

Estrutura de Dados

Filas

Prof. Antonio
PEREIRA

Filas

- Uma fila é um tipo especial de lista linear em que as inserções são realizadas num extremo enquanto as remoções são feitas no outro. O extremo em que os elementos são inseridos é denominado final da fila e aquele em que são removidos é denominado começo da fila.
- A ordem de saída dos elementos corresponde diretamente à ordem de entrada dos mesmos na fila, de modo que os primeiros elementos que entram serão os primeiros a sair, caracterizando uma estrutura FIFO (*First-In/First-Out*).

Filas

- A palavra **queue**, da língua inglesa, significa fila. As duas operações básicas que uma fila suporta são:
 - **enqueue**: insere um elemento no final da fila.
 - **dequeue**: remove um elemento do começo da fila.
- Assim como nas pilhas a manipulação é feita apenas nas extremidades da estrutura, não havendo inserção e remoção no meio dela, podemos implementar as filas de maneira estática-sequencial ou dinâmica-encadeada.

Filas

- Ao tentarmos implementar uma Fila Estática Sequencial, deparamos com dois problemas:
 - as inserções são feitas sempre no fim da fila e as remoções no começo. Como é inviável o deslocamento de todos os elementos para a esquerda na remoção, existirão espaços em branco no início da fila que não podem ser reaproveitados, pois a inserção é sempre à direita;
 - quando inserimos elementos até ela ficar cheia ($\text{fim} = \text{MAX}$), temos que não há mais possibilidades de inserção. Porém, mesmo que removamos todos os elementos, ainda assim ela vai achar que está cheia por não ter possibilidade de inserção e vazia por não ter mais elementos.

Filas

- Para resolver o primeiro problema, podemos criar uma Fila Circular, em que, após o último elemento voltar para a primeira posição e se estiver disponível para inserção, podemos colocar mais elementos.
- O segundo problema, para descobrir se a Fila está cheia ou vazia, basta criar um contador de elementos e, se estiver zerado, é porque a fila está vazia e, se estiver com MAX, ela está cheia.

Filas

- Um outra solução seria usar a alocação Dinâmica-Encadeada. Nesta abordagem vamos cada elemento da Fila será um nó contendo um dado e o endereço do próximo nó da Fila.
- A Fila, por ser uma estrutura FIFO, deverá manipular as duas extremidades (começo e final), pois as inserções são feitas no final da Fila e as remoções no começo.
- Além disso, usaremos um contador para armazenar a quantidade de elementos que a Fila possui.

Lista Simplesmente Encadeada

```
public class Node {  
    private Object dado;  
    private Node prox;  
  
    public Object getDado() {  
        return dado;  
    }  
  
    public void setDado (Object dado) {  
        this.dado = dado;  
    }  
  
    public Node getProx() {  
        return prox;  
    }  
  
    public void setProx(Node prox) {  
        this.prox = prox;  
    }  
}
```

Fila

```
public class Fila {  
  
    private Node comeco;  
    private Node fim;  
    private int total;  
  
    //Comeco da Fila  
    //Fim da Fila  
    //Contador de Elementos  
  
    public Fila() {  
        comeco = null;  
        fim = null;  
        total = 0;  
    }  
  
    public boolean isEmpty() {  
        return (total == 0);  
    }  
  
    public void enqueue(Object x) {  
        Node novo = new Node();  
        novo.setDado(x);  
        novo.setProx(null);  
        if (isEmpty()) {  
            comeco = novo;  
            fim = comeco;  
        } else {  
            fim.setProx(novo) {  
                fim = novo;  
            }  
        }  
        total++;  
        //Cria um novo nó  
        //Coloca o dado dentro do nó  
        //Como esse será o último, não tem próximo  
        //Como estava vazia, esse nó será começo e fim  
        //Depois do fim da fila atual, o novo nó  
        //O novo fim é o novo nó  
        //Incrementa o número de elementos  
    }  
}
```


Fila

```
public Object dequeue() {  
    Object resp = null;  
    if (!isEmpty()) {  
        resp = comeco.getDado();           //Captura o dado do comeco da fila  
        comeco = comeco.getProx();         //O comeco anda para o próximo  
        total--;                           //Decrementa número de elementos  
        return resp;  
    }  
    return resp;  
}
```

```
public void print() {  
    if (isEmpty()) {  
        System.out.println("Fila Vazia!");  
    } else {  
        Node aux;  
        String saida = "";  
        aux = comeco;  
        while (aux != null) {  
            saida += aux.getDado().toString() + ", ";  
            aux = aux.getProx();  
        }  
        System.out.println("F: [ " + saida + " ]");  
    }  
}
```