

Filas

TIPO DE DADOS FILA
UTILIZANDO
ARRANJOS 1D EM
JAVA

Concluído
anteriormente...

Estudamos TD Fila

- Lista Linear com restrição de acesso tipo FIFO – ***First in First Out***
- Definimos principais operações sobre Filas

Estudamos TD Pilhas em Java.

- Criação e manipulação de Pilhas por meio de arranjos 1D em JAVA

Objetivo deste Tutorial...

Implementar o TD Fila em Java.

- Criar uma classe Fila
- Implementar métodos da classe Fila que realizarão as operações básicas de uma TAD Fila.
- Utilizar conhecimentos obtidos com a implementação do TD Pilha em Java.

Operações básicas sobre Filas

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

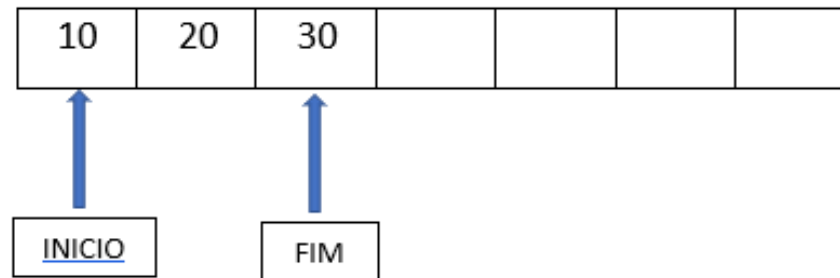
```
1 public class Principal{
2     public static void main(String args[]){
3         int aux=1;
4         Fila f = new Fila(5); // construtor substitui initialize
5         while(!f.isFull()){
6             f.push(aux);
7             aux++;
8         }
9         while(!f.isEmpty()){
10             System.out.println(f.pop());
11         }
12     }
13 }
```

Utilização desejada da Classe Fila

```
1 public class Principal{
2     public static void main(String args[]){
3         int aux=1;
4         Fila f = new Fila(5); // construtor substitui initialize
5         while(!f.isFull()){
6             f.push(aux);
7             aux++;
8         }
9         while(!f.isEmpty()){
10             System.out.println(f.pop());
11         }
12     }
13 }
```

Utilização idêntica à da TD PILHA

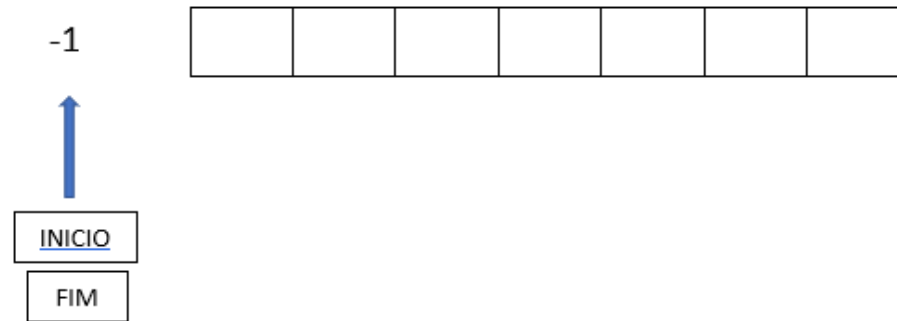
Atributos de uma TAD FILA



Requisitos:

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

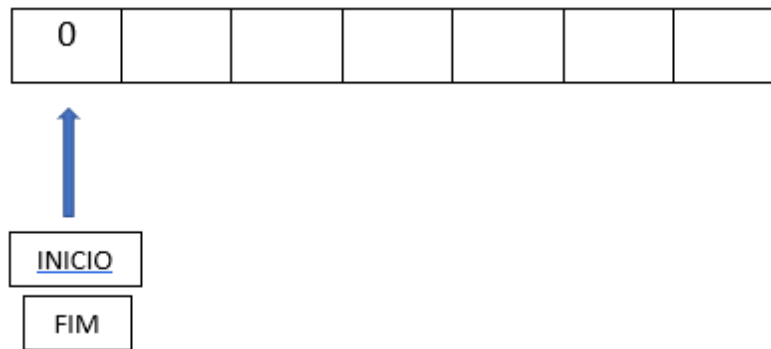
Operações sobre as FILAS



FILA vazia: INICIO = FIM = -1

Operações sobre as FILAS

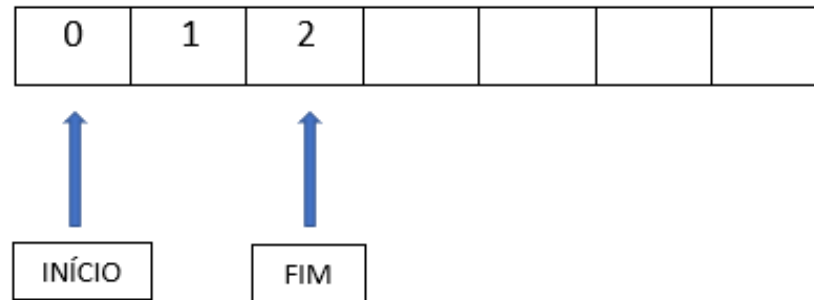
push(x)



FILA: INICIO = FIM = 0

Operações sobre as FILAS

push(y)
push(z)

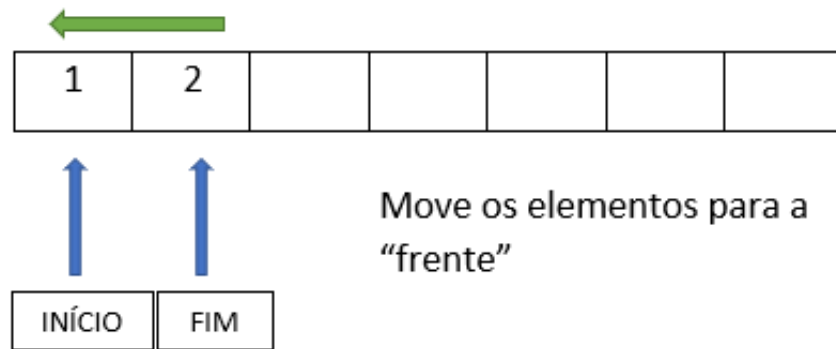


FILA: INICIO = 0 FIM = 2

Operações sobre as FILAS

pop()

sai: 0

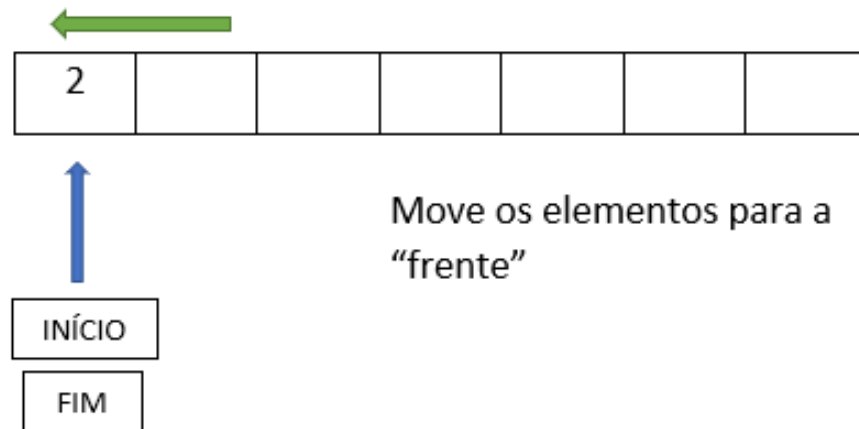


FILA: INICIO = 0 FIM = 1

Operações sobre as FILAS

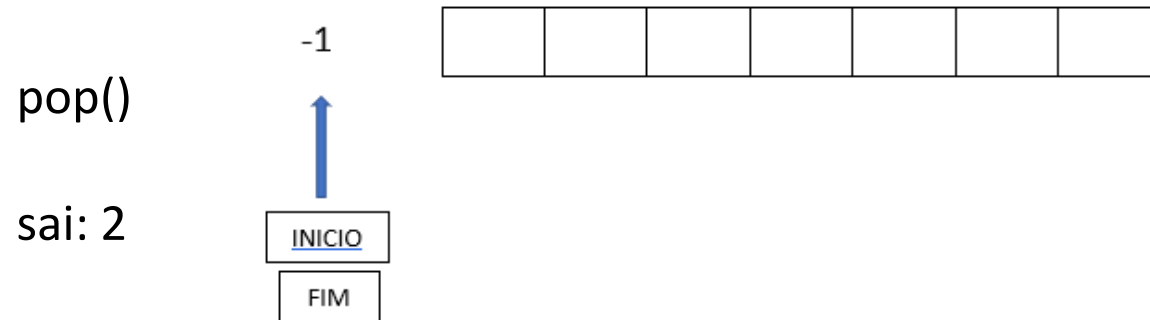
pop()

sai: 1



FILA: INICIO = 0 FIM = 0

Operações sobre as FILAS



FILA: INICIO = -1 FIM = -1

Considerações sobre o funcionamento da Fila

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

início ocupa sempre a primeira posição do arranjo, se a pilha não está vazia. Não há necessidade de se armazenar essa posição (0).

