UniEVANGÉLICA
UNIVERSIDADE EVANGÉLICA DE GOIÁS

Engenharia de Software

Estrutura de Dados II

Aula 1: Estruturas de dados básicas – Pilhas e Filas

Professor: M.e. William P. Santos Júnior

UniEVANGÉLICA
UNIVERSIDADE EVANGÉLICA DE GOIÁS

Estruturas de Dados - Definição

- Estruturas de Dados é a disciplina que estuda as técnicas computacionais para a organização e manipulação eficiente de quaisquer quantidades de informação.

Estruturas de Dados - Aspectos

Em um projeto de software, 2 aspectos devem ser considerados:

- De que forma estão organizados os dados - qual a sua estrutura;
- Quais procedimentos atuam sobre estes dados - que operações podem ser realizadas sobre eles.

Estruturas de Dados - Aspectos

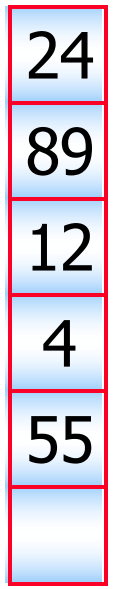
Ao estudar estruturas de dados teremos sempre este par:

- Um conjunto estruturado de informações:
  - uma classe de objetos ou um tipo de dados;
- Um conjunto definido de operações sobre estes dados:
  - um conjunto de métodos ou funções.

PILHAS

Pilhas

- A Pilha é uma estrutura de dados cujo funcionamento é inspirado no de uma pilha "natural".



Pilhas usando Vetores

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa pilha;
- Necessitamos de um indicador de qual elemento do vetor é o atual topo da pilha.

Implementação de Pilhas

- Implementaremos a Estrutura de Dados Pilha utilizando a técnica da Programação Estruturada.
  - Programação Estruturada: inventada por Niklaus Wirth (década 70) - também chamada de “Programação sem GoTo”
- Procedimento Didático:
  - Revisão/introdução de Programação Estruturada;
  - Modelagem da Pilha e de seus algoritmos usando esta técnica.

Programação Estruturada

- Baseada na expressão de algoritmos única e exclusivamente através de 4 grupos de **estruturas de controle**:
  - Bloco: comando ou conjunto de comandos *sempre* executados em seqüência.
    Ex.: (Pascal): begin … end;
  - Estrutura condicional: SE-ENTÃO-SENÃO
    Ex.: (Pascal): if (cond) then bloco1 else bloco2;
  - Estrutura de repetição: ENQUANTO COND FAÇA BLOCO
    Ex.: (Pascal): while (cond) do bloco;
  - Estrutura de abstração: procedimento ou função.
    Agrupamento de comandos com um nome e eventualmente também parâmetros nomeados.

Programação Estruturada

- Para nós parece um retrocesso.
- Na época:
  - Fazia-se programas completamente ininteligíveis;
  - Foi um grande avanço no sentido de:
    - Produzir código mais fácil de se manter e entender;
    - Produzir código com mais qualidade.

- A Programação Estruturada definiu:
  - Uma nova disciplina na programação;
  - Um novo grupo de linguagens de programação, a 3ª Geração.
    - Exemplos: Pascal, Algol, C, PL/1

- Ainda muito utilizada hoje em dia:
  - Sistemas Operacionais;
  - Análise Numérica/Computação Gráfica;
  - Redes de Computadores.

Programação Estruturada

- Não existem objetos:
  - dados e seu comportamento são considerados em separado;
  - unificar uma estrutura de dados com as operações definidas sobre a mesma é função do programador.

- Existem variáveis e tipos (dados):
  - variáveis podem ser globais ou locais (escopo);
  - tipos podem ser primitivos, derivados ou estruturados.

- Existem procedimentos e funções (comportamento):
  - conjunto de comandos referenciados por um nome;
  - um procedimento é especial e se chama Programa Principal.

- Modelamos as estruturas de dados propriamente ditas como um tipo estruturado:
  - Estrutura é uma coleção de variáveis referenciada por um mesmo nome;
  - Imagine um objeto sem métodos;
  - Chamamos a cada elemento desta coleção de *campo*.

- Algoritmicamente:

```
tipo Empregado {
  caracter nome[100];
  caracter cargo[20];
  caracter endereço[200];
  inteiro salário;
};

Variáveis
  Empregado chefe;
```

Campos

Variável global

Programação Estruturada

- Modelamos as operações sobre uma estrutura de dados como procedimentos ou funções:
  - variáveis globais valem dentro de qualquer função (*escopo global*).
    - Antes de vermos alocação dinâmica de memória e ponteiros vamos trabalhar com funções sem alguns parâmetros.

