 INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA PARAÍBA Campus Campina Grande	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA – CAMPUS CAMPINA GRANDE		
	CURSO:	Bacharelado em Engenharia de Computação	
	PERÍODO:	3º	TURMA: 20182.3.125.1V
	DISCIPLINA:	Laboratório de Programação Orientada a Objetos	
	PROFESSOR:	JAINDSON VALENTIM SANTANA	SEMESTRE LETIVO: 2018

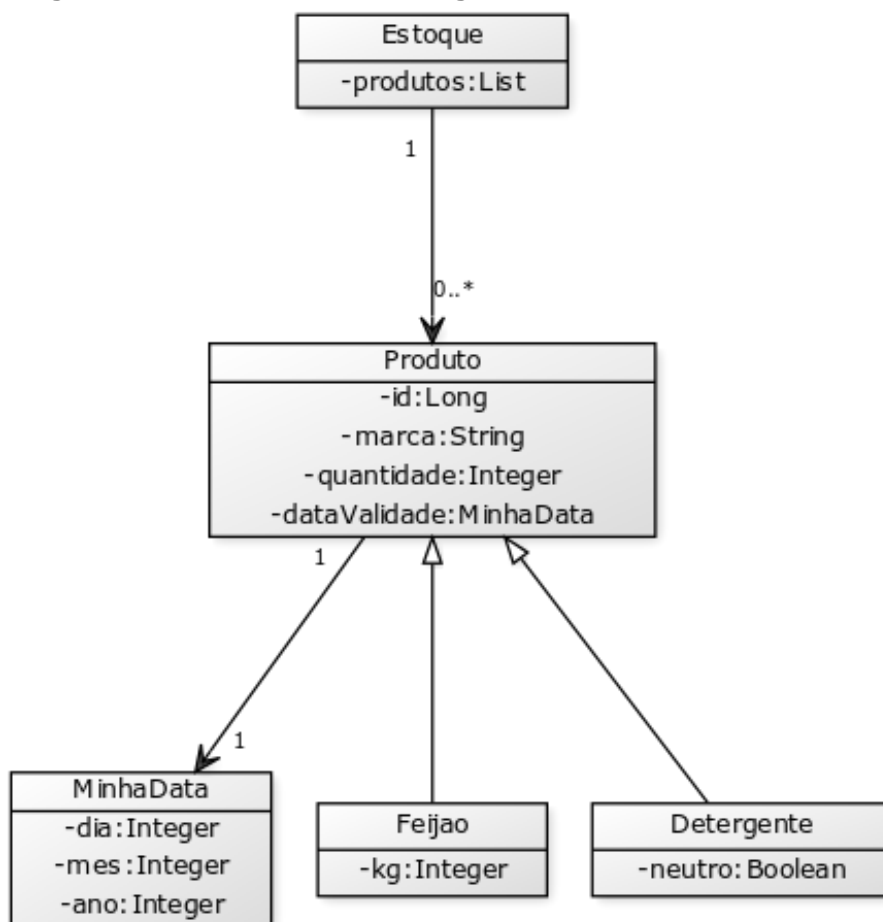
OBSERVAÇÕES

- A atividade é individual e deve ser feita por todos os alunos.
- O prazo para a entrega será **03/12/2018**. A atividade deve ser entregue por e-mail para o professor: `j a i n d s o n v s @ g m a i l . c o m`. Ao enviar o e-mail, colocar “Atividade – 20181130” no título do e-mail e seu nome no corpo do e-mail.

Atividade - 20181130

- 1) Desenvolva uma aplicação via linha de comando de Controle de Estoque de um Supermercado com Produtos a serem gerenciados. Esta aplicação é semelhante ao gerenciador de Zoológico compartilhado pelo professor em sala de aula. OBS: façam uso de Interfaces e Classes Abstratas conforme adequado.

a) **Diagrama de Classes** mínimo sugerido:



b) Operações do Menu Principal:

