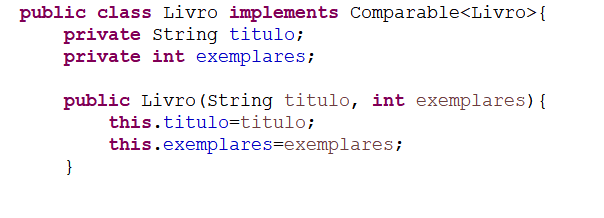
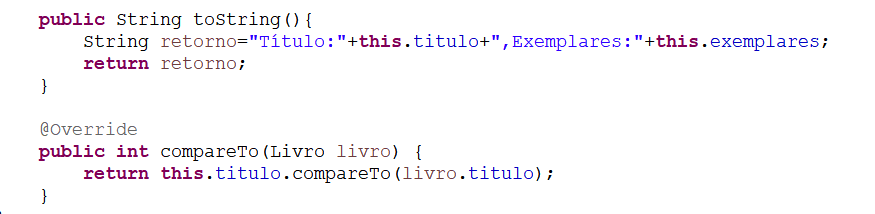
**Atividades iguais serão zeradas.**

**Exercícios que envolvam codificações iguais serão zerados.**

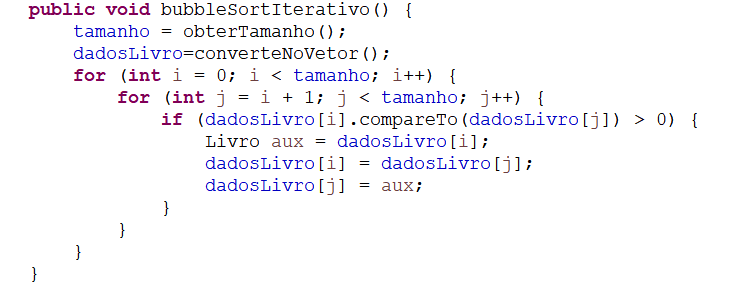
1) Implemente uma Lista de Livros em alocação dinâmica de memória com os atributos titulo e quantidade de exemplares, **que utilize somente métodos recursivos em todas as implementações que envolvam estruturas de repetições**. Realize as seguintes operações:

* verificar se a lista está vazia, retornando true se estiver vazia e false se não estiver;
* adicionar um livro no início da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
* adicionar um livro no final da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
* adicionar um livro em determinada posição da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
* remover um livro do início da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando.
* remover um livro do final da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando.
* remover um livro de determinada posição da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando.
* ordenar a lista por meio do bubble sort pelo título do livro para a lista original. Primeiramente é necessário que a classe Livro implemente a interface Comparable.

 Em seguida é necessário que sejam implementados os métodos toString() e compareTo()



Por último é necessário converter o código abaixo para que o mesmo funcione de maneira recursiva.

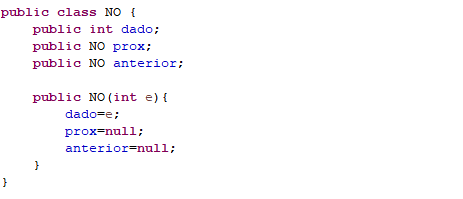


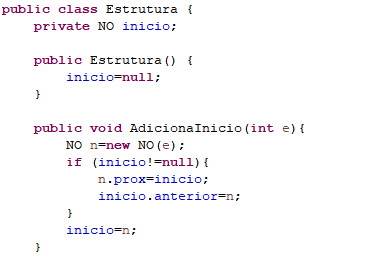
* implementar um método de busca sequêncial para retornar o livro encontrado na lista a partir do titulo
* percorrer e apresentar cada um dos elementos da lista

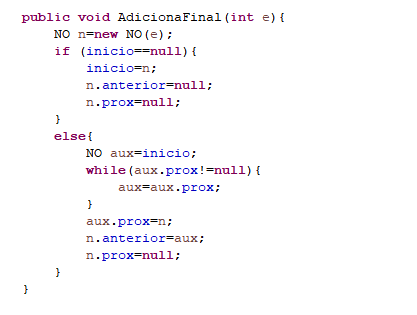
2) Implemente uma Lista de Alunos em alocação dinâmica de memória duplamente encadeada com os atributos id, nome e curso, **que utilize somente métodos recursivos em todas as implementações que envolvam estruturas de repetições.** Realize as seguintes operações

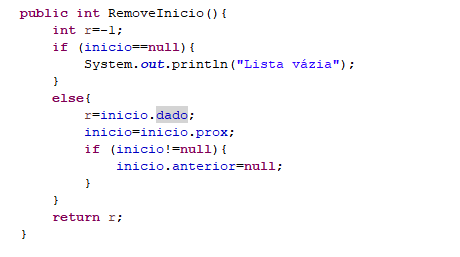
* verificar se a lista está vazia, retornando true se estiver vazia e false se não estiver;
* adicionar um aluno no início da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
* adicionar um aluno no final da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
* adicionar um aluno em qualquer posição da lista, caso a operação não possa ser realizada, mostre mensagem avisando;
* remover um aluno do início da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando.
* remover um aluno do final da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando.
* remover um aluno em qualquer posição da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando.
* ordenar a lista por meio do bubble sort pelo nome do aluno para a lista original;
* implementar um método de busca sequêncial para retornar pelo nome do aluno
* percorrer e apresentar cada um dos elementos da lista

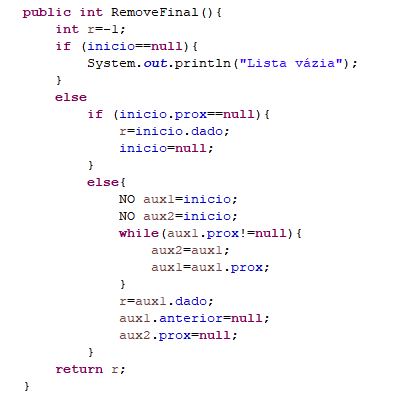
3) Dada as classes abaixo, explique qual estrutura de dados elas pertencem, o funcionamento de cada método e realize pelo menos 5 simulações para cada um deles.

