

Vamos agora inserir um controle de possíveis erros em nosso trabalho. Nessa atividade entraremos no controle de exceções no nosso sistema de Controle de Estoque.

Inicialmente pedimos que você crie um pacote com o nome “estoqueComProdutoPerecivelExcecoes” para implementar esta versão do trabalho.

Reforçamos também que você deverá submeter a pasta com o seu nome, compactada, dentro da pasta src do projeto. Exemplo: Poo20212/src/pedroSantosNeto à essa pasta como nome “pedroSantosNeto” é que eu deveria compactar e enviar como trabalho. Ela vai incluir diversas outras pastas, incluindo estoque e estoqueComProdutoPerecivelExcecoes.

Neste trabalho a grande diferença é a presença de exceções. Continuaremos com nossa classe ProdutoPerecivel, que são produtos que devem possuir lotes internamente e cada lote deve possuir a indicação da quantidade de produtos no lote e sua data de validade.

Um ProdutoPerecivel tem tudo que um produto tem. Mas a compra de produtos deve ser feita incluindo lotes internamente ao produto. O método compra da classe Estoque também foi alterado. Agora ele exige que seja sempre passado a data de validade do produto. Caso a data seja nula, indica que o produto não tem validade.

A compra deve continuar atualizando o preço de compra, bem como o preço de venda (via acréscimo do lucro).

A venda também terá uma mudança importante: no caso de venda de produtos perecíveis devemos sempre começar vendendo o mais próximo do vencimento. Se um lote com 10 produtos for o mais próximo do vencimento e tiver sido solicitado a venda de 15 produtos, devemos vender esse lote inteiro e depois procurar o próximo lote disponível para realizar a venda do restante dos produtos.

A venda só pode ser feita para produtos dentro da validade. Não é permitida a venda de produtos fora da validade. Também não devemos vender produtos perecíveis se não houve a quantidade solicitada de itens válidos.

É possível saber a quantidade de itens vencidos de um produto. Deve ser criado um método específico para isso, listado a seguir.

Não deve ser possível cadastrar um produto com dados inválidos. Isso deve gerar uma exceção com o nome DadosInvalidos. Isso significa que os campos numéricos devem ser maiores que zero, os campos de texto devem ser preenchidos (não podem ser nulos nem vazios).

Métodos Classe Estoque Atualizado

```
public interface InterfaceEstoqueComExcecoes {
```

```
    public void incluir(Produto p) throws ProdutoJaCadastrado, DadosInvalidos;
```

//Se for enviado validade e o produto não for perecível, exceção ProdutoNaoPerecivel

public void comprar(**int** cod, **int** quant, **double** preco, Date val) **throws**
ProdutoInexistente, DadosInvalidos, ProdutoNaoPerecivel;

public double vender(**int** cod, **int** quant) **throws** ProdutoInexistente, ProdutoVencido,
DadosInvalidos;

public Produto pesquisar (**int** cod) **throws** ProdutoInexistente;

public ArrayList<Produto> estoqueAbaixoDoMinimo();

public ArrayList<Produto> estoqueVencido();

public int quantidadeVencidos(**int** cod) **throws** ProdutoInexistente;

}

Construtores:

public Produto(**int** cod, String desc, **int** min, **double** lucro, Fornecedor forn);

public ProdutoPerecivel(**int** cod, String desc, **int** min, **double** lucro, Fornecedor forn);

public Lote(**int** quant, Date validade);