- Algoritmicamente:

```
Variáveis
  Empregado chefe;

Procedimento baixaSalário (inteiro porcentagem)
  variáveis
    real auxiliar;
  início
    auxiliar <- chefe.salário * (porcentagem / 100);
    chefe.salário <- chefe.salário - auxiliar;
  fim;
```

Modelagem da Pilha

- Aspecto Estrutural:
  - Necessitamos de um vetor para armazenar as informações;
  - Necessitamos de um indicador da posição atual do topo da pilha;
  - Necessitamos de uma constante que nos diga quando a pilha está cheia e duas outras para codificar erros.

- Pseudo-código:

```
constantes MAXPILHA = 100;

classe Pilha {
  inteiro  m_dados[MAXPILHA];
  inteiro topo;
};
```

Modelagem da Pilha

- Aspecto Funcional:
  - Colocar e retirar dados da pilha;
  - Testar se a pilha está vazia ou cheia;
  - C++ ou python.
- Colocar e retirar dados da pilha:
  - Empilha(dado)
  - Desempilha(dado)
  - Topo
- Testar se a pilha está vazia ou cheia:
  - PilhaCheia
  - PilhaVazia
- Inicializar ou limpar:
  - InicializaPilha

Algoritmo InicializaPilha

```
CONSTRUTOR inicializaPilha()
  início
      topo <- -1;
  fim;
```

Algoritmo PilhaCheia

```
Booleano MÉTODO pilhaCheia()
 início
    SE (topo = MAXPILHA - 1) ENTÃO
        RETORNE(Verdadeiro)
    SENÃO
        RETORNE(Falso);
 fim;
```

Algoritmo PilhaVazia

```
Booleano MÉTODO pilhaVazia()
 início
     SE (topo = -1) ENTÃO
          RETORNE(Verdadeiro)
     SENÃO
          RETORNE(Falso);
 fim;
```

Algoritmo Empilha

```
Constantes
        ERROPILHACHEIA = -1;
        ERROPILHAVAZIA = -2;

Inteiro MÉTODO empilha(inteiro dado)
 início
     SE (pilhaCheia) ENTÃO
          RETORNE(ERROPILHACHEIA);
     SENÃO
          topo <- aPilha.topo + 1
          dados[topo] <- dado;
          RETORNE(topo);
     FIM SE
 fim;
```

Algoritmo Desempilha

```
Inteiro MÉTODO desempilha()
 início
     SE (pilhaVazia) ENTÃO
          RETORNE(ERROPILHAVAZIA);
     SENÃO
          topo <- topo - 1;
          RETORNE(topo);
     FIM SE
 fim;
```

Algoritmo Desempilha - Variante

```
Inteiro MÉTODO desempilha()
 início
    SE (pilhaVazia) ENTÃO
        ESCREVA("ERRO: Pilha vazia ao tentar
 desempilhar!");
        RETORNE(ERROPILHAVAZIA);
    SENÃO
        topo <- topo - 1;
        RETORNE(dados[topo + 1]);
    FIM SE
 fim;
```

Algoritmo Topo

```
Inteiro MÉTODO topo()
 início
     SE (pilhaVazia) ENTÃO
          ESCREVA("ERRO: Pilha vazia ao acessar!");
          RETORNE(ERROPILHAVAZIA);
     SENÃO
          RETORNE(dados[topo]);
     FIM SE
 fim;
```

Modelagem da Pilha com Vetor - Trabalho 1

- Implemente todas as operações vistas sobre pilha;
- Implemente um programa principal que utilize a pilha através de um menu com os seguintes itens: empilhar, desempilhar, limpar, mostrar pilha, sair do programa.
- Ao mostrar a pilha, o programa deve colocar embaixo de cada dado, a sua posição no vetor.
- A pilha possuirá tamanho máximo 30, definido como uma constante chamada MAXPILHA.
- A pilha será referenciada por uma variável global.

FILAS

Filas usando Vetores

- A Fila é uma estrutura de dados que simula uma fila da vida real.

