



Operações com Fila

Serão apresentadas a seguir as definições das operações de Fila, a implementação das operações na Linguagem Java, as aplicações com as operações de Fila e as aplicações de Fila resolvendo problemas na linguagem Java.

DEFININDO AS OPERAÇÕES DE FILA

A operação Fila Vazia é uma função sem parâmetros da operação de Fila que retorna verdadeiro se a Fila estiver vazia e falso se a Fila não estiver vazia.

`lógico FilaVazia()`

`início_módulo`

`se (total = 0)`

`então`

`retornar verdadeiro;`

`senão`

`retornar falso;`

fimse;



fim_módulo;

A operação Fila Cheia é uma função sem parâmetros da operação de Fila que retorna verdadeiro se a Fila estiver cheia e falso se a Fila não estiver cheia.

lógico FilaCheia()

início_módulo

se (total >= tamanho)

então

retornar verdadeiro;

senão

retornar falso;

fimse;

fim_módulo;

A operação Enfileirar é um procedimento das operações de Fila que recebe um elemento como parâmetro. Este é o elemento que será enfileirado. A operação enfileirar verifica se a fila não está cheia antes de enfileirar o elemento. O elemento não poderá ser enfileirado se a fila estiver cheia e mostrará uma mensagem para o

usuário. Caso contrário, há espaço para enfileirar, logo o elemento é enfileirado na fila.



Enfileirar (elemento numérico_inteiro)

início_módulo

se (não FilaCheia())

então

vetor[fim] \leftarrow elemento;

fim \leftarrow fim + 1;

total \leftarrow total + 1;

se (fim \geq tamanho)

então

fim \leftarrow 0;


fimse;

senão

escrever ("Fila Cheia");

fimse;

fim_módulo;

A operação Desenfileirar é uma função sem parâmetros das operações e verifica se a fila não está vazia antes de desenfileirar o elemento. O elemento não poderá  desenfileirado se a fila estiver vazia, pois não há elemento a ser desenfileirado. Caso a fila não esteja vazia, então o elemento é desenfileirado e retornado.

numérico_inteiro Desenfileirar ()

início_módulo

Declarar

desenfileirado \leftarrow 0 numérico_inteiro;

se (FilaVazia()) então

 escrever (“Fila Vazia”);

senão

 desenfileirado \leftarrow vetor[início];

 início \leftarrow início + 1;

 total \leftarrow total – 1;

 se (início \geq tamanho) então

 início \leftarrow 0;

 fimse;

retornar desenfileirado;

fim_módulo;



A operação elemento início é uma operação de manipulação da fila, que é um procedimento que mostra o elemento do início da fila.

ElementoInicio()

início_módulo

se (não FilaVazia())

então

escrever ("O primeiro elemento da fila é " , vetor[inicio];

senão

escrever("Fila Vazia");

fimse;

fim_módulo;

A operação Mostrar Fila é uma operação de manipulação da fila, que é um procedimento que mostra os elementos da fila para o usuário.

MostrarFila()

início_módulo

Declarar



i, aux numérico_inteiro;

aux \leftarrow início;

para i de 1 até total passo +1 faça

escrever ("Elemento " , vetor[aux] , " posição " , i);

aux \leftarrow aux + 1;

se (aux \geq tamanho)

então

aux \leftarrow 0;

fimse;

fimpara;

fim_módulo;

MANIPULANDO AS OPERAÇÕES DE FILA

A operação FilaVazia é um método função sem parâmetros da estrutura de dados Fila que retorna 'true' se a Fila estiver vazia e 'false' se a Fila não estiver vazia.

```
public boolean FilaVazia( )
```



```
{  
    if (total == 0)  
    {  
        return true;  
    }  
    else  
    {  
        return false;  
    }  
}
```

A operação Fila Cheia é um método função sem parâmetros da estrutura de dados Fila que retorna true se a Fila estiver cheia e false se a Fila não estiver cheia.

```
public boolean FilaCheia( )
{
    if (total >= tamanho)
    {
        return true;
    }
    else
    {
        return false;
    }
}
```



O método Enfileirar é um procedimento da estrutura de dados Fila que recebe um elemento como parâmetro. Este é o elemento que será enfileirado. A operação enfileirar verifica se a fila não está cheia antes de enfileirar o elemento. O elemento não poderá ser enfileirado se a fila estiver cheia e mostrará uma mensagem para o usuário. Caso contrário, há espaço para enfileirar e o elemento é enfileirado na fila.


```
public void Enfileirar (int elemento)
{
    if (!FilaCheia( ))
    {
        vetor[fim] = elemento;
        fim = fim + 1;
        total = total + 1;
        if (fim >= tamanho)
        {
            fim = 0;
        }
    }
    else
    {
        System.out.println ("Fila Cheia");
    }
}
```



