

BCC202 - Estrutura de Dados I

Aula 10: Pilhas

Reinaldo Fortes

Universidade Federal de Ouro Preto, UFOP
Departamento de Ciência da Computação, DECOM

Website: www.decom.ufop.br/reifortes
Email: reifortes@iceb.ufop.br

Material elaborado com base nos slides do [Prof. Túlio Toffolo](#) (curso de 2013/01).

2013/02

Conteúdo

- 1 **Introdução**
- 2 **TAD Pilha**
 - TAD Pilha
 - Operações
 - Implementações
- 3 **Implementação por ARRAY**
- 4 **Implementação por PONTEIRO**
- 5 **Exemplo de uso**
- 6 **Conclusão**
- 7 **Exercícios**

Conteúdo

- 1 **Introdução**
- 2 TAD Pilha
 - TAD Pilha
 - Operações
 - Implementações
- 3 Implementação por ARRAY
- 4 Implementação por PONTEIRO
- 5 Exemplo de uso
- 6 Conclusão
- 7 Exercícios

O que é uma Pilha?



O que é uma Pilha?

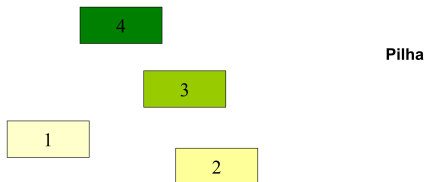


- Em computação:
 - Quando ouvir o termo “**pilha**”, pense primeiro nos pratos!!!
 - Mas, avalie o contexto, às vezes representa pilha alcalina mesmo, ou alguma outra coisa.

O que é uma Pilha?

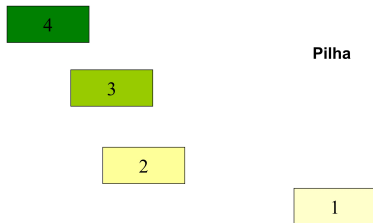
- Tipo Abstrato de Dados com a seguinte característica:
 - O **último** elemento a ser inserido é o **primeiro** a ser retirado.
 - **LIFO - Last in First Out.**
- TAD conhecida como **stack**.
- **Analogia:** pilha de pratos, livros, etc.
- **Usos:** Chamada de subprogramas, avaliação de expressões aritméticas, etc.

O que é uma Pilha?



Pilha vazia.

O que é uma Pilha?



Empilhou.

O que é uma Pilha?

4

3

Pilha

2
1

Empilhou.

O que é uma Pilha?

4

Pilha

3

2

1

Empilhou.

O que é uma Pilha?

4

3

Pilha

2
1

Desempilhou.

O que é uma Pilha?

4

Pilha

3

2

1

Empilhou.

O que é uma Pilha?

Pilha

4
3
2
1

Empilhou.

O que é uma Pilha?

4

Pilha

3

2

1

Desempilhou.

O que é uma Pilha?

4

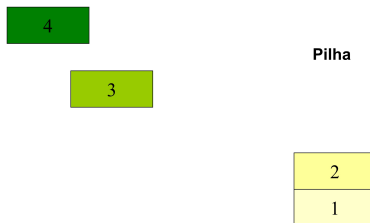
3

Pilha

2
1

Desempilhou.

O que é uma Pilha?



Pilha nada mais é do que uma **Lista** com uma restrição:

O **último** elemento a ser inserido é o **primeiro** a ser retirado.

Conteúdo

- 1 Introdução
- 2 TAD Pilha**
 - TAD Pilha
 - Operações
 - Implementações
- 3 Implementação por ARRAY
- 4 Implementação por PONTEIRO
- 5 Exemplo de uso
- 6 Conclusão
- 7 Exercícios

TAD Pilha

O que o TAD Pilha deveria conter?

- Representação do **tipo** da pilha.
- Conjunto de **operações** que atuam sobre a pilha.

Quais operações deveriam fazer parte da pilha?

- **Depende de cada aplicação.**
- Mas, um conjunto *padrão* pode ser definido.

Protótipo de uma Pilha

- Operações necessárias à grande maioria das aplicações:
 - `Pilha_Inicia(Pilha)`: inicia uma pilha vazia.
 - `Pilha_EhVazia(Pilha)`: retorna 1 se a pilha está vazia; caso contrário, retorna 0.
 - `Pilha_Push(Pilha, x)`: empilha o item `x` no topo da pilha.
 - `Pilha_Pop(Pilha, x)`: desempilha o item `x` do topo da pilha, retirando-o da pilha.
 - `Pilha_Tamanho(Pilha)`: retorna o número de itens da pilha.

