

Prova 2

- **Instruções Gerais:** Leia com atenção os enunciados. A interpretação das questões é parte da prova. Implemente as soluções em Java, observando as convenções apresentadas em aula.

1. Considere a classe:

```
public class AppProduto{
    public static void main(String args[]){
        String descricao = "Chocolate 200g";
        double precoDeCusto = 2.70;
        int estoque = 300;
        Produto produto1 = new Produto(descricao,preco);
        produto1.setPrecoDeCusto(produto1.getPrecoDeCusto()+1);
        produto1.acrescentaEstoque(estoque);
        System.out.println("Dados do Produto " + produto1 );
        produto1.retiraEstoque(50);
        System.out.println("Total em estoque " + produto1.getEstoque());
        System.out.println("Preco de venda " + produto1.calcularPreco());
    }
}
```

Considere também a descrição dada a seguir: Todo o objeto da classe **Produto** tem descrição, preço (de custo) e quantidade em estoque. A descrição do produto não pode ser nula e o preço unitário de custo deve iniciar diferente de zero (mínimo R\$1,00). O estoque, por outro lado, pode iniciar vazio. Além dos métodos básicos que aparecem explicitamente na classe dada, implemente um método “calcularPreco”. Esse método calcula o preço unitário de venda do produto considerando a tabela abaixo. Implemente ainda o método “acrescentaEstoque”, que atualiza o estoque, adicionando ao mesmo o valor informado com parâmetro. Já o método “retiraEstoque” faz o oposto, diminui do estoque (se possível) o valor informado. Implemente os métodos pedidos de forma que a classe AppProduto execute corretamente. **(3,5 pts)**

preço unitário de custo	preço de venda
valor abaixo de R\$ 10,00	lucro de 70%
de R\$ 10,00 a menos de R\$ 30,00	lucro de 50%
de R\$ 30,00 a menos de R\$ 50,00	lucro de 40%
valor acima ou igual a R\$ 50,00	lucro de 30%

2. Complemente a classe **AppProduto**, incluindo um segundo objeto Produto com dados informados pelo usuário, incluindo o estoque. Após a instanciación, escreva também o estado atual do objeto na tela. **(1,0 pt)**

3. Implemente um **método de classe recursivo** que indica quantos espaços em branco uma String qualquer possui. **(2,5 pts)**
4. Implemente uma classe com um **método de classe** que retorna a soma os n primeiros termos da série dada a seguir (use repetição): $1 + \frac{3}{2!} + \frac{5}{4!} + \frac{7}{6!} + \dots$. A seguir, crie uma outra classe e, no método main dessa nova classe, inclua um exemplo de chamada do método criado. **(3,0 pts)**