

Tipos de Dado Abstrato: Listas, Filas e Pilhas

Estruturas de Dados

Prof. Me Elenilton J Dezengrini

Introdução

- Tradicionalmente conhecidos como Tipos de Dado Abstrato, são algumas Estruturas de Dados básicas e importantes para a construção de algoritmos mais bem elaborados;

Listas, Filas e Pilhas

LISTAS - INTRODUÇÃO

Listas

- Listas são conjuntos de elementos, objetos, variáveis, tarefas, ou qualquer coisa que se possa enumerar e formar um conjunto;
- As listas estão presentes em nossa vida, desde o nosso nascimento, por exemplo, com a lista de compras que nossos pais tiveram que fazer para nós.

Listas

- Exemplo de Lista de Compras:
 - 5Kg de farinha;
 - 2Kg de açúcar;
 - 500g de carne moída;
 - 2Kg de arroz;
 - 4L de leite;
 - 1Kg de feijão;
 - Etc..

Listas

- Exemplo de Lista Telefônica:
 - Asdf de Zxcv: (44) 4444-4444
 - Beutrano Cruz: (33) 3333-3333
 - Ciclano da Silva: (22) 2222-2222
 - Fulano de Tal: (11) 1111-1111

Listas, Filas e Pilhas

COMPORTAMENTO DE UMA LISTA

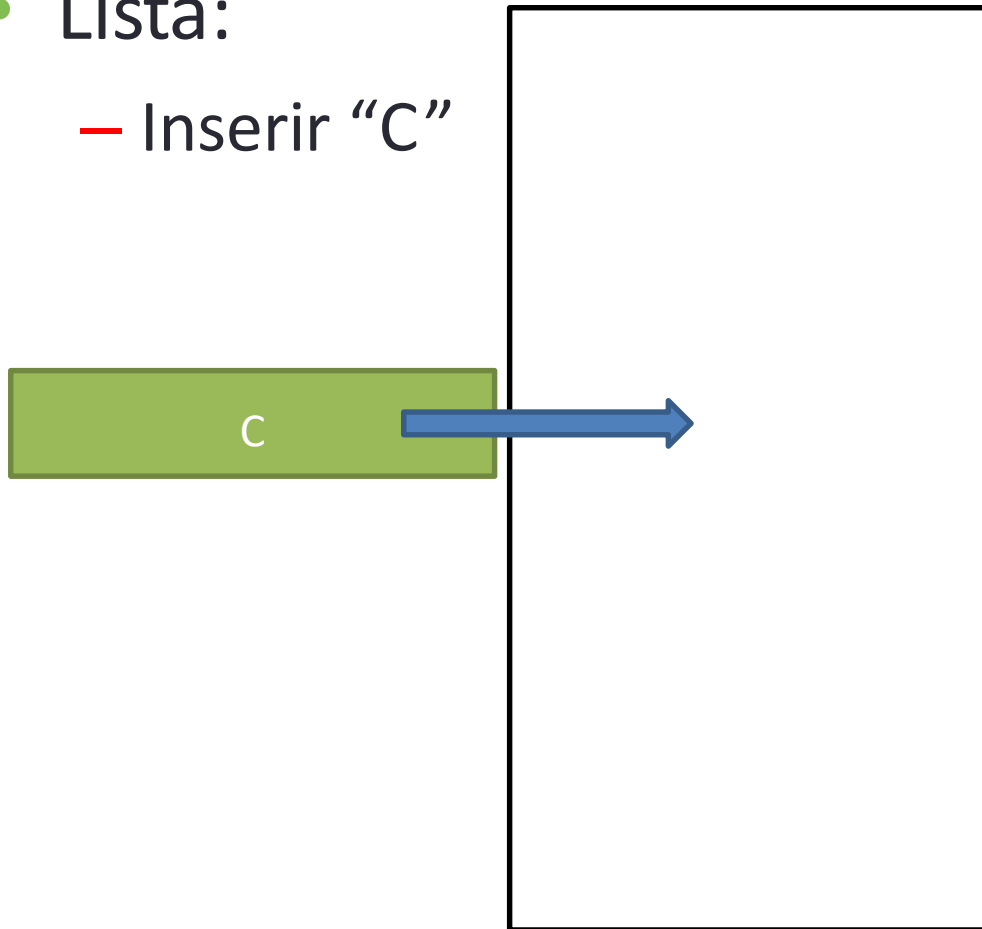
Comportamento de uma Lista

- Lista: Vazia!



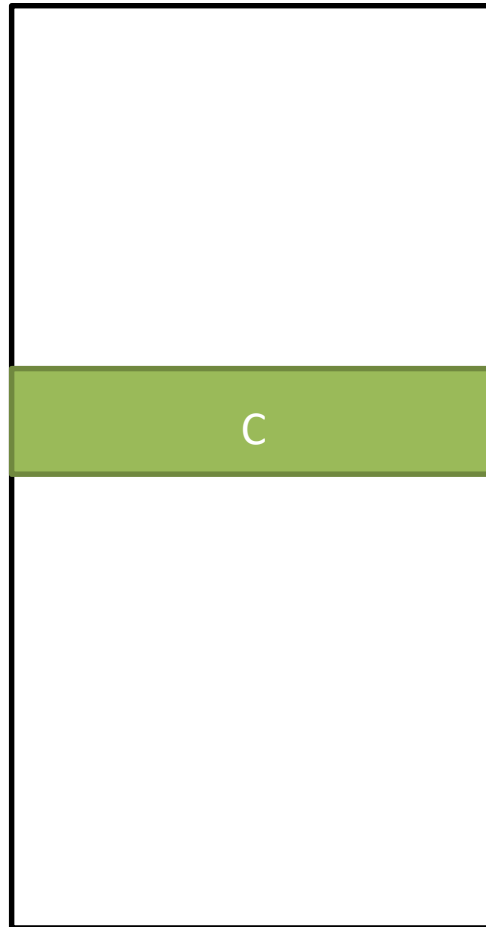
Comportamento de uma Lista

- Lista:
 - Inserir “C”



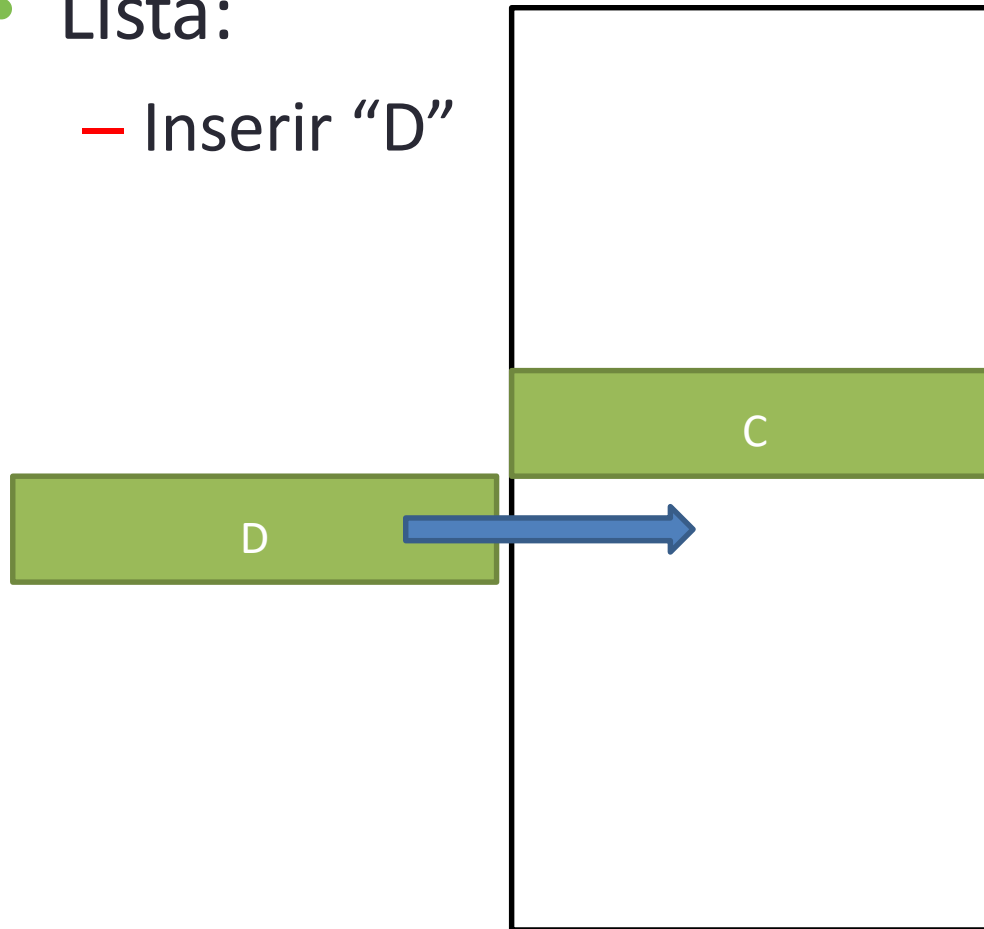
Comportamento de uma Lista

- Lista:



