

LP1A3 – Exercícios

Aula 10 – Introdução ao Desenvolvimento guiado por Testes

Instruções para entrega das listas de exercícios:

Meio de Entrega: As resoluções das listas de exercícios devem ser entregues exclusivamente por meio do ambiente Moodle (<http://eadcampus.spo.ifsp.edu.br>).

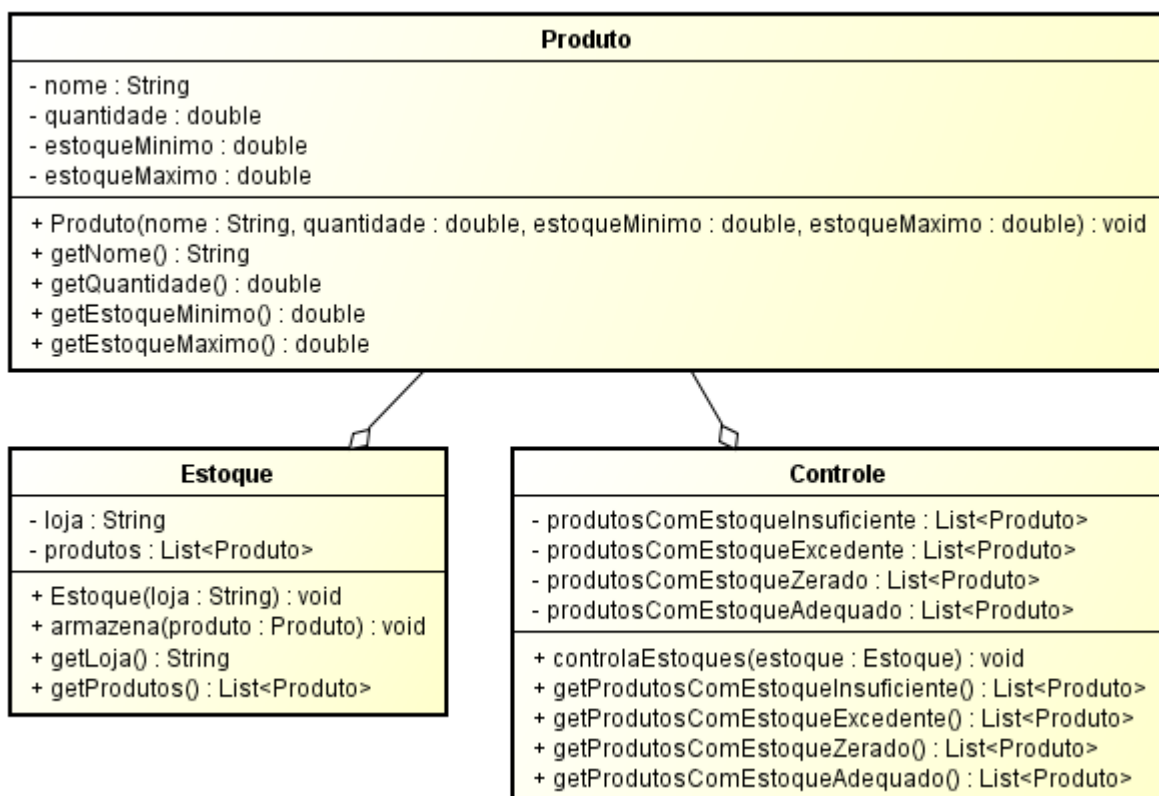
Forma de Entrega: Para exercícios com uma única classe, deve ser entregue o arquivo da classe (extensão JAVA) referente ao exercício. Por exemplo: Exercício3.java. Para exercícios com mais de uma classe, cada exercício deve ter uma pasta, na qual serão colocados os arquivos JAVA referentes ao exercício. Por exemplo: Para o Exercício 4, deve existir uma pasta “Exercicio4” contendo todos os arquivos JAVA deste exercício. **Entregue apenas os arquivos JAVA.** Todos os arquivos da lista devem ser compactados em um único arquivo (extensão RAR ou ZIP), cujo nome deverá conter a aula, o nome e um sobrenome do aluno. Por exemplo: Aula2_JoaoSilva.zip.

Prazo de Entrega: O prazo de entrega está definido na própria página de exercícios do Moodle, lembrando que o sistema bloqueia o envio de arquivos após a data e horário indicados.

Obs.: A resolução deste(s) exercício(s) deve ser feita de forma INDIVIDUAL. Listas de exercícios com uma ou mais respostas idênticas serão desconsideradas integralmente para efeitos de nota de participação.

1. Faça um programa console em Java que contemple o diagrama de classes a seguir. Se necessário, inclua outros atributos e implemente métodos auxiliares (a fim de evitar métodos contendo códigos muito extensos). Em seguida, use o **JUnit** para implementar e executar os métodos de teste descritos na sequência.

Obs 1.: Considere as seguintes regras de negócio: (a) o estoque de uma loja não deve aceitar produtos com o mesmo nome; (b) um produto não deve aceitar estoque mínimo menor que 5; e (c) um produto não deve aceitar estoque máximo maior que 100. **Para essas três regras de negócio, tente implementar primeiro os códigos de teste e, somente depois, escreva os códigos de produção. Lembre-se, esta é a forma correta de aplicar o TDD (escrever primeiro o código de teste, depois o de produção).**



- Classe **ControleTest**:
 - deveRetornarProdutosComEstoqueInsuficiente
 - deveRetornarProdutosComEstoqueExcedente
 - deveRetornarProdutosComEstoqueZerado
 - deveRetornarProdutosComEstoqueAdequado
 - deveRetornarListaVaziaParaEstoqueSemProdutos
 - deveRetornarListaVaziaParaEstoqueSemProdutosComEstoqueZerado
 - naoDeveControlarEstoquesSemNenhumProduto (testar lançamento de exceção)
- Classe **EstoqueTest**:
 - naoDeveAceitarProdutosComMesmoNome
- Classe **ProdutoTest**:
 - naoDeveAceitarProdutosComEstoqueMinimoMenorQueCinco (testar lançamento de exceção)
 - naoDeveAceitarProdutosComEstoqueMaximoMaiorQueCem (testar lançamento de exceção)

Obs 2.: Nas classes de teste, use ao menos a anotação **@Before** e crie uma classe **Test Data Builder** para criação de estoques.