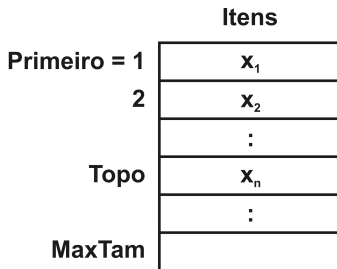
Implementações

- Existem várias opções de estruturas de dados que podem ser usadas para representar pilhas.
- As duas representações mais utilizadas são:
 - Implementação por **arrays**.
 - Implementação por **ponteiros**.

Conteúdo

- 1 Introdução
- 2 TAD Pilha
 - TAD Pilha
 - Operações
 - Implementações
- 3 Implementação por ARRAY**
- 4 Implementação por PONTEIRO
- 5 Exemplo de uso
- 6 Conclusão
- 7 Exercícios

TAD Pilha: Implementação por ARRAY



- Os itens são armazenados em posições contíguas de memória.
- Como as **inserções** e as **retiradas** ocorrem no **topo** da pilha, um campo chamado topo é utilizado para controlar a posição do item no topo da pilha.

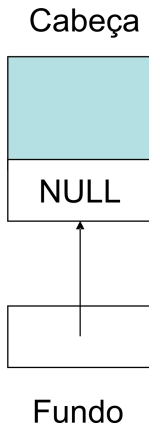
Conteúdo

- 1 Introdução
- 2 TAD Pilha
 - TAD Pilha
 - Operações
 - Implementações
- 3 Implementação por ARRAY
- 4 Implementação por PONTEIRO**
- 5 Exemplo de uso
- 6 Conclusão
- 7 Exercícios

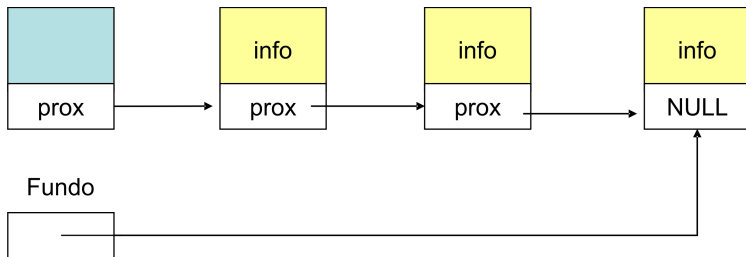
TAD Pilha: Implementação por PONTEIRO

- Criar um campo **tamanho** evita a contagem do número de itens na função tamanho.
- Cada célula de uma pilha contém um item da pilha e um apontador para outra célula.
- O registro (struct) TPilha contém um apontador para o **topo** da pilha (célula **cabeça**) e um apontador para o **fundo** da pilha.
 - Funcionam como **início** e **fim** de uma lista

TAD Pilha: Criar Pilha Vazia (usando célula Cabeça)

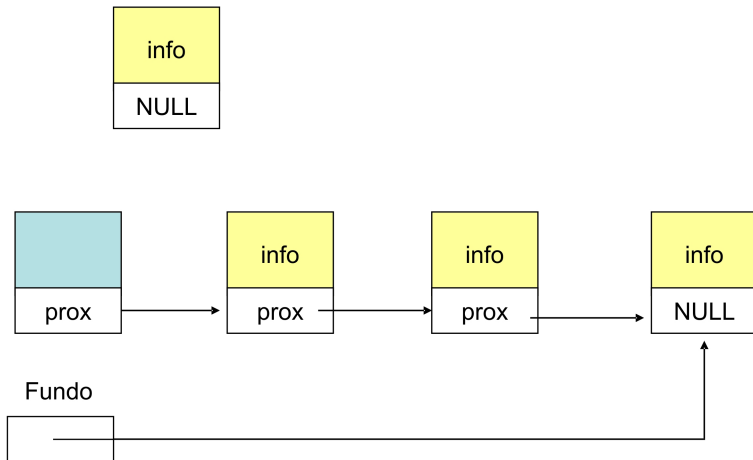


TAD Pilha: INSERÇÃO de Novos Elementos



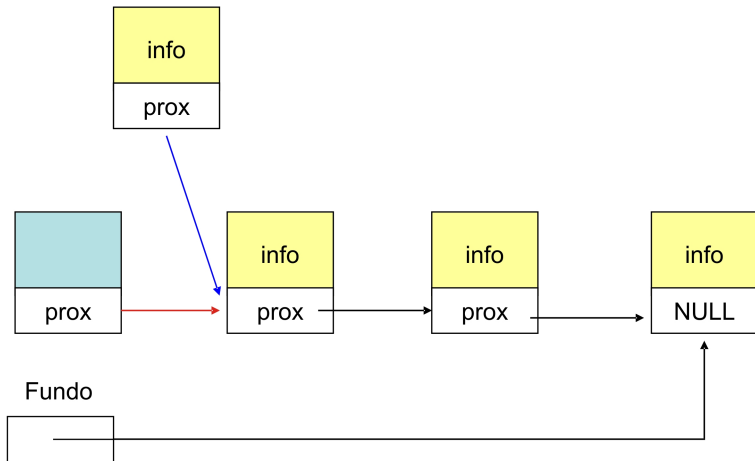
- Opção **única** de posição onde se pode inserir:
 - **Topo** da pilha, ou seja, **primeira** posição.

