BCC202 - Estruturas de Dados I

Aula 11: Pilhas

Pedro Silva

Universidade Federal de Ouro Preto, UFOP
Departamento de Computação, DECOM
Email: **silvap@ufop.edu.br**

2021

Conteúdo

Introdução

TAD Pilha

Implementação com Vetor

Implementação via Ponteiros

Considerações Finais

Bibliografia

Conteúdo

Introdução

TAD Pilha

Implementação com Vetor

Implementação via Ponteiros

Considerações Finais

Bibliografia

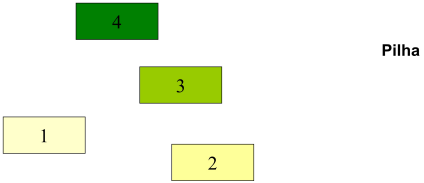
Tipo Abstrato de Dados com as seguintes características:

- O **último** elemento a ser inserido é o **primeiro** a ser retirado/ removido
- LIFO - *Last in First Out*
- TAD conhecida como stack.

Cada novo elemento é inserido no topo e só temos acesso ao elemento do topo.

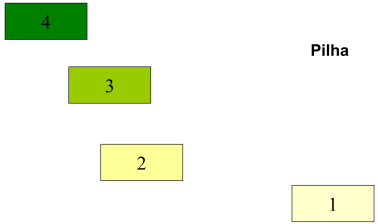
- Analogia natural com o conceito de pilhas de pratos que usamos no dia a dia
  - Pilha de pratos.
  - Pilha de livros.
- Usos:
  - Chamada de subprogramas, avalição de expressões aritméticas, etc.

O que é uma Pilha?



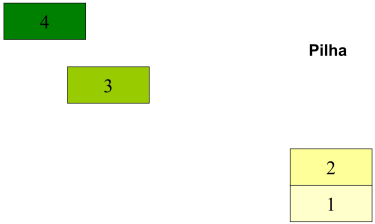
Pilha vazia.

O que é uma Pilha?



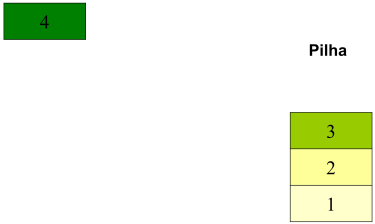
Empilhou.

O que é uma Pilha?



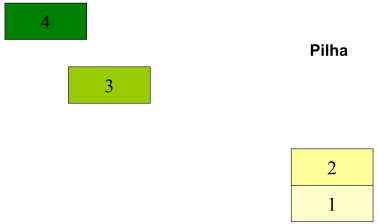
Empilhou.

O que é uma Pilha?



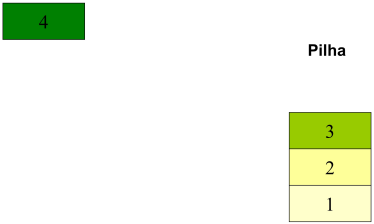
Empilhou.

O que é uma Pilha?



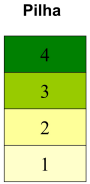
Desempilhou.

O que é uma Pilha?



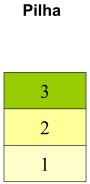
Empilhou.

O que é uma Pilha?



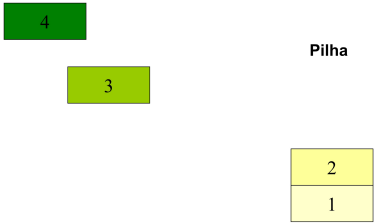
Empilhou.

O que é uma Pilha?



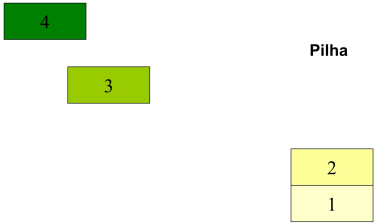
Desempilhou.

O que é uma Pilha?



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.

Exemplo: Pilha de Execução

Pilha de execução da linguagem *C*.

- As variáveis locais são dispostas numa pilha, a pilha de execução.
- Uma função só tem acesso às variáveis que estão no topo.
- Não é possível acessar variáveis locais de outras funções.