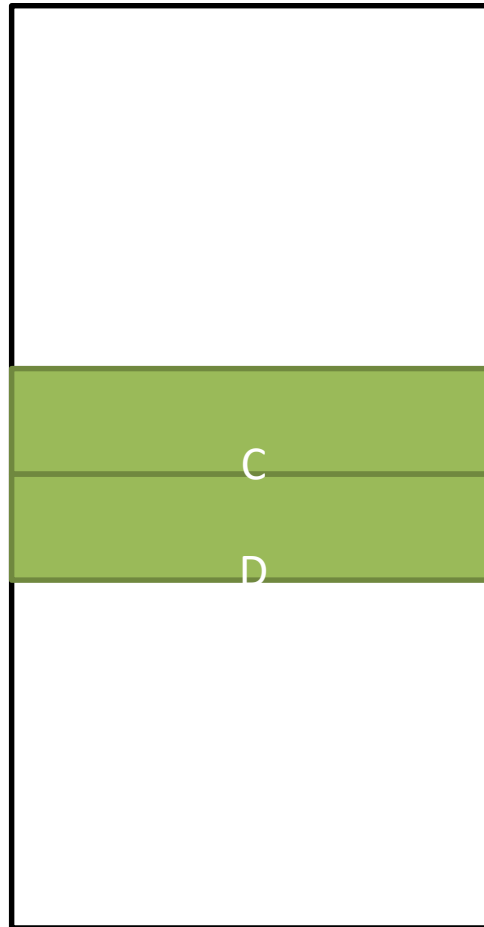
Comportamento de uma Lista

- Lista:
 - Inserir “D”



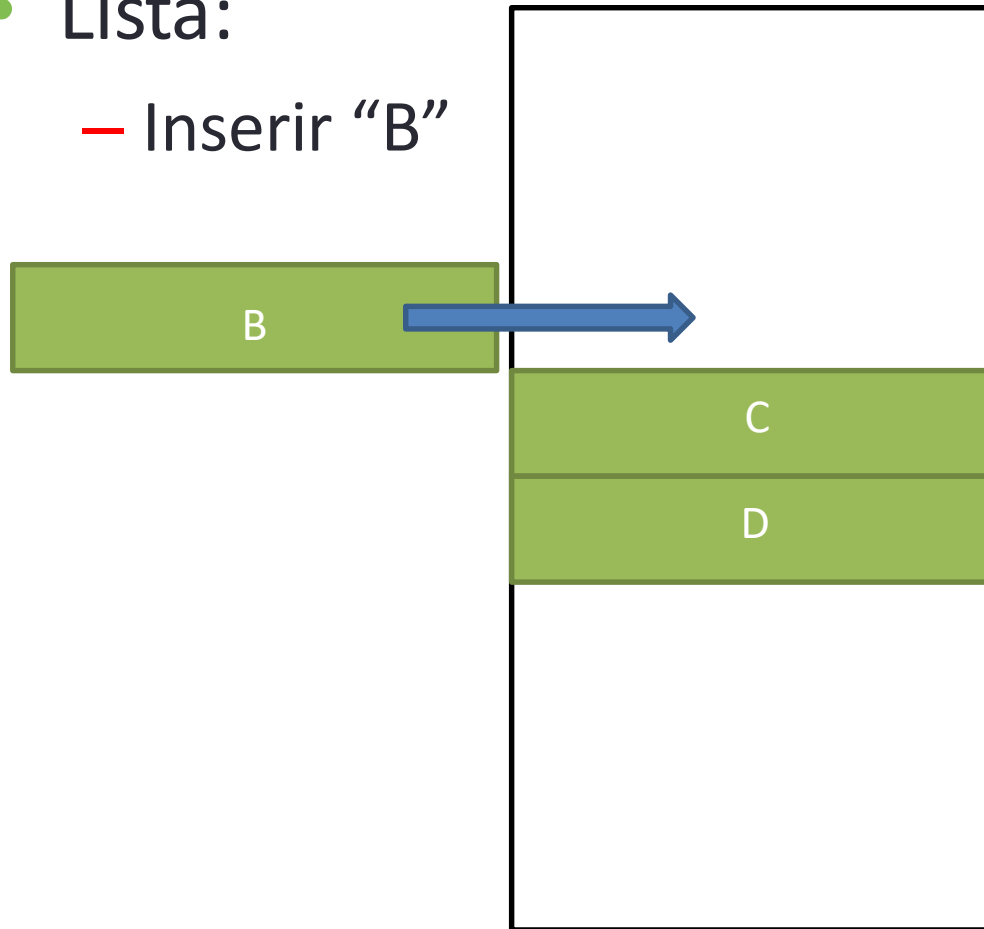
Comportamento de uma Lista

- Lista:



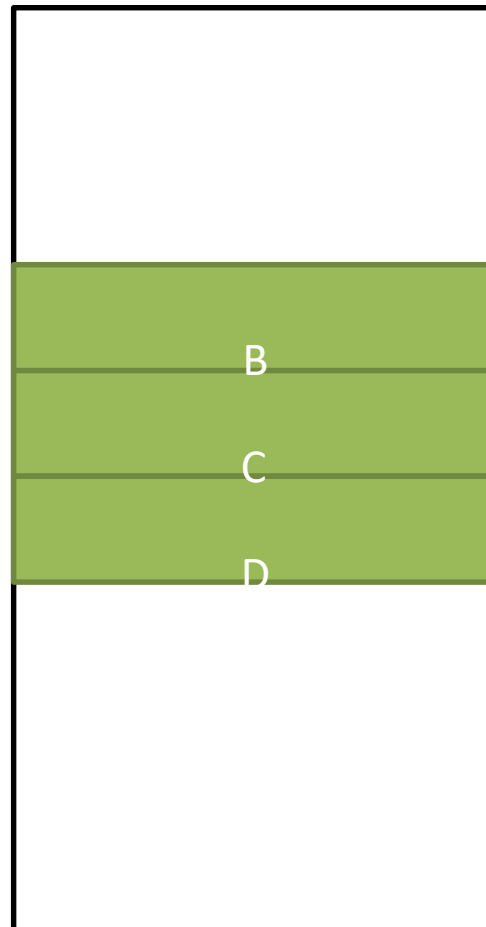
Comportamento de uma Lista

- Lista:
 - Inserir “B”



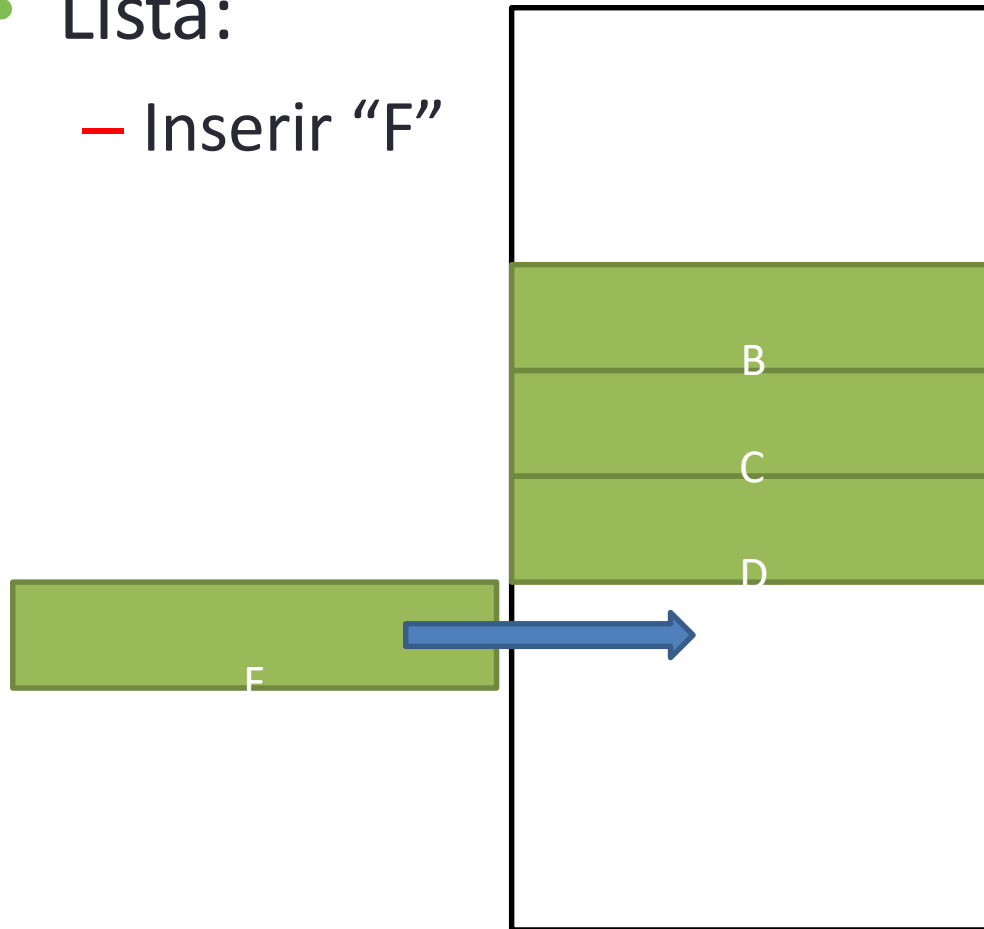
Comportamento de uma Lista

- Lista:



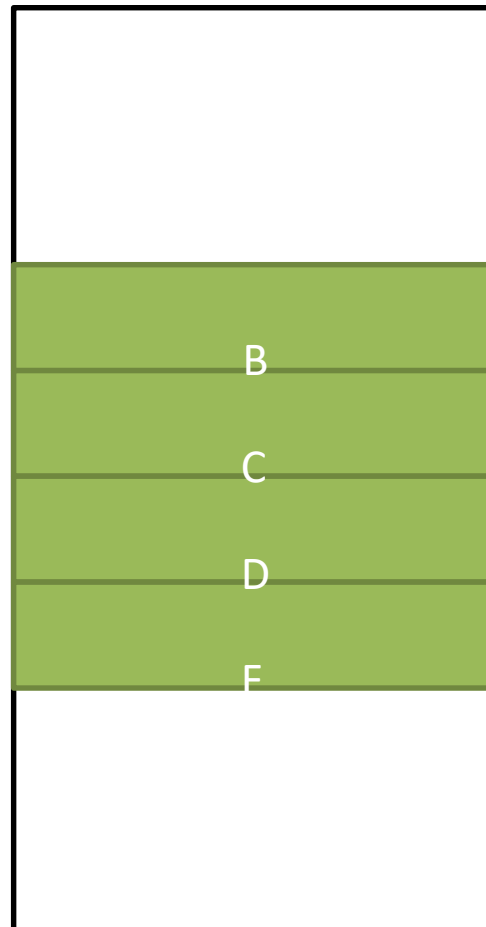
Comportamento de uma Lista

- Lista:
 - Inserir “F”



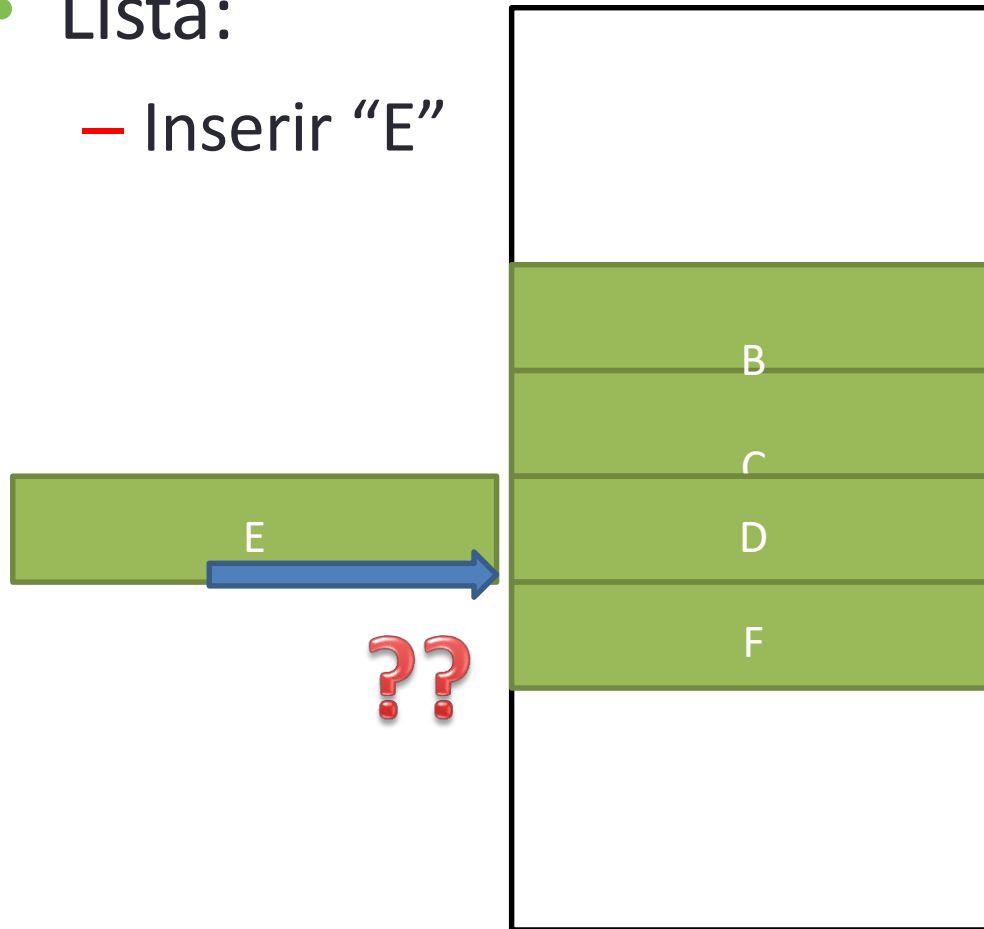
Comportamento de uma Lista

- Lista:



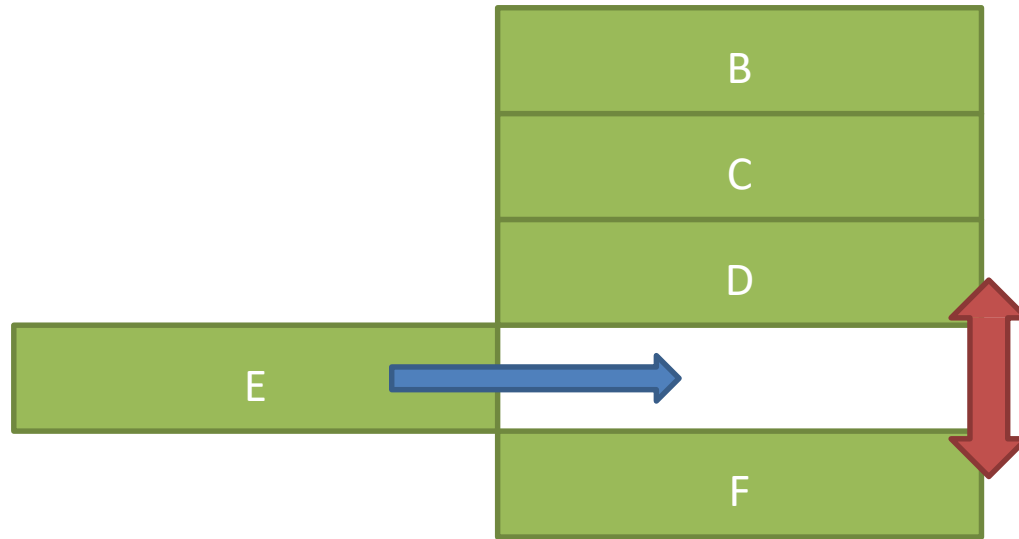
Comportamento de uma Lista

- Lista:
 - Inserir “E”



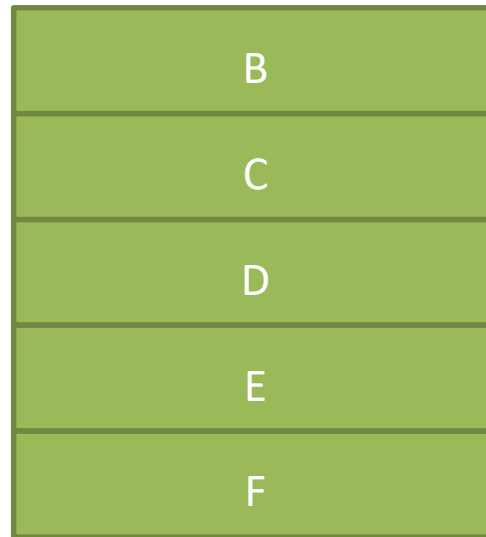
Comportamento de uma Lista

- Lista:



Comportamento de uma Lista

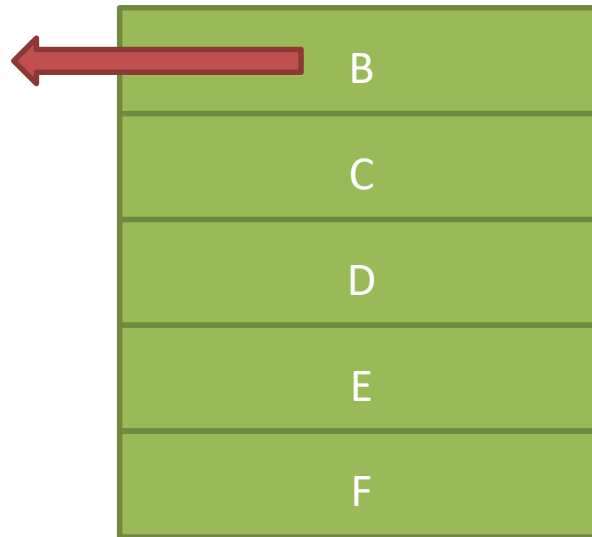
- Lista:



