



Universidade do Estado do Rio Grande do Norte
Departamento de Informática
Curso de Ciência da Computação

Lista de atividades para as disciplinas Lab. Estrutura de Dados e Lab. de POO
Prof. Antônio Oliveira Filho
Monitor: João Marcelo Nunes de Souza

Data: 27/09/2024
Entrega: até 11/10/2024

1. Implemente a classe Fila adicionando os métodos push e pop.
2. Adicione o método "topo" na classe Pilha que retorna o topo da Pilha.
3. Implemente a classe PilhaFila que utiliza Filas para simular uma Pilha.
4. Implemente a classe FilaPilha que utiliza Pilhas para simular uma Fila.
5. Defina um método em Pilha que retorna uma Fila equivalente.
6. Defina um método em Fila que retorna uma Pilha equivalente.
7. Na classe Lista, implemente o método inverte que retorna uma Lista com os elementos em ordem invertida.
8. Implemente em Lista o método para addElemento(int valor, int pos) que adiciona um elemento na posição especificada. A posição pode ou não existir.
9. Implemente uma classe ListaDupla que implementa o duplo encadeamento dos blocos. Sendo assim o Bloco precisa ser também modificado.
10. Implemente a Lista com array ao invés de uma lista encadeada. No array, blocos também serão adicionados.
11. Implemente a classe ListaRecursiva que define métodos recursivos de localizar e de tamanho. Lembre-se que os métodos devem ser declarados como private para não ficarem disponíveis para o usuário programador.
12. Utilize uma das estruturas para resolver o seguinte problema de soma:
 $((1+3)*(3+4))+7)/35$. Esse problema pode ser modelado assim: $1\ 3\ +\ 3\ 4\ +\ * \ 7\ +\ 35\ /$.
13. Implemente o método "concatena(Lista l2)" na classe Lista. O método deve concatenar os elementos da lista atual com os elementos da lista passada como parâmetro, retornando uma nova lista.
14. Implemente um método "removeDuplicados" na classe Lista. Esse método deve percorrer a lista e remover todos os elementos duplicados, mantendo apenas a primeira ocorrência.
15. Crie um método "clone" na classe Lista. Esse método deve retornar uma nova instância de Lista com os mesmos elementos da lista original.

16. Implemente a classe FilaPrioridade. os elementos serão retirados com base em sua prioridade (maior prioridade sai primeiro).
17. Implemente o método "removeElemento(int pos)" na classe Lista. Esse método deve remover o elemento na posição especificada.
18. Adicione o método "localizarMaior(int valor)" na classe Lista. Esse método deve retornar uma Lista com todos os blocos que contenham um valor maior do que o valor passado .
19. Implemente um jogo de baralho com dois decks, um para cartas jogador. O jogador vencedor é o que retira as 5 primeiras cartas com a maior soma.
20. Tipicamente os jogos da velha são implementados através de matrizes, porém o desafio aqui é implementar utilizando uma única lista.

Questão especial:

21. Implemente as classes "Aluno" com os atributos "nome" e "matrícula" e "Disciplina" com o atributo nome. Implemente também a classe "Turma" com os atributos "alunos", "disciplina" e "código". A lista de alunos deve ser implementada com a classe Lista que vem sendo desenvolvida até então.
22. Para complementar a questão 21, implemente uma solução para o cadastro das notas dos alunos. Lembre-se, sempre que houver a necessidade de uso de lista, utilize a classe Lista que vem sendo desenvolvida até então.