



Programação O. O. II - Fila

Prof. Ronaldo Serpa da Rosa
ronaldo.rosa@bento.ifrs.edu.br

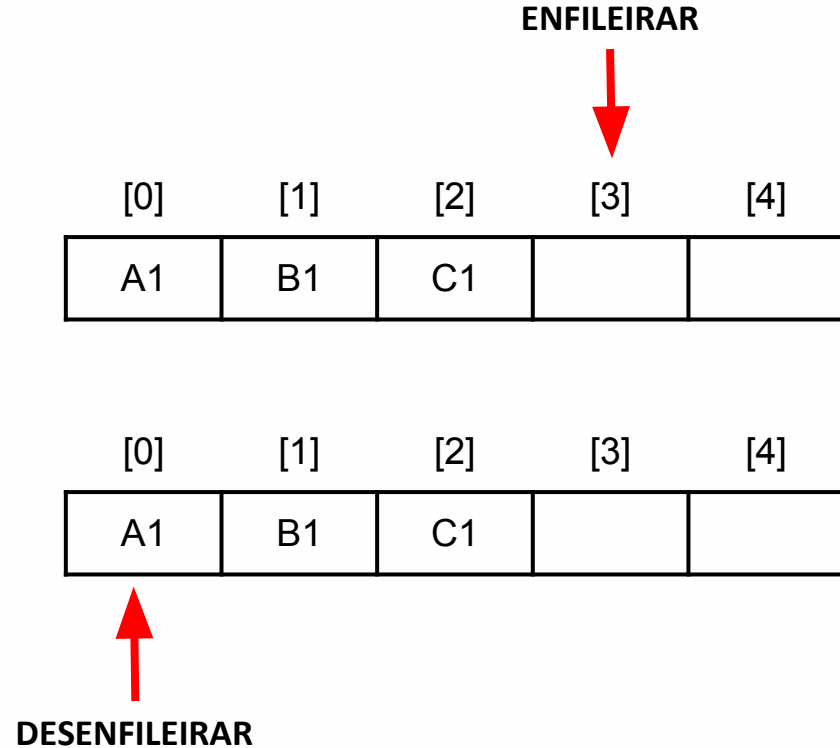
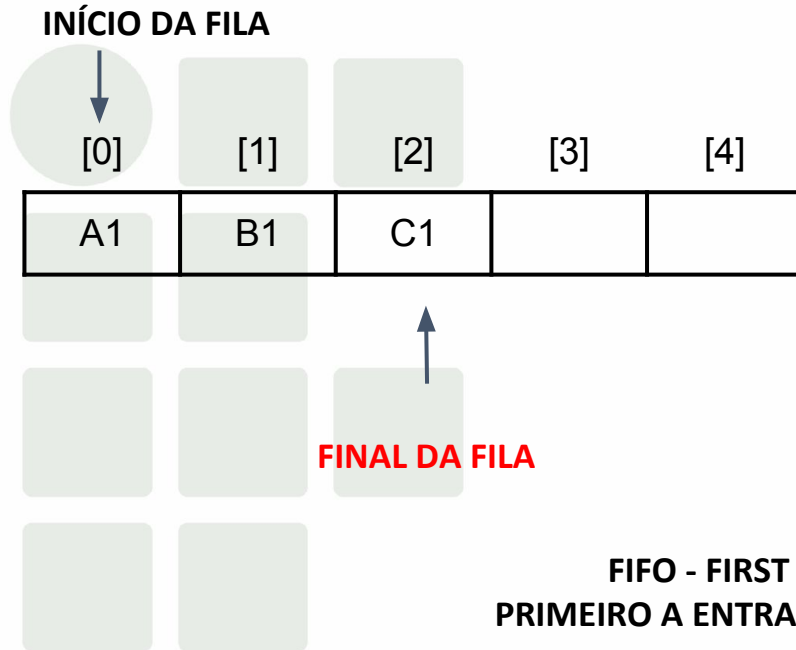
Fila



DERAL

riçalves

FILA



FIFO - FIRST IN FIRST OUT
PRIMEIRO A ENTRAR - PRIMEIRO A SAIR

Implementação da FILA

```
public class Fila {  
    private String [] elementos;  
    private int tamanho;  
    public int getTamanho() {  
        return tamanho;}  
    public Fila(int capacidade) {  
        this.elementos = new String[capacidade];  
        this.tamanho = 0;  
    }  
}
```

Enfileirar elementos

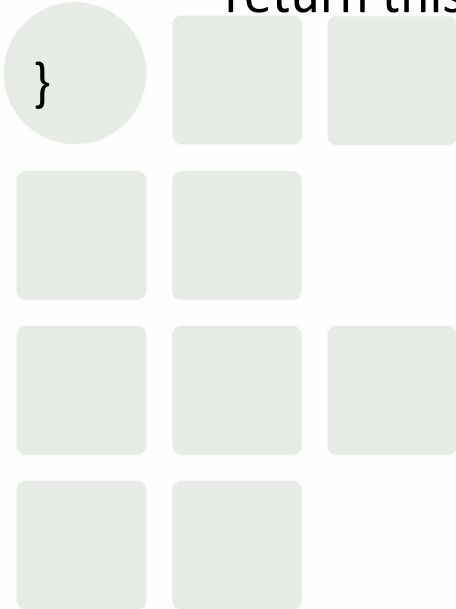
```
public boolean enfileirar(String elemento) {
```

