



GOVERNO DO ESTADO DO PIAUÍ
UNIVERSIDADE ESTADUAL DO PIAUÍ
CAMPUS PROFESSOR ANTONIO GEOVANNE ALVES DE SOUSA
COORDENAÇÃO DO CURSO DE CIÊNCIA DA COMPUTAÇÃO



LISTA DE EXERCÍCIOS - 2

01. Uma classe *Lampada* com atributo *ligada* (tipo *boolean*) e métodos *liga()* e *desliga()* que nada retornam. O método *liga* torna o atributo *ligada* *true* e o método *desliga* torna o atributo *ligada* *false*. Crie também o método *observa()* que retorna a *String* “ligada” se a lâmpada estiver ligada e “desligada” se a lâmpada estiver desligada.
02. Acrescente um método *main* à classe *Lampada* da questão 01. O método *main* deve:
- Instanciar 2 objetos do tipo *Lampada* (*lampa1* e *lampa2*)
 - Ligar o objeto *lampa1* e desligar o objeto *lampa2*.
 - Usar o método *observa()* para mostrar se os objetos *lampa1* e *lampa2* estão “ligados” ou “desligados”.
03. Escreva uma classe *Quadrado* com atributo *lado* do tipo *double*. A classe deve ter um construtor que recebe como parâmetro o lado do quadrado. Deve também ter os métodos *area()* e *perimetro()* que retornam respectivamente a área e o perímetro do quadrado, cujas fórmulas são as seguintes:
- $area = lado^2$
 - $perimetro = 4 \times lado$
04. Escreva uma classe *UsaQuadrado* cujo método *main* instancia os 3 objetos a seguir:

q1	q2	q3
lado:	lado:	lado:
2	4	5

Observe que os 3 objetos são instâncias da classe *Quadrado* criada na questão anterior. Após instanciar os 3 objetos, o método *main* ainda deve mostrar a área e o perímetro dos 3 quadrados instanciados.

05. Escreva uma classe em Java chamada *Estoque*. Ela deverá possuir:
- Os atributos *nome* (*String*), *qtdAtual* (*int*) e *qtdMinima* (*int*).
 - Um construtor sem parâmetros e um outro contendo os parâmetros *nome*, *qtdAtual*, e *qtdMinima*.
 - os métodos com as seguintes assinaturas:
 - void mudarNome(String nome)*
 - void mudarQtdMinima(int qtdMinima)*
 - void repor(int qtd)*
 - void darBaixa(int qtd)*

- v. String mostra()
- vi. boolean precisaRepor()

06. Os atributos qtdAtual e qtdMinima jamais poderão ser negativos. O método repor aumenta qtdAtual de acordo com o parâmetro qtd. O método darBaixa diminui qtdAtual de acordo com o parâmetro qtd. O método mostra() retorna uma String contendo o nome do produto, sua quantidade mínima, sua quantidade atual. O método precisaRepor retorna true caso a quantidade atual esteja menor ou igual à quantidade mínima e false, caso contrário.

07. Escreva uma classe UsaEstoque cujo método main instancia os 3 objetos a seguir:

estoque1 nome: Impressora Jato de Tinta qtdAtual: 13 qtdMinima: 6	estoque2 nome: Monitor LCD 17 polegadas qtdAtual: 11 qtdMinima: 13	estoque3 nome: Mouse Ótico qtdAtual: 6 qtdMinima: 2
--	---	--

Depois disso, execute as seguintes operações na seguinte ordem:

- a. a) Dar baixa em 5 unidades de estoque1.
- b. b) Fazer a reposição de 7 unidades de estoque2.
- c. c) Dar baixa em 4 unidades de estoque3.
- d. d) Exibir a saída do método precisaRepor dos 3 objetos.
- e. Exibir a saída do método mostra para apresentar as informações sobre os 3 objetos.

08. Implemente uma classe Java que represente uma conta Conta Bancária, digamos Conta, que carrega as características de uma conta real e tudo o que ela faz. Ela deve ter o nome do titular (String), o número (int), a agência (String), o saldo (double) e uma data de abertura (String). Além disso, ela deve fazer as seguintes ações: saca, para retirar um valor do saldo; deposita, para adicionar um valor ao saldo; calculaRendimento, para devolver o rendimento mensal dessa conta. Teste-a, usando uma outra classe que tenha o main. A classe Conta deve conter pelo menos os seguintes métodos:

- a. saca que recebe um valor como parâmetro e retira esse valor do saldo da conta
- b. deposita que recebe um valor como parâmetro e adiciona esse valor ao saldo da conta
- c. calculaRendimento que não recebe parâmetro algum e devolve o valor do saldo multiplicado por 0.1

09. Crie um método recuperaDadosParaImpressao(), que não recebe parâmetro mas devolve o texto com todas as informações da nossa conta para efetuarmos a impressão. Dessa maneira, você não precisa ficar copiando e colando um monte de System.out.println() para cada mudança e teste que fizer com cada um de seus funcionários, você simplesmente vai fazer:

```
Conta c1 = new Conta();
// brincadeiras com c1....
System.out.println(c1.recuperaDadosParaImpressao());
```

Bom treino!