

LISTAS
Parte 2

Profa. Me. Andréa Zotovici

São Paulo
2018

ALOCAÇÃO ESTÁTICA

Dada a Lista Vazia

L1:ListaDeInteiros			
dados	0	1	2
			tamanho: 0

Adicione o elemento 15 no final da lista, processo:

- Lista está cheia?
- Não:
 - guarde 15 no vetor denominado dados, no próximo índice disponível (aquele indicado por tamanho)
 - some 1 em tamanho

L1:ListaDeInteiros			
dados	0	1	2
	15		
			tamanho: 1

ALOCAÇÃO ESTÁTICA

Dada a Lista Vazia

L1:ListaDeInteiros			
dados	0	1	2
	15		
tamanho: 1			

Adicione o elemento 30 no final da lista, processo:

- Lista está cheia?
- Não:
 - guarde 30 no vetor denominado dados, no próximo índice disponível (aquele indicado por tamanho)
 - some 1 em tamanho

L1:ListaDeInteiros			
dados	0	1	2
	15	30	
tamanho: 2			

ALOCAÇÃO ESTÁTICA

Dada a Lista Vazia

L1:ListaDeInteiros			
dados	0	1	2
	15	30	
tamanho: 2			

Adicione o elemento 53 no final da lista, processo:

- Lista está cheia?
- Não:
 - guarde 53 no vetor denominado dados, no próximo índice disponível (aquele indicado por tamanho)
 - some 1 em tamanho

L1:ListaDeInteiros			
dados	0	1	2
	15	30	53
tamanho: 3			

ALOCAÇÃO ESTÁTICA

Dada a Lista

L1:ListaDeInteiros			
	0	1	2
dados	15	30	53
tamanho: 3			

Adicione o elemento 47 no início da lista, processo:

- Lista está cheia?

- Sim:

- Mostre a mensagem "Lista cheia"

L1:ListaDeInteiros			
	0	1	2
dados	15	30	53
tamanho:3			

ALOCAÇÃO ESTÁTICA

1. Lista está cheia?

Sim:

2. Mostre a mensagem “Lista cheia”, não altere o estado do objeto

Não:

3. guarde o novo elemento no vetor denominado dados, no próximo índice disponível (aquele indicado por tamanho)

4. some 1 em tamanho

Implementação em Java

```
public void adicionaFinal(int e) {  
1. if (tamanho == dados.length)  
2.     System.out.println("ERRO! A Lista Está Cheia");  
    else {  
3.         dados[tamanho] = e;  
4.         tamanho++;  
    }  
}
```

ALOCAÇÃO ESTÁTICA

Dada a Lista

L1:ListaDeInteiros				
dados	0	1	2	tamanho: 3
	15	30	53	

Remova o elemento do final da lista, processo:

- Lista está vazia?
- Não
 - guarde o elemento do final da lista em uma variável
 - decrémente um de tamanho
 - retorne o elemento removido

L1:ListaDeInteiros				
dados	0	1	2	tamanho: 2
	15	30		

retorna 53

ALOCAÇÃO ESTÁTICA

Dada a Lista

L1:ListaDeInteiros			
	0	1	2
dados	15	30	
tamanho: 2			

Remova o elemento do final da lista, processo:

- Lista está vazia?
- Não
 - guarde o elemento do final da lista em uma variável
 - decrémente um de tamanho
 - retorne o elemento removido

L1:ListaDeInteiros			
	0	1	2
dados	15		
tamanho: 1			

retorna 30

ALOCAÇÃO ESTÁTICA

Dada a Lista

L1:ListaDeInteiros			
	0	1	2
dados	15		
tamanho: 1			

Remova o elemento do final da lista, processo:

- Lista está vazia?
- Não
 - guarde o elemento do final da lista em uma variável
 - decrémente um de tamanho
 - retorne o elemento removido

L1:ListaDeInteiros			
	0	1	2
dados			
tamanho: 0			

retorna 15

ALOCAÇÃO ESTÁTICA

Dada a Lista

L1:ListaDeInteiros			
	0	1	2
dados			
tamanho: 0			

Remova o elemento do início da lista, processo:

- Lista está vazia?

- Sim

- mostre a mensagem “ERRO! Lista Vazia”

Implemente o método removeFinal e teste na classe ListaDeInteiros

EXERCÍCIO

2) Mostre detalhadamente o processo de cada método invocado a seguir para uma Lista com alocação estática de memória, com capacidade máxima 4, que inicia vazia. Adicione todas as mensagens geradas pelos métodos e informe todos os valores devolvidos:

- a) l1.adicionaFinal(74)
- b) l1.adicionaInicio(73)
- c) l1.adicionaFinal (72)
- d) l1.adicionaInicio (71)
- e) l1.adicionaFinal (70)
- f) l1.removeFinal()
- g) l1.removeInicio()
- h) l1.removeFinal ()
- i) l1.removeFinal ()
- j) l1.removeFinal ()

EXERCÍCIO

3) Implemente uma Lista de caracteres em alocação estática com todas as operações indicadas a seguir. A lista possui um vetor para guardar os caracteres e um atributo para guardar a quantidade de elementos presentes na lista.

- adicionar caractere no final da lista, caso a operação não possa ser realizada, mostre mensagem de erro;
- remover caractere do final da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem de erro;
- adicionar caractere no início da lista, caso a operação não possa ser realizada, mostre mensagem de erro;
- remover caractere do início da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem de erro;
- concatenar os caracteres em uma String e retorná-la;
- verificar se os caracteres da lista formam um palíndromo retornando true, se não for retorna false;
- gerar e retornar uma String codificada somando um em cada caractere.

EXERCÍCIO

4) Implemente uma lista de números reais em alocação estática com todas as operações indicadas a seguir. A lista possui um vetor para guardar os caracteres e um atributo para guardar a quantidade de elementos presentes na lista.

- adicionar um novo elemento no final da lista, caso a operação não possa ser realizada, mostre mensagem de erro;
- remover um elemento do final da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem de erro;
- adicionar um novo elemento no início da lista, caso a operação não possa ser realizada, mostre mensagem de erro;
- remover um elemento do início da lista, retornando o elemento que foi removido, caso a operação não possa ser realizada, mostre mensagem de erro;
- concatenar os elementos em uma String e retorná-la.

REFERÊNCIAS BIBLIOGRÁFICAS

EDELWEISS, N; GALANTE, R. **Estruturas de Dados**. Livros Didáticos UFRGS, V.18. Bookman, 2009.

GUIMARÃES, A. de M.; LAGES, N. A. **Algoritmos e Estruturas de Dados**. Livros Técnicos e Científicos, 1994.

LAFORE, Robert. **Estrutura de Dados & Algoritmos em Java**. Rio de Janeiro: Ed. Ciência Moderna, 2004.

KOFFMANN, E. B. **Objetos, abstração, estrutura de dados e projeto**. LTC, 2008.

SHILDT, Hebert. **C Completo e Total**. McGraw Hil, 1991.

TENENBAUM, Aaron; LANGSAM, Yedidyah; AUGENSTEIN, Moshe J. **Estruturas de Dados Usando C**. 1a. Ed. São Paulo: Makron Books, 1995.