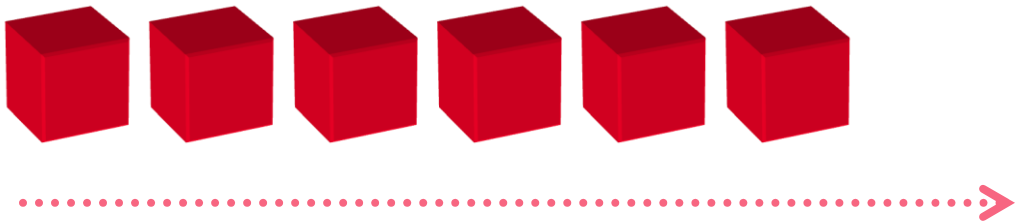
- Possui duas operações básicas:

  - Incluir no fim da fila;

  - Retirar do começo da fila;

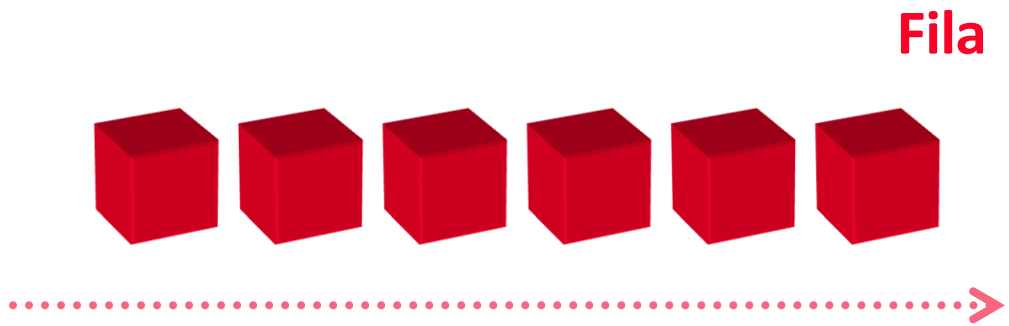
  - Chamada de Estrutura-FIFO:
    First-In, First-Out - O primeiro que entrou é o primeiro a sair...

Fila



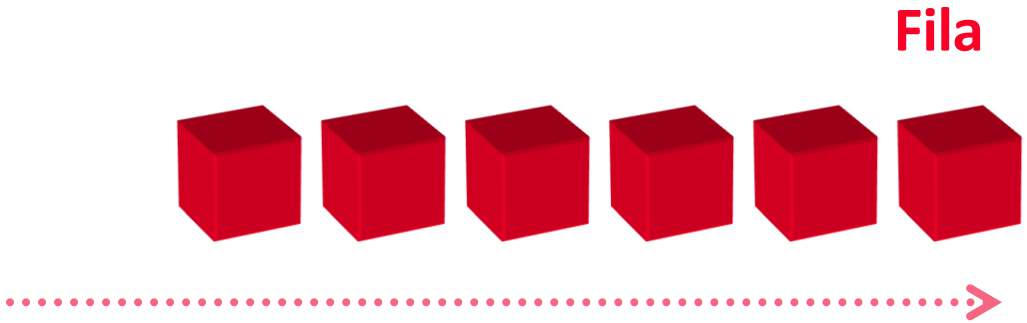
Filas usando Vetores

- É uma estrutura de dados importantíssima para:
  - Gerência de dados/processos por ordem cronológica:
    - Fila de impressão em uma impressora de rede;
    - Fila de pedidos de uma expedição ou tele-entrega.
  - Simulação de processos seqüenciais:
    - chão de fábrica: fila de camisetas a serem estampadas;
    - comércio: simulação de fluxo de um caixa de supermercado;
    - tráfego: simulação de um cruzamento com um semáforo.



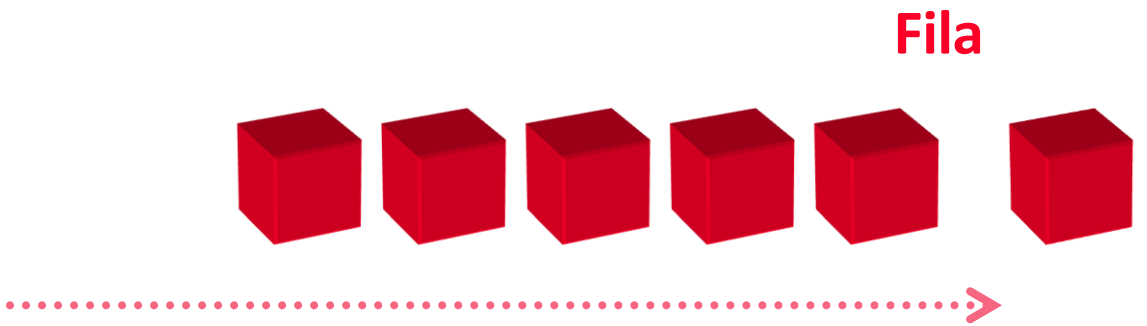
Filas usando Vetores

- É uma estrutura de dados importantíssima para:
  - Gerência de dados/processos por ordem cronológica:
    - Fila de impressão em uma impressora de rede;
    - Fila de pedidos de uma expedição ou tele-entrega.
  - Simulação de processos seqüenciais:
    - chão de fábrica: fila de camisetas a serem estampadas;
    - comércio: simulação de fluxo de um caixa de supermercado;
    - tráfego: simulação de um cruzamento com um semáforo.



