

Atividades iguais serão zeradas.

Exercícios que envolvam codificações iguais serão zerados.

1. Defina cada uma das estruturas de dados abaixo, descreva quais são as operações aplicáveis e apresente o teste de mesa para um conjunto de entrada de 15 elementos para cada estrutura:

- Lista Estática
- Lista Circular
- Pilha
- Fila de Prioridades
- Fila Circular
- Lista Ligada

2. Implemente uma Lista de temperaturas em alocação dinâmica com todas as operações indicadas a seguir:

-verificar se a lista está vazia, retornando true se estiver vazia e false se não estiver;

-adicionar uma temperatura no início da lista, caso a operação não possa ser realizada, mostre mensagem avisando;

-adicionar uma temperatura no final da lista, caso a operação não possa ser realizada, mostre mensagem avisando;

-adicionar uma temperatura em determinada posição da lista, caso a operação não possa ser realizada, mostre mensagem avisando;

-remover a temperatura do início da lista, retornando o valor que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;

-remover a temperatura do final da lista, retornando o valor que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;

-remover a temperatura de determinada posição da lista, retornando o valor que foi removido, caso a operação não possa ser realizada, mostre mensagem avisando;

-percorrer e apresentar cada um dos elementos da lista.

3. A classe Aluno possui atributos que são criados na classe concreta e acessados por meio dos métodos públicos da classe (getters e setters). Dado o cenário implemente uma Lista Encadeada:

- Sejam criados em uma classe os atributos **RA** de tipo de dado **inteiro**, **nome** de

tipo de dado `String`, `turma` do tipo de dado `String` e `semestre` do tipo de dado `String`. Estes atributos devem ser acessados por meio de seus métodos públicos (getters e setters).

- Implemente um método que adicione um aluno no final da lista
- Implemente um método que adicione um aluno no início da lista
- Implemente um método que adicione um aluno no meio da lista
- Implemente um método que remova um aluno no final da lista
- Implemente um método que remova um aluno no início da lista
- Implemente um método que remova um aluno no meio da lista
- Implemente um método que apresente cada um dos elementos da lista
- Realize pelo menos um teste de mesa para cada estrutura com 15 elementos como conjunto de entrada.