



# Filas Estáticas

Prof. Allan Rodrigo Leite  
Estruturas de dados



# Estrutura de dados

- Estruturas definem estratégias para
  - Armazenar um conjunto de dados
  - Visitar e recuperar os dados armazenados
- Fila
  - É uma das estruturas de dados simples e similar à pilha
  - Define uma política onde o acesso aos elementos é realizado sempre pelo início da fila
    - Os novos elementos são introduzidos no fim da fila (cauda)
    - A remoção dos elementos é realizada no início da fila (cabeça)



# Filas

- Aplicações

- ☐ Spool de impressão
- ☐ Ordenação dos pacotes enviados e recebidos por um dispositivo de rede
- ☐ Fila de atendimento de um hospital

- Política de acesso

- ☐ Os elementos da fila são retirados na mesma ordem em que foram introduzidos
- ☐ **FIFO – First In, First Out**



# Filas

- Tipos de filas

- Simples

- Filas de espera

- queue

- Filas duplamente terminada

- deque – double ended queue

- Filas com prioridade

- priority queue

# Filas

- Operações de uma fila

- Adicionar

- Insere um novo elemento, introduzindo-o na cauda da fila

- Remover

- Remove o elemento contido na cabeça da fila

 cauda ← NULL  
cabeça ← NULL

# Filas

## ■ Operações de uma fila

### □ Adicionar

- Insere um novo elemento, introduzindo-o na cauda da fila

### □ Remover

- Remove o elemento contido na cabeça da fila



# Filas

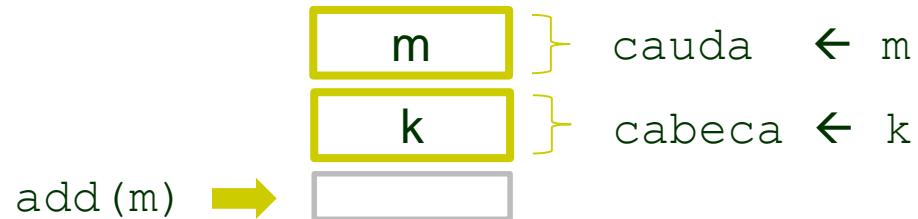
## ■ Operações de uma fila

### □ Adicionar

- Insere um novo elemento, introduzindo-o na cauda da fila

### □ Remover

- Remove o elemento contido na cabeça da fila



# Filas

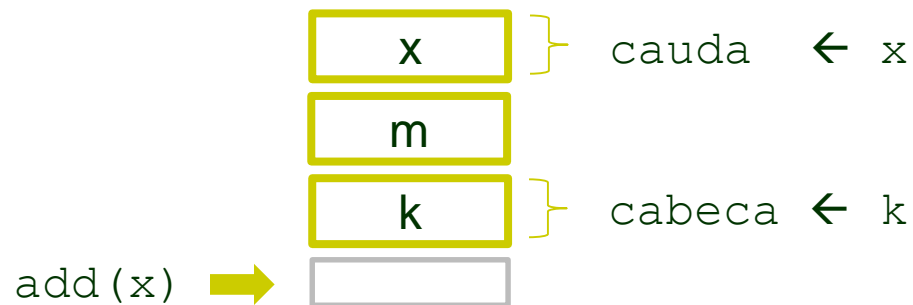
## ■ Operações de uma fila

### □ Adicionar

- Insere um novo elemento, introduzindo-o na cauda da fila

### □ Remover

- Remove o elemento contido na cabeça da fila





# Filas

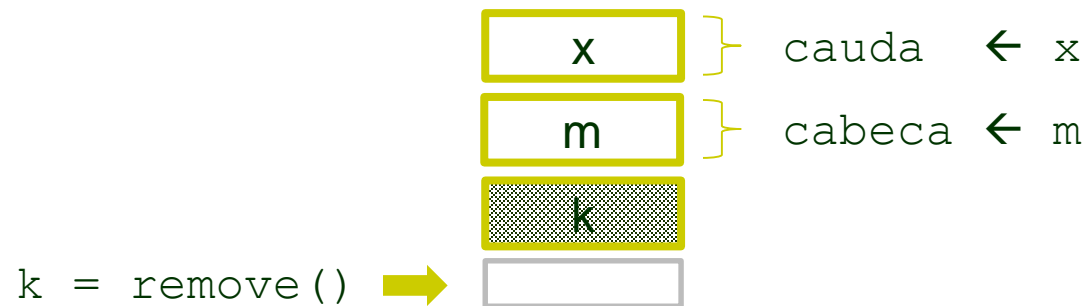
## ■ Operações de uma fila

### □ Adicionar

- Insere um novo elemento, introduzindo-o na cauda da fila

### □ Remover

- Remove o elemento contido na cabeça da fila





# Operações de filas

- Demais operações de uma fila
  - Criação e destruição
  - Inserção e remoção de elemento
  - Localização de um elemento
    - Consulta ou alteração
  - Concatenação ou divisão de uma fila

# Implementação de filas

- Implementação com vetor estático
  - Fila será armazenada em um vetor chamado `fila[N]`
    - Onde `N` é o tamanho máximo da fila
  - A parte ocupada pela fila poderá se deslocar ao longo do vetor