O método Desenfileirar é uma função sem parâmetros da estrutura de dados fila. Ele verifica se a fila não está vazia antes de desenfileirar o elemento. Esse elemento não poderá ser desenfileirado se a fila estiver vazia, pois não há elemento na fila para ser desenfileirado. Caso a fila não esteja vazia, então o elemento é desenfileirado e retornado.

```

public int Desenfileirar ()
{
    int desenfileirado = 0;
    if (FilaVazia())
    {
        System.out.println("Fila Vazia");
    }
    else
    {
        desenfileirado = vetor[inicio];
        inicio = inicio + 1;
        total = total - 1;
        if (inicio >= tamanho)
        {
            inicio = 0;
        }
    }
    return desenfileirado;
}

```



A operação elemento início é um método de manipulação da estrutura de dados fila que é um procedimento que mostra o elemento do início da fila.

```

public void ElementoInicio( )
{
    if (!FilaVazia())
    {
        System.out.println("O primeiro elemento é "+ vetor[inicio]);
    }
    else
    {
        System.out.println("Fila Vazia");
    }
}

```

A operação Mostrar Fila é um método de manipulação da estrutura de dados da fila. que é um procedimento que mostra os elementos para o usuário.



```
public void MostrarFila( )
{
    int i, aux;

    aux = inicio;
    for (i = 1 ; i <= total ; i++)
    {
        JOptionPane.showMessageDialog(null, "Elemento " + vetor[aux] +
        " posição " + i);
        aux = aux + 1;
        if (aux >= tamanho)
        {
            aux = 0;
        }
    }
}
```

APLICANDO FILAS

Desenvolva um algoritmo que recebe do usuário dez números inteiros para armazenar apenas os números pares numa Fila com capacidade de dez elementos e mostra os números da Fila.

```
tipo Fila = registro
    tamanho ← 0 numérico_inteiro;
    inicio ← 0 numérico_inteiro;
    fim ← 0 numérico_inteiro;
    total ← 0 numérico_inteiro;
    vetor [ tamanho] numérico_inteiro;
fimregistro;
```

`lógico FilaVazia()`

`lógico FilaCheia()`

`Enfileirar (elemento numérico_inteiro)`

`numérico_inteiro Desenfileirar ()`

`ElementoInicio()`

`MostrarFila()`

Algoritmo Exemplo1

início_algoritmo

Declarar

intFila ← novo Fila(10);

entrada ← 0 numérico_inteiro;

i numérico_inteiro;

para i de 1 até 10 passo +1faça

entrada ← escrever("Digite um valor inteiro");

se (entrada mod 2 = 0)

então

intFila.Enfileirar(entrada);

fimse;

fimpara;

intFila.MostrarFila();

fim_algoritmo.



APLICANDO FILAS NO JAVA

Para exemplificar o uso de Fila na linguagem de programação Java, temos um exemplo em que construímos uma fila com capacidade de dez elementos. Enfileira números inteiros pares e mostra os elementos da fila.

```
class Fila
```

```
{
    int tamanho, inicio, fim, total;
    int vetor[];
    Fila(int tam)
    {
        inicio = 0;
        fim = 0;
        total = 0;
        tamanho = tam;
        vetor = new int[tam];
    }
}
```



```
class ExemploPares {
    public static void main (String arg []) {
        Fila intFila = new Fila(10);
        int i, entrada = 0;

        for (i = 1 ; i <= 10 ; i++) {
            entrada = Integer.parseInt(JOptionPane.showInputDialog
                ("Digite um valor inteiro"));
            if (entrada % 2 == 0) {
                intFila.Enqueue(entrada);
            }
        }
        intFila.MostrarFila( );
        System.exit(0);
    }
}
```

```
import javax.swing.*;
```

```
class ExemploPares {
```

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

```
Fila intFila = new Fila(10);
```



```
int i, entrada = 0;
```

```
for (i = 1; i <= 10; i++) {
```

```
    entrada = Integer.parseInt(JOptionPane.showInputDialog ("Digite um valor  
inteiro"));
```

```
    if (entrada % 2 == 0) {
```

```
        intFila.Enfileirar(entrada);
```

```
    }
```

```
}
```

```
intFila.MostrarFila();
```

```
System.exit(0);
```

```
}
```

```
}
```

```
class Fila
```

```
{
```

```
    int tamanho, inicio, fim, total;
```

```
    int vetor[];
```

