



Estrutura de Dados

Prof. Dr. Bruno Aguilar da Cunha
bruno.cunha@facens.br



01

FILAS

FILA



FILA é uma lista em que as inserções são feitas num extremo, que chamaremos de **final** e as remoções são feitas no extremo oposto, denominado **início**.

Quando um novo item é inserido numa fila, ele é colocado em seu final e, em qualquer instante, apenas o item no início da fila pode ser removido.



FILA

Devido a essa política de acesso, os itens de uma fila são removidos na mesma ordem em que foram inseridos, ou seja, o primeiro a entrar é o primeiro a sair. Por isso, as filas também são denominadas listas FIFO (First-In/First-Out)

A principal propriedade de uma fila é sua capacidade de manter a ordem de uma sequência. Essa propriedade é útil em várias aplicações em computação.

EXEMPLO DE APLICAÇÃO 1

Em um sistema operacional, cada solicitação de impressão feita pelo usuário é inserida no final de uma fila de impressão. Quando a impressora fica livre, o gerenciador de impressão sempre atende a próxima solicitação de impressão removendo-a do início da fila. Ou seja, solicitações de impressão são atendidas na mesma ordem em que são feitas.

EXEMPLO DE APLICAÇÃO 2

Em um sistema operacional, a fila é utilizada para gerenciar a entrada de dados via teclado. A medida que as teclas são pressionadas pelo usuário, os caracteres correspondentes são inseridos numa área de memória chamada buffer de teclado. Então, quando um caractere é lido por um programa, o primeiro caractere inserido no buffer de teclado é removido e devolvido como resposta. Assim, os caracteres são processados na mesma ordem em que são digitados pelo usuário.

SIMULADOR DE PILHAS E FILAS

LINK PARA O SIMULADOR DE PILHA:

<https://yongdanielliang.github.io/animation/web/Stack.html>

LINK PARA O SIMULADOR DE FILA:

<https://liveexample.pearsoncmg.com/dsanimation13java/QueueBook.html>

OPERAÇÕES EM FILAS

As funções associadas à fila são:

- **enfileira(item)** insere um novo item no final da fila. Precisa o item e não retorna nada.
- **desenfileira()** remove o item do início da fila. Não precisa de parâmetros e retorna o item. A fila é modificada.
- **cheia()** testa para ver se a fila está cheia. Não precisa de parâmetros e retorna um valor booleano (**bool**); **True** se a fila está cheia e **False** em caso contrário.
- **vazia()** testa para ver se a fila está vazia. Não precisa de parâmetros e retorna um valor booleano (**bool**); **True** se a fila está vazia e **False** em caso contrário.
- **exibeFila()** mostra os elementos existentes na fila. Não precisa de parâmetros.

A classe Fila()

Parte 1

Lembre-se que quando executarmos o código nada acontece além da definição da classe. Nós devemos criar um objeto **FILA** e então usá-lo.

```
class Fila{
    int tamanho;
    int inicio;
    int fim;
    int total;
    int vetor[];
    Fila(int tam) {
        inicio = 0;
        fim = 0;
        total = 0;
        tamanho = tam;
        vetor = new int [tam];
    }
    public boolean vazia() {
        if (total == 0)
            return true;
        else
            return false;
    }
    public boolean cheia() {
        if (total >= tamanho)
            return true;
        else
            return false;
    }
    public void enfileirar(int elem) {
        if (!cheia())
        {
            vetor[fim] = elem;
            fim++;
            total++;
            if (fim >= tamanho)
                fim = 0;
        }
        else
            System.out.println("Fila Cheia");
    }
}
```

A classe Fila()

Parte 2

Lembre-se que quando executarmos o código nada acontece além da definição da classe. Nós devemos criar um objeto **FILA** e então usá-lo.

```
public void enfileirar(int elem) {
    if (!cheia())
    {
        vetor[fim] = elem;
        fim++;
        total++;
        if (fim >= tamanho)
            fim = 0;
    }
    else
        System.out.println("Fila Cheia");
}

public String desenfileirar(){
    String elem ;
    if (vazia() == false)
    {
        elem = String.valueOf(vetor[inicio]);
        inicio++;
        if (inicio >= tamanho)
            inicio = 0;
        total --;
    }else
        elem = "Fila vazia";
    return elem;
}

public void exhibeFila(){
    for (int i = 0; i < total; i++)
        System.out.println("posicao " + i + " valor " + vetor[i]);
}
}
```



02

EXERCÍCIOS - FILAS

FILA - EXERCÍCIOS

1) Escreva um programa em Java que leia números inteiros e armazene em uma fila. A entrada de dados deve ser interrompida quando o usuário informar o número zero ou esgotar a quantidade definida de elementos a serem armazenados na estrutura. Por último, imprima os elementos na ordem em que forem removidos da fila.

FILA - EXERCÍCIOS

2) Considere uma fila contendo números inteiros que foram informados pelo usuário. Escreva uma função que calcule o maior, o menor e a media aritmética dos seus elementos.

3) Considere que você tenha duas filas (Denominadas F1 e F2) contendo números inteiros que foram informados pelo usuário. Verifique e informe se a fila F1 tem mais elementos do que uma fila F2.

FILA - EXERCÍCIOS

4) Existem partes de sistemas operacionais que cuidam da ordem em que os programas devem ser executados. Por exemplo, em um sistema de computação de tempo compartilhado (“time-shared”) existe a necessidade de manter um conjunto de processos em uma fila, esperando para serem executados.

Escreva um programa que seja capaz de ler uma série de solicitações para:

- a) Incluir novos processos na fila de processo;
- b) Retirar da fila o processo com o maior tempo de espera;
- c) Imprimir o conteúdo da lista de processo em determinado momento. Assuma que cada processo é representado por um registro composto por um número identificador do processo.

REFERÊNCIA BIBLIOGRÁFICA

SLIDES DE AULA

Lógica de Programação e estrutura de dados com aplicações em JAVA
Sandra Puga e Gerson Rissetti
2ª. Edição

Dúvidas?

Dúvidas fora do horário de aula:
bruno.cunha@facens.br