

Introdução à Programação (IF-669)

2º Exercício Escolar

Adriano Sarmiento
Centro de Informática-UFPE
03 de dezembro de 2008

1ª Questão

Considere que exista uma classe Venda que possua dois métodos: Date getDataVenda(); e abstract double getValorVenda() e que existam as classes VendaAVista e VendaAPrazo que são especializações de Venda

Agora considere a classe ColecaoOrdenadaDataVenda, que representa uma coleção de vendas ordenadas pela data de forma decrescente, e a interface IteratorOrdenadaVenda:

```
/*
 * Classe que representa uma coleção de Vendas
 ordenada da maior data para a menor data
 */
public class ColecaoOrdenadaDataVenda {
    ...
    public IteratorOrdenadoVenda getIterator() {...}
    ...
}
```

```
public interface IteratorOrdenadaVenda {
    public Venda next();
    public boolean hasNext();
}
```

Finalmente, considere a classe Vendedor abaixo que possui uma coleção ordenada de Vendas associado a ela:

```
public class Vendedor {

    private String nome;
    private ColecaoOrdenadaDataVenda colecao;
    private double salarioBase;

    public Vendedor(String nome, double salario){
        this.nome = nome;
        this.salarioBase = salario;
        colecao = new ColecaoOrdenadaDataVenda();
    }

    public double calcularSalario(){
        return salarioBase;
    }

    public ColecaoOrdenadaDataVenda getColecao() {
        return colecao;
    }
    ...
}
```

- (a) Defina o método getValorMaiorVenda que retorna o valor da maior venda efetuada pelo vendedor (2,0p)

- (b) Defina o método `ultimaVenda` que retorna a última venda realizada pelo vendedor. (2,0p)
- (c) Defina a classe `VendedorTemporario` que é uma especialização de `Vendedor`. Esta classe deve armazenar uma comissão. O método `calcularSalario` deve levar em conta o salário base, o valor total de todas as vendas A VISTA e a comissão. Por exemplo, se a comissão for de 0,01, o método deve retornar a soma do salário base com 1% do valor total de vendas A VISTA do vendedor. Não se esqueça de definir o construtor e o método `getComissao()` (2,0p).

2ª Questão

Examine o trecho de programa abaixo escrito em Java:

```
Vendedor v1 = new VendedorTemporario("Maria",200,0.01);
Vendedor v2 = new Vendedor("Joao",400);
//Codigo para associar aos vendedores Maria e Joao uma coleção
//de vendas cujo total de vendas a vista é 1000
...
```

Para cada linha de comando abaixo adicionado a este trecho de programa, descreva o que ocorrerá, justificando a resposta. No caso de erro, escreva qual é o tipo de erro (compilação ou execução) e explique o porquê do erro (2,0p).

- a) `System.out.println("Salario Maria: " + v1.calcularSalario());`
- b) `System.out.println("Comissao Maria: " + v1.getComissao());`
- c) `System.out.println("Salario Maria: " + ((Vendedor)v1).calcularSalario());`
- d) `System.out.println("Comissao Joao: " + ((VendedorTemporario)v2).getComissao());`

3ª Questão

Discuta benefícios e problemas de concorrência. Qual é o suporte que Java dá para o programador que o permite utilizar a concorrência de forma segura? (2,0 p)