

Universidade Católica de Pernambuco

Professor: Augusto César Oliveira

Disciplina: Programação III

Aluno(a): _____ data: ____/____/____

Observações:

* A prova deve ser feita na linguagem Java e deve ser entregue até o final da aula. A prova é individual e sem consulta; os entregáveis são os arquivos executáveis.

* Por padrão, todos os atributos devem possuir visibilidade privada e os métodos públicos. O construtor de cada classe também deve ser implementado, além dos gets e sets de cada atributo.

Questão 01. Crie as seguintes classes:

a. A classe **Cliente**, com os seguintes atributos (peso 1,0):

- i. nome – String;
- ii. idade – Inteiro;
- iii. cpf – String;
- iv. endereço – String;
- v. bomCliente – Boolean;
- vi. carrinhoDeCompras – Vetor de 100 posições do tipo **Item**.

b. A classe **Item**, com os seguintes atributos (peso 1,0):

- i. nome – String;
- ii. codigo – Inteiro;
- iii. valor – Double;
- iv. quantidadeEmEstoque – Inteiro;
- v. disponibilidade – Boolean.

Questão 02. Na classe **Cliente**, crie os seguintes métodos (peso 2,0):

- a. adicionar() – Recebe um item como parâmetro e adiciona ele no vetor de compras realizadas;
- b. cancelar() – Recebe um código de produto e remove ele do vetor em questão;
- c. finalizarCompra() – **Soma o valor de todos os itens** que estão no carrinho de compras; **reduz a quantidade de estoque de todos eles em (1)** e retorna **o gasto total da compra**.

Questão 03. Crie a classe **ClienteEspecial**. A classe deve herdar da classe **Cliente** e ter os seguintes atributos a mais (peso 4,0):

- a. pontos – Inteiro;
- b. saldoDeCompras – Double.

O método finalizarCompra() deve ser sobrescrito com os seguintes detalhes:

- a. O cliente recebe 5% de desconto nas compras. Este saldo (valor de desconto) deve ser somado ao saldo de compras já existente. Caso o saldo seja acima de 100 reais, o cliente ganhará 5 pontos e, logo em seguida, o saldo de compras deverá ser reduzido em 100;

- b. O método `finalizarCompra` também deverá sofrer uma sobrecarga. Sobrecarregue-o adicionando o número de pontos que cada **item** custa. Sendo assim, caso o cliente possua pontos disponíveis, ele não pagará nada pelo novo item, os pontos serão utilizados automaticamente para comprar o item em questão. Os itens comprados através dos pontos não devem entrar no gasto total das compras, ou seja, é uma compra única.

Questão 04. Crie a classe **Funcionario**, ela deve conter os seguintes atributos (peso 2,0):

- a. `setor` – String;
- b. `id` – Inteiro;
- c. `salario` – Double.

A classe deve herdar da classe **ClienteEspecial**, mas possui os seguintes métodos a mais:

- a. `renovarEstoque()` – O método deve receber um **inteiro** e o **item**, onde o estoque dele deve ser acrescentado.
- b. `bloquearItem()` – Recebe um **item** como parâmetro e deixa ele indisponível ou não.