

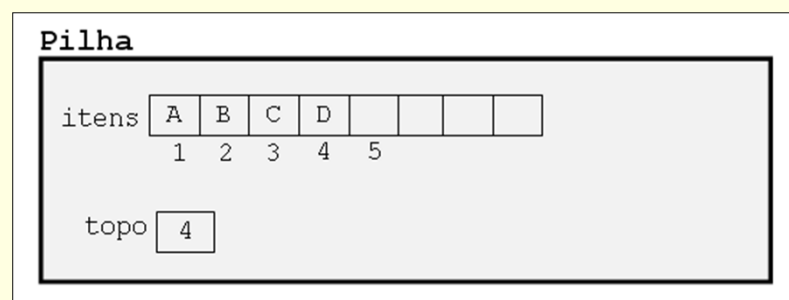
Pilha

SCC0202 – Algoritmos e Estruturas de
Dados I

Prof. Thiago A. S. Pardo

Implementação da pilha

- Seqüencial e estática



Implementação da pilha

- Declaração em C

```
#define TamPilha 100

typedef int elem;

typedef struct {
    int topo;
    elem itens[TamPilha];
} Pilha;

Pilha P;
```

3

Exercício

- Implementar operações da pilha

- Create
- Empty
- IsEmpty
- IsFull
- Push
- Pop
- Top

- Atenção: considerações sobre TAD

- Arquivos .c e .h, parâmetros, mensagens de erro

4

Exercício

- Adicionar uma rotina ao TAD para verificar se duas pilhas são iguais

5

Exercício

- Adicionar uma rotina ao TAD para inverter a posição dos elementos de uma pilha

6

Editor de texto: exercício

- Considere que um editor de texto representa os caracteres digitados como uma pilha, sendo que o último caractere lido fica no topo
- Alguns comandos apagam caracteres
 - # representa *backspace* (apaga só o último caractere digitado)
 - @ indica “apagar tudo”
- Faça um programa que execute essas ações usando o TAD pilha

7

```
#include <stdio.h>
#include "pilha.h"
int main(void) {
    elem c, x;
    int erro;
    Pilha P;
    Create(&P);
    printf("Digite seu texto: ");
    while ((c=getche())!='\r') {
        if (c=='#') {
            Pop(&P,&x,&erro);
            if (erro) printf("erro ");
            else printf("(%c desempilhado) ",x);
        }
        else if (c=='@') {
            Empty(&P);
            printf("(pilha esvaziada) ");
        }
        else {
            Push(&P,&c,&erro);
            if (erro) printf("erro ");
        }
    }
    ...
    printf("\n\nDesempilhando tudo: ");
    while (!IsEmpty(&P)) {
        Pop(&P,&x,&erro);
        if (erro) printf("erro ");
        else printf("%c ",x);
    }
    system("pause");
    return 0;
}
```

POSSÍVEL SOLUÇÃO

8

Notação posfixa: exercício

- Avaliação de expressões aritméticas
 - Às vezes, na aritmética tradicional, faz-se necessário usar parênteses para dar o significado correto à expressão
 - $A*B-C/D \rightarrow (A*B)-(C/D)$
 - Notação polonesa (prefixa): operadores aparecem antes dos operandos e dispensam parênteses
 - $-*AB/CD$
 - Notação polonesa reversa (posfixa): operadores aparecem depois dos operandos
 - $AB*CD/-$

9

Notação posfixa: exercício

- **Interpretação** da notação posfixa usando **pilha**
 - Empilha operandos até encontrar um operador
 - Retira os operandos, calcula e empilha o resultado
 - Até que se chegue ao final da expressão

10

Notação posfixa: exercício

- $AB^*CD/-$

A					
A	B				
A	B	*			
A^*B					
A^*B	C				
A^*B	C	D			
A^*B	C	D	/		
A^*B	C/D				
A^*B	C/D	-			
A^*B-C/D					

Exercício

- Implemente uma função que calcule o valor de uma expressão posfixa passada por parâmetro utilizando o TAD pilha

Exercício – resposta algorítmica

```

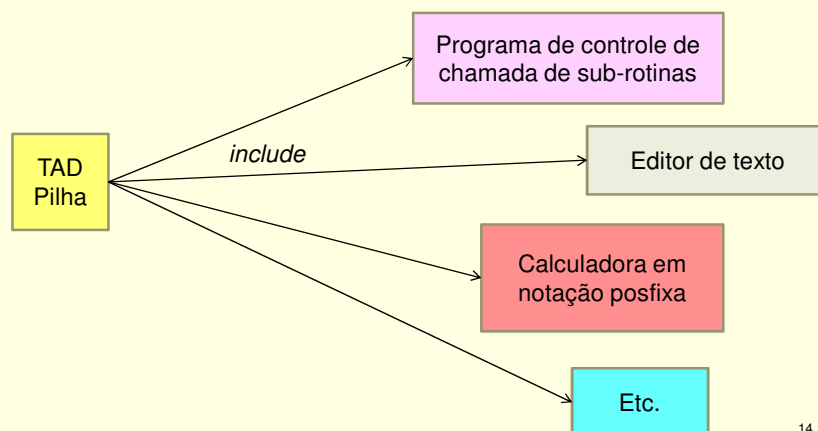
função valor(E: expressão): retorna real;
declare x real;
declare P pilha;
início
  Create(P)
  enquanto não acabou(E) faça
    início
      x=proxsimb(E);
      se x é operando então Push(P,x)
      senão início
        remove operandos; {dois pops, em geral}
        calcula o resultado da operação;
        empilhe resultado; {push}
      fim
    fim
  fim
  valor=Top(P);
fim

```

13

Pilha

■ TAD bem implementado



14