Fila(int tam)



{

 inicio = 0;

 fim = 0;

 total = 0;

 tamanho = tam;

 vetor = new int[tam];

}

public boolean FilaVazia()

{

 if (total == 0)

 {

 return true;

 }

 else

 {

 return false;

 }

```
}
```



```
public boolean FilaCheia( )
```

```
{
```

```
    if (total >= tamanho)
```

```
    {
```

```
        return true;
```

```
    }
```

```
    else
```

```
    {
```

```
        return false;
```

```
    }
```

```
}
```

```
public void Enfileirar (int elemento)
```

```
{
```

```
    if (!FilaCheia( ))
```

```
    {
```

```
        vetor[fim] = elemento;
```



```
fim = fim + 1;
```



```
total = total + 1;
```

```
if (fim >= tamanho)
```

```
{
```

```
    fim = 0;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
    System.out.println ("Fila Cheia");
```

```
}
```

```
}
```

```
public int Desenfileirar ()
```

```
{
```

```
    int desenfileirado = 0;
```

```
    if (FilaVazia())
```

```
{
```

```
        System.out.println("Fila Vazia");
```

```
}
```



```
else
```

```
{
```

```
    desenfileirado = vetor[inicio];
```

```
    inicio = inicio + 1;
```

```
    total = total - 1;
```

```
    if (inicio >= tamanho)
```

```
    {
```

```
        inicio = 0;
```

```
    }
```

```
}
```

```
return desenfileirado;
```

```
}
```

```
public void ElementoInicio( )
```

```
{
```

```
    if (!FilaVazia())
```

```
    {
```

```
        System.out.println("O primeiro elemento é "+ vetor[inicio]);
```

```
}
```



```
else
```

```
{
```

```
    System.out.println("Fila Vazia");
```

```
}
```

```
}
```

```
public void MostrarFila( )
```

```
{
```

```
    int i, aux;
```

```
    aux = inicio;
```

```
    for (i = 1 ; i <= total ; i++)
```

```
    {
```

```
        JOptionPane.showMessageDialog(null, "Elemento " + vetor[aux] +
```

```
            " posição " + i);
```

```
        aux = aux + 1;
```

```
        if (aux >= tamanho)
```

```
        {
```

```
aux = 0;
```

```
}
```

```
}
```

```
}
```

```
}
```

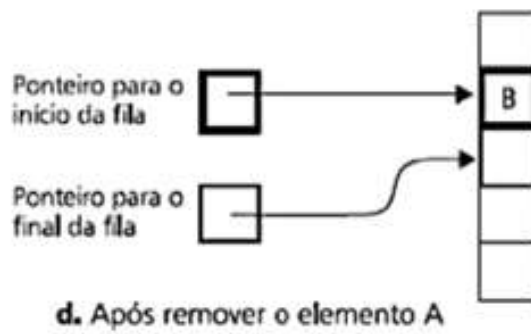
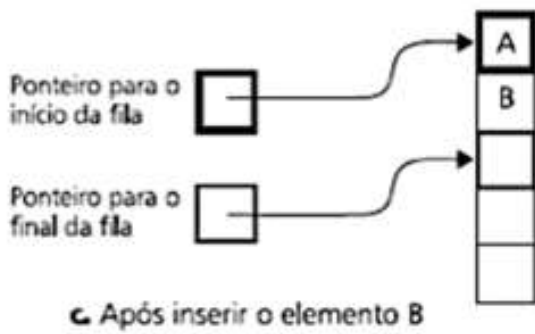
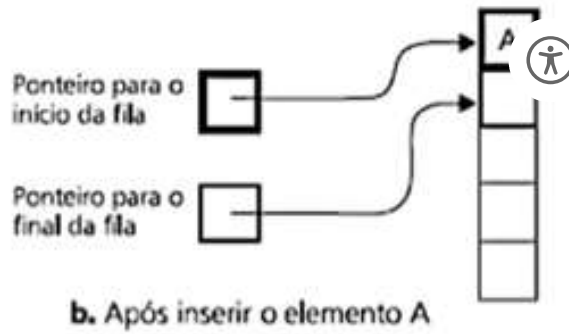
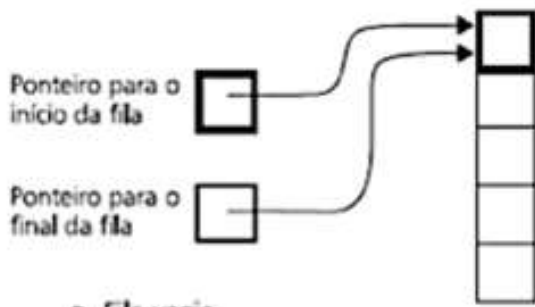