Conteúdo

Introdução

TAD Pilha

Implementação com Vetor

Implementação via Ponteiros

Considerações Finais

Bibliografia

pilha.h

```
1  #ifndef pilha_h
2  #define pilha_h
3
4  #include <stdio.h>
5
6  /*definindo um novo tipo*/
7  typedef struct { ... } Pilha;
8
9  void TPilha_Inicia(Pilha**);
10 void TPilha_EhVazia(Pilha*);
11 void TPilha_Push(Pilha*, int); /* insere no mesmo lugar que remove */
12 int TPilha_Pop(Pilha*); /* retira do mesmo lugar que insere */
13 void TPilha_Libera(Pilha**);
14
15 #endif /* pilha_h */
```

Existem várias opções de estruturas de dados que podem ser usadas para representar pilhas. As duas representações mais utilizadas são: vetor e ponteiros.

Conteúdo

Introdução

TAD Pilha

Implementação com Vetor

Implementação via Ponteiros

Considerações Finais

Bibliografia

struct

Devemos ter um vetor para armazenar os elementos da pilha.

- um novo elemento é armazenado na primeira posição vazia do vetor (posição vazia de menor índice).
- a pilha pode ter um limite conhecido (vetor estático): é necessário verificar a sua capacidade a cada inserção.
- a pilha pode usar um vetor dinâmico, dimensão atualizada (via uso de *realloc*) sempre que necessário).

Struct Pilha utilizando Vetores

```
1  #define MAXTAM 1000 /*verificar capacidade a cada inserção*/
2
3  typedef struct{
4    int n; /*número de elementos armazenados*/
5    int vet[MAXTAM]; /*alocação estática*/
6  } TPilha;
7
8  /*implementar as funções de pilha.h para manipular os dados especificados.*/
```

Conteúdo

Introdução

TAD Pilha

Implementação com Vetor

Implementação via Ponteiros

Considerações Finais

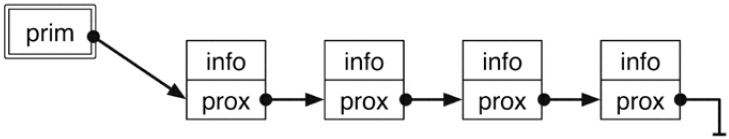
Bibliografia

Ponteiros

A implementação será similar à lista encadeada.

- Elementos serão inseridos e retirados da primeira posição da pilha (topo).
- Será mantido um ponteiro: prim ou topo.

prim é um ponteiro que corresponde ao topo da pilha.



E se mantivéssemos como topo a última posição?

O mesmo que a *Lista*: nós encadeados com informações e ponteiro para o próximo nó.

pilha_ponteiro.c

```c
#ifndef pilha_h
#define pilha_h
#include <stdio.h>
/* Decisão de implementação, poderia ser vetor.*/
typedef struct {
    int chave;
} TItem;

typedef struct Celula {
    TItem item;
    struct Celula *prox;
} TCelula;

typedef struct {
    TCelula *cabeca;
} TPilha;

/*implementar as funções de "pilha.h"*/
```

Conteúdo

Introdução

TAD Pilha

Implementação com Vetor

Implementação via Ponteiros

Considerações Finais

Bibliografia

- Pilha: Tipo Abstrato de Dado.
  - Vetor.
  - Ponteiro.
- Protocolo bem definido: (Last In, First Out - FIFO).
- Qual a complexidade de tempo para empilhar (*push*) e para desempilhar (*pop*)?

Tipo Abstrato de Dado: Fila.

Conteúdo

Introdução

TAD Pilha

Implementação com Vetor

Implementação via Ponteiros

Considerações Finais

Bibliografia

Bibliografia

Os conteúdos deste material, incluindo figuras, textos e códigos, foram extraídos ou adaptados do livro-texto indicado a seguir:


Celes, Waldemar and Cerqueira, Renato and Rangel, José
Introdução a Estruturas de Dados com Técnicas de Programação em C.
Elsevier Brasil, 2016.
ISBN 978-85-352-8345-7.

Exercício

Implemente o *TAD Pilha* utilizando ponteiros.

```
1 void TPilha_Inicia(Pilha**);
2 void TPilha_EhVazia(Pilha*);
3 void TPilha_Push(Pilha*, int);/* insere no inicio */
4 int TPilha_Pop(Pilha*);/* retira do inicio */
5 void TPilha_Libera(Pilha**);
```