fim indica a **posição do último elemento da Fila**

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

Remoção (**pop**) é feita sempre na primeira posição do vetor e todos os valores da fila, a partir do segunda posição, são movidos uma posição à frente.

fim ocupa a última posição do arranjo, se a fila estiver cheia.

Considerações sobre o funcionamento da fila

Apenas com o atributo fim é possível gerenciar uma fila.

- $\text{fim} = -1 \Rightarrow$ fila vazia
- $\text{fim} = \text{length} - 1 \Rightarrow$ fila cheia
- fim entre 0 e $\text{length} - 1 \Rightarrow$ posição do último elemento da fila.

Começando a Classe Fila...

Requisitos:

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

```
1 public class Fila{  
2     private int[] vetor;  
3     private int fim;
```


Inicialização da Fila com o método construtor

Programa Principal:

```
4      Fila f = new Fila(5); // construtor substitui initialize
```

Construtor na Classe Fila:

```
5      public Fila(int tam){  
6          vetor = new int[tam];  
7          fim = -1;  
8      }
```

Inserção na Fila – método push()

Programa Principal:

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

push() na Classe Fila:

```
10     public void push(int valor){
11         if(fim == vetor.length-1)
12             System.out.println("Fila Cheia!");
13         else{
14             fim++;
15             vetor[fim]=valor;
16         }
17     }
```

Remoção da Fila – método pop()

Programa Principal:

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

pop() na Classe Fila:

```
19      public int pop(){
20          int aux = vetor[0];
21          if(fim>=0){
22              for(int i=0; i<fim; i++)
23                  vetor[i] = vetor[i+1];
24              fim--;
25              return aux;
26          } else{
27              System.out.println("Fila Vazia!");
28              return -99999;
29          }
30      }
```

Remoção da Fila – método pop()

Programa Principal:

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

pop() na Classe Fila:

```
19      public int pop(){
20          int aux = vetor[0];
21          if(fim>=0){
22              for(int i=0; i<fim; i++)
23                  vetor[i] = vetor[i+1];
24              fim--;
25              return aux;
26          } else{
27              System.out.println("Fila Vazia!");
28              return -99999;
29          }
30      }
```



Valor a ser removido

Remoção da Fila – método pop()

Programa Principal:

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

pop() na Classe Fila:

```
19      public int pop(){
20          int aux = vetor[0];
21          if(fim>=0){
22              for(int i=0; i<fim; i++)
23                  vetor[i] = vetor[i+1];
24              fim--;
25              return aux;
26          } else{
27              System.out.println("Fila Vazia!");
28              return -99999;
29          }
30      }
```

Valor a ser removido

Move a fila para frente

Remoção da Fila – método pop()

Programa Principal:

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

pop() na Classe Fila:

```
19  public int pop(){
20      int aux = vetor[0];
21      if(fim>=0){
22          for(int i=0; i<fim; i++)
23              vetor[i] = vetor[i+1];
24          fim--;
25          return aux;
26      } else{
27          System.out.println("Fila Vazia!");
28          return -99999;
29      }
30  }
```

Valor a ser removido

Move a fila para frente

Erro por fila vazia

Métodos isFull() e isEmpty()

```
32     public boolean isFull(){
33         return (fim == vetor.length-1);
34     }
35
36     public boolean isEmpty(){
37         return (fim == -1);
38     }
39 }
```

Utilização da Classe Fila

```
1 public class Principal{
2     public static void main(String args[]){
3         int aux=1;
4         Fila f = new Fila(5); // construtor substitui initialize
5         while(!f.isFull()){
6             f.push(aux);
7             aux++;
8         }
9         while(!f.isEmpty()){
10             System.out.println(f.pop());
11         }
12     }
13 }
```

```
C:\Users\eduar\Desktop\ED\Filas>java Principal
```

```
1
2
3
4
5
```


Conclusões:

Apresentou-se a implementação de uma Fila.

Foi utilizado Programação Orientada a Objetos

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

Implementação do pop movimenta todos elementos da fila para a frente -> elevada carga computacional.

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