TAD Pilha: INSERÇÃO de um elemento

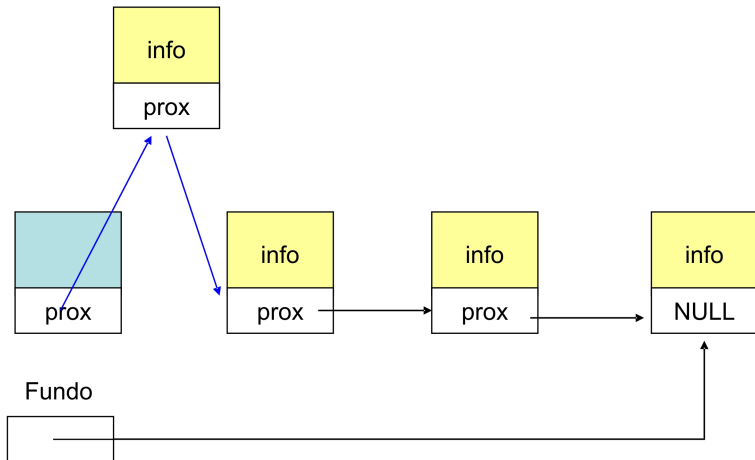


Implementação por PONTEIRO

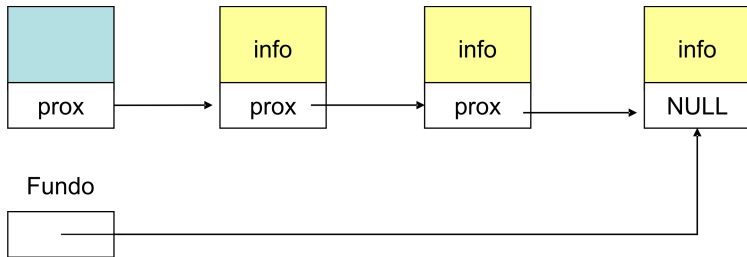
TAD Pilha: INSERÇÃO de um elemento



TAD Pilha: INSERÇÃO de um elemento

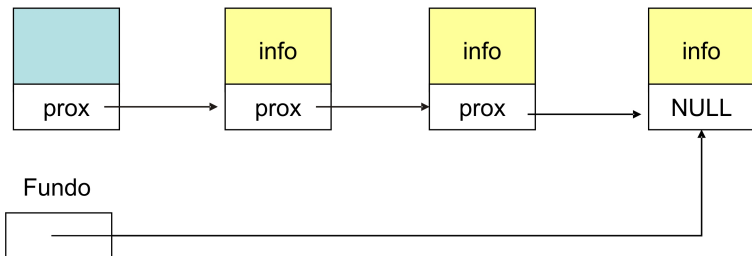


TAD Pilha: RETIRADA de Elementos

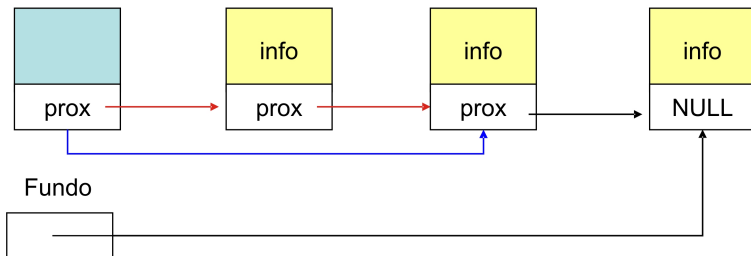


- Opção **única** de posição onde se pode retirar:
 - **Topo** da pilha, ou seja, **primeira** posição.