B
C
D
E
F

Comportamento de uma Lista

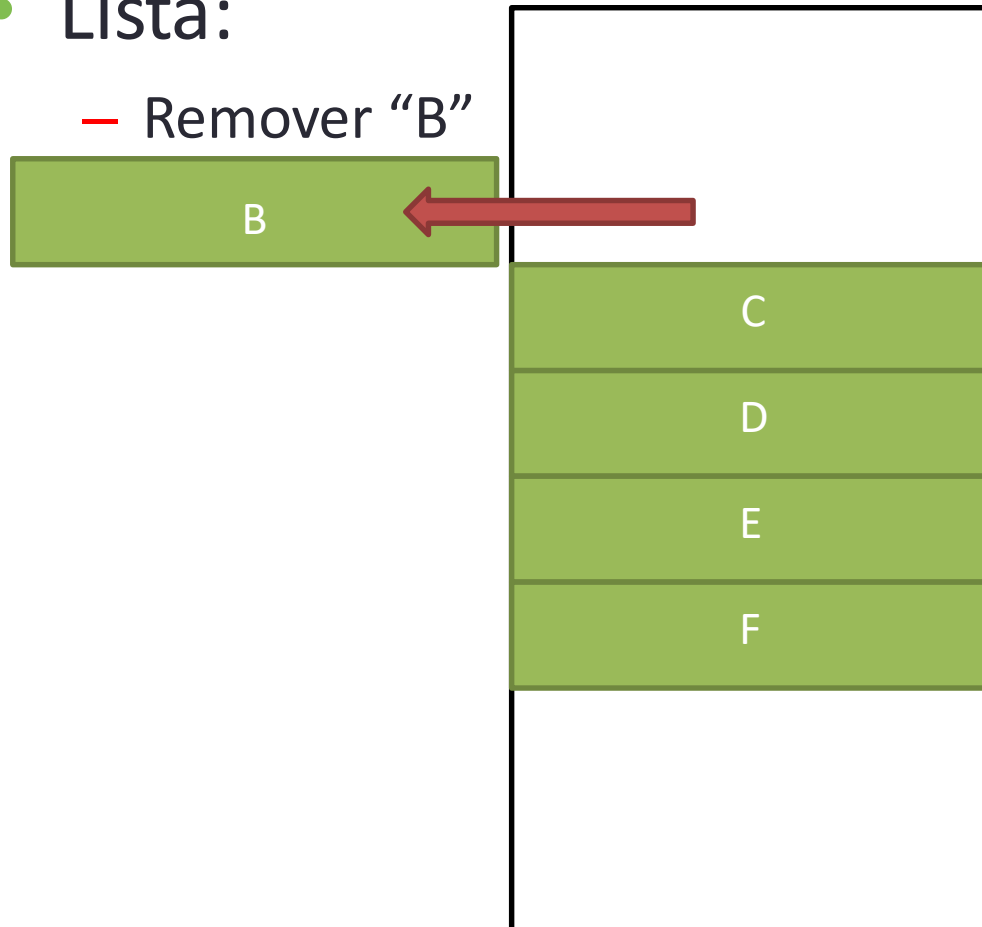
- Lista:
 - Remove “B”



Comportamento de uma Lista

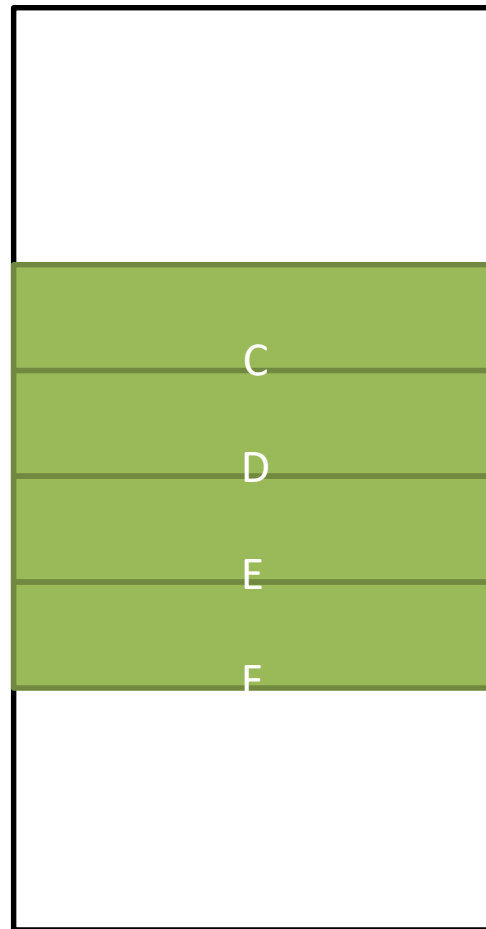
- Lista:

— Remover “B”



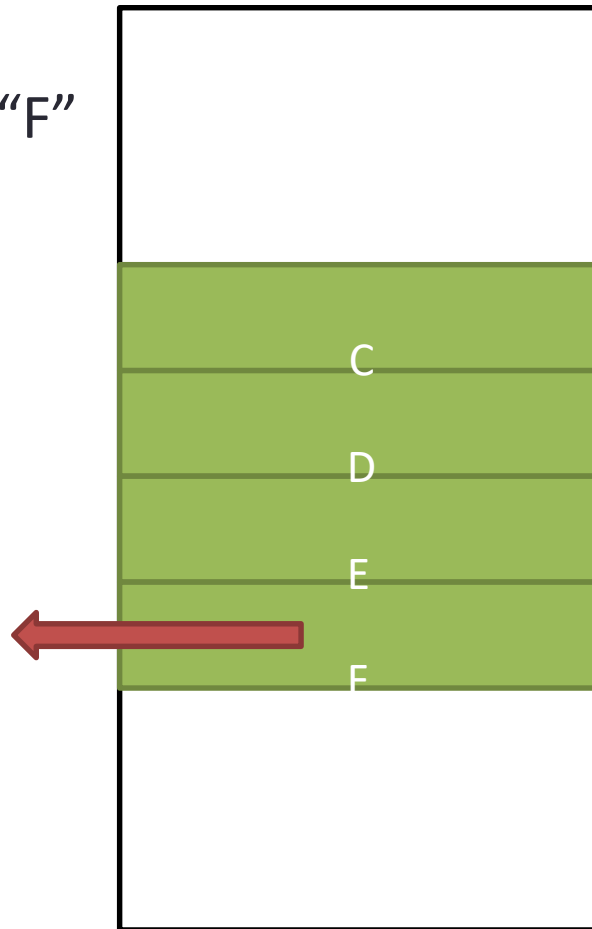
Comportamento de uma Lista

- Lista:



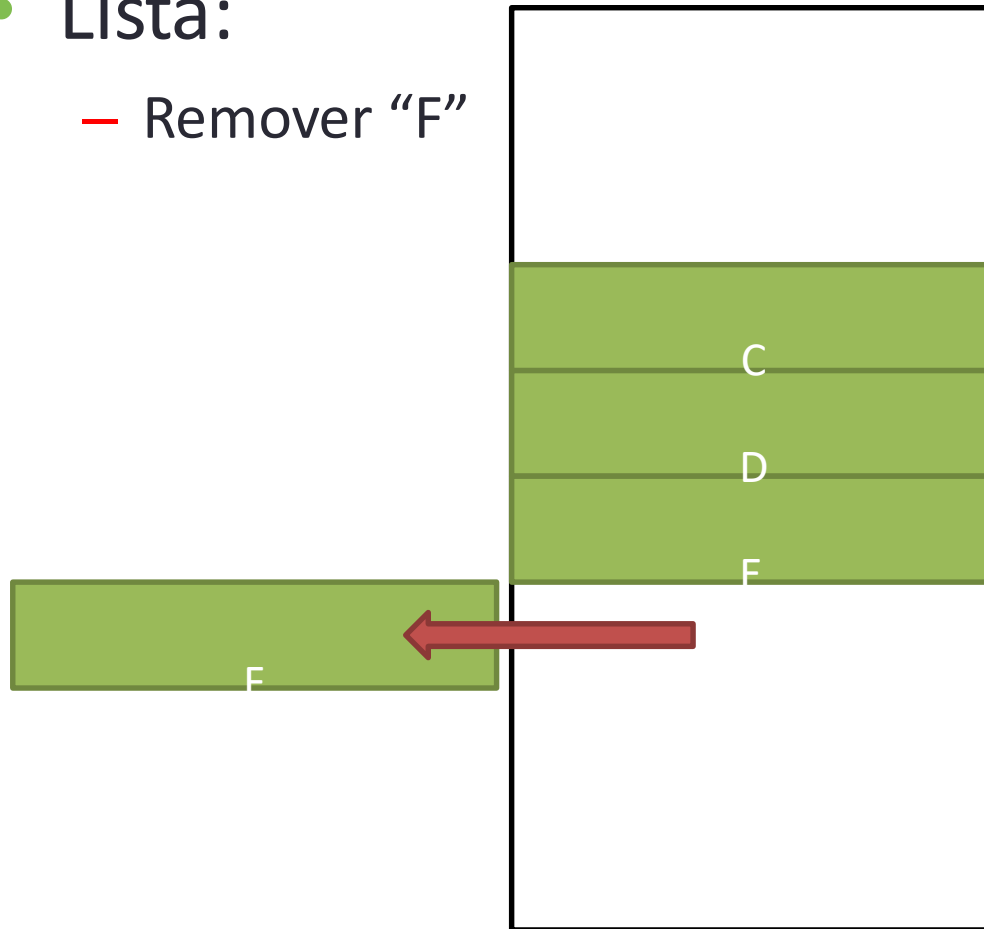
Comportamento de uma Lista

- Lista:
 - Remover “F”



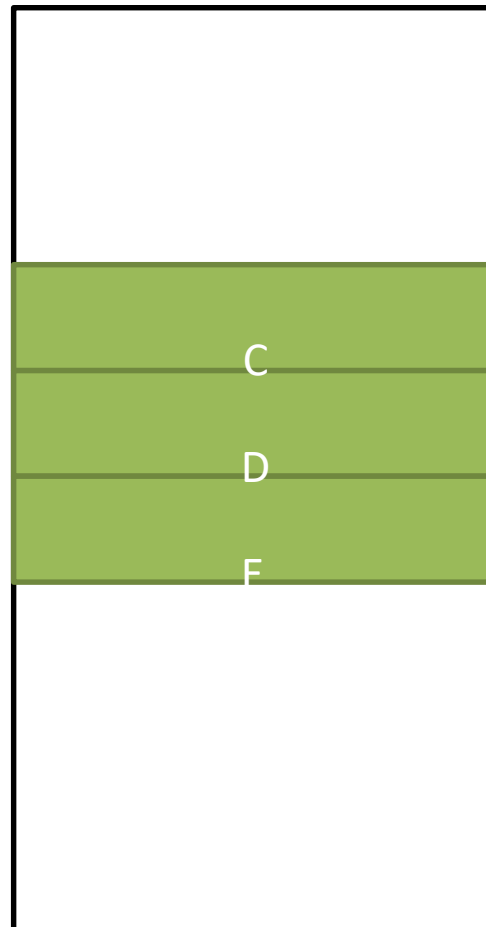
Comportamento de uma Lista

- Lista:
 - Remover “F”