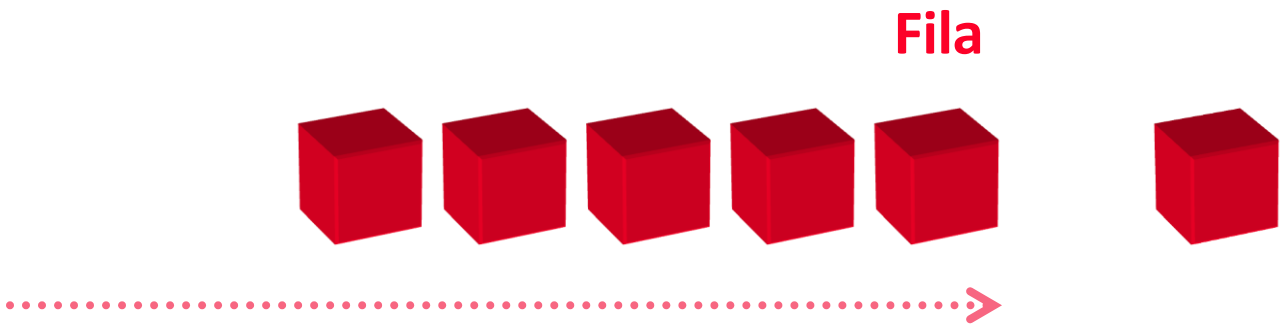
Filas usando Vetores

- É uma estrutura de dados importantíssima para:
  - Gerência de dados/processos por ordem cronológica:
    - Fila de impressão em uma impressora de rede;
    - Fila de pedidos de uma expedição ou tele-entrega.
  - Simulação de processos seqüenciais:
    - chão de fábrica: fila de camisetas a serem estampadas;
    - comércio: simulação de fluxo de um caixa de supermercado;
    - tráfego: simulação de um cruzamento com um semáforo.



