



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS AGRÁRIAS
DEPARTAMENTO DE ENGENHARIA RURAL
Programação II

Descrição do problema

Uma empresa comercializa produtos diversos e mantém um cadastro destes produtos com seus respectivos códigos, nome, tipo e preço unitário, conforme exemplo da tabela 1.

Código do Produto	Tipo	Nome do Produto	Preço unitário
0001A	1	Pen drive 2 Gb	12,05
0002B	2	CD	1,50
0D04D	2	Blu-Ray	15,20

Tabela 1 – Exemplo de Tabela de Preços

A venda destes produtos é feita por meio de pedidos, que são emitidos para clientes da empresa. Cada pedido contém as seguintes informações:

- N° do Pedido;
- Código do Cliente;
- Condição de Pagamento;
- Código do Produto;
- Quantidade vendida do produto;
- Preço unitário do produto;
- Preço total do item;
- Valor total do pedido.

Sabe-se que um pedido pode ter no máximo 10 produtos, ou seja, é possível que existam pedidos com dois produtos (conforme exemplo apresentado na tabela 2), como também pedidos com 3, 4 até 10 produtos. Cada venda pode ser feita para apenas um cliente.

Nº do Pedido	001A	Código do Cliente	12	Condição de Pagamento	A vista
Código do Produto		Quantidade Vendida	Preço unitário	Preço total	
0001A		2	12,05	24,1	
0D04D		1	15,20	15,20	
Desconto				2,41	
Valor total				36,89	

Tabela 2 – Exemplo de um pedido com 2 produtos

A empresa mantém um cadastro de clientes, com seu código, nome, renda e tipo (Prata e Ouro).

a) Histórias de usuários:

1. O cliente chega ao balcão com os produtos que deseja comprar, o atendente confere os produtos e preenche a “notinha” informando os produtos pelo código, o cliente - caso ele seja cadastrado, e a condição de pagamento. Se o cliente for cadastrado, o atendente pode permitir uma venda a prazo. Vendas a prazo têm 5% de acréscimo no preço total;
2. Caso o cliente não deseja se cadastrar, as vendas só podem ser a vista e seu registro é feito para o cliente 001, com o nome fictício de “Venda sem cadastro”;



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS AGRÁRIAS
DEPARTAMENTO DE ENGENHARIA RURAL
Programação II

3. Os clientes podem se cadastrar na empresa e participar de 2 categorias (tipos) distintos. Clientes tipo Ouro são aqueles com renda superior a 1000 reais. Estes têm 10% de desconto no valor das suas faturas. Clientes prata são aqueles com renda entre 300 e 1000 reais e tem desconto de 5%. Os demais clientes cadastrados com renda inferior a 300 reais não tem classificação (tipo) e não possuem desconto;
4. Quando o cliente chega à empresa para pagar suas faturas (de vendas a prazo) o atendente procura pelos pedidos realizados (e assinados pelo cliente) e anota o valor total de cada pedido para calcular o valor total a pagar pelo cliente. De posse do valor total a ser pago o cliente realiza o pagamento e o atendente registra o valor total pago, a data do pagamento e o número dos pedidos que foram pagos;
5. Ao final de cada mês o gerente da empresa faz a baixa (realiza um pagamento "teórico") das vendas que foram registradas para o cliente 001.
6. Produtos do tipo 1 tem 10% de desconto para vendas a vista. Os demais produtos não têm qualquer tipo de desconto.