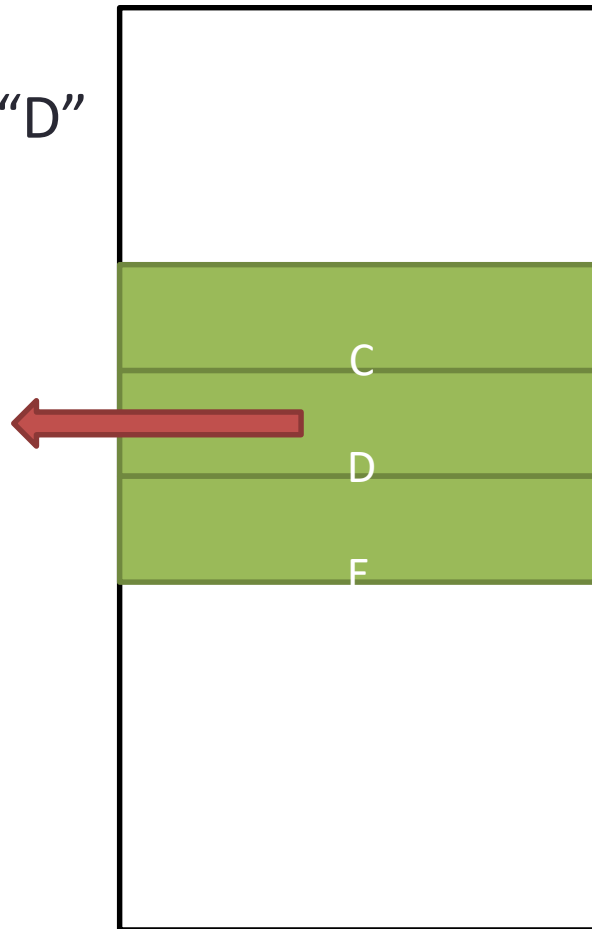
Comportamento de uma Lista

- Lista:



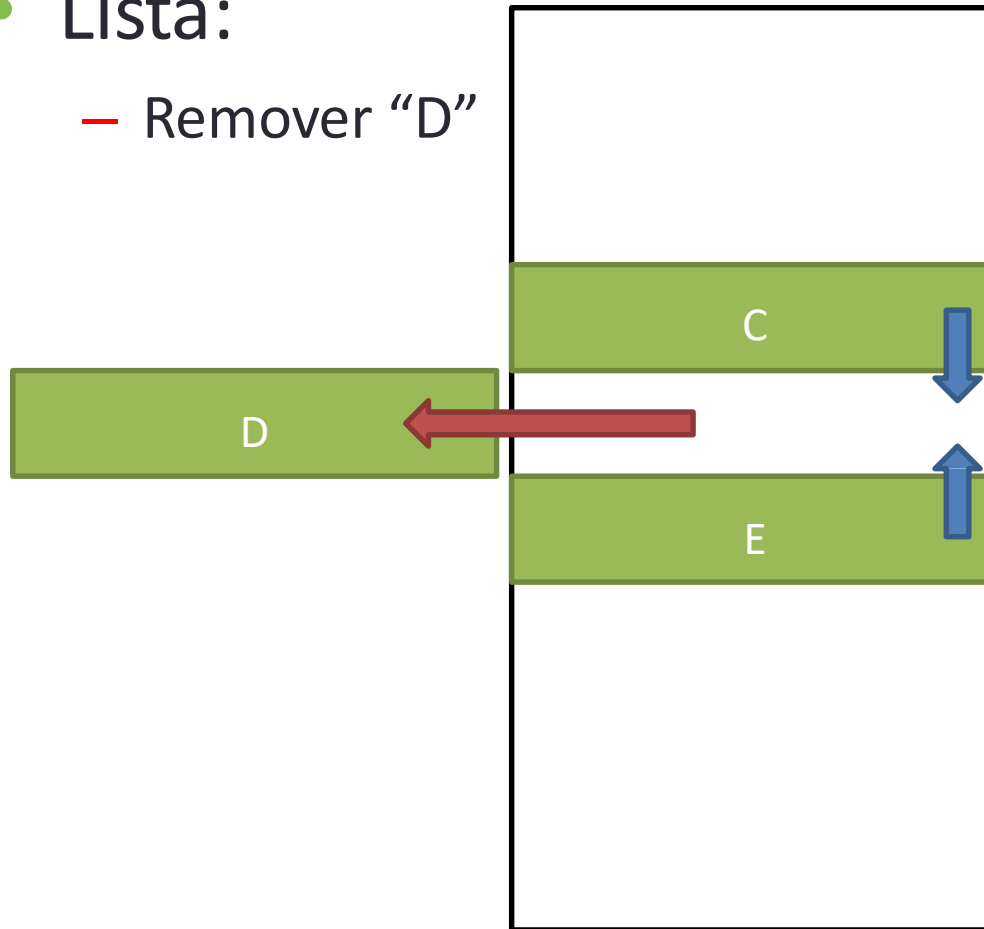
Comportamento de uma Lista

- Lista:
 - Remover “D”



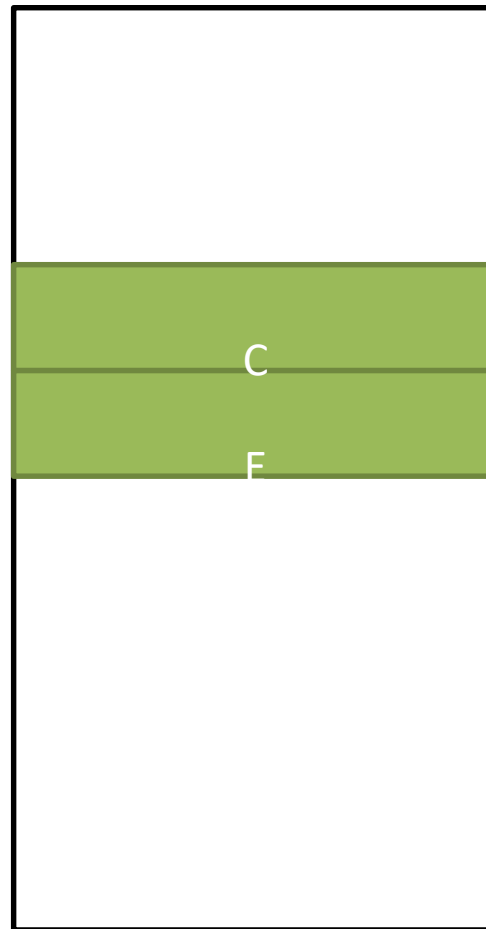
Comportamento de uma Lista

- Lista:
 - Remover “D”



Comportamento de uma Lista

- Lista:



Listas, Filas e Pilhas

LISTAS - IMPLEMENTAÇÃO

Listas

- Implementando as listas:
 - As listas podem ser implementadas de várias formas, mas num aspecto mais geral podemos separar em duas principais:
 - Em **Arrays**; ou
 - **Encadeadas**.

Listas em Arrays

- **Em Arrays:**

- Imagine que a lista anterior tinha posições fixas e pré-determinadas:
 - Um array é uma estrutura com posições fixas, cada elemento da lista deve ser colocado em uma posição no array;
 - Ao inserir ou excluir um elemento, talvez seja necessário realocar todos os demais elementos.

[illegible]

Listas em Arrays

- Prós:

- Criar um array de qualquer tamanho é muito simples;
- Não há necessidade de compreender ponteiros ou referências;

- Contras:

- Limitações quanto ao tamanho de memória;
- Custo computacional maior;
- Alocação de memória exagerada.

Listas Encadeadas

- Encadeado, Dicionário Houaiss:
 - *adjetivo*
 - 1. disposto ou ligado por ou como por cadeias; ordenado, junto;
 - 2. preso, submetido;

Listas Encadeadas

- Prós:
 - Extremamente eficiente no custo de memória e de processamento;
 - Nunca acarreta em movimentar todos os elementos;
- Contras:
 - Envolve conceitos mais avançados de programação:
 - Ponteiros ou Referências.

Listas Encadeadas

- Para criarmos uma lista encadeada, precisamos primeiro definir o que será armazenado nela;
- Por exemplo, para criarmos uma lista de contatos, gostaríamos de armazenar os nomes, telefones e e-mails de diversas pessoas:

Listas Encadeadas

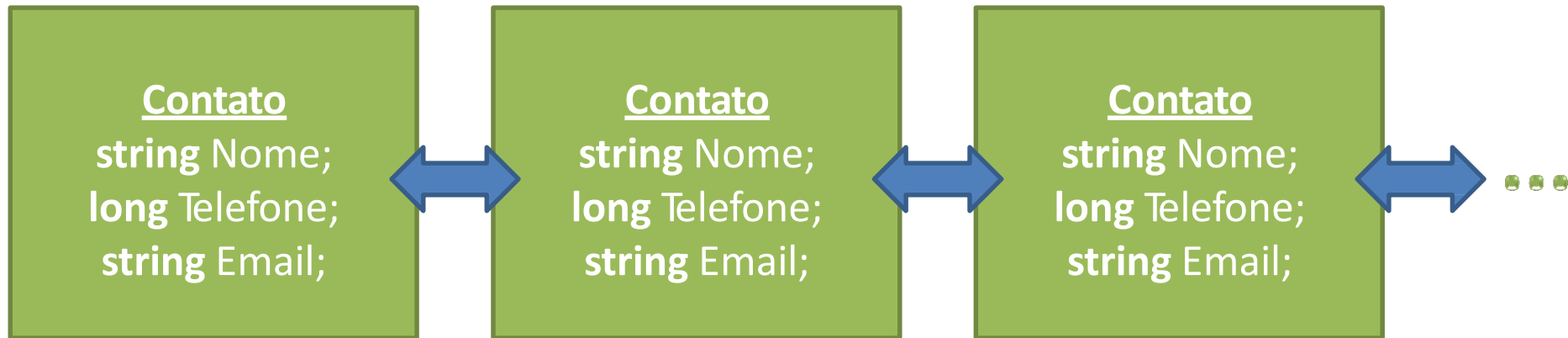
- Exemplo de elemento “Contato” da lista:

Contato

```
string Nome;  
long Telefone;  
string Email;
```

Listas Encadeadas

- Exemplo da Idéia de Encadeamento:



- Mas como fazer isto?