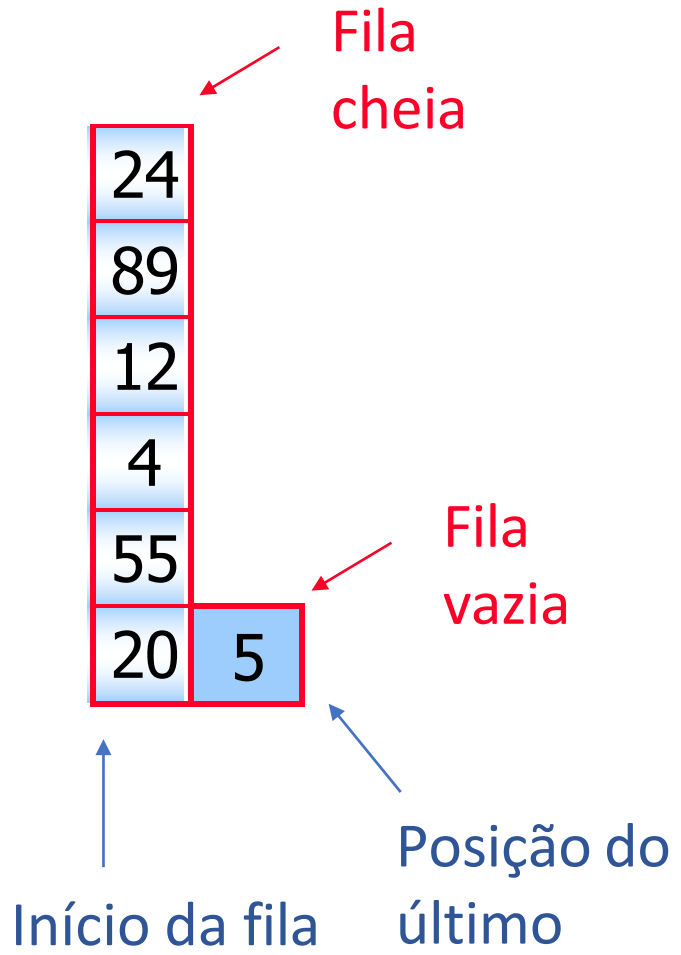
Filas usando Vetores

- É uma estrutura de dados importantíssima para:
  - Gerência de dados/processos por ordem cronológica:
    - Fila de impressão em uma impressora de rede;
    - Fila de pedidos de uma expedição ou tele-entrega.
  - Simulação de processos seqüenciais:
    - chão de fábrica: fila de camisetas a serem estampadas;
    - comércio: simulação de fluxo de um caixa de supermercado;
    - tráfego: simulação de um cruzamento com um semáforo.



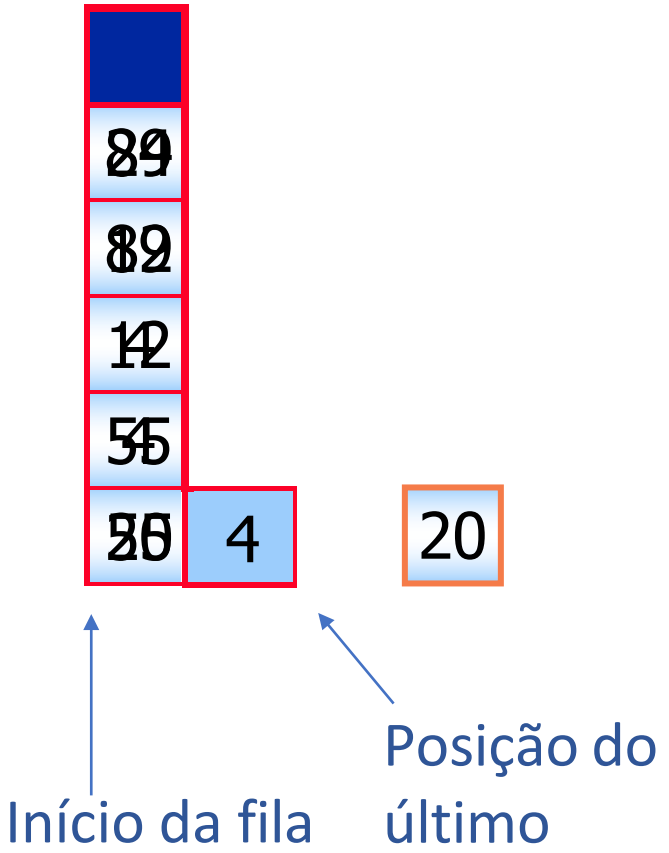
Filas usando Vetores

- Vetores possuem um espaço limitado para armazenar dados;
- Necessitamos definir um espaço grande o suficiente para a nossa fila;
- Necessitamos de um indicador de qual elemento do vetor é o atual fim da fila (último);
- Incluímos sempre no fim.



Algoritmo Retira

- Procedimento:
  - Testamos se há elementos;
  - Decrementamos o fim da fila (último);
  - Salvamos o primeiro elemento em variável auxiliar;
  - Empurramos tudo para a frente.
- Parâmetros:
  - Fila (global).



Modelagem da Fila com Vetor

- Aspecto Funcional:
  - Colocar e retirar dados da fila;
  - Testar se a fila está vazia ou cheia;

- Colocar e retirar dados da fila:
  - Inclui(dado)
  - Retira
  - Último

- Testar se a fila está vazia ou cheia:
  - FilaCheia
  - FilaVazia

- Inicializar ou limpar:
  - InicializaFila

Modelagem da Fila com Vetor - Exercício

- Inserir e retirar dados da fila: elabore um algoritmo para retirar um elemento de uma fila com vetor:
  - Utilize a mesma filosofia definida na implementação da pilha;
  - Utilize um laço para percorrer o vetor;
  - Lembre-se de testar antes se há elementos;
  - Lembre-se de que se há só um elemento, a fila ficará vazia.

Modelagem da Fila com Vetor - Trabalho 2

- Implemente todas as operações vistas sobre fila;
- Implemente um programa principal que utilize a fila através de um menu com os seguintes itens: enfileirar, desenfileirar, limpar, mostrar fila, sair do programa.
- A fila possuirá tamanho máximo 100, definido como uma constante chamada MAXFILA.
- A fila será referenciada por uma variável global;
- Para implementar a estrutura de dados defina um tipo **elementoDaFila** que será char[40] e defina a sua fila como um vetor de 100 **elementoDaFila**.

Dúvidas?