Histórico de Revisões

Data	Modificação	Feito
30/09/2010	Exercícios 1..9	
05/10/2010	Exercícios 10..12	
14/10/2010	Exercícios 13..16	
21/10/2010	Exercícios 17..21	

Exercícios

1. Modifique o código que modo que a venda a prazo não permita qualquer tipo de desconto;
2. Crie o atributo `desconto` (real) na classe `Venda` e refatore (modifique) o método `calculaTotal`, criando um método `calculaDesconto`;
3. Crie um breakpoint no método `main` na linha onde o fechamento da venda é executado e "passe por dentro do código", observando no painel (normalmente localizado na parte debaixo da IDE) os valores de cada atributo das instâncias;
4. Crie um método `adicionaItens`, que deve receber um array como parâmetro. Faça uso deste método para incluir outros itens (no `main`);
5. Crie uma venda à prazo;
6. Modifique o método `calculaTotalVenda` para usar o comando `DO`;

7. Remova todos os `sets` e `gets` desnecessários;
8. Crie uma classe chamada `Estoque` (classe de persistência) para guardar os produtos. Essa classe deve permitir que um produto seja encontrado pelo nome ou pelo código e que produtos com o mesmo nome não possam existir em estoque. Não deve também permitir a inclusão de produtos com o mesmo código. Implemente-a e faça uso da mesma;
9. Implementar o método `removeItem` na classe `Venda`;
10. Crie as classes `VendaPrazo` e `VendaVista`, utilizando os conceitos de herança, polimorfismo, sobrecarga e sobreposição nos métodos adequados, observando o enunciado do problema;
11. Limite Crédito de vendas à Prazo
O limite de crédito apresenta o percentual que um cliente pode comprar a prazo considerando sua renda, conforme tabela

Tipo	Limite
Outros	0
Prata	10%
Ouro	20%

Tabela 1 - Limite de Crédito para compras à prazo

Um cliente ouro que tenha uma renda de 1500 reais, pode ter um limite de crédito de 150 reais. É necessário saber o saldo utilizado (do limite permitido) por cada cliente, e não deve ser permitido que um cliente compre à prazo, acima do seu limite.

12. Faça uma modificação no programa de Vendas de modo que ele possibilite a entrada de dados pelo usuário, e crie os objetos a partir destes dados. Conforme vimos em aula, o uso da classe `Scanner` é bem simples, conforme código 1:

```
Scanner s = new Scanner(System.in);  
System.out.println("Digite um nome:");  
String nome = s.next();
```

Código 1 – Exemplo de uso da classe `Scanner`

O sistema deverá exibir um *menu* de opções para:

- Criar vendas (máximo de 10 vendas);
- Fazer o fechamento das vendas;
- Apresentar o saldo devedor de cada cliente;
- Apresentar os produtos e a quantidade em estoque;
- Sair do sistema.

13. Faça uma implementação que utilize campos de classes (estáticos);
14. Crie um método `toString` em todas as classes;



UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO DE CIÊNCIAS AGRÁRIAS
DEPARTAMENTO DE ENGENHARIA RURAL
Programação II

15. Avalie se há aplicabilidade para criar uma Fábrica de Instâncias no programa feito até o momento e faça a implementação;
16. Faça uso de delegação ou composição para criar a classe Quitação, com o código da classe Data (livro Rafael Santos);
17. Incluir o campo data da emissão da venda nas classes em que se fazem necessárias
18. Incluir o campo data do vencimento em vendas a prazo (vencimento = emissão + 30 dias)
19. Criar um relatório de vendas que deve listar todas as vendas feitas em um período compreendido entre a data inicial e uma data final. O relatório deve apresentar
 - a) Nome do cliente
 - b) Tipo da venda
 - c) Valor total
 - d) Valor de desconto ou acréscimo
 - e) Data da venda
 - f) Se a venda foi paga ou não
 - g) No final deve apresentar o total geral vendido no período, o total de desconto e acréscimos;
20. Criar uma opção onde o usuário possa fazer quitações de vendas a prazo para um cliente. O usuário deverá informar uma data e o sistema deve apresentar todas as vendas a prazo não pagas até aquela data. Após isso o usuário deverá registrar o pagamento

Exemplo

Vendas não pagas inferiores a 23/08/10

Cod	Cliente	Total	Vencimento
1	José da silva	R\$ 2.300,00	22/08/10
2	João almeida	R\$ 1.200,00	20/08/10

Seleciona quais vendas serão pagas e confirma o pagamento

21. O sistema deverá registrar as vendas selecionadas como pagas.