1. **Listar Produtos:** seu programa deve imprimir todos os produtos cadastrados em ordem de inserção deles no estoque;
2. **Listar Produtos em ordem alfabética:** seu programa deve imprimir todos os produtos cadastrados em ordem alfabética (dica: fazer a classe Produto implementar Comparable<Produto>);
3. **Listar Produtos em ordem crescente de quantidade:** seu programa deve imprimir todos os produtos cadastrados em ordem crescente de quantidade (dica: criar um Comparator<Produto> para implementar esta lógica de comparação);
4. **Listar Produtos em ordem crescente de data de validade:** seu programa deve imprimir todos os produtos cadastrados em ordem crescente de data de validade (dica: criar um Comparator<Produto> para implementar esta lógica de comparação, além de fazer a classe MinhaData implementar Comparable<MinhaData> para facilitar ainda mais esta lógica);
5. **Listar produtos que estarão vencidos em <data>:** listar os produtos cuja data de validade é anterior a uma determinada data fornecida pelo usuário (dica: fazer a classe MinhaData implementar Comparable<MinhaData>). Para isso, solicite do usuário a data (primeiro o dia, depois o mês, depois o ano, um por vez), em seguida, crie uma instância de MinhaData com base nestes dados e utilize ele para verificar no estoque quais produtos estão com a data de validade anterior a esta data. Ou seja, esta funcionalidade serve para o dono do supermercado saber que produtos irão vencer numa determinada data;
6. **Adicionar Detergente:** o programa deve solicitar a marca, quantidade, dia, mês e ano da validade, neutro (cada uma destas informações, uma por vez), criar uma instância de Detergente com base nestas informações e adicionar no Estoque;
7. **Adicionar Feijão:** o programa deve solicitar a marca, quantidade, dia, mês e ano da validade, kg (cada uma destas informações, uma por vez), criar uma instância de Feijao com base nestas informações e adicionar no Estoque;
8. **Remover Produto:** liste os produtos e peça pro usuário o ID do produto que ele deseja remover. Trate o caso do ID não ser encontrado;
9. **Incrementar quantidade de Produto do Estoque:** liste os produtos e peça pro usuário o ID do produto que ele deseja incrementar a quantidade no estoque. O incremento sempre será de uma unidade. Trate o caso do ID não ser encontrado;
10. **Diminuir quantidade de Produto do Estoque:** liste os produtos e peça pro usuário o ID do produto que ele deseja diminuir a quantidade no estoque. A diminuição sempre será de uma unidade. Trate o caso do ID não ser encontrado;
11. **Sair:** salvar os dados e finalizar o programa.

c) Telas

Sigam a mesma ordem do menu principal fornecida em "Operações do Menu Principal". Fiquem a vontade para os demais submenus / solicitações do usuário necessários.

d) Observações:

1. A app deve armazenar o Estoque em um arquivo usando o formato binário (Streams em Java), de modo que ao finalizar e iniciar a app novamente o Estoque permaneça com os produtos cadastrados anteriormente. Ela também deve ser capaz de armazenar o Estoque em memória, sendo possível alterar entre uma implementação e outra, exigindo alteração mínima de código (usar Interface?);
2. Não precisa validar se o dia realmente pode fazer parte do mês escolhido pelo usuário na vida real. Por exemplo, se o usuário informou o dia 31, do mês 1 (Janeiro), do ano de 2018, não precisa lançar erro e o sistema deve aceitar esta data;
3. O identificador do Produto deve ser atribuído no momento da adição dele no Estoque. A geração do ID não deve ser de responsabilidade do Estoque. Forneça pelo menos duas implementações de geradores de ID diferentes. Faça de modo que seja fácil alterar entre uma implementação e outra, exigindo alteração mínima de código (usar Interface?);
4. Não faz sentido adicionar / instanciar um Produto genérico (usar Classe Abstrata?).

e) Regras de negócio de validação / tratamento de erro:

1. O sistema deve tratar o caso quando usuário tentar remover um produto que não existe (i.e. ID não encontrado no estoque);
2. Não deve ser permitido adicionar no Estoque um Produto cuja data de validade tenha o dia fora do intervalo entre 1 e 31 (não precisa validar a quantidade de dias específico que cada mês de fato tem, ou seja, o sistema deve permitir adicionar um produto cuja data de validade seja, por exemplo, dia 31 de janeiro de 2019);
3. Não deve ser permitido adicionar no Estoque um Produto cuja data de validade tenha o mês fora do intervalo entre 1 e 12;
4. Não deve ser permitido diminuir o estoque de um determinado produto caso ele já esteja zerado;
5. Não deve ser permitido aumentar o estoque de um determinado produto fazendo com que ele fique acima de 5. O espaço disponível para estocar os produtos é pequeno;