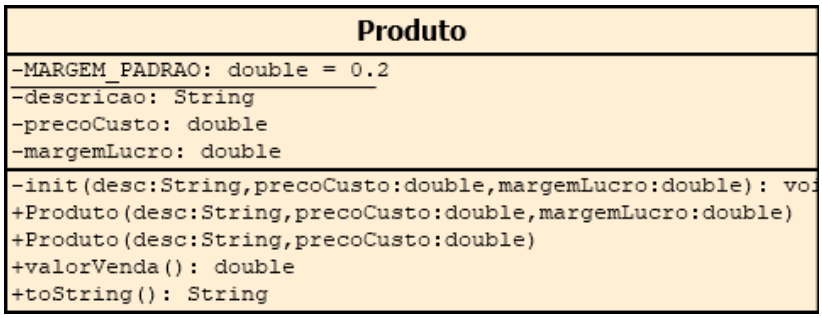
**Pontifícia Universidade Católica de Minas Gerais**
**Bacharelado em Engenharia de *Software* – *Campus* Lourdes**
**Algoritmos e Estruturas de Dados II (Laboratório)**
**Profs. Eveline Alonso Veloso e João Caram – Semestre 1/2025**
**Oficina: Revisão e Nivelamento (Java, Classes, POO e Polimorfismo)**

Este documento é um guia da atividade desenvolvida em conjunto no laboratório de AEDs II. Nesta atividade relembramos e construímos um código simples envolvendo os conceitos de modularidade, orientação por objetos e polimorfismo em Java. A atividade é individual e seu conteúdo servirá como base para atividades avaliativas futuras. Portanto, atenção ao que será desenvolvido em aula e procure fazer sua parte na tarefa, enviando as atualizações para o repositório da atividade no GitHub.

**Tema**: Produtos

*Um pequeno comércio vende produtos de tipos diversos. Cada produto é cadastrado com sua descrição, o preço de custo para a loja e uma margem de lucro definida pelo comerciante. A margem de lucro padrão adotada é de 20%, porém qualquer valor diferente deste pode ser informado no momento do cadastro do produto. Quando a venda é realizada, este produto deve responder o preço a ser cobrado a partir do preço de custo e da margem de lucro.*

O diagrama de classe UML ao lado mostra o projeto inicial para uma classe ***"Produto"*** com os requisitos descritos.

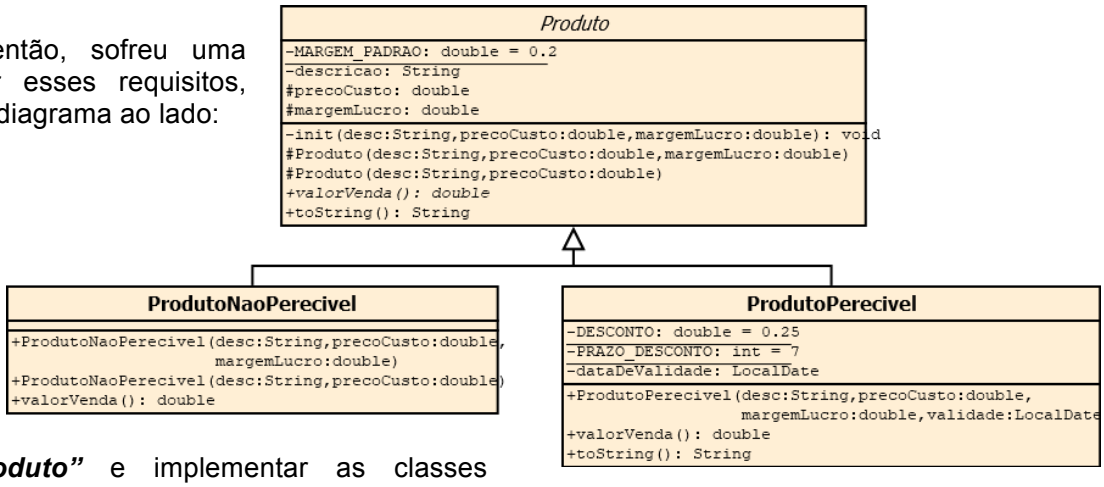



**Tarefa 0:**
Observar a classe ***"Produto"*** implementada de acordo com o modelo proposto. Analisar também o código JUnit contendo um teste simples para esta classe.

---

A loja foi notificada de que, para vender gêneros alimentícios, precisa registrar a data de validade do produto. Este produto não pode ser cadastrado com uma data de validade anterior ao dia atual. Também não pode ser solicitado seu valor de venda (que representa uma venda) se estiver fora da data de validade. Para incentivar a venda de produtos que estejam perto do final do prazo de validade, a loja decidiu conceder um desconto de 25% no preço de venda, caso o vencimento esteja com prazo de 7 dias ou menos a partir de hoje.

O modelo de classes, então, sofreu uma evolução para contemplar esses requisitos, conforme pode ser visto no diagrama ao lado:




**Tarefa 1:**
Refatorar a classe ***"Produto"*** e implementar as classes ***"ProdutoNaoPerecivel"*** e ***"ProdutoPerecivel"***, contemplando corretamente os requisitos por meio do uso de polimorfismo com herança.

**Instruções e observações:**

- O projeto deve estar hospedado na tarefa correspondente do GitHub Classroom. Endereço para aceitar a tarefa: https://classroom.github.com/a/pEVEnAgS
- As atividades pontuadas da disciplina podem depender direta ou indiretamente dos códigos desenvolvidos nas aulas. Portanto, é essencial o comprometimento no acompanhamento das atividades semanais.
- Para a correção das atividades pontuadas, serão considerados todos os *commits/pushes* realizados ao longo das semanas, não somente o último com a resposta final do exercício.