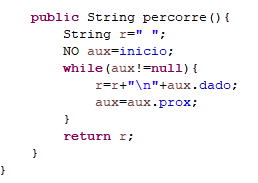




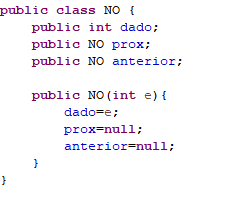


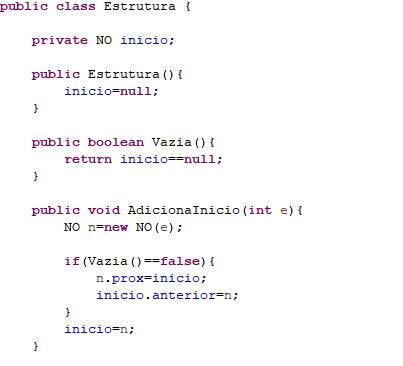


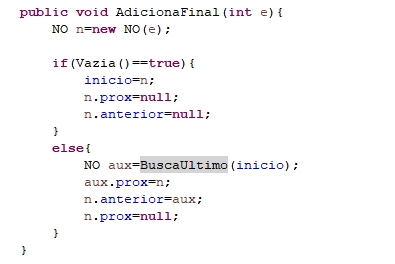


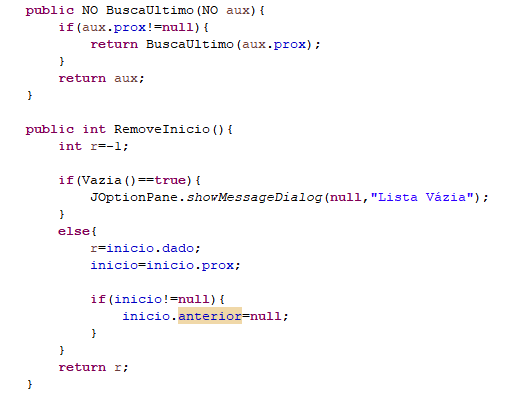


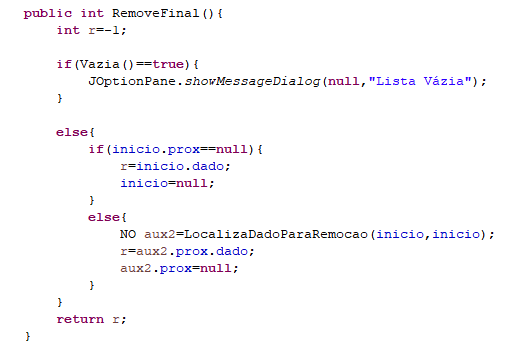
4) Dada as classes abaixo, explique qual estrutura de dados elas pertencem, o funcionamento de cada método e explique a principal diferença entre os métodos AdicionaFinal e RemoveFinal em relação aos métodos apresentados no exercício 1.

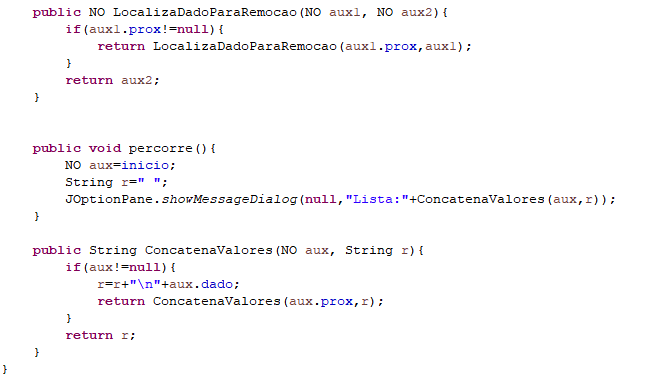












5) Explique cada uma das afirmações abaixo, descrevendo em qual estrutura de dados elas pertencem. Justifique sua resposta.

I - Inserção e remoção de elementos acontecem apenas na “cabeça” da estrutura.

II - Inserção de um nó no meio da estrutura pode ser realizada com custo computacional constante.

III - Respeito à política FIFO: o primeiro elemento que entra é o primeiro a sair.

6) Explique se a frase abaixo se refere a uma pilha, lista encadeada, fila, matriz ou vetor. Justifique sua resposta.

“Na alocação dinâmica de memória, os dados são armazenados em posições de memória referenciadas e dispostos em uma dada organização não linear, sendo possível, a partir de um elemento, encontrar os próximos.”

7) Considerando as definições para listas, pilhas e filas, explique cada uma das frases abaixo. Justifique se cada uma das afirmações é verdadeira ou falsa.

A) Uma lista é um tipo de fila que se caracteriza por considerar que o primeiro elemento a entrar é o primeiro a sair.

B) Lista é um conjunto de filas e pilhas e se compõe por elementos que podem ser ligados ou não.

C) Lista é uma sequência finita de elementos ligados entre si. Podem ser organizada de tal forma que implemente uma fila ou uma pilha.