

Lista de Exercícios 4

Disciplina: Projeto de Sistemas Orientados a Objetos

Aula: Análise e Projeto Orientado a Objetos

Conteúdo Formativo: Diagrama UML, implementação de software em Java, coesão, acoplamento, encapsulamento, tratamento de exceções e serialização de objetos.

Exercício: Sistema de Vendas para Supermercado

Objetivo:

Implementar o sistema de vendas para supermercado, utilizando a linguagem Java e seguindo os princípios de encapsulamento, coesão, acoplamento e tratamento de exceções. O sistema deve possuir uma interface de linha de comando com menu de opções e permitir a serialização de objetos.

Requisitos Mínimos:

- Interface de Linha de Comando (CLI): O sistema deve ter uma interface de linha de comando (CLI) com um menu de opções para interagir com as funcionalidades do sistema.
- Tratamento de Exceções: O sistema deve lidar com erros e exceções de forma adequada, garantindo a estabilidade e a robustez do sistema. Utilize blocos try-catch para tratar as exceções e mensagens informativas para o usuário.
- Encapsulamento: As classes devem ser projetadas com encapsulamento, ocultando os detalhes internos e expondo apenas as informações e funcionalidades necessárias para o uso externo. Utilize modificadores de acesso (private, protected, public) para controlar a visibilidade dos membros das classes.
- Coesão: As classes devem ser coesas, com métodos relacionados ao mesmo propósito, evitando duplicação de código e aumentando a legibilidade.
- Acoplamento: O sistema deve ter um acoplamento fraco entre as classes, utilizando interfaces e minimizando a dependência entre elas. Utilize interfaces para definir contratos entre as classes e abstrair dependências.
- Serialização de Objetos: O sistema deve permitir a serialização dos objetos, permitindo que os dados sejam salvos em arquivos e carregados posteriormente, garantindo a persistência das informações. Utilize a interface Serializable para permitir que os objetos sejam serializados.

Interface de Linha de Comando (CLI):

A interface de linha de comando (CLI) deve conter um menu principal e submenus de opções oferecendo ao usuário o acesso às funcionalidades do sistema.

Menu Principal: O menu principal deve apresentar as seguintes opções:

- Cadastro de Clientes
- Cadastro de Atendentes de Caixa
- Realização de Vendas
- Histórico de Compras e
- Histórico de Vendas
- Gerenciamento de Estoque
- Pedidos de Compra
- Sair

Cadastro de Clientes:

- Incluir novo cliente
- Consultar cliente
- Atualizar cliente
- Excluir cliente

Cadastro de Atendentes de Caixa:

- Incluir novo atendente
- Consultar atendente
- Atualizar atendente
- Excluir atendente

Opções de Venda:

- Iniciar nova venda
- Adicionar produtos à venda
- Finalizar venda

Histórico de Compras:

- Consultar histórico de compras de um cliente

Histórico de Vendas:

- Consultar histórico de vendas de um caixa

Gerenciamento de Estoque:

- Cadastrar novo produto
- Consultar estoque de um produto
- Atualizar estoque de um produto
- Excluir produto

Pedidos de Compra:

- Gerar novo pedido de compra

Sair:

- Finalizar o programa.

Etapas de Implementação:

- Implementação das Classes: Implementar as classes do sistema de acordo com o diagrama de classes UML desenvolvido na Lista 3, seguindo os princípios de encapsulamento, coesão e acoplamento.
- Tratamento de Exceções: Implementar o tratamento de exceções para garantir que o sistema lide com erros de forma robusta e segura, informando ao usuário sobre o problema e permitindo que a execução do sistema continue.
- Interface de Linha de Comando (CLI): Implementar uma interface de linha de comando (CLI) com um menu de opções, que permita ao usuário interagir com as funcionalidades do sistema. Utilize a classe Scanner para ler a entrada do usuário.

- **Serialização de Objetos:** Implementar a serialização dos objetos, permitindo que os dados do sistema sejam salvos em arquivos e carregados posteriormente. Utilize a interface `Serializable` para permitir a serialização dos objetos.

Entrega:

O aluno deve entregar o código fonte do sistema em um arquivo ZIP.

Dicas:

- Documente o código com comentários e descrições claras para facilitar a compreensão e a manutenção.
- Faça testes para garantir a qualidade do sistema.

Observações:

O aluno deve usar o diagrama de classes UML desenvolvido na Lista 3 como base para a implementação do sistema.

O código fonte deve ser organizado e comentado, de forma a facilitar a leitura e a compreensão do código.

Este trabalho visa auxiliar os alunos no processo de aprendizagem da implementação de sistemas de software utilizando a programação orientada a objetos na linguagem Java. O aluno deve se dedicar ao desenvolvimento do sistema, garantindo que ele atenda aos requisitos e tenha uma estrutura organizada e eficiente.