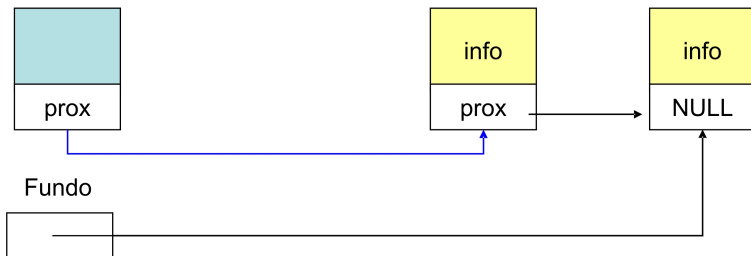
TAD Pilha: RETIRADA de um elemento



TAD Pilha: RETIRADA de um elemento



TAD Pilha: RETIRADA de um elemento



Conteúdo

- 1 Introdução
- 2 TAD Pilha
 - TAD Pilha
 - Operações
 - Implementações
- 3 Implementação por ARRAY
- 4 Implementação por PONTEIRO
- 5 Exemplo de uso**
- 6 Conclusão
- 7 Exercícios

Editor de Textos (ET): Definição

- Escreveremos um **Editor de Texto**, carinhosamente chamado de **ET**, contendo os seguintes comandos:
 - **Cancela caracter.**
 - **Cancela linha.**
 - **Imprime linha.**
- O **ET** deverá ler um caractere de cada vez do texto de entrada e produzir a impressão linha a linha, cada linha contendo no máximo 70 caracteres de impressão.
- O **ET** utilizará o TAD TPilha definido anteriormente.

Editor de Textos (ET): Caracteres de comando

- “#”: cancelar caractere anterior na linha sendo editada.
 - Ex.: UFM##FOB#P DCC##ECOM!
- “\”: cancela todos os caracteres anteriores na linha sendo editada.
- “*”: salta a linha.
- “!”: imprime os caracteres que pertencem à linha sendo editada, iniciando uma nova linha de impressão a partir do caractere imediatamente seguinte ao caractere salta-linha.

Editor de Textos (ET): Texto exemplo

Este et# um teste para o ET, o extraterrestre em C.*Acabamos de
 testar a capacidade de o ET saltar de linha, utilizando seus poderes
 extras (cuidado, pois agora vamos estourar a capacidade máxima
 da linha de impressão, que é de 70 caracteres.)*O k#cut#rso
 dh#e Estruturas de Dados et# h#um cuu#rsh#o #x#
 x?*#?#+.* Como et# bom n#nt#ao### r#ess#tt#ar
 mb#aa#triz#cull#ado nn#x#ele\ Sera que este funciona\\\? O
 sinal? não### deve ficar!

Editor de Textos (ET): Implementação

- O programa apresentado a seguir utiliza um tipo abstrato de dados sem conhecer detalhes de sua implementação.
- Assim, uma implementação do **TAD Pilha** que utiliza array pode ser substituída por uma implementação que utiliza ponteiros sem causar impacto no programa.

Exemplo de uso de Pilha

Editor de Textos (ET): Implementação

```

1 int main(int argc, char* argv[]) {
2     TPilha Pilha;
3     TItem x;
4     TPilha_Inicia(&Pilha);
5     x.Chave = getchar();
6     while (x.Chave != MarcaEof) {
7         if (x.Chave == CancelaCaractere) {
8             if (!TPilha_EhVazia(&Pilha)) TPilha_Pop(&Pilha, &x);
9         } else if (x.Chave == CancelaLinha)
10            TPilha_Inicia(&Pilha);
11        else if (x.Chave == SaltaLinha)
12            TPilha_Imprime(&Pilha);
13        else if (TPilha_Tamanho(Pilha) == MaxTam)
14            TPilha_Imprime(&Pilha);
15        else
16            TPilha_Push(&Pilha, x);
17        x.Chave = getchar();
18    }
19    if (!TPilha_EhVazia(&Pilha)) TPilha_Imprime(&Pilha);
20    return 0;
21 }
```

Conteúdo

- 1 Introdução
- 2 TAD Pilha
 - TAD Pilha
 - Operações
 - Implementações
- 3 Implementação por ARRAY
- 4 Implementação por PONTEIRO
- 5 Exemplo de uso
- 6 Conclusão**
- 7 Exercícios

Conclusão

- Nesta aula tivemos contato com um tipo especial de *Lista* denominado **Pilha**.
- Esta é uma estrutura muito comum na solução de determinados problemas.
- *Próxima aula*: Fila.
- **Dúvidas?**

Conteúdo

- 1 Introdução
- 2 TAD Pilha
 - TAD Pilha
 - Operações
 - Implementações
- 3 Implementação por ARRAY
- 4 Implementação por PONTEIRO
- 5 Exemplo de uso
- 6 Conclusão
- 7 Exercícios**

Exercício 01

- Implemente uma **TAD Pilha** utilizando **arrays** e teste o programa **ET** definido no slide 25.

Exercício 02

- Implemente uma **TAD Pilha** utilizando **ponteiros** e teste o programa **ET** no slide 25.
- Utilize o mesmo arquivo `main` utilizado no exercício 01. Ou seja, você deve compilar o mesmo arquivo `.c` do programa **ET**, usando as implementações diferentes da mesma TAD, implementadas nos dois exercícios.