

Filas Circulares

TIPO DE DADOS FILA
CIRCULAR
UTILIZANDO
ARRANJOS 1D EM
JAVA



Concluído
anteriormente...

Estudamos TD Fila

- Verificamos o impacto que causado pela movimentação de toda a fila quando um elemento é removido.

Estudamos TAD Fila Circular.

- Melhoria no TD Fila implementado:
 - Muda a indicação do início da Fila para evitar a movimentação de todos os elementos da Fila.
- Estudamos como definir:
 - novoInicio
 - novoFim
 - isEmpty()
 - isFull()



Objetivo deste Tutorial...

Implementar o TD Fila Circular em Java.

- Criar uma classe Fila Circular
- Implementar as operações básicas de uma TAD Fila Circular por meio de métodos da classe Fila Circular



Operações básicas sobre Filas Circulares

push(valor) – insere o “valor” no final da fila, se ela não estiver cheia e atualiza a fila.

pop() – retorna o valor que está no início da fila, se ela não estiver vazia, e atualiza a fila.

isEmpty() – retorna se a fila está vazia

isFull() – retorna se a fila está cheia

initialize() – cria ou inicializa a fila.

destroy() – apaga o conteúdo da fila.



Utilização desejada da Classe Fila Circular

```
1  public class Principal_a{
2      public static void main(String args[]){
3          FilaCircular f = new FilaCircular(5); // construtor substitui initialize
4
5          for(int i=0; i<3; i++) f.push(i+1);
6          f.status();
7          System.out.println(f.pop());
8          System.out.println(f.pop());
9          f.status();
10         for(int i=3; i<6; i++) f.push(i);
11         f.status();
12     }
13 }
```



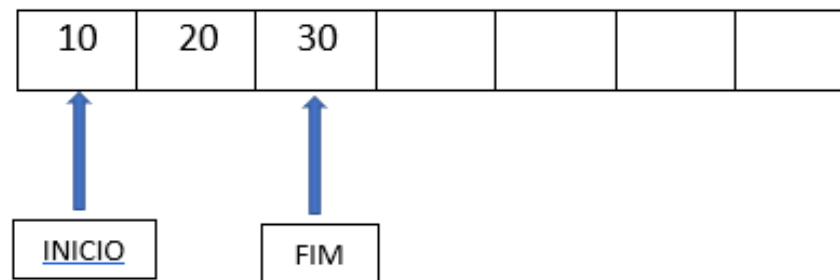
Utilização desejada da Classe Fila Circular

```
1  public class Principal_a{
2      public static void main(String args[]){
3          FilaCircular f = new FilaCircular(5); // construtor substitui initialize
4
5          for(int i=0; i<3; i++) f.push(i+1);
6          f.status();
7          System.out.println(f.pop());
8          System.out.println(f.pop());
9          f.status();
10         for(int i=3; i<6; i++) f.push(i);
11         f.status();
12     }
13 }
```

← Apresenta os atributos da Fila Circular



Atributos de uma TAD FILA CIRCULAR



Requisitos:

- Arranjo 1D para armazenar valores
- Posição do INÍCIO e FIM da FILA CIRCULAR.



Atributos da Classe Fila Circular

Requisitos:

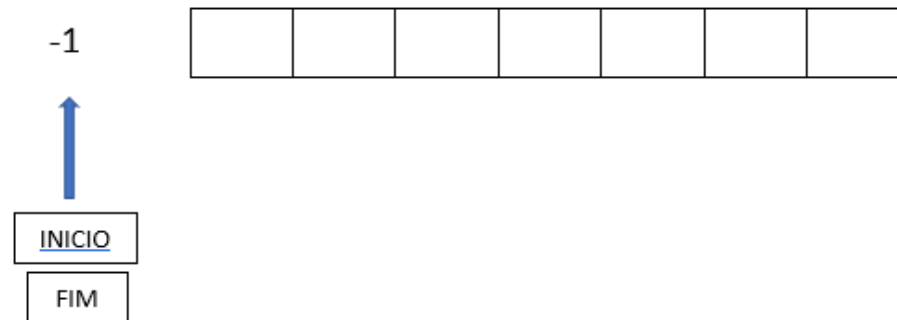
- Arranjo 1D para armazenar valores
- Posição do topo da Fila.

```
1  public class FilaCircular{  
2      private int[] vetor;  
3      private int inicio, fim;
```



Operações sobre uma FILA CIRCULAR

Inicialização:



FILA vazia: $\text{INICIO} = \text{FIM} = -1$



Inicialização da Fila Circular com o método construtor

Programa Principal:

```
3      FilaCircular f = new FilaCircular(5); // construtor substitui initialize
```

