### **LISTAS**

Profa. Me. Andréa Zotovici

São Paulo 2018

Tipo Abstrato de Dados.

É descrito por propriedades e operações.

Quando utilizado não é necessário conhecer detalhes da sua implementação, apenas as suas operações.

Encapsula os dados necessários e as operações que podem ser realizadas sobre os dados.

Referências Complementares:

ICMC-USP. Disponível em <a href="http://www.lcad.icmc.usp.br/~nonato/ED/node6.html">http://www.lcad.icmc.usp.br/~nonato/ED/node6.html</a>.

Acesso em: 05/03/2018.

Quais informações manipulamos diariamente como uma coleção?

Quais informações manipulamos diariamente como uma coleção?

- Alunos
- Cinemas
- Contatos Pessoais
- Contatos Profissionais
- Escolas
- Livros
- Presentes

Quais operações podem ser realizadas sobre as coleções citadas?

### Operações:

- Busca;
- Combinação de duas listas em uma;
- Intercalação;
- Inserção;
- Ordenação;
- Remoção;
- etc.....

Qual é a relação entre listas e tipo abstrato de dados?

Lista é um tipo abstrato de dados.

A Lista pode possuir elementos:

- com tipo primitivo de dados, como uma lista de números inteiros, uma lista de letras ou uma lista de números reais;
- com tipo abstrato de dados, como uma lista de alunos, lista de contatos, lista de cinemas e etc...

### LISTA COM ELEMENTOS DO TIPO PRIMITIVO DE DADOS

### ListaDeInteiros

- dados: int[]

- tamanho: int

+ ListaDeInteiros(c: int)

+ vazia(): boolean

+ cheia(): boolean

+ adicionaInicio(e: int): void

+ adicionaFinal(e: int): void

+ adiciona(int e, int posicao): void

+ removeInicio(): int

+ removeFinal(): int

+ remove(int posicao): int

+ obtemPrimeiro(): int

+ obtemUltimo(): int

L1:ListaDeInteiros							
	0	1	2	3			
dados	10	8	12				
tamanho	3						

### LISTA COM ELEMENTOS DO TIPO ABSTRATO DE DADOS

Dada a lista de alunos

	listaAlunos: I	ListaDeAlunos		
	0	1	2	
dados	Nome: "Adriana" RA: 12135567	Nome: "Adriano" RA: 13100314		
tamanho	2			

### LISTA COM ELEMENTOS DO TIPO ABSTRATO DE DADOS

### ListaDeAlunos

- dados: Aluno[]

- tamanho: int

+ ListaDeAlunos(c: int)

+ vazia(): boolean

+ cheia(): boolean

+ adicionalnicio(e: Aluno): void

+ adicionaFinal(e: Aluno): void

+ adiciona(Aluno e, int posicao): void

+ removeInicio(): Aluno

+ removeFinal(): Aluno

+ remove(int posicao): Aluno

+ obtemPrimeiro(): Aluno

+ obtemUltimo(): Aluno

### ESTÁTICA X DINÂMICA

Alocação Estática

Quantidade constante de elementos.

Aloca espaço de acordo com a quantidade de elementos.

Usa arrays.

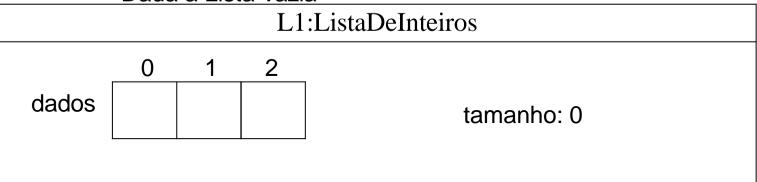
Alocação Dinâmica

Não há quantidade máxima de elementos (o limite é a memória do computador).

Utiliza somente o espaço de memória suficiente para determinado elemento.

Ponteiros ou Referências.

Dada a Lista Vazia



Adicione o elemento 15 no início da lista, processo:

- Lista está cheia?
- Não:
  - guarde 15 no vetor denominado dados, índice 0
  - some 1 em tamanho

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

Dada a Lista

L1:ListaDeInteiros

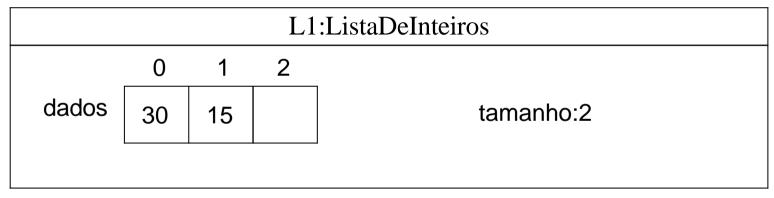
dados

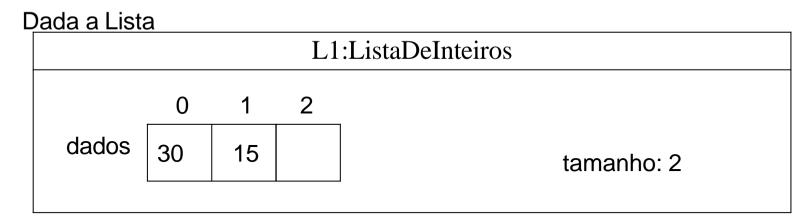
15

tamanho: 1

Adicione o elemento 30 no início da lista, processo:

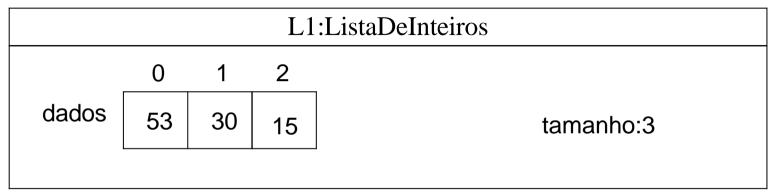
- Lista está cheia?
- Não:
  - passe, a partir do último elemento, todos os elementos uma posição à frente
  - guarde 30 no vetor denominado dados, índice 0
  - some 1 em tamanho



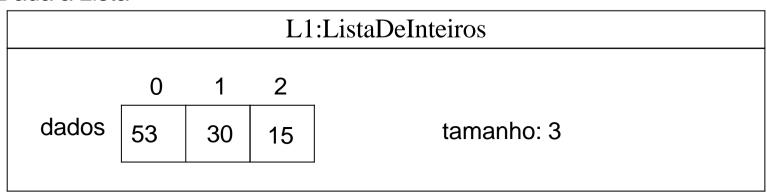


Adicione o elemento 53 no início da lista, processo:

- Lista está cheia?
- . Não:
  - passe, a partir do último elemento, todos os elementos uma posição à frente
  - guarde 53 no vetor denominado dados, índice 0
  - some 1 em tamanho



#### Dada a Lista

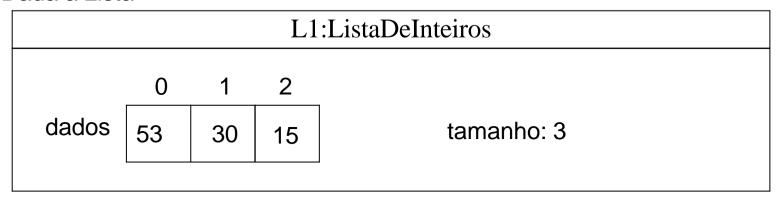


Adicione o elemento 47 no início da lista, processo:

- Lista está cheia?
- Sim:
  - Mostre a mensagem "Lista cheia"

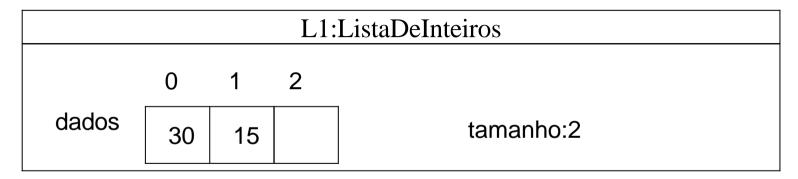
L1:ListaDeInteiros						
	0	1	2			
dados	53	30	15	tamanho:3		

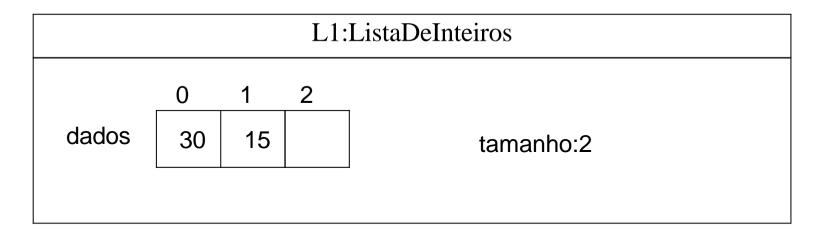
#### Dada a Lista



Remova o elemento do início da lista, processo:

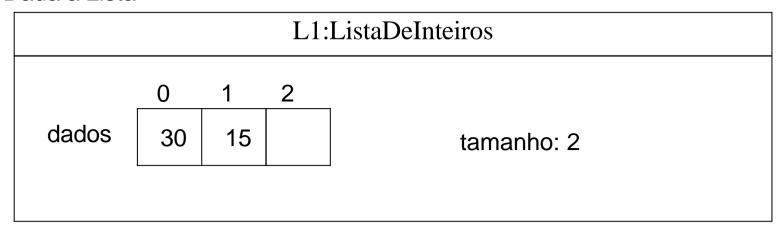
- Lista está vazia?
- Não
  - guarde o elemento do início da lista em uma variável
  - passe cada elemento um índice à esquerda, a partir do índice 1
  - decremente um de tamanho





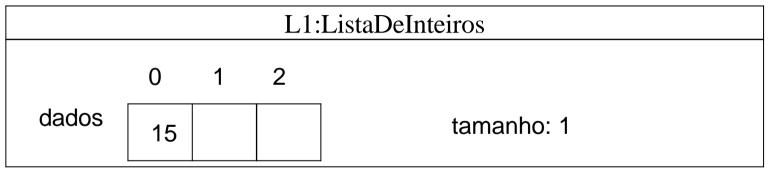
Os elementos da lista são os que estão a partir do índice 0 (zero) até o índice equivalente tamanho – 1, portanto: 30 e 15

