

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

Sem consulta e sem telemóvel
Duração: 90 minutos, 13.0 valores

1 – Suponha a classe Produto listada abaixo:

```
public class Produto {
    private int cod;
    private String nome;
    private double custo;

    public Produto(int cod, String nome) {
        this.cod = cod;
        this.nome = nome;
        this.custo = 0.0;
    }
    public int getCod() { ... }
    public void setCod(int cod) { ... }
    public String getNome() { ... }
    public void setNome(String nome) { ... }
    public double getCusto() { ... }
    public void setCusto(double custo) { ... }
    @Override
    public String toString() { ... }
    @Override
    public boolean equals(Object obj) { ... }
    public Object clone(){ ...}
}
```

- a) Construa o código para os métodos getNome e setNome.
 - b) Construa o código para o método toString.
 - c) Construa o código do método equals.
 - d) Construa o código do método clone.
 - e) Construa um método de permita subir o custo de um produto. O método deve receber como parâmetro o valor com a percentagem a subir. Se o valor dado como parâmetro não estiver entre 0 e 100, o método deverá gerar uma exceção do tipo ValorException, com a seguinte mensagem de erro associada: “ o valor X deve ser maior que 0 e menor que 100”, onde X representa o valor dado como parâmetro. O método deve lançar a exceção para ser tratada no programa que invocar o método.
 - f) Construa a classe de exceção ValorException.
-

2 – Suponha o seguinte programa:

```
public static void main(String[] args) {  
    Produto p1, p2, p3;  
    p1 = new Produto(1, "Produto1");  
    p2 = (Produto)p1.clone();  
    p3 = p2;  
    p1.setCusto(100);  
    p2.setCusto(500);  
    p3.setCusto(700);  
    System.out.println (p1==p2);  
    System.out.println (p2==p3);  
    System.out.println (p1.equals(p2));  
    System.out.println (p2.equals(p3));  
    System.out.println(p2);  
}
```

- a) Qual o output deste programa?
- b) Indique graficamente que variáveis que existem neste programa e quais os seus valores.
- c) Para este programa escreva o código necessário para aumentar 20% o custo do produto p2.
- 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.

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

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

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;

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.

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?

Classe java.util.ArrayList:

```
ArrayList() // construtor vazio, dimensão inicial zero.  
boolean add(Object element)  
// adiciona o elemento especificado ao final da lista  
void add( int index, Object obj)  
//insere o elemento especificado na posição index  
Object remove(int index )//remove o elemento da posiçã index  
boolean remove( Object o)  
//remove a primeira ocorrência do objecto dado como parâmetro  
Object set (int position, Object obj )  
// substitui o elemento da posição index pelo elemento dado  
Object get (int position)//devolve o elemento da posição index  
void clear() // remove todos os elementos da lista  
Object clone() // devolve uma cópia da lista  
boolean contains(Object element)  
// devolve true se a lista contém o elemento especificado  
boolean equals ( Object obj)  
// permite comparar duas listas  
int indexOf(Object element)  
// procura o índice da 1ª ocorrência de elemento  
boolean isEmpty() // verifica se a lista não tem componentes  
int size() // devolve a dimensão actual  
String toString ()
```