Exercícios em aula

Pilhas, Filas e Listas

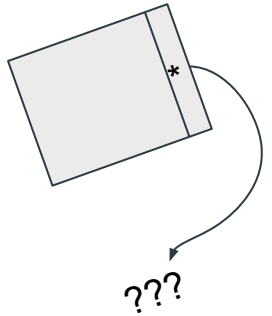
Revisão dos conceitos

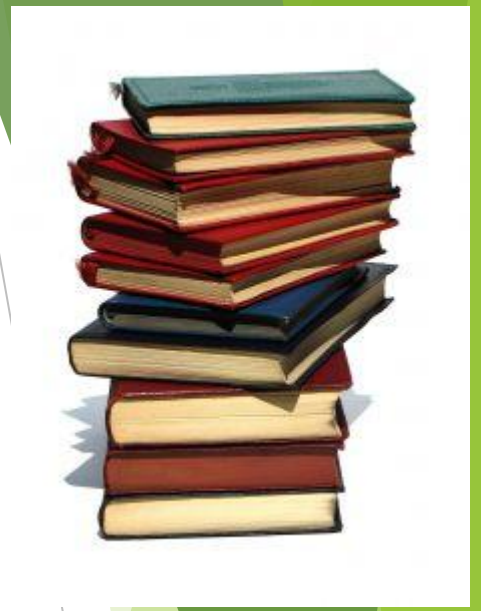
Pilha

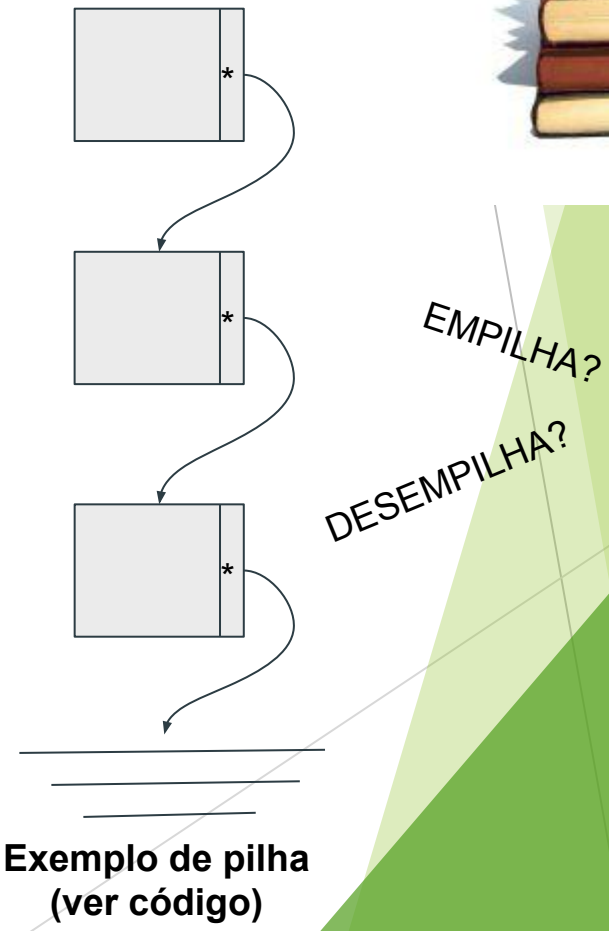
- novos elementos entram no conjunto exclusivamente no topo da pilha;

- o único elemento que pode sair da pilha em um dado momento, é o elemento do topo. Por esse motivo, as Pilhas são conhecidas como LIFO (*last in, first out*), isto é, o último a entrar é o primeiro a sair.

Ex.: Recurso *desfazer* de um editor de texto







Exemplo de pilha (ver código)

Revisão dos conceitos

Fila

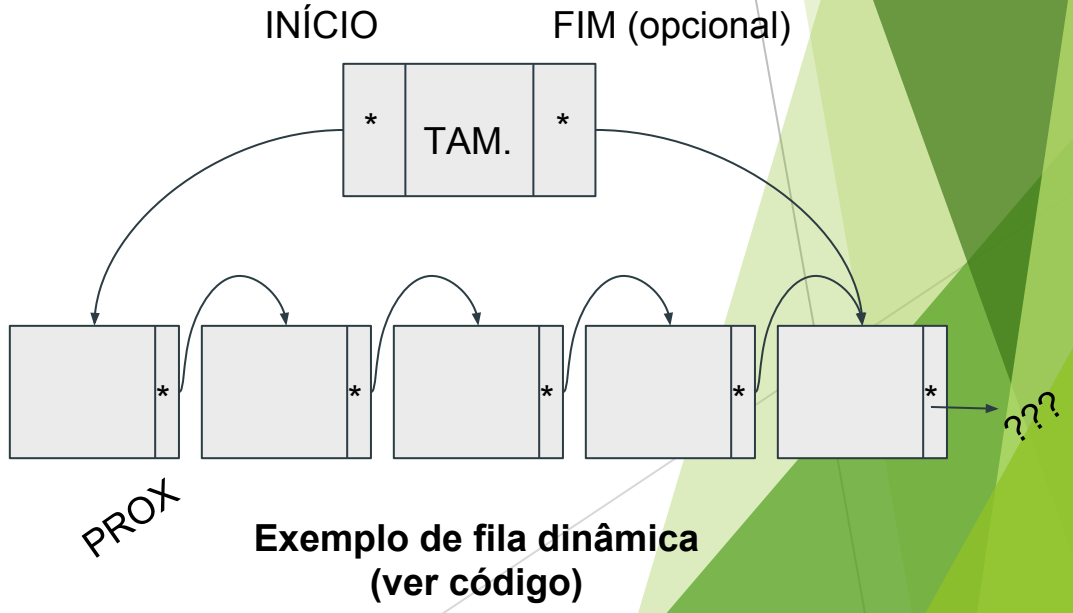
A fila é uma estrutura para armazenar um conjunto de elementos de mesmo tipo que funciona da seguinte forma:

- novos elementos sempre entram no fim da fila.

