ALGORITMOS E LÓGICA DE PROGRAMAÇÃO II

Prof. Wilson Lourenço

wilson.slourenco@sp.senac.br

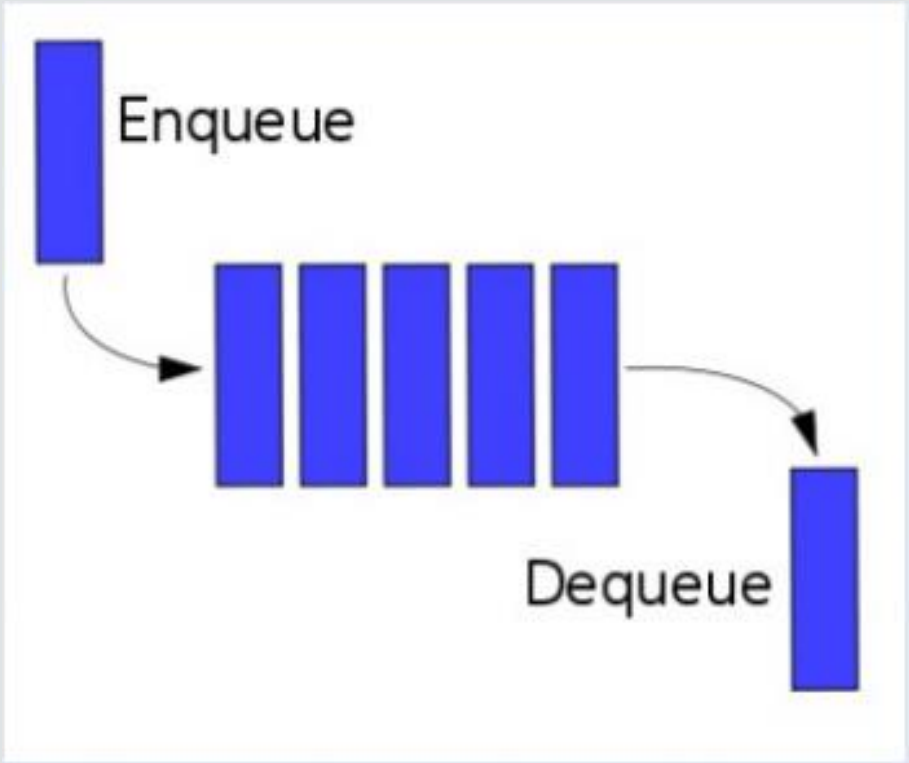
FILAS

FILAS

1. Lista FIFO (First In, First Out);

2. Os elementos são colocados e retirados por ordem de chegada:
   - Inserção apenas no final da fila;
   - Remoção apenas no início da fila.

FILAS



FILAS

- Uma fila permite várias operações:

    1. Criar uma fila vazia;
    2. Inserir um novo item (no final);
    3. Remover um item (do início);
    4. Esvaziar a fila;
    5. Etc.

FILAS

- Implementação de filas
  - Usando vetores(Estática):
    1. Pode-se implementar uma fila de tamanho fixo usando vetores. Este tamanho determinará o número máximo de elementos que poderão estar na fila ao mesmo tempo
    2. É necessário dois inteiros para armazenar o valor das posições do vetor onde se encontram o início e o final da fila

FILAS

- Usando uma lista ligada(Dinâmica):

    1. A implementação de uma fila que usa como estrutura básica uma lista ligada é mais simples, pois a lista não é uma estrutura de tamanho fixo

    2. Basicamente, os dados devem ser colocados (enfileirados) no final da lista e retirados (desenfileirados) do início da lista.

FILAS

Aplicações

- As filas são utilizadas quando desejamos processar itens de acordo com a ordem de chegada (o primeiro a chegar será o primeiro a ser processado).

