

Relatório EP2-COO

Membros

Nome: Gustavo Felisberto de Oliveira	NºUSP:13672300
Nome: Marcelo de Menezes Nascimento	NºUSP:13727078
Nome: Gabriel Lacerda Domingues	NºUSP:13781987

Introdução

O objetivo deste EP é refatorar o código de um “Gerador de Relatórios” para que ele obedeça os padrões de projeto Strategy e Decorator, além de implementações como o carregador de csv e outras opções de filtro e formatação. Também foi feita atualização do array de produtos substituindo pelo arcabouço de coleções de java.

Padrões de Projeto Strategy e Decorator

Para implementar o padrão strategy, que consiste de um padrão de projeto que delega as responsabilidades adquiridas pelas entidades, geralmente criando uma interface e classes que a implementam. Para fazer isso, nós criamos a pasta “criterios” que contém a interface criterioOrdenacao que contém a função comparar que recebe como parâmetro dois produtos, também criamos a CriterioFromString para determinar o critério de ordenação que será utilizado, a partir de uma string passada como parametro no terminal, e as subclasses CriterioDescricao, CriterioEstoque, CriterioPreco que implementam a interface criterioOrdenacao. Para formatação do texto html, foi feita a implementação do padrão Decorator que facilita a adição de novas funcionalidades sem alteração do código original. Para isso, foi criada a pasta Decorator contendo todas as possíveis formatações, sendo elas: itálico, negrito, itálico e negrito e a cor que o texto deve exibir. A interface produto foi utilizada como base para implementação da classe ProdutoDecorator e a partir dela foram feitas extensões para que todas as possíveis formatações fossem contempladas. Com isso, houve a separação de função, pois a classe GeradorDeRelatorio deixou de ser responsável pela formatação, atribuição essa que foi passada para o tipo Produto com o uso do Decorator. Também criamos uma classe contendo as constantes e fizemos uma classe própria para executar a Main, além de remover funções que não deveriam ser responsabilidade da GeradorDeRelatorio.

Implementações

Foi implementada a função csvReader que é responsável por ler o csv, cuja rota deverá ser passada nos parâmetros, e adicionar cada produto em uma collection ArrayList, as filtragens e ordenações também foram adaptadas para funcionar usando a collection.

Além disso, o `csvReader` também verifica os campos negrito, itálico e cor para chamar as classes decorators correspondentes de cada produto.

Conclusão

Ao final do EP, o código ficou muito mais legível e fácil de se entender, as responsabilidades que estavam todas agrupadas na classe `GeradorDeRelatorios` foram separadas em classes próprias e a responsabilidade da `GeradorDeRelatorios` se tornou unicamente gerar os relatórios. Pudemos observar como a utilização de boas práticas e padrões de projetos tornaram possíveis implementações ou manutenções no código tarefas simples e rápidas de serem executadas.