Gabriel Monteiro de Souza - 14746450
Vitor Ferreira Sacchi - 12542776

Relatório da refatoração do EP 2 de COO

1. Algoritmo de ordenação

Foram aplicados os conceitos de Strategy em ambos os quick sort e insertion sort: Foi criada a interface sort e, com isso, o quick e o insertion sort se tornam implementações dessa interface. Dessa forma se torna mais fácil para criar outros algoritmos de ordenação como sugere o padrão Strategy, uma vez que apenas necessita implementar a interface sort e implementar o restante do algoritmo.

2. Critério de ordenação

Nesse quesito, também foi aplicado o Strategy, separando uma interface chamada Criteria que especifica o método compare, que compara os critérios, para assim, facilmente adicionar novos critérios para a ordenação, sempre que necessário.

Com isso, foi adicionado o critério de ordenação decrescente, para os critérios já existentes.

3. Critério de filtragem

Aqui também foi aplicado o Strategy, e foi criada a interface Filter, que verifica o funcionamento de cada filtro separado em arquivos já existentes.

Foram, também, implementados dois filtros novos, sendo eles, descricao_contendo, especifica uma sub string que tenha encontrar uma descrição que contenha tal sub string, e preco_intervalo, em que se especifica dois valores, com duas casas decimais, e o filtro entrega os produtos que se encontram em tal intervalo de preço.

Por exemplo, você pode colocar os valores “30.40,50.50”, nesse caso o filtro entregaria todos os produtos com valores entre R$30,40 e R$50,50.

4. Opções de formatação

Nesse caso foi aplicado o padrão Decorator, foi criada a interface Format, que especifica o método format e um decorador para o format, ou seja, uma classe abstrata FormatDecorator,

Os diferentes formatações que podem ser aplicadas aos textos, utilizam do decorador, e por padrão utilizam o FormatDefault, que define o padrão para impressão. Além disso existem as formatações bold, italic e color.

As formatações por padrão são iguais a tabela CSV, mas caso seja especificado alguma das opções de formatação, negrito e/ou itálico, faz com que todo o texto fica nesse opção de formatação.

5. Arquivo de entrada

Foi criada uma classe separada, chamada ProductFormatted que extende a classe produto padrão para que tenha uma formatação. Existe também a nova classe LoadProducts que serve para ler as entradas de um arquivo .csv que contém uma tabela. A cada nova entrada que é lida, ele cria um ProductFormatted com a devida formatação e adiciona a uma coleção do tipo lista.

6. Outras mudanças

O vetor de produtos, produto[ ], foi alterado para uma coleção tipo lista, List<Produto>, o que facilita o uso dos métodos do Java para o tornar mais orientado a objetos, já que é possível utilizar os métodos, tanto, de lista, quanto, de coleção

7. Conclusão

As mudanças de forma geral, foram feitas com o intuito de utilizar os padrões de projeto, Strategy e Decorator, e ainda por cima, os princípios SOLID, para deixar o código mais fácil de ser lido, compreendido, modificado e mantido. Com isso o código se torna, facilmente, adaptável, caso haja a necessidade de novos recursos,