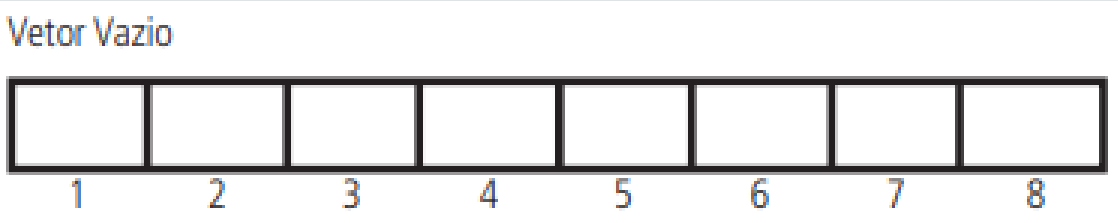
FILAS

- Sistemas operacionais, por exemplo, usam filas para regular a ordem em que as tarefas devem receber processamento e recursos devem ser alocados.

  Exemplos:
  - Escalonamento de serviços submetidos ao Sistema Operacional.
  - Sequência de trabalhos submetidos à impressora.
  - Fila de pacotes a serem transmitidos em uma rede.
  - Fila de mensagens que serão enviadas através de um servidor de e-mail.

Fila estática

- Operações em uma fila implementada com vetores.
- As filas podem usar vetor, ou seja, uma estrutura estática para armazenar os dados, como pode ser constatado na Figura a seguir.



- As operações básicas que podem ser implementadas em uma fila são:

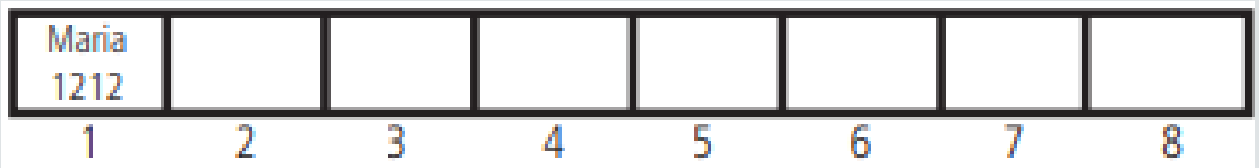
  I. criar uma fila vazia;
  II. inserir elemento no final da fila;
  III. remover o elemento do início da fila;
  IV. consultar o primeiro da fila;
  V. listar todos os elementos da fila.

- Para um melhor entendimento do que é uma fila, temos abaixo uma implementação de uma SIMULAÇÃO de fila de matrículas em um curso.

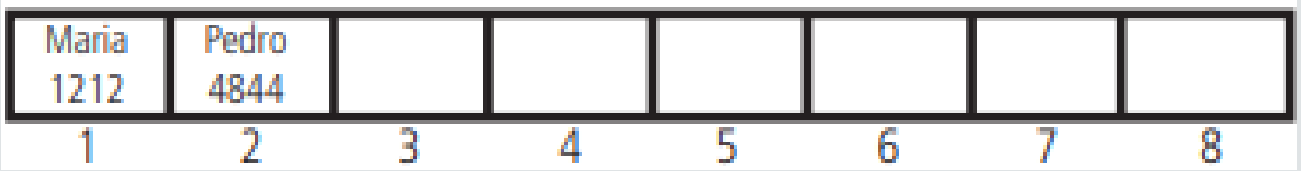
Fila vazia

- No início o vetor está vazio, ou seja, fila vazia;

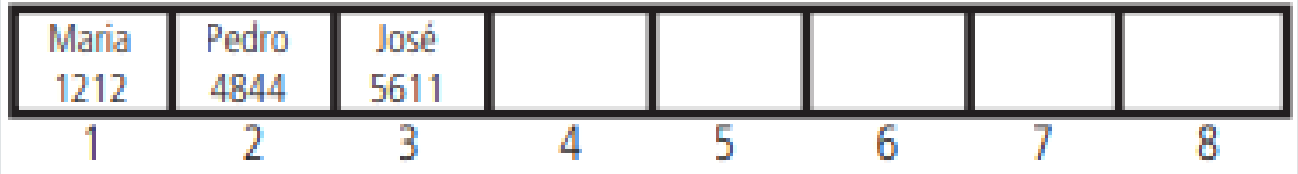
Inserção da aluna de matrícula 1212 e nome Maria