Listas Encadeadas

- Conforme vamos criando elementos na memória do computador, estes elementos vão ficando espalhados e desconexos;
- Para criar listas encadeadas precisamos criar elementos que façam **referência** a outro elemento, ou seja, indiquem onde podemos encontrar um outro elemento.

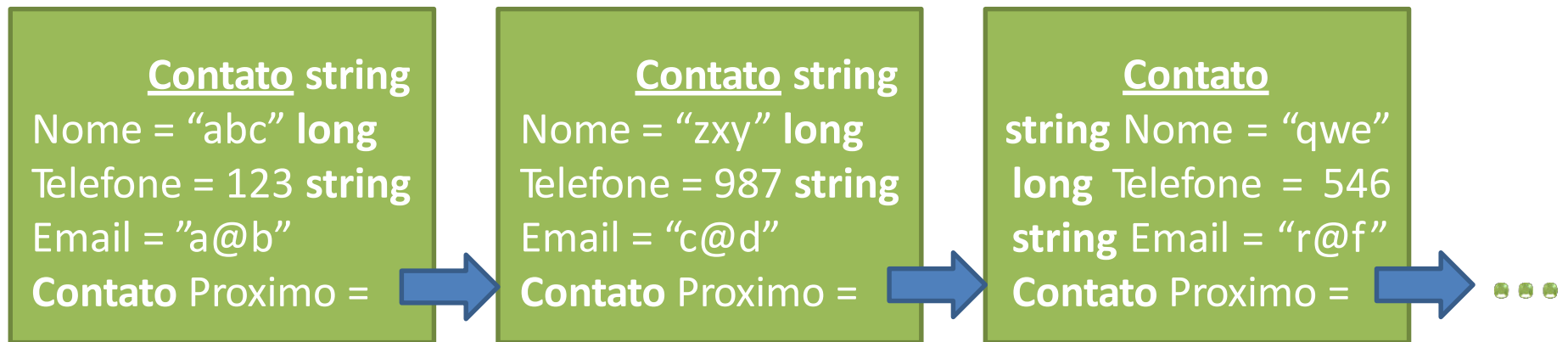
Listas Encadeadas

- Exemplo de elemento encadeado:

```
Contato  
string Nome;  
long Telefone;  
string Email;  
Contato Proximo;
```

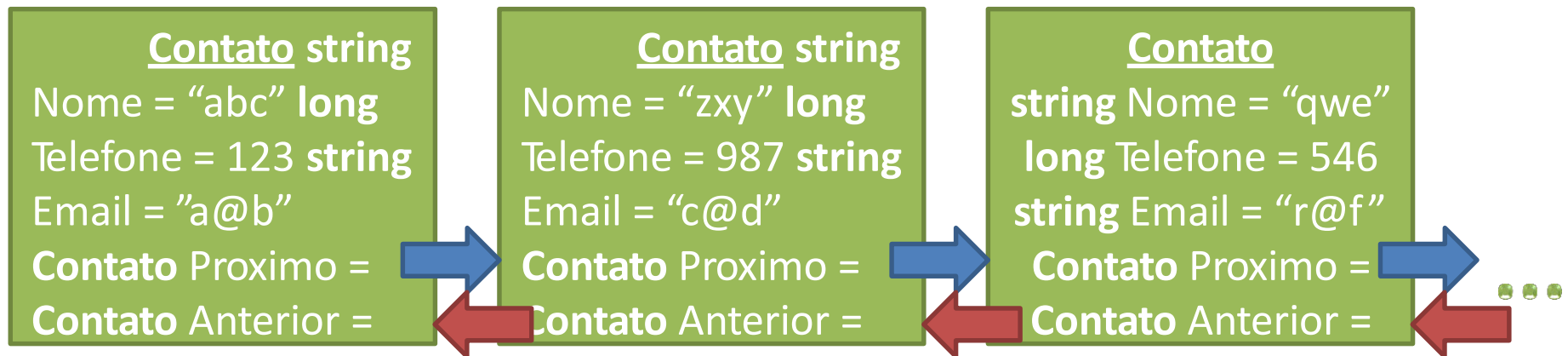
Listas Encadeadas

- Exemplo com Elemento **Encadeado**:



Listas Encadeadas

- Exemplo **Duplamente** Encadeado:



Listas Encadeadas

- Iniciando uma lista vazia:
 - **Contato** Inicio_Lista = **null**;
 - **Contato** Fim_Lista = **null**;
 - O “*valor*” de referência **null** é usado para quando ainda **não existe um objeto** na memória para qual a variável irá fazer referência;
 - O último elemento da lista aponta para **null**.
- Iniciando uma lista com 1 elemento:
 - **Contato** Inicio_Lista = **new Contato()**;

Listas Encadeadas

— Criando a Lista:

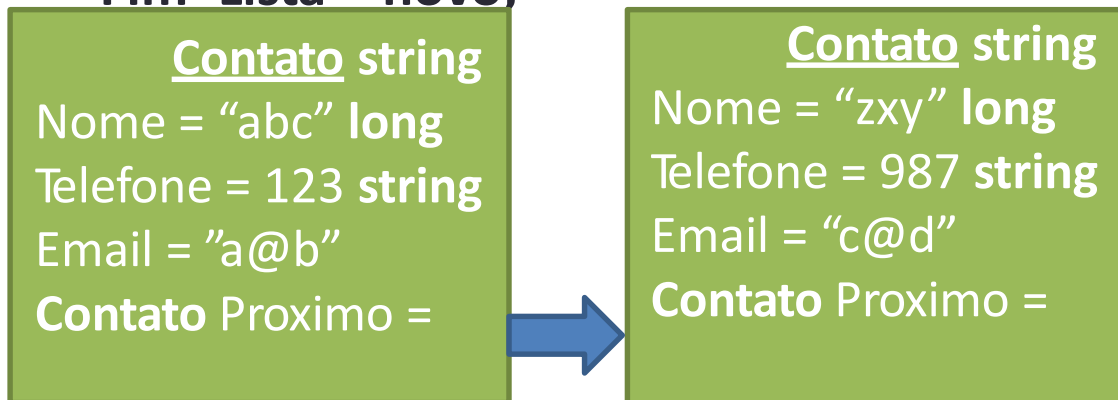
- **Contato** Inicio_Lista = **new Contato()**;
- **Contato** Fim_Lista = Inicio_Lista;
- Inicio_Lista.Nome = "abc";
- Inicio_Lista.Telefone = 123;
- Inicio_Lista.Email = "a@b";
- Inicio_Lista.Proximo = **null**;

```
Contato string  
Nome = "abc" long  
Telefone = 123 string  
Email = "a@b"  
Contato Proximo =
```

Listas Encadeadas

– Adicionando um segundo elemento:

- **Contato** novo = **new Contato()**;
- novo.Nome = “zxy”;
- novo.Telefone = 987;
- novo.Email = “c@d”;
- novo.Proximo = **null**;
- **Fim_Lista.Proximo = novo;**
- **Fim_Lista = novo;**



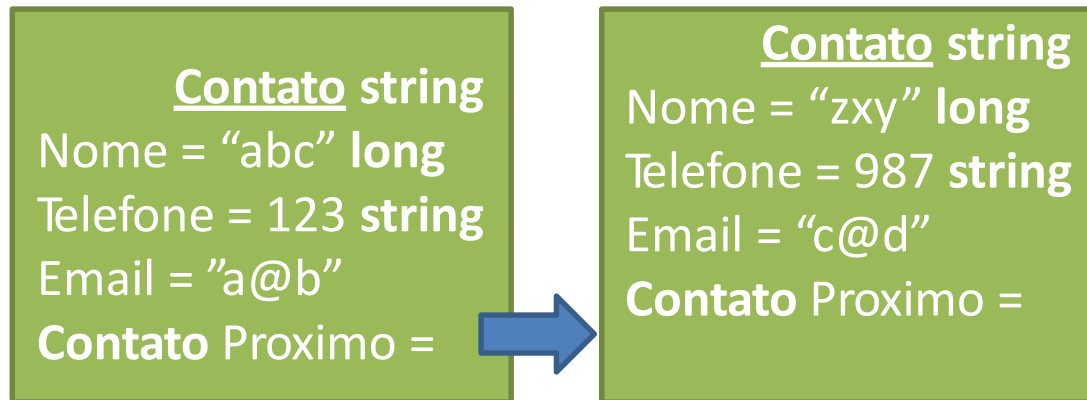
Listas Encadeadas

- Percorrendo a lista:

```
Contato aux = Inicio_Lista;  
while (aux != null) {  
    //Faz alguma tarefa com o elemento aux  
    aux = aux.Proximo;  
}
```

