TÉCNICO LISBOA

					1/7

Programação com Objectos

1º Teste • 2011/2012 • 1º Semestre

19 de Novembro de 2012, 08:00 (90 minutos)

Nome: _____

Primeira Parte (7 valores)

PERGUNTA	NOTA
1.1	
1.2	
1.3	
1.4	

1ª PARTE	

Segunda Parte (3 valores)

PERGUNTA	RESPOSTA
2.1	
2.2	
2.3	
2.4	
2.5	

2ª PARTE	

certas 0.6 / erradas -0.15 / ausentes 0

Para alterar uma resposta: riscar a resposta antiga e escrever à frente a nova resposta. Considera-se ausência de resposta a apresentação de várias repostas a uma pergunta, mesmo que uma delas esteja certa.

Consideram-se não respondidas as perguntas cujas respostas não estejam assinaladas nesta tabela.

1.1. (3.0) Considere o seguinte domínio:

Uma empresa de mobiliário produz vários tipos de móvel: peças individuais, tais como mesas, cadeiras, etc.; e conjuntos de peças relacionadas, tais como as que fazem parte da mobília de uma sala de jantar. Cada conjunto pode ser composto por peças individuais e/ou por subconjuntos (e.g., um conjunto de sofás pode fazer parte do mobiliário de uma sala de estar, juntamente com uma mesa). Todos os tipos de peça têm uma referência única no catálogo da empresa (cadeia de caracteres). Uma vez que a empresa é multinacional, existe ainda um código numérico correspondente ao país de origem de cada peça, assim como um nome comercial, para efeitos de publicidade.

Para cada peça, a empresa mantém o número de existências em stock, assim como o preço de catálogo. As peças podem ainda ter um desconto associado. Estes factores condicionam, juntamente com o tipo de cliente, o preço final da peça.

Os clientes da empresa são identificados pelo nome (cadeia de caracteres) e possuem ainda um endereço de correio electrónico e um número de telefone. Para cada cliente, a empresa mantém um registo das vendas realizadas e o desconto a aplicar nas vendas a esse cliente. Os clientes inicialmente não usufruem de descontos, mas à medida que vão fazendo compras, o processo de fidelização vai introduzindo benefícios para clientes frequentes ou que gastem quantias significativas. Assim, para clientes que façam pelo menos cinco compras (independentemente do valor gasto), o desconto passa para 30%; para clientes que gastem pelo menos 10000 unidades monetárias (independentemente do número de compras), o desconto passa para 50%; se o cliente gastar mais de 100000 (independentemente de outros factores), o desconto passa para 70%. A empresa permite aos seus clientes fazer devoluções com restituição parcial do valor gasto: assim, para clientes que tenham desconto 0%, a devolução é de 10%; se o cliente tiver 30% ou 50% de desconto, o valor devolvido é de 20%; para clientes que tenham 70% de desconto, o valor devolvido é de 50%, mas essa parcela é abatida ao valor vendido que, se descer abaixo de 100000, faz com que o cliente passe a usufruir apenas de 50% de desconto.

A empresa permite obter a lista de vendas, a lista de clientes, a lista de 10 melhores clientes, e o total de vendas. É possível saber quantas unidades foram vendidas de cada peça, assim como o valor facturado. Para cada cliente, é possível saber o valor pago e as peças que comprou e que devolveu, assim como o nível de desconto a que tem direito.

Desenhe o diagrama de classes UML correspondente ao domínio apresentado. Represente as classes (seus nomes, métodos e atributos). Indique também as relações de herança, associação e agregação.

