

INSTITUTO FEDERAL
Sertão Pernambucano
Campus Salgueiro

Estrutura de Dados e Algoritmos com Java

Prof. Heraldo Gonçalves Lima Junior
heraldo.junior@ifsertao-pe.edu.br



1.1. Características de uma Fila

- ◎ **Filas são listas** nas quais o acesso somente pode ser feito em uma das extremidades, ao **final da fila**.

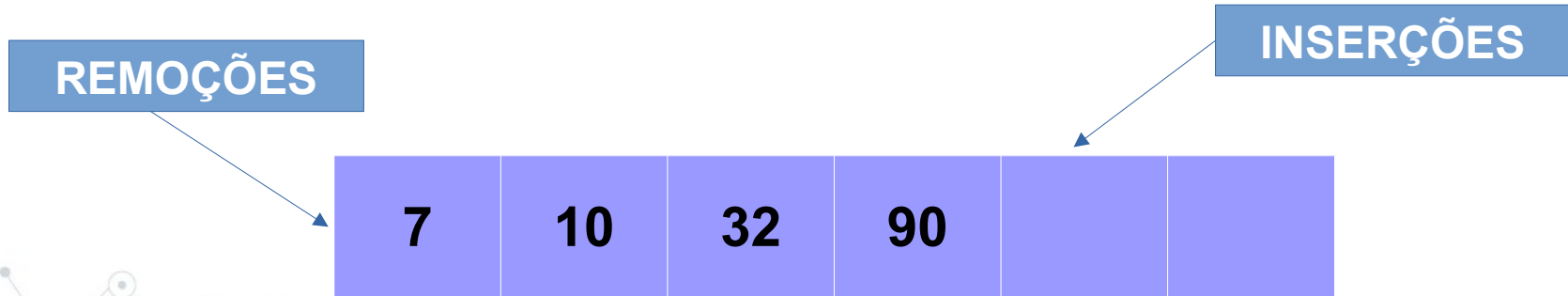
◎ **FIFO**

(o PRIMEIRO a entrar é o PRIMEIRO a sair, “**Last In, First Out**”, em inglês).



1.1. Características de uma Fila

- ⦿ As inserções são sempre no final, na primeira posição livre (**final da fila**);
- ⦿ As **remoções** acontecem na primeira posição (**início da fila**).



1.2. Pilhas x Filas

- ⊙ A implementação de Filas estáticas é bem semelhante às Pilhas.
- ⊙ O que muda é apenas onde são realizadas as operações de inserção e remoção.

1.4. Operações Básicas

- ⊙ As operações que podem ser realizadas sobre uma fila são limitadas pela disciplina de acesso que apresentam. Assim, somente as seguintes operações podem ser executadas:

- ☐ **Inserir**
- ☐ **Remover**
- ☐ **Verificar se é vazia**
- ☐ **Tamanho**
- ☐ **Verificar qual é o primeiro da fila**



2.1. Implementação da Fila Estática

- ◎ Para criar a nossa classe Fila, utilizaremos um **vetor do tipo Object** onde guardaremos os **elementos da fila**, um atributo para armazenar a **posição atual do último elemento**.

```
public class Pilha {  
    private Object elementos[];  
    private int topo;
```


2.2. Criando uma pilha

- ⦿ A criação de uma pilha resultará em uma pilha vazia, devolvendo ao usuário as informações necessárias para seu posterior acesso.
- ⦿ Deverá ser utilizada alguma convenção para indicar que a pilha está vazia. Optamos pela seguinte estratégia: **a indicação de que a pilha está vazia é feita quando o índice de topo = -1**, pois a posição zero do array já armazena informação.

2.2. Criando uma pilha

```
public Pilha(int capacidade) {  
    this.topo=-1;  
    this.elementos = new Object[capacidade];  
}
```

2.3. Inserindo um novo elemento

- ⦿ Para inserirmos um novo elemento, primeiramente devemos **verificar se a pilha não está cheia**. Feito isso, **basta inserir o novo elemento e incrementar o topo**.

```
public boolean push(Object elemento) {  
    if(this.topo < this.elementos.length-1) {  
        this.topo++;  
        this.elementos[topo] = elemento;  
        return true;  
    }  
    return false;  
}
```

2.4. Verificando se a pilha está vazia

- ⦿ Para verificarmos se a pilha está vazia, basta verificar o valor do atributo topo. Se for menor que zero, nenhum elemento está contido na pilha.

```
public boolean isEmpty(){  
    return topo < 0;  
}
```

2.5. Verificando o tamanho da pilha

- ⦿ O tamanho da pilha pode ser verificado através do atributo **topo**, que traz a posição do último elemento da pilha(o que está no topo).

```
public int size() {  
    if(this.isEmpty()) {  
        return 0;  
    }  
    return this.topo+1;  
}
```

2.6. Verificando o elemento do topo

- ⦿ O elemento do topo é o único que pode ser manipulado, então, para verificarmos quem é esse elemento, basta acessar a posição **topo** no nosso vetor.

```
public Object top() {  
    if(this.isEmpty()) {  
        return null;  
    }  
    return this.elementos[this.topo];  
}
```

2.7. Removendo um elemento

- ⦿ Na estrutura Pilha, só podemos remover o elemento do topo.

```
public Object pop() {  
    if(this.isEmpty()) {  
        return null;  
    }else {  
        return this.elementos[this.topo--];  
    }  
}
```

2.8. Conclusão

- ⦿ Como pode-se imaginar, a implementação estática de pilhas, assim como as listas, impõe várias restrições no que se refere a capacidade de crescimento da estrutura de dados.
- ⦿ Na próxima aula, veremos como implementar pilhas através de alocação dinâmica.

2.9. Exercícios

- ① 1. Implemente os códigos vistos em aula e envie através do Classroom.
- ① 2. Altere o método **pop()** para não retornar nenhuma elemento e só remover o elemento do topo da pilha.

MA, RESP IRA!



Obrigado!

Perguntas?



heraldo.junior@ifsertao-pe.edu.br



heraldolimajr.com.br