Listas Encadeadas

- Removendo o elemento “zxy”:
 - Inicio_Lista.Proximo = **null**;



Listas, Filas e Pilhas

LISTAS EM JAVA

Collection - ArrayList

- Java disponibiliza diversas classes que implementam diversas funcionalidades de diferentes estruturas de dados:
 - Conjunto chamado de Collections;
- Para a estrutura de dados Lista, iremos utilizar a classe disponível em:
 - `java.util.ArrayList`

Collections

- Lista:
 - Localizada no pacote **java.util**
 - Nome da classe: **ArrayList**
 - Construção do objeto lista:

```
ArrayList<ClasseArmazenada> lista = new ArrayList();
```

- Onde iremos criar uma lista dinâmica que armazena objetos da “*ClasseArmazenada*”.
- Exemplo para lista de int:

```
ArrayList<Integer> lista = new ArrayList();
```

Collections

- Com o objeto criado, utilizamos os seus métodos para executar ações:
 - Adicionar um elemento:
`lista.add(99);` //99 é o valor armazenado
 - Recuperar um elemento:
`int valor = lista.get(0);` //0 é o índice
 - Verificar a quantidade de elementos:
`int qdade = lista.size();` //Neste caso será 1
 - Esvaziar a lista:
`lista.clear();`

Collections

- Com o objeto criado, utilizamos os seus métodos para executar ações:
 - Pesquisar por um elemento:
`int pos = lista.indexOf(99);` **//retorna a posição do elemento. Se não for encontrado, retorna -1**
 - Ordenar a Lista:
`Collections.sort(lista);`
 - Remover um elemento:
`lista.remove(0);` **//0 é o índice do elemento**

Collections - Exemplo

```
Scanner entrada = new Scanner(System.in);
ArrayList<Integer> lista = new ArrayList();
int numero;
//Inserir elementos na lista
do {
    System.out.print("Digite um número: ");
    numero = Integer.parseInt(entrada.nextLine());
    if (numero != 0) {
        lista.add(numero);
    }
} while (numero != 0);
//Exibir todos os elementos da lista
System.out.println("=== Os números inseridos foram ===");
for (int i = 0; i < lista.size(); i++) {
    System.out.println(lista.get(i));
}
```

Listas, Filas e Pilhas

FILAS

Filas

- O que é uma fila em nosso cotidiano?
- As filas são diferentes das listas?
 - Em quais sentidos?
- Onde usamos filas em nosso cotidiano?
- Detalhe o funcionamento de uma fila.

Filas

- Existem muitos exemplos de fila no mundo real:
 - Uma fila de banco;
 - No ponto de ônibus;
 - Um grupo de carros aguardando sua vez no pedágio;
 - Entre outros.

Filas

- Uma fila é um conjunto de itens a partir do qual podem-se eliminar itens numa extremidade (chamada início da fila) e no qual podem-se inserir itens na outra extremidade (chamada final da fila).



Filas

- Filas são casos especiais de listas;
- **Obs:** Nas listas, quando precisávamos criar um novo elemento, poderíamos inseri-lo ou removê-lo de qualquer posição da lista, exemplos:
 - Na primeira posição;
 - Na última posição; ou
 - Em qualquer parte no meio da lista.

Filas

- Numa **fila** existe uma regra básica a ser seguida:
 - Primeiro a Chegar é o Primeiro a Sair;
 - Do inglês: FIFO – *First In, First Out*;
- Um novo elemento da **fila** somente pode ser **inserido** na última posição(fim da fila);
- Um elemento só pode ser **removido** da **primeira** posição (inicio da fila).

Filas

- Tem um sentido de chegada:
 - Fila vazia.



Filas

- Inserindo Elementos:
 - Inserir o elemento “G”



Fim da Fila

Início da Fila

Filas

- Inserindo Elementos:
 - O elemento entra na última posição.



Filas

- Inserindo Elementos:
 - E avança até a primeira posição disponível.



Filas

- Inserindo Elementos:
 - Inserir o elemento “B”



Filas

- Inserindo Elementos:
 - O elemento entra na última posição



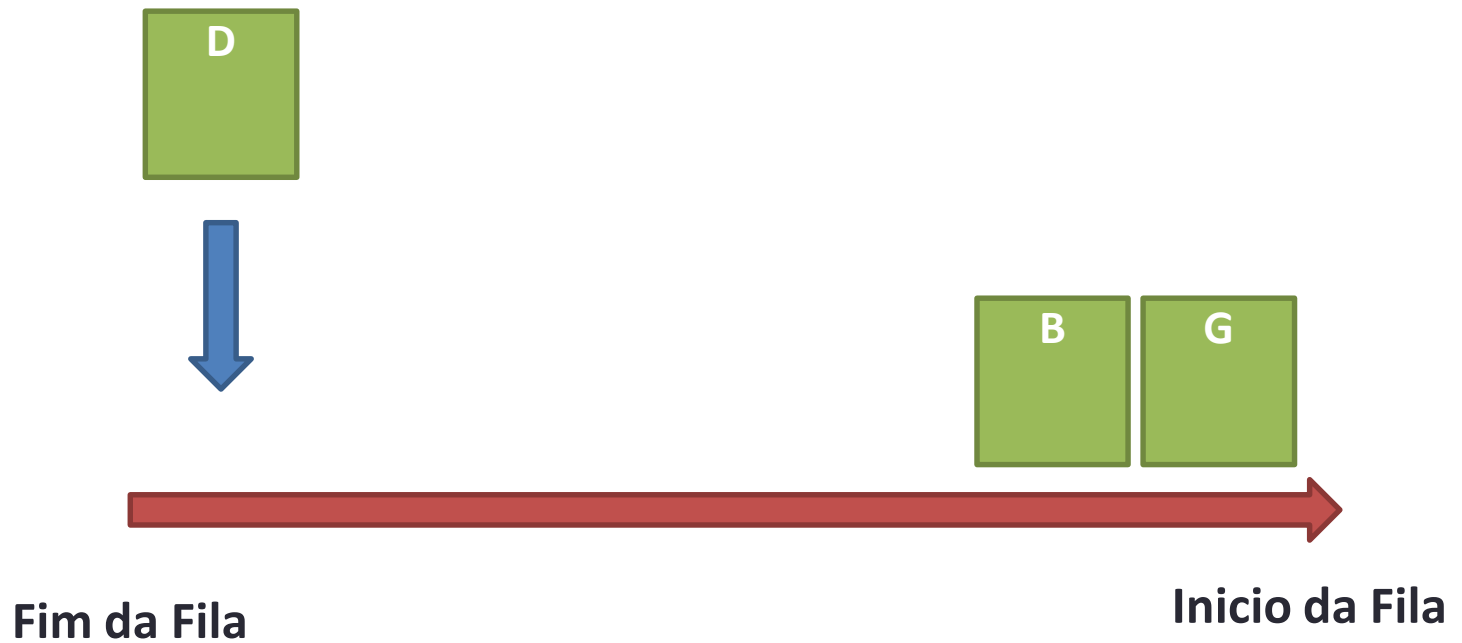
Filas

- Inserindo Elementos:
 - E avança até a primeira posição disponível.



Filas

- Inserindo Elementos:
 - Inserir o elemento “D”



Filas

- Inserindo Elementos:
 - O elemento entra na última posição

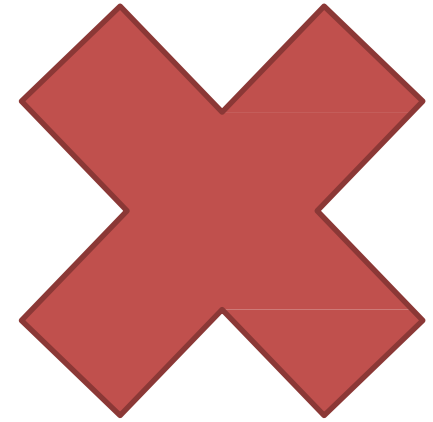


Filas

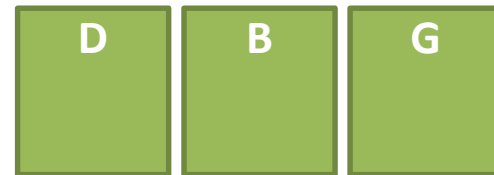
- Inserindo Elementos:
 - E avança até a primeira posição disponível.



Filas



- Removendo Elementos:
 - Remover o elemento B?
 - Não podemos remover elementos que não estejam no início da fila!
 - Da mesma forma, o elemento D não pode ser removido!



Fim da Fila

Início da Fila

Filas

- Removendo Elementos:
 - Remover o elemento da fila:
 - Retiramos o primeiro elemento da fila;
 - Neste momento o elemento deve ser utilizado.

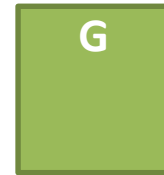


Fim da Fila

Início da Fila

Filas

- Removendo Elementos:
 - Remover o elemento da fila:
 - E os demais elementos avançam na fila.



Fim da Fila

Início da Fila

Filas

- Exemplos de uso de filas na computação:
 - Filas de impressão:
 - Impressoras tem uma fila, caso vários documentos sejam impressos, por um ou mais usuários, os primeiros documentos impressos serão de quem enviar primeiro;
 - Filas de processos:
 - Vários programas podem estar sendo executados pelo sistema operacional. O mesmo tem uma fila que indica a ordem de qual será executado primeiro;
 - Filas de tarefas:
 - Um programa pode ter um conjunto de dados para processar. Estes dados podem estar dispostos em uma fila, onde o que foi inserido primeiro, será atendido primeiro.