Atividade Extra

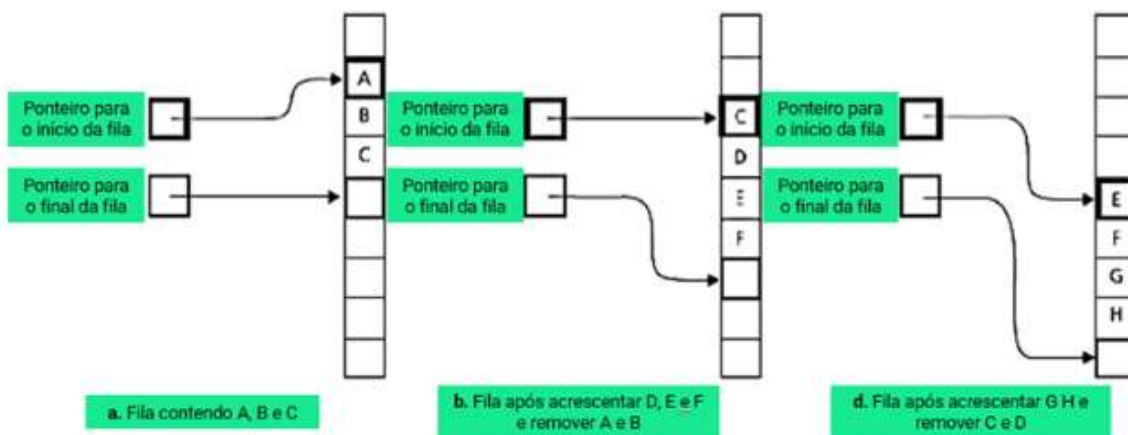
Uma outra visão sobre Filas

Em uma fila, as inserções são realizadas em uma extremidade e as remoções na outra. É o esquema first-in, first-out – FIFO. Outra forma de considerar fila é pensar que nela os objetos são retirados de acordo com a ordem com que chegaram. A extremidade na qual são retirados os elementos é chamada de início da fila, onde eles são inseridos é chamado de fim da fila.

Abaixo uma fila em diferentes momentos de processamento:



Outro exemplo da dinâmica de uma fila:



Referência Bibliográfica

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. Algoritmos Teoria e Prática. 3ª ed. Editora Cammpus, 2012.



Atividade Prática 09 – Entendendo as aplicações de Fila

Título da Prática: Aplicações com Fila em Java

Objetivos: Entender como utilizar o netbeans para desenvolver programas em Java para manipular e aplicar as operações com Fila

Materiais, Métodos e Ferramentas: Computador, netbeans, Java.

Atividade Prática

O Algoritmo de manipulação das operações de Fila para desenvolver um algoritmo que recebe do usuário cinco números inteiros numa Fila com capacidade para cinco números e mostra esses números, pode ser escrito como segue.

Desenvolva o programa em Java deste algoritmo no NetBeans.

Algoritmo Fila



Definir

// definição do tipo registro Fila com os campos abaixo

tipo Fila = **registro** // o tipo registro chama-se Fila

tamanho \leftarrow 0 **numérico_inteiro**; // armazena o tamanho da fila

início \leftarrow 0 **numérico_inteiro**; // armazena a posição inicial da fila

fim \leftarrow 0 **numérico_inteiro**; // armazena a posição final da fila

total \leftarrow 0 **numérico_inteiro**; // armazena a quantidade de elementos da fila

vetor [tamanho] **numérico_real**; // armazena os elementos da fila

fimregistro;

lógico FilaVazia()

início_módulo

se (total = 0)

então

retornar verdadeiro;

senão

retornar falso;

fimse;

fim_módulo;



lógico FilaCheia()

início_módulo

se (total >= tamanho)

então

retornar verdadeiro;

senão

retornar falso;

fimse;

fim_módulo;

