No ambiente competitivo do comércio eletrônico, oferecer benefícios aos clientes é uma estratégia crucial para aumentar as vendas e fidelizar consumidores. Com isso em mente, uma loja virtual implementou uma regra de desconto progressivo, incentivando os clientes a gastarem mais para obter maiores descontos em suas compras. A lógica por trás dessa estratégia é simples: quanto maior o valor da compra, maior o percentual de desconto concedido.  
  
Para garantir que essa regra fosse aplicada corretamente nas transações, a loja contratou um desenvolvedor Java para criar um programa que calculasse o desconto com base no valor total da compra. A expectativa era que o sistema funcionasse perfeitamente, garantindo que os clientes recebessem os descontos corretos de acordo com as faixas de valor definidas:  
  
- Compras entre R$ 500,00 e R$ 1.000,00: 5% de desconto.  
- Compras entre R$ 1.000,01 e R$ 2.000,00: 10% de desconto.  
- Compras acima de R$ 2.000,00: 15% de desconto.  
  
No entanto, após a implementação, os testes mostraram que o cálculo do desconto não estava correto, resultando em valores que não correspondiam aos esperados para determinadas faixas de preço. Agora, é necessário analisar o código para identificar onde o erro pode estar ocorrendo e corrigir o problema.  
  
Analise o código implementado pelo desenvolvedor Java e identifique os erros que estão causando o cálculo incorreto dos descontos progressivos:

import java.util.List;

import java.util.ArrayList;

class Produto {

String nome;

double valorUnitario;

int quantidade;

Produto(String nome, double valorUnitario, int quantidade) {

this.nome =nome;

this.valorUnitario = valorUnitario;

this.quantidade = quantidade;

}

}

public class Venda {

private List<Produto> produtos;

public Venda () {

this.produtos = new ArrayList<>();

}

public void adicionarProduto(Produto produto) {

this.produtos.add(produto);

}

public double calcularTotal(){

double total = 0;

for (Produto produto : produtos) {

total += produto.valorUnitario \* produto.quantidade;

}

if (total >= 500 || total < 1000) {

total = total \* 0.05;

} else if (total >= 1000 || total < 2000) {

total = total \* 0.10;

} else if (total >= 2000) {

total = total \* 0.15;

}

return total;

}

}

Em seguida, descreva qual a solução necessária para corrigir o método calcularTotal.

Ao analisar o código, é possível identificar alguns problemas na implementação do cálculo do desconto. Vamos analisar os principais erros:

1. **Operadores Lógicos Incorretos**: No método calcularTotal(), os operadores lógicos usados para definir as faixas de desconto estão incorretos. O operador || (ou lógico) foi usado no lugar do operador && (e lógico) nas condições if, o que leva a uma interpretação incorreta das faixas de valor. As condições precisam ser ajustadas para garantir que o valor total da compra esteja dentro do intervalo correto de cada faixa.

O trecho:

java

Copiar código

if (total >= 500 || total < 1000) {

deveria ser:

java

Copiar código

if (total >= 500 && total <= 1000) {

O mesmo problema se repete para as outras condições.

1. **Aplicação do Desconto**: O cálculo atual aplica o desconto diretamente ao total, mas isso substitui o valor total pelo valor do desconto. Para calcular o valor final da compra com desconto, precisamos subtrair o desconto do valor total original.

Ao invés de:

java

Copiar código

total = total \* 0.05;

a expressão correta seria:

java

Copiar código

total = total - (total \* 0.05);

Aqui está o código corrigido:

java

Copiar código

import java.util.List;

import java.util.ArrayList;

class Produto {

String nome;

double valorUnitario;

int quantidade;

Produto(String nome, double valorUnitario, int quantidade) {

this.nome = nome;

this.valorUnitario = valorUnitario;

this.quantidade = quantidade;

}

}

public class Venda {

private List<Produto> produtos;

public Venda() {

this.produtos = new ArrayList<>();

}

public void adicionarProduto(Produto produto) {

this.produtos.add(produto);

}

public double calcularTotal() {

double total = 0;

for (Produto produto : produtos) {

total += produto.valorUnitario \* produto.quantidade;

}

if (total >= 500 && total <= 1000) {

total = total - (total \* 0.05); // 5% de desconto

} else if (total > 1000 && total <= 2000) {

total = total - (total \* 0.10); // 10% de desconto

} else if (total > 2000) {

total = total - (total \* 0.15); // 15% de desconto

}

return total;

}

}

### Explicação das Alterações:

* **Condições Corrigidas**: Alteramos as condições para usar &&, assegurando que o total esteja dentro do intervalo de valores correto para cada faixa.
* **Cálculo do Desconto**: Subtraímos o desconto calculado do valor total, em vez de substituí-lo pelo valor do desconto.

Com essas alterações, o programa deverá calcular o desconto corretamente com base nas faixas definidas.