# GERADOR DE RELATÓRIOS – MUDANÇAS

Foram feitas duas interfaces Strategy: StrategyFiltro e StrategyOrdena. Como os nomes sugerem, uma tem as classes concretas implementado os filtros e a outra os algoritmos de ordenação. Abaixo um esquema simples de com estão dispostas no código.

Interface StrategyFiltro{

Classe Filtro implementa StrategyFiltro{

FiltroCategoria extende Filtro

FiltroContem extende Filtro

FiltroIntervalo extende Filtro

FiltroQtdEst extende Filtro

}

}

Interface StrategyOrdena{

Classe InsertionSort implementa StrategyOrdena{

InsertionSortDec extende InsertionSort

InsertionEstCresc extende InsertionSort

InsertionEstCresc extende InsertionSort

InsertionPrecoCresc extende InsertionSort

InsertionPrecoCresc extende InsertionSort

}

Classe QuickSort implementa StrategyOrdena{

QuickSortDec extende QuickSort

QuickEstCresc extende QuickSort

QuickEstCresc extende QuickSort

QuickPrecoCresc extende QuickSort

QuickPrecoCresc extende QuickSort

}

}

Em seguida foi feita a classe DecoratorFormatting, que implementa a interface produto e tem como base a classe ProdutoPadrao, com os herdeiros ProdutoItalico, ProdutoNegrito e ProdutoColorido, cada um adicionando uma característica ao método formataParaImpressao() da base para utilizar o padrão Decorator na formatação da saída.

Com isso, foram feitas as adaptações no construtor e nas variáveis globais da classe GeradorDeRelatorios, e os algoritos originas do método GeraRelatorios() foram substituídos pelos métodos presentes em objetos do tipo StrategyOrdena e StrategyFiltro que a classe recebe no construtor. A formatação da saída foi passada para o método carregaProdutos() usando o padrão Decorator na instanciação dos produtos.

No método main foram adicionadas instâncias de exemplo de todas as classes presentes em StrategyOrdena e e StrategyFiltro.