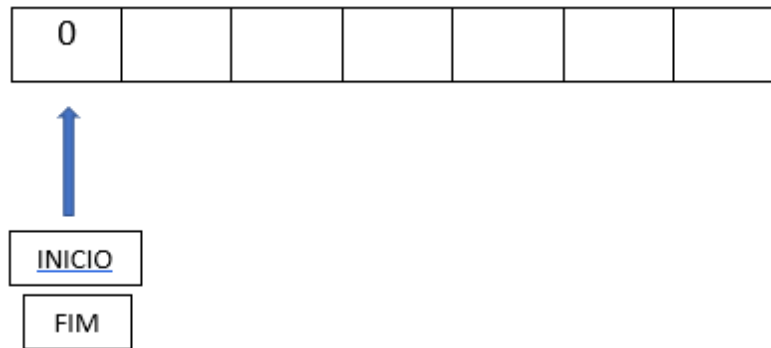
Construtor na Classe Fila:

```
5      public FilaCircular(int tam){  
6          vetor = new int[tam];  
7          inicio = fim = -1; // fila vazia  
8      }
```



Operações sobre uma FILA CIRCULAR

push(x)

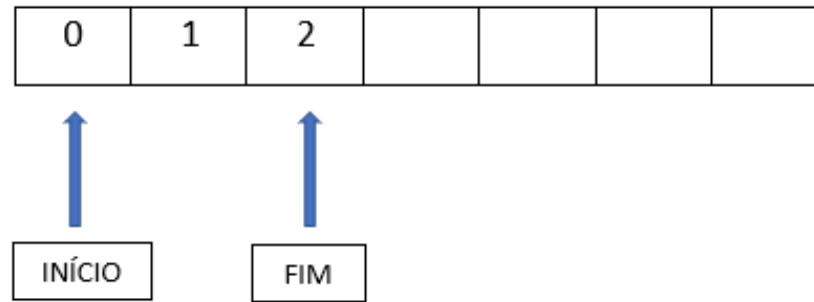


FILA: INICIO = FIM = 0



Operações sobre uma FILA CIRCULAR

push(y)
push(z)



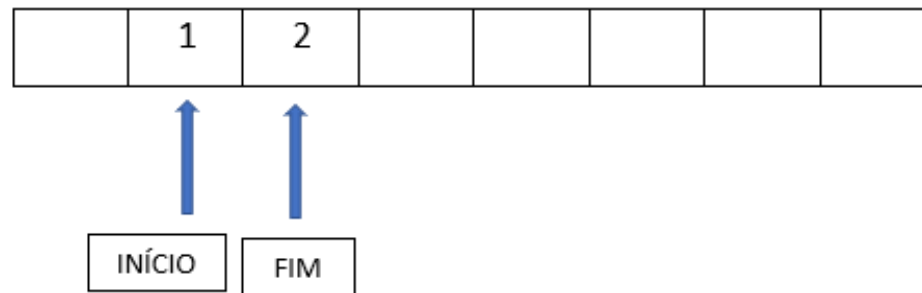
FILA: INICIO = 0 FIM = 2



Operações sobre uma FILA CIRCULAR

pop()

sai: 0



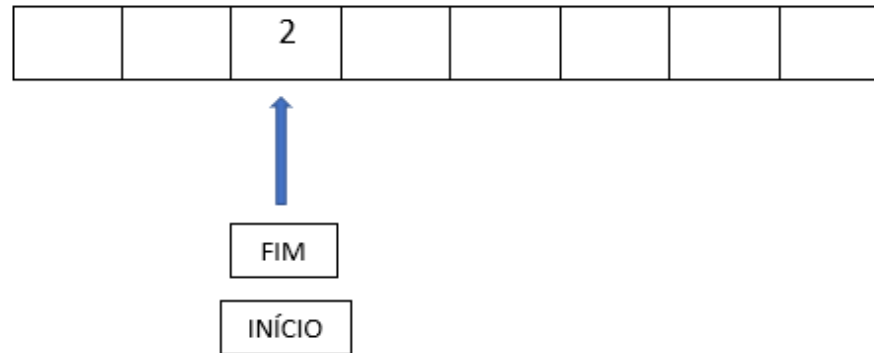
FILA: INICIO = 1 FIM = 2



Operações sobre as FILAS

pop()