Enfileirar (elemento **numérico_real**)

início_módulo

se (**não** FilaCheia())

então

vetor[fim] \leftarrow elemento;

fim \leftarrow fim + 1;

total \leftarrow total + 1;

se (fim > = tamanho)



então

fim \leftarrow 0;

fimse;

senão

escrever (“Fila Cheia”);

fimse;

fim_módulo;

numérico_real Desenfileirar ()

início_módulo

Declarar

desenfileirado \leftarrow 0.0 **numérico_real;**

se (FilaVazia())

então

escrever (“Fila Vazia”);

senão

desenfileirado \leftarrow vetor[inicio];

início ← início + 1;



total ← total – 1;

se (início >= tamanho)

então

início ← 0;

fimse;

retornar desenfileirado;

fim_módulo;

ElementoInício()

início_módulo

se (**não** FilaVazia())

então

escrever ("O primeiro elemento da fila é ", vetor[início];

senão

escrever("Fila Vazia");

fimse;

fim_módulo;

MostrarFila()



início_módulo

Declarar

i, aux **numérico_inteiro**;

aux \leftarrow início;

para i de 1 até total passo +1 faça

escrever ("Elemento " , vetor[aux] , " posição " , i);

aux \leftarrow aux + 1;

se (aux \geq tamanho)

então

aux \leftarrow 0;

fimse;

fimpara;

fim_módulo;

fim_algoritmo.

Algoritmo Exemplo1

início_algoritmo

Declarar



```
realFila ← novo Fila(5);
```

```
entrada ← 0.0 numérico_real;
```

```
i numérico_inteiro;
```

```
para i de 1 até 5 passo +1faça
```

```
    entrada ← escrever(“Digite um valor real”);
```

```
    realFila.Enfileirar(entrada);
```

```
fimpara;
```

```
realFila.MostrarFila( );
```

```
fim_algoritmo.
```

Gabarito Atividade Prática

```
import javax.swing.*;
```

```
class Fila
```



```
{
```

```
    int tamanho, inicio, fim, total;
```

```
    double vetor[];
```

```
    Fila(int tam)
```

```
    {
```

```
        inicio = 0;
```

```
        fim = 0;
```

```
        total = 0;
```

```
        tamanho = tam;
```

```
        vetor = new double[tam];
```

```
    }
```

```
    public boolean FilaVazia()
```

```
    {
```

```
        if (total == 0)
```

```
        {
```

```
            return true;
```

```
        }
```

```
    else
```



```
{
```

```
    return false;
```

```
}
```

```
}
```

```
public boolean FilaCheia()
```

```
{
```

```
    if (total >= tamanho)
```

```
    {
```

```
        return true;
```

```
    }
```

```
    else
```

```
    {
```

```
        return false;
```

```
    }
```

```
}
```

```
public void Enfileirar (double elemento)
```

```
{
```

if (! FilaCheia())



{

vetor[fim] = elemento;

fim = fim + 1;

total = total + 1;

if (fim >= tamanho)

{

fim = 0;

}

}

else

{

System.out.println ("Fila Cheia");

}

}

// retorna o valor desenfileirado

public double Desenfileirar ()

{

double desenfileirado = 0.0;



if (FilaVazia())

{

System.out.println("Fila Vazia");

}

else

{

desenfileirado = vetor[inicio];

inicio = inicio + 1;

total = total - 1;

if (inicio >= tamanho)

{

inicio = 0;

}

}

return desenfileirado;

}

public void ElementoInicio()



```
{
```

```
if (!FilaVazia())
```

```
{
```

```
    System.out.println("O primeiro elemento é "+ vetor[inicio]);
```

```
}
```

```
else
```

```
{
```

```
    System.out.println("Fila Vazia");
```

```
}
```

```
}
```

```
public void MostrarFila( )
```

```
{
```

```
    int i, aux;
```

```
    aux = inicio;
```

```
    for (i = 1 ; i <= total ; i++)
```

```
{
```

```
JOptionPane.showMessageDialog(null, "Elemento " + vetor[aux]  
    + " posição " + i);
```



```
aux = aux + 1;
```

```
if (aux >= tamanho)
```

```
{
```

```
    aux = 0;
```

```
}
```

```
}
```

```
}
```

```
}
```

```
class Exemplo1
```

```
{
```

```
public static void main (String arg [])
```

```
{
```

```
Fila realFila = new Fila(5);
```

```
double entrada = 0.0;
```

```
int i;
```

```
for (i = 0 ; i < 5 ; i++)
```



```
{
```

```
    entrada = Double.parseDouble(JOptionPane.showInputDialog(  
        "Digite um valor real"));
```

```
    realFila.Enqueueirar(entrada);
```

```
}
```

```
realFila.MostrarFila( );
```

```
System.exit(0);
```

```
}
```

```
}
```

Ir para exercício