Inserção do aluno de matrícula 4844 e nome Pedro



Inserção do aluno de matrícula 5611 e nome José



Exemplo: (Ex_Fila_01.java)

```
import java.util.Scanner;

public class Ex_Fila_01 {

    static Scanner in = new Scanner(System.in);
    static int tam_max=10, tam=0;

    public static void main(String[] args) {
```

```
int vet[] = new int[10], op, elem;
 do {
     System.out.println("\n 1 - Insere\n");
     System.out.println("\n 2 - Remove\n");
     System.out.println("\n 3 - Exibe lista\n");
     System.out.println("\n 0 - Sair\n");

     op = in.nextInt();
     switch(op)
     {
         case 1 :{
             System.out.println("\n Digite o elemento: \n");
```

```
    elem = in.nextInt();
    enfileirar (vet, elem);
    break;
}
case 2: {
    elem = desenfileirar (vet);
    System.out.println("\n Elemento removido da fila = " + elem);
    break;
}
case 3: {
    exibe(vet);
    break;
}
```

```
        }//case
    } while (op!=0); //while

}

static void exibe(int vet[])
{
    int i;
   System.out.println("\n");
    if (tam >=1 )
    {
```

```
for(i=0;i<tam ;i++)
            System.out.println("\n vet[" + i + "]= " + vet[i]);
    }
    else
        System.out.println("\n fila vazia");
}

//**************************************
static void enfileirar (int vet[], int elem)
{
```

```
    if (tam == tam_max)
    {
        System.out.println("\n fila cheia");
    }
    else
    {
        vet[tam]=elem;
        tam++;
    }
}
```

```
//**************************************
   static int desenfileirar (int vet[])
   {
      int i, elem=-1;
      if (tam >= 1)
      {
            elem=vet[0];
            tam--;
```

```
for (i=0; i<tam; i++)
            {
                  vet[i]=vet[i+1];
            }
      }
      return elem;
   }
}
```

Exemplo: (Ex_Fila_02.java)

```
public class Queue {
    int itens[] = new int[100];
    int front,rear;
}
```

```
public class Ex_Fila_02 {

    static final int TAMANHO_MAXIMO=100;

    public static void main(String[] args) {

       Queue q = new Queue();
       q.front = 0; q.rear = 0;
```

```
System.out.println("Valor " + enqueue(q,1) + " inserido na fila");
System.out.println("Valor " + enqueue(q,2) + " inserido na fila");
System.out.println("Valor " + enqueue(q,3) + " inserido na fila");
System.out.println("Valor " + enqueue(q,4) + " inserido na fila\n");

System.out.println("Fila com " + (size(q)-1) + " itens\n");

System.out.println("Proximo valor da fila " + front(q));
System.out.println("Tirando o valor " + dequeue(q) + " da fila ...\n");
```

```
    System.out.println("Proximo valor da fila " + front(q));
    System.out.println("Tirando o valor " + dequeue(q) + " da fila ...\n");

    System.out.println("Proximo valor da fila " + front(q));
    System.out.println("Tirando o valor " + dequeue(q) + " da fila ...\n");

    System.out.println("Proximo valor da fila " + front(q));
    System.out.println("Tirando o valor " + dequeue(q) + " da fila ...\n");

    System.out.println("Fila vazia, fila com " + (empty(q)-1) + " itens\n");
}
```

```
static int empty(Queue pq)
  {
     /* se o início da fila for igual ao final da fila,
        a fila está vazia */
     if( pq.front == pq.rear )
     {
        return 1;
     }
     return 0;
  }
```

```
static int enqueue(Queue pq, int x)
  {
     if( pq.rear + 1 >= TAMANHO_MAXIMO )
     {
        System.out.println("\nEstouro da capacidade da fila");
     }
     pq.itens[ pq.rear++ ] = x;
     return x;
  }
```

```
static int size(Queue pq)
  {
     return (pq.rear + 1);
  }

  static int front(Queue pq)
  {
     /* o primeiro elemento sempre está no início do vetor */
     return pq.itens[0];
  }
```

```
static int dequeue(Queue pq)
  {
        int x, i;
        if(empty(pq)==1)
        {
            System.out.println("\nFila vazia");
        }
```

```
 /* Salva o primeiro elemento e refaz o arranjo dos itens,
                puxando o segundo elemento para o primeiro, o terceiro
                    para o segundo e assim sucessivamente. */
            x = pq.itens[0];
            for(i=0; i < pq.rear; i++)
            {
                pq.itens[i] = pq.itens[i+1];
            }
            pq.rear--;
            return x;
    }
}
```

