Universidade da Beira Interior

Programação Orientada a Objectos

Cursos: 1º ciclo:
Eng.ª Informática, Matemática e Aplicações; 2º ciclo em EEC
Frequência, 2021/01/05

---

1 – Suponha a classe Produto listada abaixo:

a) Construa o código para os métodos getNome e setNome.

```
public String getNome() {
      return nome;
   }
public void setNome(String nome) {
      this.nome = nome;
   }
```

b) Construa o código para o método toString.

```
@Override
   public String toString() {
      return "Produto{" + "cod=" + cod + ", nome=" + nome + ", custo=" + custo + '}';
   }
```

c) Construa o código do método equals.

```
public boolean equals(Object obj) {
     if (obj != null && this.getClass()== obj.getClass()){
        Produto p = (Produto)obj;
        return this.cod == p.cod && this.nome.equals(p.nome) && this.custo== p.custo;
     }
     return false;
   }
```

d) Construa o código do método clone.

```
public Object clone(){
      Produto p = new Produto(this.cod, this.nome);
      p.custo = this.custo;
      return p;
   }
```

e)

```
public void subirCusto (double perc) throws ValorException{
      if (perc >0 && perc < 100)
         this.custo = this.custo + this.custo*perc /100.0;
      else
         throw new ValorException ("O valor" +  perc + " deve ser maior que 0 e menor que 100");
   }
```

f) Construa a classe de exceção ValorException.

```
public class ValorException extends Exception{
   public  ValorException(){ super();}
   public  ValorException(String s){ super(s);}
}
```

2 – Suponha o seguinte programa:

a) Qual o output deste programa?
false
true
false
true
Produto{cod=1, nome=Produto1, custo=700.0} // **ver toString**

b) Indique graficamente que variáveis que existem neste programa e quais os seus valores.

cod	nome	custo
1	Produto1	100.0

<= p1

cod	nome	custo
1	Produto2	700.0

<= p2
<= p3

c) Escreva o código necessário para aumentar 20% o custo do produto p2.

```
try{
    p2.subirCusto(20);
}
catch (ValorException e){
     System.out.println(e.getMessage());
}
```

d) Construa um método de classe (public static) que receba como parâmetro um objeto do tipo ArrayList<Produto> , e determine qual o produto mais barato dessa lista de produtos. O método deve devolver como resultado o nome (ou nomes no caso de serem vários) dos produtos que têm o preço mais baixo.

```
public static ArrayList<String> custoMaisBaixo(ArrayList<Produto> lista) {
      ArrayList<String> nomes = new ArrayList<String>();
      if (lista.size() > 0) {
         double min = lista.get(0).getCusto();
         nomes.add(lista.get(0).getNome());
         for (int i = 1; i < lista.size(); i++) {
            if (lista.get(i).getCusto() < min) {
               min = lista.get(i).getCusto();
               nomes.clear();
               nomes.add(lista.get(i).getNome());
            } else if (lista.get(i).getCusto() == min) {
               nomes.add(lista.get(i).getNome());
            }
         }
      }
      return nomes;
   }
```

3 – Um produto de vestuário (ProdutoVest), é um Produto que tem como atributos adicionais o género a que se destina (masculino ou feminino) e o tamanho.
a) Defina o cabeçalho e os atributos da classe ProdutoVest.

```
public class ProdutoVest extends Produto{
   private String genero;
   private int tam;
```

b) Defina um construtor que receba como parâmetro um objeto do tipo Produto.

```
public ProdutoVest (Produto p){
     super (p.getCod(), p.getNome());
     super.setCusto(p.getCusto());
     genero = "";
     tam = 0;
   }
```

4 – Uma nota de encomenda tem um número de encomenda (número sequencial atribuído automaticamente) tem também um número de contribuinte e uma lista de produtos (objeto do tipo ArrayList<Produto>).
a) Defina o cabeçalho e os atributos da classe NotaEncomenda.

```
public class NotaEnc {
   private static int ultimo = 0;
   private int numNE;
   private int nif;
   private ArrayList<Produto> lista;
```

b) Defina um construtor para a classe NotaEncomenda, que tenha como parâmetro, o número de contribuinte.

```
public NotaEnc(int nif) {
      ultimo++;
      numNE = ultimo;
      this.nif = nif;
      lista = new ArrayList<Produto>();
   }
```

c) Para a classe NotaEncomenda defina um método que permita adicionar um objeto do tipo Produto à nota de encomenda. Se o produto já fizer parte da encomenda, o método não deve fazer nada.

```
public void adicionaProd(Produto p) {
      if (!lista.contains(p)) {
         lista.add(p);
      }
   }
```

5 – Considerando o programa do exercício 2, adicione um bloco de código que:
a) Declare e instancie um objeto do tipo NotaEncomenda para o cliente com número de contribuinte 123456789:
b) Adicione os produtos p1 e p2 à nota de encomenda;
c) Construa um objeto do tipo ProdutoVest com o código=123; nome = "calça" e custo=50. A calça é de homem e tem tamanho 42. Suponha que os getters e setters para a classe ProdutoVest foram implementados.
d) Adicione à nota de encomenda o produto que criou na alínea anterior

```
   a)  NotaEnc ne = new NotaEnc (123456789);
   b)
       ne.adicionaProd(p1);
       ne.adicionaProd(p2);
   c)
```

```
        ProdutoVest pv = new ProdutoVest(new Produto(123, "calça"));
        pv.setCusto(50);
        pv.setGenero("Masculino");
        pv.setTam(42);
    d)
      ne.adicionaProd(pv);
```

6 – Construa para a classe de teste um método de classe (public static) que receba como parâmetro um objeto do tipo ArrayList<NotaEncomenda>. O método deve contar quantos produtos do tipo ProdutoVest existem na lista de encomendas. Suponha que os getters e setters para a classe NotaEncomenda foram implementados.

```
public static  int contaVestuario(ArrayList<NotaEnc> lista){
      int conta = 0;
      for (int i =0; i<lista.size (); i++){
         NotaEnc ne = lista.get(i);
         for (int j = 0; j < ne.getLista().size(); j++) {
            if (ne.getLista().get(j) instanceof ProdutoVest)
                conta ++;
         }
      }
      return conta;
   }
```

7 – Um programa em Java para ser executado é compilado ou interpretado? Justifique a sua resposta.

8 – Indique quais as principais características da orientação a objectos e como são implementadas na linguagem Java?

9 – O que é uma interface em java e para que serve?

10 – O que significa a cláusula "Implements" e como é usada?

1 – a) 0.2
b) 0.3
c) 0.5
d) 0.5
e) 0.75
f) 0.25 2.5

2 –
a) 0.5
b) 0.5
c) 0.5
d) 1.75 3.25

3 –
a) 0.25
b) 0.75 1.0

4 –
a) 0.25
b) 0.5
c) 0.5 1.25

5 – 1.0

6 - 2.0

7 –	0.25
8 –	0.75
9 –	0.25
10 -	0.75