

# Pilhas

---

ESTRUTURAS DE DADOS APLICADAS

PROF. FELIPE A. PRZYSIADA

# Introdução

---

- É uma lista linear em que todas as inserções, retiradas e, geralmente, todos os acessos são feitos em apenas um extremo da lista;
- Os itens são colocados um sobre o outro;
- O item inserido mais recentemente está no topo e o inserido menos recentemente no fundo;

# Introdução



## Pilha

Dado 05
Dado 04
Dado 03
Dado 02
Dado 01

# Introdução

---

- Insere-se elementos no topo da pilha;
- Remove-se ou utiliza-se apenas o elemento que estiver no topo da pilha;
- Propriedade: o último item inserido é o primeiro item que pode ser retirado;
- São chamadas listas LIFO (“last-in, first-out” – “Último a entrar, primeiro a sair”);

# Aplicações

---

- Existe uma ordem linear para pilhas, do “mais recente para o menos recente”;
- É ideal para processamento de estruturas aninhadas de profundidade imprevisível;
- Uma pilha contém uma sequência de obrigações adiadas. A ordem de remoção garante que as estruturas mais internas serão processadas antes das mais externas;

# Aplicações

---

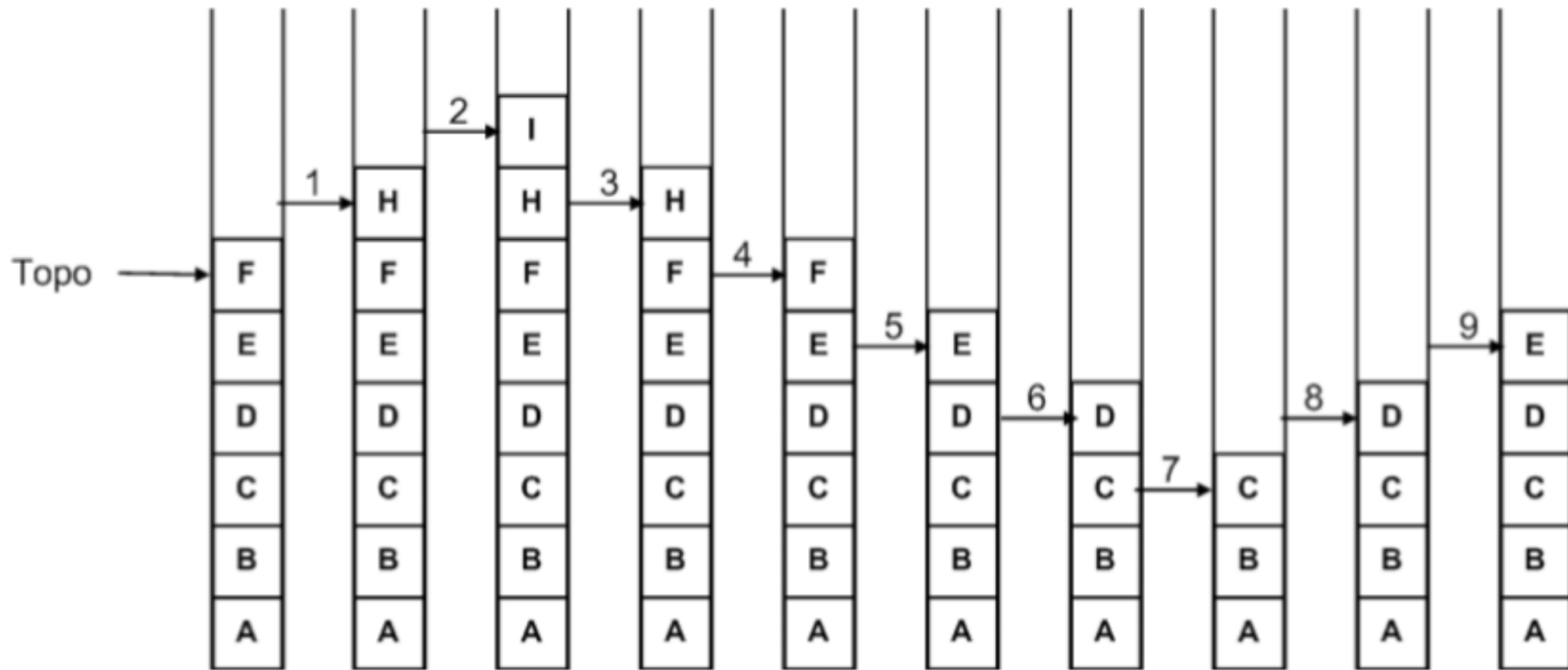
- Aplicações em estruturas aninhadas:
  - Quando é necessário caminhar em um conjunto de dados e guardar uma lista de coisas a fazer posteriormente;
  - O controle de sequências de chamadas de subprogramas;
  - A sintaxe de expressões aritméticas;
- As pilhas ocorrem em estruturas de natureza recursiva (como árvores);
- Elas são utilizadas para implementar a recursividade;

# Implementação

---

- Existem várias opções de estruturas de dados que podem ser usadas para representar pilhas;
- As duas representações mais utilizadas são as implementações por meio de vetores e de estruturas encadeadas;
- Os itens da pilha são armazenados em posições contíguas de memória;
- Como as inserções e as retiradas ocorrem no topo da pilha, um cursor chamado Topo é utilizado para controlar a posição do item no topo da pilha;

# Implementação





# Implementação

---

## ➤ Adicionar:

- `push();`

## ➤ Remover:

- `pop();`

## ➤ Examinar:

- `peek();`

## ➤ Esvaziar:

- `clear();`

## ➤ Tamanho:

- `size();`

## ➤ Listar:

- `list();`

# Exemplo 1

---

- Implementar uma pilha com dez posições de elementos inteiros;

# Exemplo 2

---

- Implementar uma pilha com dez posições de elementos estruturados;

# Exemplo 3

---

- Implementar uma pilha com  $n$  posições de elementos estruturados;

# DÚVIDAS, PERGUNTAS, QUESTIONAMENTOS

---

