

LISTAS
Parte 3

Profa. Me. Andréa Zotovici

São Paulo
2018

ALOCAÇÃO ESTÁTICA

L1:ListaDeInteiros			
Posição	1ª	2ª	3ª
Dados	0	1	2
tamanho: 0			

Quando a lista está vazia, o novo elemento pode ser adicionado apenas na 1ª. Posição. Quando houver algum elemento na lista, ele poderá ser adicionado desde a 1ª posição até a posição = tamanho + 1.

ALOCAÇÃO ESTÁTICA

Dada a Lista Vazia

L1:ListaDeInteiros			
dados	0	1	2
tamanho: 0			

Adicione o elemento 15 na 1ª posição da lista, processo:

- A posição é válida?
- Sim
- 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

L1:ListaDeInteiros			
	0	1	2
dados	15		
tamanho: 1			

Adicione o elemento 30 na 2ª posição da lista, processo:

- A posição é válida?
- Sim
- Lista está cheia?
- Não:
 - guarde 30 no vetor denominado dados, no índice equivalente a posição - 1
 - some 1 em tamanho

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

ALOCAÇÃO ESTÁTICA

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

Adicione o elemento 53 na 1ª posição da lista, processo:

- A posição é válida?
- Sim
- Lista está cheia?
- Não:
 - passe cada elemento para o próximo índice, a partir do último, até o índice posição - 1
 - guarde 53 no vetor denominado dados, no índice equivalente a posição - 1
 - some 1 em tamanho

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

ALOCAÇÃO ESTÁTICA

Dada a Lista

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

Adicione o elemento 47 na 2ª posição da lista, processo:

- A posição é válida ?

Sim

- Lista está cheia?

- Sim:

- Mostre a mensagem “Lista cheia”

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

ALOCAÇÃO ESTÁTICA

L1:ListaDeInteiros			
Posição	1ª	2ª	3ª
Dados	0	1	2
tamanho: 0			

Se a lista não estiver vazia, a operação remove poderá ser realizada desde a 1ª posição até a posição = tamanho .

ALOCAÇÃO ESTÁTICA

Dada a Lista

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

Remova o elemento da 2ª. posição da lista, processo:

- A posição é válida?
- Sim
- Lista está vazia?
- Não
 - guarde o elemento da 2ª. posição (índice = posição - 1) da lista em uma variável
 - passe cada elemento para um índice anterior, a partir de índice = posição
 - decremente um de tamanho
 - retorne o elemento removido

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

retorna 30

Dada a Lista

ALOCAÇÃO ESTÁTICA

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

Remova o elemento da 1ª. posição da lista, processo:

- A posição é válida?
- Sim
- Lista está vazia?
- Não
 - guarde o elemento da 1ª. posição (índice = posição - 1) da lista em uma variável
 - passe cada elemento para um índice anterior, a partir de índice = posição
 - decrémente um de tamanho
 - retorne o elemento removido

L1:ListaDeInteiros			
	0	1	2
dados	53		
tamanho: 1			

retorna 15

Dada a Lista

ALOCAÇÃO ESTÁTICA

L1:ListaDeInteiros			
	0	1	2
dados	53		
tamanho: 1			

Remova o elemento da 1ª. posição da lista, processo:

- A posição é válida?
- Sim
- Lista está vazia?
- Não
 - guarde o elemento da 1ª. posição (índice = posição - 1) da lista em uma variável
 - decremente um de tamanho
 - retorne o elemento removido

L1:ListaDeInteiros			
	0	1	2
dados	53		
tamanho: 1			

retorna 15

EXERCÍCIO

1) 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.adiciona(74,2)
- b) l1.adicionaFinal(74)
- c) l1.adiciona(73, 1)
- d) l1.adicionaInicio (72)
- e) l1.remove (2)
- f) l1.adiciona (70, 4)
- g) l1.remove(1)
- h) l1.removeInicio()
- i) l1.removeFinal ()
- j) l1.adiciona (67,2)
- k) l1.removeFinal ()

EXERCÍCIO

- 2) Implemente os métodos adiciona e remove de posição específica para cada lista que foi implementada em exercícios anteriores.
- 3) Implemente uma classe lista de números inteiros que adicione elementos em ordem crescente. Aplique essa lista para resolver o problema 1259 – Pares e Ímpares do urionlinejudge.
- 4) Utilize a classe ListaDeInteiros para resolver o problema 1110 – Throwing Cards Away do urionlinejudge. Escolha adequadamente as operações para reduzir o tempo de execução do algoritmo.

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.