?????

```
}
```

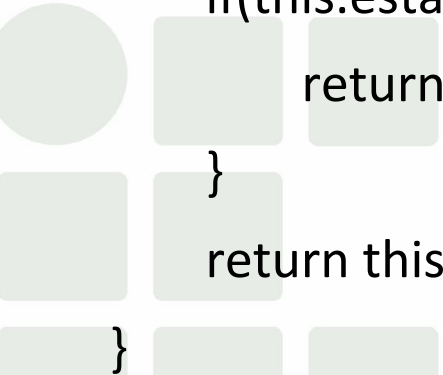
Verificando se a fila está vazia

```
public boolean estaVazia() {  
    return this.tamanho == 0;  
}
```



Visualizando o próximo elemento

```
public String proximoElemento() {  
    if(this.estaVazia()) {  
        return null;  
    }  
    return this.elementos[????????????];  
}
```



The diagram illustrates a queue structure with 10 slots. The first slot is a circle, and the others are squares. The first four slots are occupied by elements, and the last six are empty.

Desenfileirando um elemento

```
public String desenfileirar() {
```

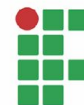
?????

}

Imprimindo os elementos da fila

@Override

```
public String toString() {  
    StringBuilder s = new StringBuilder();  
    s.append("[");  
    for(int i=0;i<tamanho;i++) {  
        s.append(this.elementos[i]);  
        s.append(",");    }  
    if(this.tamanho>0) {  
        s.append(this.elementos[this.tamanho-1]);    }  
    s.append("]");  
    return s.toString();}
```



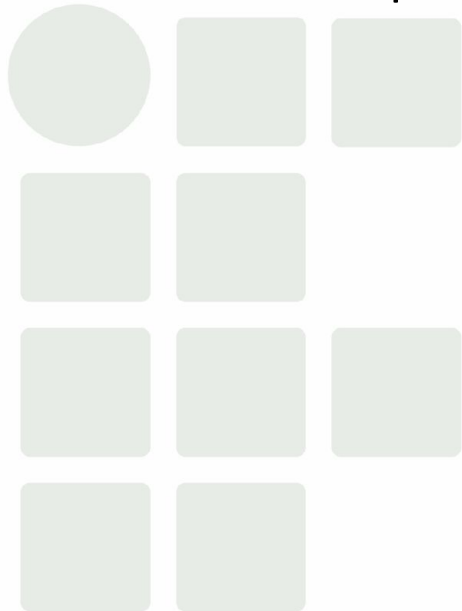
Aumentando a capacidade da fila

```
private void aumentaCapacidade() {  
    if(this.tamanho == this.elementos.length) {  
        String[] elementosNovos = new String[this.elementos.length*2];  
        for(int i=0;i<this.elementos.length;i++) {  
            elementosNovos[i] = this.elementos[i];  
        }  
        this.elementos = elementosNovos;  
    }  
}
```

*******Chamar este método na primeira linha do método enfileirar*******

Exercício 1

- Criar uma classe chamada Fila e implementar os métodos de acordo com conteúdo apresentado do slide 5 até o slide 10



Exercício 2

- Criar uma classe chamada TestaFila (com método main);
- Importar a classe Fila (criada anteriormente);
- Realizar as operações abaixo utilizando os métodos da classe Vetor;
 - Criar uma fila com capacidade para 5 elementos;
 - Verificar e imprimir se a fila está vazia;
 - Adicionar 3 elementos na fila;
 - Imprimir o tamanho da fila;
 - Imprimir todos os elementos da fila;
 - Imprimir o próximo elemento a ser desenfileirado;
 - Adicionar mais 3 elementos na fila;
 - Verificar e imprimir se a fila está vazia;
 - Desenfileirar 1 elemento da fila;
 - Imprimir o próximo elemento a ser desenfileirado;
 - Desenfileirar 1 elemento da fila;
 - Imprimir todos os elementos da fila novamente;
 - Imprimir o tamanho da fila novamente;

Exercício 3 - Fila de atendimento

- Escreva um programa para controlar a fila de atendimento de um estabelecimento comercial de acordo com as especificações abaixo:
 - Deverão existir duas filas, uma para atendimento comercial e outra para atendimento financeiro;
 - O cliente irá ter a opção de escolher em qual fila ele deseja entrar (comercial ou financeiro).
 - De acordo com a fila que ele entrar ele irá receber uma senha. Se for comercial a senha será composta pela letra "C" e mais três dígitos numéricos, por exemplo, C001. No caso de atendimento financeiro a senha deverá ter no início a letra "F" e mais três dígitos numéricos, seguindo o mesmo padrão da senha comercial.
 - Quando o atendente (comercial ou financeiro) for realizar um atendimento, ele deverá chamar sempre o primeiro da respectiva fila. Após a fila deverá ser atualizada.

Referências

- GOODRICH, Michael T.; TAMASSIA, Roberto. Estruturas de dados e algoritmos em Java. 5. ed. Porto Alegre: Bookman, 2013
- <https://www.youtube.com/playlist?list=PLGxZ4Rq3BOBrgumpzz-l8kFMw2DLERdxi>