

Universidade Federal de Uberlândia  
Faculdade de Computação

**Disciplina:** Programação Orientada a Objetos II – BSI

**Professor:** Fabiano Azevedo Dorça

**Valor:** 25,0 pontos

**[Apresente diagrama de classes e código fonte da solução para as seguintes questões.]**

**[Atividade avaliativa INDIVIDUAL]**

**[Cópias serão anuladas]**

**Questão 01)** Desenvolva um aplicativo que controla uma máquina de venda automática de refrigerantes. Cada moeda inserida segue passando por uma fila de manipuladores. Ao encontrar um manipulador (fenda) com diâmetro compatível, a moeda é retida e o valor da moeda é informado ao usuário. Se a moeda não pode ser reconhecida por nenhum manipulador, deve ser rejeitada, e uma mensagem deve ser emitida. Utilize o padrão cadeia de responsabilidades para resolver o problema, considerando moedas de R\$0,01 (1 cm); R\$0,05 (1,2 cm); R\$0,10 (1,4 cm); R\$0,50 (1,6 cm) e R\$1,00 (1,8 cm). Mostre o funcionamento da cadeia em um método `main()`. (VALOR: 8,0 pontos).

**Questão 02)** Em um sistema de vendas, um produto pode estar em 3 diferentes estados:

- **disponível:** quantidade > 10
- **estoqueCrítico:** quantidade entre 1 e 10
- **indisponível:** quantidade igual a zero [ESTADO INICIAL]

Um produto pode sofrer operações de compras (reposição de estoque) e vendas (abate de estoque). A cada operação sofrida, o estado do produto pode ser verificado e alterado caso necessário, de acordo com a nova quantidade em estoque.

Um produto não pode ficar com estoque negativo (estado inexistente).

Estando em estoque crítico, a operação de venda deve emitir um alerta para que o gerente reponha o estoque (`println`).

Utilize o padrão State para modelar e implementar adequadamente a solução para esta situação.

Apresente um método `main()` para simular sua execução (8,0 pontos).

**Questão 03)** Em uma aplicação gráfica, deseja-se permitir a modelagem de formas geométricas (quadrado, círculo, retângulo, triângulo...) e decorá-las com diferentes efeitos (preenchimento, textura, luz, sombreado, relevo, wireframe, 3D, etc...). O método `desenhar()` de um objeto geométrico deve simular a renderização do objeto, retornando uma *string* com sua descrição (por exemplo: “círculo, luz, sombreado, 3D.”). Utilizando o **padrão Decorator**, crie esta aplicação. É suficiente apresentar na implementação apenas 1 concrete component e 2 concrete decorators. Crie um método `main()` para demonstrar a instanciação de um objeto decorado. (9,0 pontos)