Centro Universitário Facens

Engenharia da Computação
Programação Orientada a Objetos

Prof. Renato Moraes Silva

Avaliação Continuada 2 (AC2)

Instruções gerais

1. Leia atentamente a descrição dos exercícios para garantir que você está executando o que foi compra.
2. O trabalho pode ser feito em grupo de até 3 integrantes. Somente um integrante do grupo deve enviar a atividade.
3. Sempre que for adequado, deve-se usar os conceitos de encapsulamento, herança e polimorfismo. Verifique também a necessidade de aplicar outros conceitos relacionados, como classes e métodos abstratos, sobrecarga e sobrescrita de métodos, relacionamento de objetos, entre outros. Por fim, você deverá deixar o software com o máximo possível de **coesão** e o mínimo possível de **acoplamento**.
4. Siga boas práticas de programação, pois, caso contrário, poderá causar perda de nota. .
   - dar nomes intuitivos para as variáveis
   - dar nomes intuitivos para as funções
   - comentar o código
5. Cuidado com plágio. Se for detectado plágio, a punição será dada para todos os envolvidos: todos do grupo que copiou e todos do grupo que forneceu a cópia.

Descrição do projeto

A empresa *Organizações Tabajara* precisa de um sistema para **controle de compras** feitas por clientes.

Um **cliente** contém **endereço** e **data de cadastro**. O endereço de um cliente contém **rua, número, bairro, CEP, cidade e estado** e um **método** chamado **paraString** que **retorna uma string com as informações organizadas**. Além disso, existem clientes **pessoa física** e **pessoa jurídica**. Para o primeiro tipo de cliente é armazenado seu **CPF** e a **quantidade máxima de parcelas** da compra. Já, para cliente pessoa jurídica, é armazenado o **CNPJ**, **razão social** e o **prazo máximo (em dias) para pagamento** da compra. Todo objeto que seja um cliente deve ter um método chamado **paraString** que deve retornar as informações do cliente como uma *string*.

Um cliente pode efetuar uma ou mais compras. Cada compra possui um ou mais itens comprados contendo a **quantidade**, **nome do produto**, **preço unitário** e **valor total** do

item. Além do **item de compra**, a compra possui um **identificador**, **data**, **valor total**, o **CPF ou CNPJ** do cliente que fez a compra e o **total que foi pago até o momento**. O objeto relativo a uma determinada compra deve possuir também um método que retorne o **valor que ainda falta para pagar**.

A empresa deseja armazenar as seguintes informações sobre os produtos comercializados: **código, nome do produto, descrição e preço**. Alguns produtos são perecíveis e, nesse caso, devem possuir **data de validade** e um **método** que retorne um valor do tipo *boolean* indicando se na data de hoje o produto está **vencido ou não**.

O sistema deve ter um menu que deve ser apresentado por meio de funções da classe JOptionPane. Ele precisa conter as seguintes funcionalidades:

1. Cadastros de Clientes
2. Deletar cliente pelo CPF
3. Deletar cliente pelo nome
4. Cadastro de Produtos
5. Efetuação de uma compra
6. Atualização da situação de pagamento de uma compra
7. Relatórios
   (a) Relação de todos os Clientes que possuem o nome iniciado por uma determinada sequência de caracteres
   (b) Relação de todos os Produtos
   (c) Busca de um produto pelo nome
   (d) Relação de produtos que são perecíveis e que estão com a data de validade vencida
   (e) Relação de todas as compras
   (f) Busca de uma compra pelo número
   (g) Relação de todas as compras que não foram pagas ainda
   (h) Relação das 10 últimas compras pagas
   (i) Apresentação das informações da compra mais cara
   (j) Apresentação das informações da compra mais barata
   (k) Relação do valor total de compras feitas em cada mês nos últimos 12 meses.
8. Sair (termina o sistema).

Nas tarefas indicadas no menu acima, sempre que for necessário, trate os casos envolvendo clientes, produtos ou compras inexistentes. Onde for necessário, trate também os casos envolvendo dados repetidos.

Você deve implementar as funções indicadas no menu acima por meio classes de gerenciamento de clientes, produtos e compras. Porém, toda a comunicação com o usuário deve ser feita na classe principal ou por meio de uma classe criada especialmente para isso. Isso inclui pedir que o usuário adicione dados ou mostrar coisas para ele.

Quando iniciar o sistema, ele deve verificar se existe um arquivo de texto para cada um dos cadastros (Clientes, Produtos e Compras) dentro de uma pasta chamada **baseDados**. Caso existe um arquivo de texto para um determinado cadastro, essas informações devem ser carregadas no respectivo objeto. Quando o usuário clicar na opção **Sair** do menu, os arquivos de texto da base de dados devem ser atualizados com as novas informações dos

objetos que foram adicionadas durante o tempo que o software ficou em uso. Todas essas tarefas envolvendo gravar e ler as informações dos arquivos devem ser feitas por uma classe especializada.

Formato de Entrega

O sistema deve ser desenvolvido em Java e entregue pelo Canvas da nossa disciplina. Deve ser enviado um arquivo compactado em formato ZIP contendo:

- Os códigos Java do sistema;
- Uma pasta chamada **baseDados** que possua:
  - um arquivo de texto contendo cadastro de pelo menos 5 Clientes para ser carregado quando iniciar o sistema.
  - um arquivo de texto contendo cadastro de pelo menos 5 Produtos para ser carregado quando iniciar o sistema.
  - um arquivo de texto contendo cadastro de pelo menos 5 Compras para ser carregado quando iniciar o sistema.
- Arquivo .txt com os nomes dos componentes do grupo e seus respectivos números de matrícula.