

(https://accounts.coursera.org/i/zendesk/courserahelp?return_to=https://learner.coursera.help/hc)

Tarefa: Diferentes tipos de produto

Enviar até Março 27, 11:59 PM PDT

Informações importantes

É especialmente importante enviar essa tarefa antes do prazo, Março 27, 11:59 PM PDT, pois ela deve receber uma nota dos colegas. Se você enviar depois dessa data, pode não haver colegas suficientes para avaliar seu trabalho. Isso dificulta (e, em alguns casos, impossibilita) a atribuição de notas. Envie no prazo para evitar esses riscos.

Instruções (/learn/orientacao-a-objetos-com-java/peer/OEIEI/diferentes-tipos-de-produto)

My submission (/learn/orientacao-a-objetos-com-java/peer/OEIEI/diferentes-tipos-de-produto/submit)

Discussões (/learn/orientacao-a-objetos-com-java/peer/OEIEI/diferentes-tipos-de-produto/discussions)

Instruções

Crie uma classe chamada Produto que deve possuir um nome, um código e um preço. Sobrescreva os métodos equals() e hashCode() de Object (veja seção sobre esses métodos), de forma a serem considerados iguais instancias de Produto que possuam o mesmo código.

Crie uma classe chamada ProdutoComTamanho que estenda a classe Produto. Essa classe deve possuir uma informação adicional de tamanho. Um exemplo seria o tamanho de uma roupa ou a numeração de um calçado. Método equals() e hashCode() devem ser sobrescritos de forma que um produto com mesmo código e tamanhos diferentes são considerados diferentes.

Crie uma classe CarrinhoDeCompras que armazene em um atributo interno do tipo HashMap cada produto adicionado no carrinho e sua respectiva quantidade. O método adicionaProduto() deve receber a instancia do produto e a quantidade. Caso o produto já exista no HashMap, a quantidade deve ser somada a que já existe no carrinho. Deve haver também um método removeProduto() que também recebe a instancia do produto e a quantidade a ser removida. Observe que produtos de tamanhos diferentes devem ser considerados como produtos diferentes no carrinho. O carrinho deve possuir um método que calcula o valor total da compra.

Crie testes com Unit para a classe Produto, para a classe ProdutoComTamanho e para a classe CarrinhoDeCompras. Os testes de cada classe devem ser colocados em classes separadas e devem estar em um diretório de código diferente das classes de produção.

Review criteria

menos ^

Serão verificados se os itens pedidos no enunciado foram cumpridos. Procure entregar um código organizado pois isso também será avaliado.

Métodos equals() e hashCode()

menos ^

O método equals() é utilizado para a comparação de objetos. Ele é definido na classe Object e pode ser sobrescrito em qualquer classe que possua uma lógica diferente.

O método hashCode() também é definido na classe Object e retorna o código hash de um objeto, que é utilizado em várias classes que implementam estruturas de dados. Ele precisa obedecer a seguinte regra: quando o método equals() retornar true, o retorno de hashCode() desses objetos precisa ser igual. Mas se o retorno do hashCode() for igual, não necessariamente os objetos são iguais, ou seja, o equals() precisa retornar true.

É importante respeitar essa regra para que algumas classes como HashSet e HashMap funcionem corretamente!