1.1. (espaço adicional)

1.2. (1.0 val.) Implemente, em Java, as classes **Tabuleiro** e **Peça**. A classe **Tabuleiro** guarda um número variável de peças e tem o método **desenhaPeças()** que é responsável por desenhar todas as peças guardadas. A classe **Peça** tem dois atributos inteiros que representam as suas coordenadas no tabuleiro, um atributo que representa a côr da peça (cadeia de caracteres) e dois métodos: o método **move()** e o método **devolveCor()**. Existem dois tipos de peças: **Cabo** e **Sargento**. Cada tipo de peça move-se de forma distinta: as peças do tipo **Cabo** avançam uma unidade em cada uma das coordenadas enquanto que as peças do tipo **Sargento** avançam 3 unidades em cada uma das coordenadas. Para simplificar o seu trabalho considere que o tabuleiro não tem limites.

Se necessário pode implementar outras classes e/ou métodos.

1.2. (espaço adicional)

1.3. (1.5 val.) Diga em que consistem e como se expressam os conceitos de sobrecarregamento (*overloading*) e redefinição (*overriding*) de métodos nas linguagens de programação com objectos. Dê exemplos práticos, se possível, relacionados com o projecto.

1.4. (1.5 val.) Explique em consiste o mecanismo de abstracção presente em linguagens como o Java e o C++. Qual é a sua relação com o polimorfismo. Que consequências têm estes dois aspectos na produção de código?

2.1. Considere o diagrama UML da figura 1 (à direita). Qual das seguintes afirmações está correcta?

(a) a implementação de `doThis` tem de estar necessariamente em `C`
(b) `doThis` não está definido para `B`
(c) `doIt` pode ser directamente invocado através de referências para `I`
(d) `I` fornece a implementação para métodos declarados por `A`
(e) nenhuma das anteriores está correcta

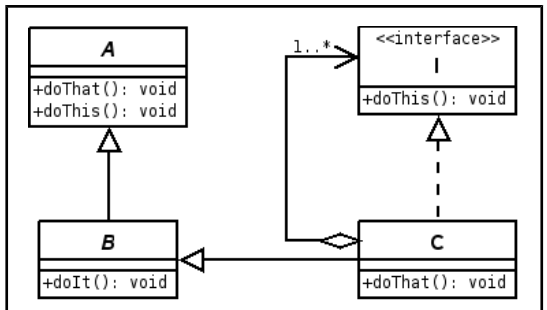

Figura 1
(todos os métodos estão implementados)

2.2. Em Java, qual das seguintes frases está correcta?

(a) Uma classe pode estender uma única classe
(b) Uma classe pode estender várias classes
(c) Uma classe só pode implementar uma única interface
(d) Uma classe não pode simultaneamente estender uma classe e implementar uma interface
(e) Uma classe não pode simultaneamente estender uma classe e implementar mais do que uma interface

2.3. Em Java, um método declarado `protected`...

(a) só pode ser acedido dentro da própria classe ou por classes derivadas
(b) não pode ser redefinido
(c) não pode chamar outros métodos que também não sejam `protected`
(d) pode ser acedido por classes da mesma *package*
(e) só pode ser usado por construtores

2.4. Supondo que está a fazer os imports correctos, qual das seguintes instruções não gera nem avisos nem erros de compilação?

(a) List<Integer> myList = new ArrayList<Integer>();
(b) List<Integer> myList = new List<Integer>();
(c) List<Object> myList = new ArrayList<Integer>();
(d) ArrayList<Integer> myList = new List<Integer>();
(e) Todas estão correctas

2.5. Quais são as possíveis saídas do seguinte programa em Java:

```
1.  public static void main(String[] args) {
2.    try {
3.      if (args.length == 0) throw new Exception();
4.    }
5.    catch (Exception e) {
6.      System.out.print("done ");
7.    }
8.    finally {
9.      System.out.println("finally ");
10.   }
11. }
```

(a) sempre "`done finally`"
(b) sempre "`finally`"
(c) sempre "`done`"
(d) ou "`finally`" ou "`done finally`"
(e) ou "`done`" ou "`finally`" ou "`done finally`"

NÃO ESQUECER DE PREENCHER GRELHA DE RESPOSTAS NA FOLHA DE ROSTO