Filas

- Variações de Filas:
 - Fila de Prioridades:
 - Cada item tem uma prioridade. Elementos mais prioritários podem ser atendidos antes, mesmo não estando no início da fila;
 - Fila Circular:
 - Neste tipo de fila os elementos nem sempre são removidos ao serem atendidos, mas voltam ao fim da fila para serem atendidos novamente mais tarde.

Listas, Filas e Pilhas

IMPLEMENTANDO FILAS

Filas

- As filas podem ser implementadas em
 - listas encadeadas ou em vetores;
- Vetores:
 - Devemos ter duas variáveis indicando a posição do início e do fim da fila;
- **Lista Encadeada:**
 - Devemos ter duas referências, uma ao elemento de início da fila e outra ou elemento do fim da fila.

Collections - Fila

- **Fila:** ([documentação](#))
 - Construir:
 - **Queue**<Integer> fila = **new** LinkedList();
 - Adicionar:
 - fila.add(20);
 - Remover:
 - **int** x = fila.remove();
 - Examinar:
 - **int** y = fila.element();
 - Esvaziar:
 - fila.clear();
 - Tamanho:
 - fila.size();

Listas, Filas e Pilhas

PILHAS

Pilhas

- Um dos conceitos mais úteis na ciência da computação é o de pilha;



Pilhas

- Como eram as listas?
 - — Insere, remove ou utiliza qualquer elemento inserido;
- Como eram as filas?
 - Insere apenas no fim da fila, utiliza e remove apenas o primeiro elemento inserido;

Pilhas

- Como são as Pilhas?
 - Insere-se elementos no topo da pilha;
 - Remove-se ou utiliza-se apenas o elemento que estiver no topo da pilha!
- LIFO (ou FILO):
 - **Last In, First Out**;
 - Último a entrar, primeiro a sair;

Pilhas

- Pilha Vazia: Topo = **null**;

Pilhas

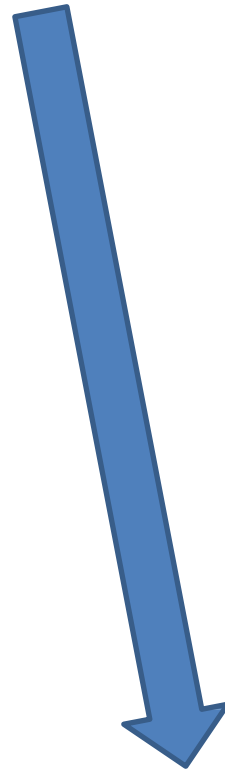
- Pilha Vazia: Topo = **null**;

- Inserindo elemento **Z**



Pilhas

- Pilha Vazia: Topo



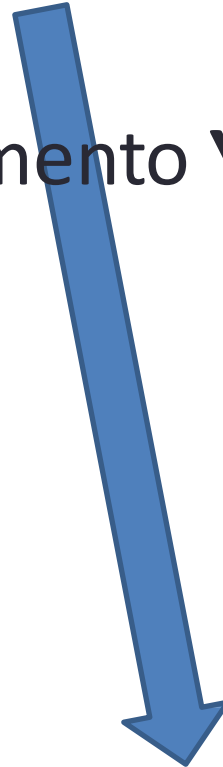
Z

Pilha p

Pilhas

- Pilha Vazia: Topo

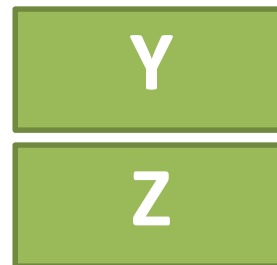
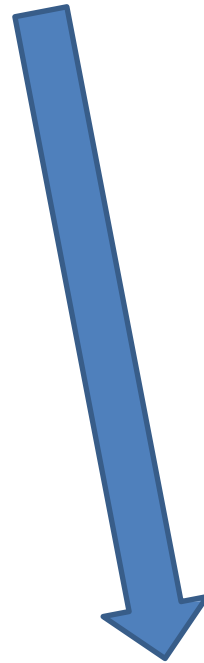
- Inserindo elemento Y



Pilha p

Pilhas

- Pilha Vazia: Topo

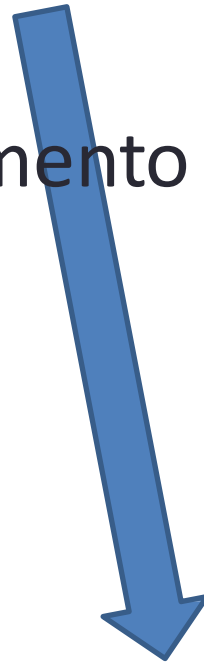


Pilha p

Pilhas

- Pilha Vazia: Topo

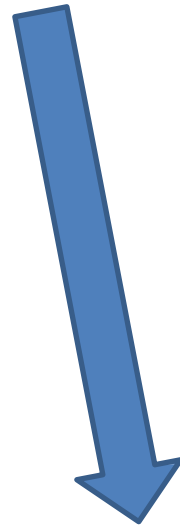
- Inserindo elemento **X**



Pilha p

Pilhas

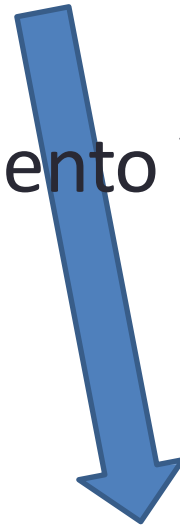
- Pilha Vazia: Topo



Pilha p

Pilhas

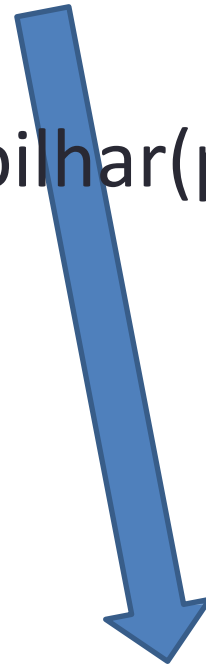
- Pilha Vazia: Topo
- Retirar o elemento Y?
 - Não.



Pilha p

Pilhas

- Pilha Vazia: **Topo**
- `Aux = Desempilhar(p);`



Aux



X

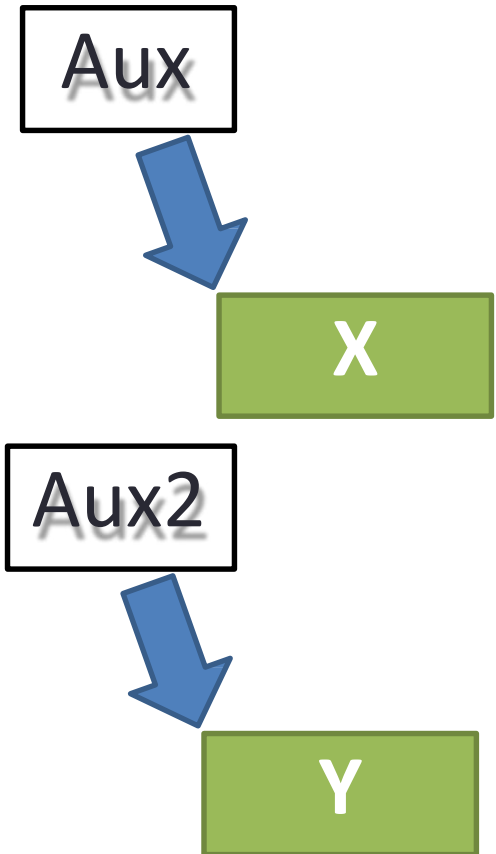
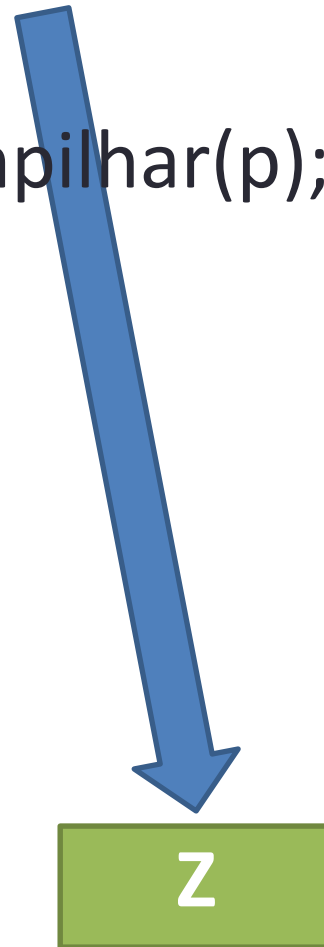
Y

Z

Pilha *p*

Pilhas

- Pilha Vazia: `Topo`
- `Aux2 = Desempilhar(p);`



Pilhas

- As pilhas podem ser implementadas em listas encadeadas ou em vetores;
- Vetores:
 - Ter uma variável indicando a posição do topo da pilha;
- Lista Encadeada:
 - Devemos ter uma referência ao elemento do topo da pilha.

Collections

- **Pilha:** ([documentação](#))
 - Construir:
 - **Stack**<Integer> pilha = **new** Stack();
 - Adicionar:
 - pilha.push(20);
 - Remover:
 - **int** x = pilha.pop();
 - Examinar:
 - **int** y = pilha.peek();
 - Esvaziar:
 - pilha.clear();
 - Tamanho:
 - pilha.size();