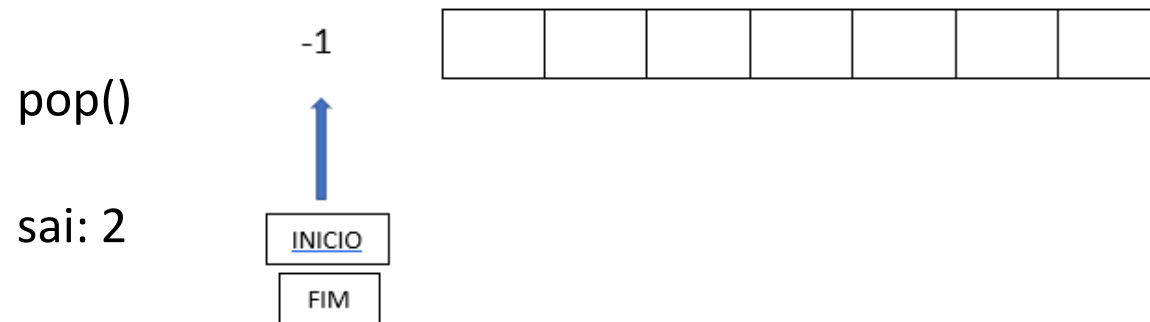
sai: 1



FILA: INICIO = 2 FIM = 2



Operações sobre as FILAS



FILA: INICIO = -1 FIM = -1



Considerações sobre o funcionamento da Fila Circular

inicio e fim são iguais a -1, se a Fila esta vazia.

início guarda a primeira posição da Fila, se a Fila não está vazia.

fim indica a posição do último elemento da Fila

Inserção (push) é feita sempre na posição posterior ao fim da Fila (novoFim).

Remoção (pop) é feita sempre na posição indicada por inicio que é atualizado para novoInicio.

novofim indica a posição do inicio, se a fila estiver cheia.



Considerações sobre o funcionamento da Fila Circular

Em uma fila circular:

- $\text{inicio} == \text{fim} == -1 \Rightarrow$ fila vazia
- $\text{novoFim} == \text{inicio} \Rightarrow$ fila cheia
- $\text{novoFim} \neq \text{inicio} \Rightarrow$ posição do último elemento da fila.



Inserção na Fila Circular – método push()

Programa Principal:

```
6      f.push(aux);
```

push() na Classe Fila Circular:

```
10     public void push(int valor){
11         int novoFim = fim + 1;
12
13         if(novoFim == vetor.length){
14             novoFim = 0;
15         }
16
17         if(novoFim != inicio){
18             vetor[novoFim] = valor;
19             if (inicio != -1)
20                 fim = novoFim;
21         else
22             inicio = fim = novoFim;
23     } else System.out.println("Fila Cheia!");
24 }
```



Inserção na Fila Circular – método push()

Programa Principal:

```
6      f.push(aux);
```

push() na Classe Fila Circular:

```
10     public void push(int valor){
11         int novoFim = fim + 1;
12
13         if(novoFim == vetor.length){
14             novoFim = 0;
15         }
16
17         if(novoFim != inicio){
18             vetor[novoFim] = valor;
19             if (inicio != -1)
20                 fim = novoFim;
21             else
22                 inicio = fim = novoFim;
23         } else System.out.println("Fila Cheia!");
24     }
```

Define
novoFim



Inserção na Fila Circular – método push()

Programa Principal:

```
6      f.push(aux);
```

push() na Classe Fila Circular:

```
10     public void push(int valor){
11         int novoFim = fim + 1;
12
13         if(novoFim == vetor.length){
14             novoFim = 0;
15         }
16
17         if(novoFim != inicio){
18             vetor[novoFim] = valor;
19             if (inicio != -1)
20                 fim = novoFim;
21         else
22             inicio = fim = novoFim;
23     } else System.out.println("Fila Cheia!");
24 }
```

Define
novoFim

Verifica se a fila
não está cheia



Inserção na Fila Circular – método push()

Programa Principal:

6 `f.push(aux);`

push() na Classe Fila Circular:

```
10    public void push(int valor){
11       int novoFim = fim + 1;
12
13       if(novoFim == vetor.length){
14           novoFim = 0;
15       }
16
17       if(novoFim != inicio){
18           vetor[novoFim] = valor;
19           if (inicio != -1)
20               fim = novoFim;
21           else
22               inicio = fim = novoFim;
23       } else System.out.println("Fila Cheia!");
24    }
```

Define
novoFim

Verifica se a fila
não está cheia

Atualiza
posição do fim



Inserção na Fila Circular – método push()

Programa Principal:

6 `f.push(aux);`

push() na Classe Fila Circular:

```
10    public void push(int valor){
11        int novoFim = fim + 1;
12
13        if(novoFim == vetor.length){
14            novoFim = 0;
15        }
16
17        if(novoFim != inicio){
18            vetor[novoFim] = valor;
19            if (inicio != -1)
20                fim = novoFim;
21            else
22                inicio = fim = novoFim;
23        } else System.out.println("Fila Cheia!");
24    }
```

Define
novoFim

Verifica se a fila
não está cheia

Atualiza
posição do fim

Erro
por fila
cheia

Remoção da Fila Circular – método pop()

