Catalogo() {
        carregarDados();
    }

    private void carregarDados() {

        // Recria a mesma estrutura de dados do seu código Python
        Setor limpeza = new Setor("LIMPEZA");
        limpeza.adicionarProduto(new Produto(101, "Água Sanitária L", 3.00));
        limpeza.adicionarProduto(new Produto(102, "Sabão em pó Omo 1Kg", 40.00));

        Setor mercearia = new Setor("MERCEARIA");
        mercearia.adicionarProduto(new Produto(201, "Feijão Kicaldo 1kg", 8.00));
        mercearia.adicionarProduto(new Produto(202, "Arroz Camil 1Kg", 5.00));

        Setor bebidas = new Setor("BEBIDAS");
        bebidas.adicionarProduto(new Produto(301, "Coca-cola 2L", 10.00));
        bebidas.adicionarProduto(new Produto(302, "Suco de uva 2L", 20.00));

        Setor padaria = new Setor("PADARIA");
        padaria.adicionarProduto(new Produto(401, "Pão de batata Kg", 16.80));
        padaria.adicionarProduto(new Produto(402, "Sonho Un", 8.00));

        this.setores.add(limpeza);
        this.setores.add(mercearia);
        this.setores.add(bebidas);
        this.setores.add(padaria);
    }

    public List<Setor> getSetores() {
```

```
        return setores;
    }

    public Setor getSetorPorIndice(int indice) {
        if (indice >= 0 && indice < setores.size()) {
            return setores.get(indice);
        }
        return null;
    }
}
```

```
import java.util.List;

import java.util.Scanner;


public class SupermercadoTerminal {


    private static Scanner scanner = new Scanner(System.in);

    private static Catalogo catalogo = new Catalogo();

    private static Carrinho carrinho = new Carrinho();


    public static void main(String[] args) {

        System.out.println("----BEM-VINDOS AO SUPERMERCADO PITHON (VERSÃO JAVA)----");


        loopPrincipalDeCompras();

        processarPagamento();


        System.out.println("\nObrigado pela preferência, volte sempre!");

        scanner.close();

    }


    private static void loopPrincipalDeCompras() {

        int continuarComprando = 1;

        while (continuarComprando == 1) {

            exibirMenuSetores();

            int escolhaSetor = scanner.nextInt();

            Setor setorEscolhido = catalogo.getSetorPorIndice(escolhaSetor - 1);

            if (setorEscolhido != null) {

                loopDeComprasNoSetor(setorEscolhido);

            } else {

                System.out.println("Opção de setor inválida.");

            }

        }


        carrinho.exibirResumo();

        System.out.print("\nDeseja escolher outro setor? (1 para SIM, 2 para NÃO): ");

        continuarComprando = scanner.nextInt();

    }

}
```



```
}  
}
```

```
private static void exibirMenuSetores() {  
    System.out.println("\nTEMOS OS SEGUINTE SETORES: ");  
    List<Setor> setores = catalogo.getSetores();  
    for (int i = 0; i < setores.size(); i++) {  
        System.out.println("[ " + (i + 1) + " ] - " + setores.get(i).getNome());  
    }  
    System.out.print("Escolha o número do setor que deseja conferir: ");  
}
```

```
private static void loopDeComprasNoSetor(Setor setor) {  
    int continuarNoSetor = 1;  
    while (continuarNoSetor == 1) {  
        System.out.println("\n--- Produtos no setor " + setor.getNome() + " ---");  
        List<Produto> produtos = setor.getProdutos();  
        for (int i = 0; i < produtos.size(); i++) {  
            System.out.println("[ " + (i + 1) + " ] - " + produtos.get(i).getNome() + " ---- R$ " + String.format("%.2f",  
produtos.get(i).getPreco()));  
        }  
    }  
}
```

```
System.out.print("Digite o número do produto que deseja levar: ");  
int escolhaProduto = scanner.nextInt();  
Produto produtoEscolhido = setor.getProdutoPorIndice(escolhaProduto - 1);
```

```
if (produtoEscolhido != null) {  
    System.out.print("Quantas unidades você deseja levar? ");  
    int quantidade = scanner.nextInt();  
    carrinho.adicionarItem(produtoEscolhido, quantidade);  
} else {  
    System.out.println("Opção de produto inválida.");  
}
```

```
System.out.print("\nDeseja levar mais algum produto deste setor? (1 para SIM, 2 para NÃO): ");
```

```
        continuarNoSetor = scanner.nextInt();
    }
}

private static void processarPagamento() {
    // Lógica de pagamento (simplificada, pode ser expandida como no seu original)
    System.out.println("\n--- PAGAMENTO ---");
    carrinho.exibirResumo();
    double total = carrinho.calcularTotal();

    System.out.println("[1] - Crédito à Vista");
    System.out.println("[2] - Pix");
    System.out.println("[3] - Cancelar");
    System.out.print("Escolha a forma de pagamento: ");
    int formaPagamento = scanner.nextInt();

    if (formaPagamento == 1 || formaPagamento == 2) {
        System.out.println("Pagamento de R$ " + String.format("%.2f", total) + " realizado com sucesso!");
    } else {
        System.out.println("Compra cancelada.");
    }
}
}
```