#### Dada a Lista



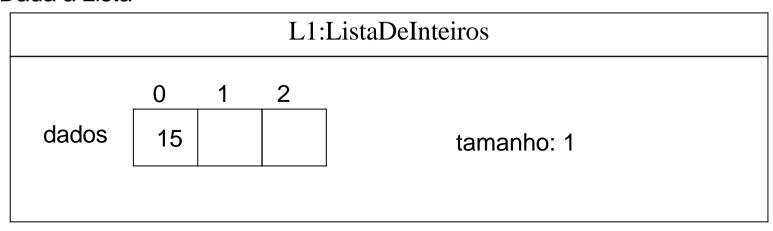
Remova o elemento do início da lista, processo:

- Lista está vazia?
- Não
  - guarde o elemento do início da lista em uma variável
  - passe cada elemento um índice à esquerda, a partir do índice 1
  - decremente um de tamanho



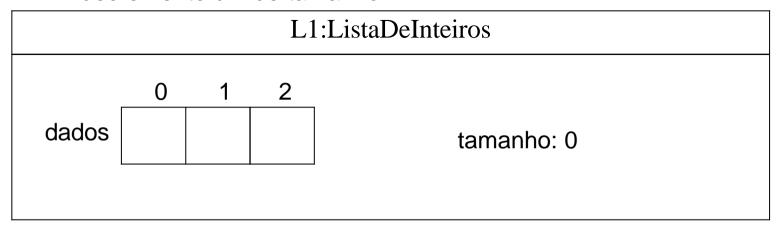
19

#### Dada a Lista

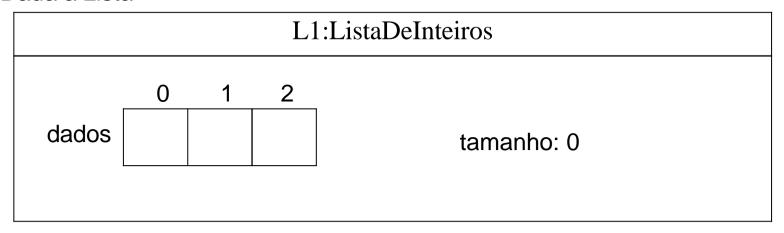


Remova o elemento do início da lista, processo:

- Lista está vazia?
- Não
  - guarde o elemento do início da lista em uma variável
  - passe cada elemento um índice à esquerda, a partir do índice 1
  - decremente um de tamanho



#### Dada a Lista



Remova o elemento do início da lista, processo:

- Lista está vazia?
- <sub>-</sub> Sim
  - mostre a mensagem "ERRO! Lista Vazia"

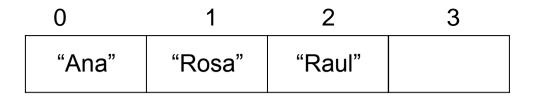
### Exemplo de Lista com RA e Nome de Alunos:

```
public class ListaDeAlunos {
         private Aluno[] dados;
         private int tamanho;
         public ListaDeAlunos(int capacidade) {
             dados = new Aluno[capacidade];
             tamanho = 0;
         public boolean vazia() {
             return tamanho == 0;
         public boolean cheia() {
             return tamanho == dados.length;
```

```
public void adicionaFinal(Aluno aluno) {
    if (!cheia()) {
       dados[tamanho] = aluno;
       tamanho++;
     } else
      System.out.println("Lista Cheia!");
public Aluno removeFinal ( ) {
    Aluno aluno=null;
     if (!vazia()) {
         tamanho--;
         aluno = dados[tamanho];
    return aluno;
```

### **EXERCÍCIO**

1) Considere a lista de nomes por alocação estática de memória:



tamanho=3

a) Qual será a situação da lista e do total de alunos da lista (ou mensagem impressa) se o método adicionaFinal("Sandra"), for invocado?

0	1	2	3
"Ana"	"Rosa"	"Raul"	

tamanho=\_\_\_\_\_

## **EXERCÍCIO**

1) Continu	ıação -	preenc	ha toda	as as po	sições
b) Qual será de alunos da método adicio	lista (	ou men	sagem i	mpressa	a) se o
	0	1	2	3	
ta	amanho=				
c) Qual será de alunos da método remove	lista (	ou men	sagem i	mpressa	
	0	1	2	3	

tamanho=\_\_\_\_

### **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) I1.adicionalnicio(74)
- b) I1.adicionalnicio(73)
- c) I1.adicionalnicio(72)
- d) I1.adicionalnicio(71)
- e) I1.adicionalnicio(70)
- f) I1.removelnicio()
- g) I1.removeInicio()
- h) I1.removeInicio()
- i) 11.removelnicio()
- j) 11.removelnicio()

### 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.