- o único elemento que se pode retirar da fila em um dado momento é seu primeiro elemento.

A fila serve para modelar situações em que é preciso armazenar um conjunto ordenado de elementos, no qual o primeiro elemento a entrar no conjunto será também o primeiro a sair.

A fila obedece ao critério FIFO (*first in, first out*).



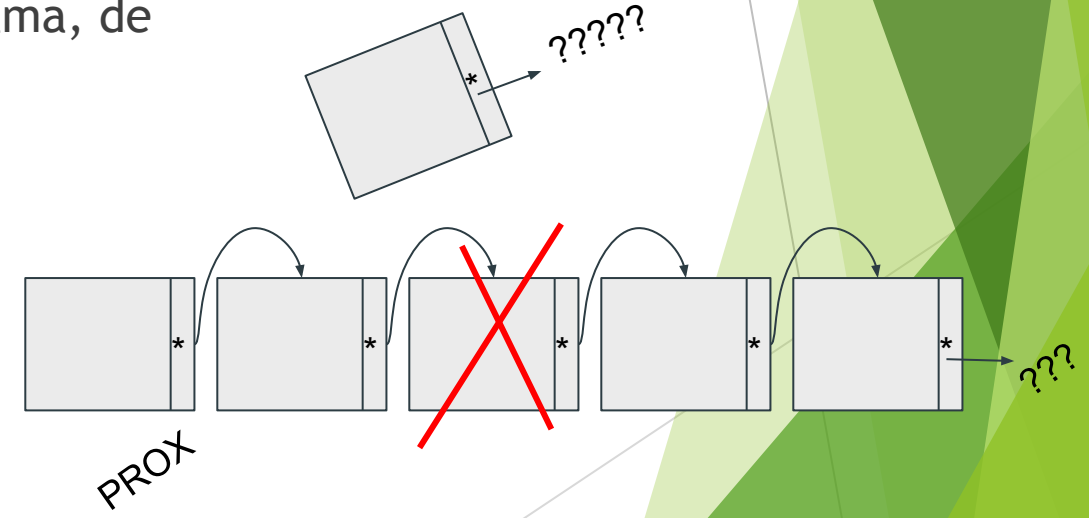
Exemplo de fila dinâmica (ver código)

Revisão dos conceitos

Lista

Uma lista é uma coleção de elementos do mesmo tipo dispostos linearmente que podem ou não seguir determinada organização.

Listas são estruturas muito flexíveis, porque podem crescer ou diminuir de tamanho durante a execução de um programa, de acordo com a demanda.



Exemplo de organização de uma lista encadeada dinâmica (ver código)

Exercícios

1) Considere que um editor de texto representa os caracteres digitados como uma **pilha**, sendo que o último caracter lido fica no topo

- Alguns comandos apagam caracteres. Por exemplo, o *backspace* apaga o último caractere lido
- Alguns comandos apagam tudo o que já foi lido anteriormente
- Considere que, no seu editor, # representa *backspace*e @ indica “apagar tudo”
- Faça um programa que execute essas ações usando o TAD pilha

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

```
...
    printf("\n\nDesempilhando tudo: ");
    while (!IsEmpty(&P)) {
        if (Pop(&P,&x))
            printf("%c ",x);
        else printf("erro ");
    }
    system("pause");
    return 0;
}
```

POSSÍVEL SOLUÇÃO

Pilha

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□(A*B)-(C/D)

Notação polonesa (prefixa): operadores aparecem antes dos operandos e dispensa parênteses

- -*AB/CD

Notação polonesa reversa (posfixa): operadores aparecem depois dos operandos e dispensa parênteses

- AB*CD/-

- 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

- AB*CD/-

Crescimento da pilha →

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					

Exercícios

2) Implemente uma função que calcule o valor de uma expressão posfixa passada por parâmetro utilizando uma pilha

```
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
    valor=Top(P);
fim
```

Exercícios

3) Faça uma função que inverta uma **fila** F1, criando-se uma nova **fila** F2

4) Implemente o sistema para a biblioteca usando uma **fila**

- Cada livro deve ser representado por um registro
- Nome do livro, disponibilidade, fila de espera
- Ao requisitar um livro, a pessoa entra na fila de espera se o livro não estiver disponível
- Quando um livro fica disponível, o primeiro da fila de espera do livro deve receber o livro
- Implemente as demais funcionalidades (cadastra livro, retira livro, etc.) que julgar necessárias

Exercícios

5) Escreva uma função que receba duas **Listas** (L1 e L2), intercale-as gerando uma terceira Lista, L3

5.1) Escreva uma função que inverte L1, colocando o resultado em L2