`add(a)`

`add(b)`

`add(c)`

# Implementação de filas

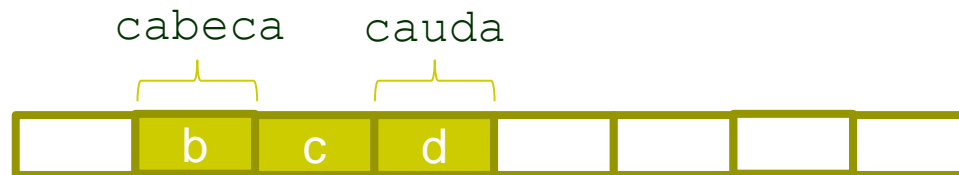
- Implementação com vetor estático
  - Fila será armazenada em um vetor chamado `fila[N]`
    - Onde `N` é o tamanho máximo da fila
  - A parte ocupada pela fila poderá se deslocar ao longo do vetor



```
a = remove()
```

# Implementação de filas

- Implementação com vetor estático
  - Fila será armazenada em um vetor chamado `fila[N]`
    - Onde `N` é o tamanho máximo da fila
  - A parte ocupada pela fila poderá se deslocar ao longo do vetor



`add(d)`

# Implementação de filas

- Implementação com vetor estático
  - Fila será armazenada em um vetor chamado `fila[N]`
    - Onde `N` é o tamanho máximo da fila
  - A parte ocupada pela fila poderá se deslocar ao longo do vetor



`add(h) ???`

# Implementação de filas

- Implementação com vetor estático
  - Fila será armazenada em um vetor chamado `fila[N]`
    - Onde `N` é o tamanho máximo da fila
  - A parte ocupada pela fila poderá se deslocar ao longo do vetor



```
add(h) //conceito de vetor circular
```

# Implementação de filas

- Estrutura para representação da fila com vetor circular

FilaEstatica
- fila Object[] - cabeca int - cauda int
+ void add(Object o) + Object remove() + boolean isFull() + boolean isEmpty() + clear()





# Implementação de filas

```
public interface FilaInterface {  
    public void add(Object o) throws Exception;  
    public Object remove() throws Exception;  
  
    public boolean isEmpty();  
    public boolean isFull();  
    public void clear();  
}
```

# Implementação de filas

```
public class FilaEstatica implements FilaInterface {  
    private Object[] fila;  
    private int cabeca;  
    private int cauda;  
  
    public FilaEstatica(int capacidade) {  
        fila = new Object[capacidade];  
        cabeca = cauda - 1;  
    }  
  
    ...  
}
```

# Implementação de filas

```
public class FilaEstatica implements FilaInterface {  
    ...  
  
    public boolean isEmpty() {  
        return cabeca == -1 && cauda == -1;  
    }  
  
    public boolean isFull() {  
        return (cauda + 1) % fila.length == cabeca;  
    }  
  
    ...  
}
```

# Implementação de filas

```
public class FilaEstatica implements FilaInterface {  
    ...  
  
    public void add(Object o) throws Exception {  
        if (this.isFull()) throw new Exception("Fila cheia");  
        System.out.println("Adicionando " + o);  
        cauda = (cauda + 1) % fila.length;  
        fila[cauda] = o;  
        if (cabeca == -1) //caso seja o 1º elemento adicionado  
            cabeca = cauda;  
    }  
  
    ...  
}
```

# Implementação de filas

```
public class FilaEstatica implements FilaInterface {
    ...
    public Object remove() throws Exception {
        if (this.isEmpty()) throw new Exception("Fila vazia");
        Object o = fila[cabeca]; //elemento na cabeça da fila
        if (cabeca == cauda)
            cabeca = cauda = -1;
        else
            cabeca = (cabeca + 1) % fila.length;
        System.out.println("Removendo " + o);
        return o;
    }
    ...
}
```

# Implementação de filas

```
public class FilaEstatica implements FilaInterface {  
    ...  
  
    public void clear() {  
        while (!this.isEmpty()) {  
            try {  
                this.remove();  
            } catch (Exception e) { e.printStackTrace(); }  
        }  
    }  
  
    ...  
}
```

# Implementação de filas

```
public class FilaEstatica implements FilaInterface {  
    ...  
  
    public void show() {  
        if (!this.isEmpty()) {  
            int i = cabeca;  
            for (i = cabeca; i != cauda; i = (i+1) % fila.length) {  
                System.out.println("Valor " + fila[i]);  
            }  
            System.out.println("Valor " + fila[i]);  
        }  
    }  
}
```

# Implementação de filas

## ■ Exercício

- ☐ Cria uma fila com capacidade máxima de 6 elementos
- ☐ Adicione os seguintes valores
  - 20,0
  - 20,8
  - 20,3
  - 44,5
  - 33,33
  - 20,9
- ☐ Exiba o conteúdo da fila
- ☐ Remova os dois primeiros elementos da fila
- ☐ Exiba o conteúdo da fila
- ☐ Esvazie a fila





# Implementação de filas

## ■ Exercício

- Faça um algoritmo que controle a sequência de atendimento de um hospital, considerando que o atendimento é realizado conforme ordem de chegada. Além disto, este hospital possui um número limite máximo de 10 atendimentos diários.
  
- Faça um algoritmo para gerenciar processos em um sistema operacional que seja capaz de realizar as seguintes operações:
  - Incluir novos processos na fila de processo
  - Executar processos da fila de processo
  - Exibir o conteúdo da fila de processo
  - Cada processo é representado por um número identificador do processo