

Projeto de Software

Atividade 07 - Sistema de compras com Padrões de Projeto

Atividade individual

Data de Entrega: 16/nov

Crie uma API de pedidos que permita a criação de diferentes tipos de pedidos (comuns e expressos) e o processamento deles. A aplicação deve interagir com um serviço de logística externo, que possui uma interface de dados legada e precisa de um adaptador para ser integrado ao sistema. O objetivo é exercitar a aplicação de diferentes padrões de projeto neste desenvolvimento

Crie as seguintes classes representando as entidades do sistema:

- Produto:
 - id (inteiro)
 - nome (String)
 - preco (ponto-flutuante)
- Pedido:
 - id (inteiro)
 - dataCriacao (Date)
 - status (enum StatusPedido)
 - itens (Lista)
- ItemPedido:
 - produto (Produto)
 - quantidade (inteiro)
- StatusPedido:
 - Enum com os valores NOVO, EM_PROCESSAMENTO, ENVIADO, ENTREGUE

Crie uma classe PedidoFactory que será responsável por instanciar diferentes tipos de pedidos, abstraindo a lógica de criação.

- Crie uma interface IPedido com o método processarPedido().
- Crie as classes PedidoComum e PedidoExpresso que implementam IPedido.

- A PedidoFactory deve ter um método criarPedido(TipoPedido tipo) que retorna a implementação correta de IPedido com base no enum TipoPedido (COMUM ou EXPRESSO).
- Inclua a PedidoFactory em um serviço de pedido (PedidoService).

Crie um adaptador para integrar o sistema de logística legado.

- Interface Legada: Crie uma interface ILogisticaLegado com um método enviarPacote(String endereço, String códigoRastreio).
- Serviço Legado: Crie uma classe LogisticaLegadoService que implemente ILogisticaLegado e simule a lógica de envio (por exemplo, imprimindo uma mensagem no console).
- Interface Nova: Crie uma interface ILogistica com o método enviar(Pedido pedido).
- Adapter: Crie a classe LogisticaAdapter que implemente ILogistica. Esta classe deve receber uma instância de ILogisticaLegado e adaptar a chamada para o novo método, convertendo o objeto Pedido para os parâmetros esperados pelo serviço legado.
- Inclua o LogisticaAdapter no PedidoService.

Crie uma classe ConfiguracaoSistema que armazena informações como o nome da loja e a taxa de desconto global.

- Utilize o padrão Singleton para garantir que haverá apenas uma instância dessa classe durante toda a execução da aplicação.
- Essa classe pode ser um enum para garantir a segurança de threads de forma simples.
- Inclua a instância de ConfiguracaoSistema no PedidoService para que ela possa ser utilizada na lógica de processamento.

Crie uma API REST com Spring Boot:

- Crie um PedidoController com os seguintes endpoints:
 - POST /pedidos: Recebe um JSON com os dados do pedido (itens, tipo) e utiliza a PedidoFactory para criar e processar o pedido.
 - GET /pedidos/{id}: Retorna os detalhes de um pedido específico.
- Se quiser, utilize o Spring Data JPA para criar um repositório (PedidoRepository) para persistir os pedidos em um banco de dados em memória, como o H2.
- O PedidoService deve:

- Receber os dados do PedidoController.
- Utilizar a PedidoFactory para criar a instância correta do pedido.
- Processar o pedido através da chamada pedido.processarPedido().
- Utilizar o LogisticaAdapter para enviar o pedido.
- Utilizar a ConfiguracaoSistema para aplicar um desconto.

Utilize o postman ou outra ferramenta para exercitar o seu sistema.

Eventualmente, após a entrega, poderá haver o sorteio de alguns alunos para me mostrar o sistema, explicando o que foi feito e as decisões tomadas.