

1 Descrição do problema:

Você deve implementar um sistema de ordenação de uma lista de números inteiros usando o algoritmo de ordenação QuickSort. No entanto, o algoritmo deve ser parametrizado para permitir que o usuário escolha a ordem de ordenação: crescente ou decrescente.

Para implementar esse sistema, você deve usar o padrão de projeto Template Method. O algoritmo de ordenação QuickSort deve ser implementado como um método abstrato na classe abstrata `Sorter`. A classe `Sorter` deve conter um método template chamado `sort`, que irá chamar o método abstrato de ordenação QuickSort e definir a ordem de ordenação de acordo com o parâmetro passado pelo usuário.

2 Requisitos

- Implemente uma classe abstrata `Sorter` com um método abstrato chamado `quickSort` que recebe como parâmetro uma lista de números inteiros e um inteiro que indica o índice do primeiro elemento da lista a ser ordenado e o índice do último elemento da lista a ser ordenado. Essa classe deve ter também um método template chamado `sort`, que recebe como parâmetro a lista de números inteiros e um parâmetro booleano que indica a ordem de ordenação: crescente ou decrescente. O método `sort` deve chamar o método `quickSort` e definir a ordem de ordenação de acordo com o parâmetro passado pelo usuário.
- Implemente duas subclasses da classe `Sorter`: `AscendingSorter` e `DescendingSorter`, que implementam o método `quickSort` de acordo com a ordem de ordenação desejada (crescente ou decrescente).
- Implemente uma classe `TestSorter` que testa o funcionamento do sistema. Essa classe deve criar uma lista de números inteiros, chamar o método `sort` da classe `Sorter` com diferentes parâmetros de ordenação e imprimir a lista ordenada.

3 Observações

- O padrão de projeto Template Method é útil para implementar algoritmos que possuem uma estrutura fixa, mas que permitem variações em certas etapas. Nesse caso, a estrutura fixa é o algoritmo de ordenação QuickSort, e a variação é a ordem de ordenação.
- Lembre-se de que o algoritmo QuickSort é recursivo. Portanto, o método `quickSort` deve ser capaz de chamar a si mesmo com sub-listas (ou subarrays) menores.
- Certifique-se de testar seu sistema com diferentes tamanhos de conjuntos de dados e diferentes ordens de ordenação.