Exemplo: (Ex_Fila_03.java)

```
public class Ex_Fila_03 {

    public static void main(String[] args) {
        FilaInteiro fila = new FilaInteiro(10);

        fila.enqueue(10);
        fila.ExibirFila();

        fila.dequeue();
        fila.ExibirFila();
```

```
 fila.enqueue(103);
        fila.ExibirFila();
        fila.enqueue(120);
        fila.ExibirFila();
        fila.enqueue(150);
        fila.ExibirFila();
        fila.dequeue(); //SAI O 103;
        fila.enqueue(110);
        fila.ExibirFila();
    }
}
```

```
public class FilaInteiro {
    private static int[] vetorEstrutura;
    private int quantidade;
    private int inicio;
    private int fim;

    public FilaInteiro(int tamanhoFila){
        vetorEstrutura = new int[tamanhoFila];
        quantidade = 0;
        inicio = -1;
        fim = -1;
    }
```

```
public void enqueue(int value) {
        if (isEmpty()) {
            inicio++;
            fim++;
            vetorEstrutura[inicio] = value;
            quantidade++;
        }
        else{
```

```
if (!isFull()) { //Se não estiver cheia
            fim++;
            vetorEstrutura[fim] = value;
            quantidade++;
        }
        else{
            System.out.println("Fila Cheia!");
        }
    }
}
```

```
public int dequeue() {
        int valorRetorno = 0;
        if (isEmpty()) {
                System.out.println("Fila Vazia!");
            }
        else{
            //Obtendo o primeiro da fila para que seja retornado.
            valorRetorno = vetorEstrutura[inicio];
```

```
if (inicio == fim) {
            inicio = -1;
            fim = -1;
            quantidade--;
        }
        else{
            inicio++;
            Reorganizar();
        }
    }
    return valorRetorno;
  }
```

```
public boolean isEmpty() {
        //se inicio == -1 então está vazia, esta forma verifica e
            //retorna o valor booleano da operação.
        return inicio == -1;
    }

public boolean isFull() {
        return fim == vetorEstrutura.length - 1;
    }
```

```
public int size() {
        return quantidade;
    }
private void Reorganizar() {
        for (int i = inicio; i <= fim; i++) {
            vetorEstrutura[i-1] = vetorEstrutura[i];
        }
       inicio--;
       fim--;
    }
```

```
public void ExibirFila(){
        if (!isEmpty()) {
             for (int i = inicio; i <= fim; i++) {
                System.out.print(vetorEstrutura[i] + " ");
            }
        }else{
            System.out.println("\nFila vazia!");
        }
        System.out.println("");
    }
}
```

Vídeo:

https://www.youtube.com/watch?v=Gh636gK-2w8

Quizizz:

https://quizizz.com/join

Dicas para Estudo

Seja “CURIOSO”:

Procure revisar o que foi estudado.

Pesquise as referências bibliográficas.

Seja “ANTENADO”:

Leia a próxima aula.

Seja “COLABORATIVO”:

Traga assuntos relevantes para a sala de aula.

Participe da aula.

Proponha discussões relevantes sobre o conteúdo.

Prof. Wilson Lourenço


Dúvidas?
Não mais..