3ª Prova de POO

Instruções: Entregar os arquivos ".java" com os fontes via e-mail para destinatário **sbacala@gmail.com**, assunto: "**prova 3 de POO – 2019-1 nome do aluno**". Não se esqueça de incluir seu nome no interior do e-mail. Entregas diferentes dessa definição serão desconsideradas. NÃO COMPACTAR. Anexar os arquivos java diretamente.

1 – (1,0) Crie uma classe **Produto**, com atributos descrição, preço unitário e quantidade em estoque. Adicione um atributo que represente o código do produto com 6 caracteres (5+DV) (trata-se de valor obrigatório, portanto deve ser pedido no construtor).

2 – (2,0) Fazer uma rotina para validar código do produto usando o algoritmo DV para base 10. Incluí-la no método *setCodigoBarra(cod)*, que atualiza o código do produto apenas se o DV for válido. Se não for válido, levantar uma exceção.

Módulo 10
Conforme esquema abaixo, cada dígito do número, começando da direita para a esquerda (menos significativo para o mais significativo) é multiplicado, na ordem, por 2, depois 1, depois 2, depois 1 e assim sucessivamente. Em vez de ser feito o somatório das multiplicações, será feito o somatório dos dígitos das multiplicações (se uma multiplicação der 12, por exemplo, será somado 1 + 2 = 3). O somatório será dividido por 10 e se o resto (módulo 10) for diferente de zero, o dígito será 10 menos este valor. Exemplo: 61533-6

```
+---+---+---+---+---+ +---+
| 6 | 1 | 5 | 3 | 3 | - | 6 |
+---+---+---+---+---+ +---+
  |   |   |   |   |
 x2  x1  x2  x1  x2
  |   |   |   |   |
 12   1  10   3   6  => 1+2+1+1+0+3+6 = 14  => (14 / 10) = 1, resto 4 => DV = (10 - 4) = 6
```

3 – (1,0) Sobrescreva o método *equals()* retornando *true* se dois produtos possuírem o mesmo código de produto.

4 – (2,0) Crie uma classe Loja com o método *main()* que adicione 5 produtos diferentes a um array e os imprima usando sobrecarga de *toString()*. Implemente um processo de busca que, dados um produto e um array de produtos, indique em que posição de encontra o produto especificado ou exiba uma mensagem de erro (*JOptionPane*) informando que o mesmo não fora encontrado.

5 – (1,0) No *main()*, após a impressão dos produtos, escolha um dos 5 produtos e crie duas novas instâncias idênticas a ele: uma com mesmo código de produto, e outra com código diferente. Efetue a busca usando as duas instâncias e exiba o resultado.

6 – (1,5) Faça com que Produto implemente a interface *Comparable (int compareTo()* – resultado menor que zero indica que *this* vêm antes, maior que zero indica que *this* vem depois, e zero indica que são iguais) e implemente a comparação por nome. Ao final, no *main()*, ordene o vetor utilizando o método *java.util.Arrays.sort()* e imprima-o novamente.

7 – (1,5) Comente o código do exercício 5 e implemente a comparação para ordenar por preço decrescente.