Programa Principal:

```
10      System.out.println(p.pop());
```

pop() na Classe Fila Circular:

```
26      public int pop(){
27          int aux;
28
29          if(inicio != -1){
30              aux = vetor[inicio];
31              if(inicio != fim){
32                  inicio = inicio + 1;
33                  if(inicio == vetor.length)
34                      inicio = 0;
35              } else{
36                  inicio = fim = -1;
37              }
38              return aux;
39          } else {
40              System.out.println("Fila Vazia!");
41              return -99999;
42          }
43      }
```



Remoção da Fila Circular – método pop()

Programa Principal:

```
10      System.out.println(p.pop());
```

pop() na Classe Fila Circular:

```
26      public int pop(){
27          int aux;
28
29          if(inicio != -1){
30              aux = vetor[inicio];
31              if(inicio != fim){
32                  inicio = inicio + 1;
33                  if(inicio == vetor.length)
34                      inicio = 0;
35              } else{
36                  inicio = fim = -1;
37              }
38              return aux;
39          } else {
40              System.out.println("Fila Vazia!");
41              return -99999;
42          }
43      }
```

Valor a ser
removido



Remoção da Fila Circular – método pop()

Programa Principal:

```
10      System.out.println(p.pop());
```

pop() na Classe Fila Circular:

```
26      public int pop(){
27          int aux;
28
29          if(inicio != -1){
30              aux = vetor[inicio];
31              if(inicio != fim){
32                  inicio = inicio + 1;
33                  if(inicio == vetor.length)
34                      inicio = 0;
35              } else{
36                  inicio = fim = -1;
37              }
38              return aux;
39          } else {
40              System.out.println("Fila Vazia!");
41              return -99999;
42          }
43      }
```

Valor a ser
removido

Define novo
inicio



Remoção da Fila Circular – método pop()

Programa Principal:

```
10      System.out.println(p.pop());
```

pop() na Classe Fila Circular:

```
26      public int pop(){
27          int aux;
28
29          if(inicio != -1){
30              aux = vetor[inicio];
31              if(inicio != fim){
32                  inicio = inicio + 1;
33                  if(inicio == vetor.length)
34                      inicio = 0;
35              } else{
36                  inicio = fim = -1;
37              }
38              return aux;
39          } else {
40              System.out.println("Fila Vazia!");
41              return -99999;
42          }
43      }
```

Valor a ser removido

Define novo inicio

Erro por fila vazia



Métodos isFull() e isEmpty()

```
45     public boolean isFull(){
46         int novoFim = fim + 1;
47         if(novoFim == vetor.length)
48             novoFim = 0;
49         return (inicio == novoFim);
50     }
51
52     public boolean isEmpty(){
53         return (fim == -1);
54     }
```



Utilização desejada da Classe Fila Circular

```
1 public class Principal_a{
2     public static void main(String args[]){
3         FilaCircular f = new FilaCircular(5); // construtor substitui initialize
4
5         for(int i=0; i<3; i++) f.push(i+1);
6         f.status();
7         System.out.println(f.pop());
8         System.out.println(f.pop());
9         f.status();
10        for(int i=3; i<6; i++) f.push(i);
11        f.status();
12    }
13 }
```

```
C:\Users\eduar\Desktop\ED\Filas circulares>java Principal_a
Tam = 5 Inicio: 0 Fim: 2
1 2 3 0 0
1
2
Tam = 5 Inicio: 2 Fim: 2
1 2 3 0 0
Tam = 5 Inicio: 2 Fim: 0
5 2 3 3 4
```



Conclusões:

Apresentou-se a implementação de uma Fila Circular.

Foi utilizado Programação Orientada a Objetos

- Construímos uma classe Fila Circular
- Atributos foram um arranjo 1D para valores inteiros e variáveis inteiro **início** e **fim**.
- Foram implementados as principais operações sobre uma Fila Circular na forma de métodos da classe.



Exercício proposto 01:

Modifique a classe Fila para que ela seja capaz de armazenar dados do tipo caractere.



Exercício proposto 02:

Usando a classe que você rescreveu no exercício proposto 02 do tutorial sobre Pilhas, construa um programa Principal que guarda uma mensagem em uma fila e inverte as letras de um texto terminado por ponto (.), preservando a ordem das palavras, usando chamadas às classes Fila e Pilha.

Por exemplo, dado o
texto:

ESTE EXERCICIO E MUITO FACIL. TENTE!!!!

A saída deve ser:

ETSE OICICREXE E OTIUM LICAF



Exercício proposto 03:

Em várias partes do código fonte de filas circulares desenvolvido para este tutorial, foi necessário calcular a posição do novoFim.

Com isso, acabamos repetindo esse processo.

Crie um método chamado novoFim() que retorna um inteiro que é a posição relativa ao novo Fim na inserção na fila circular e rescreva a classe Fila Circular para que ela utilize o método que você criou.

