1 - Conceito de Fila

Fila (*queue* em inglês) é uma estrutura de dados dinâmica para manter elementos no formato de uma fila, onde

o 1º elemento que entra é o 1º que sai.

Desta forma, ela é considerada uma estrutura do tipo FIFO (First-In First-Out).

Uma Fila pode ser implementa usando arrays ou listas encadeadas.

2 - Implementação de Fila usando Array

Um array é uma estrutura imutável, ou seja, ele não pode ser redimensionado para comportar novos elementos, mas uma fila pode comportar qualquer quantidade de elementos. Então para implementar uma fila usando array temos de estabelecer a quantidade máxima de elementos da fila, que será a quantidade de elementos do array.

Na estrutura da fila, elementos podem ser inseridos (no final) e removidos (no início) a qualquer momento, ou seja, as posições de início e fim no array são movidas para a direita à medida que os elementos são inseridos e removidos, então chegará um momento que o array terá de ser "circular", pois as primeiras posições do array estarão livres e as últimas já estarão preenchidas.

A Figura 1 mostra uma implementação circular de fila usando array. Para ter uma fila usando array é necessário ter:

- Um array para manter os dados;
- Uma variável que indica a posição de início da fila no array;
- Uma variável que indica a quantidade de elementos da fila.

Uma fila deve ter as seguintes operações:

- Criar a fila: deve criar o array e iniciar as variáveis de inicio e total de elementos na fila;
- Checar se a fila está vazia: a fila está vazia quando o total de elementos é zero;
- Checar se a fila está cheia: a fila está cheia se o total de elementos na fila é igual a quantidade de elementos do array;
- Limpar a fila: para liberar a fila sem alterar o array basta zerar o inicio e o total;
- Inserir um elemento na fila: adiciona o elemento após a última posição;
- Retirar um elemento da fila: move a posição de inicio para a próxima posição.

A classe Fila da Figura 1 contempla ainda um método para imprimir os elementos da fila, mas uma operação de imprimir não faz parte das operações padrões de uma fila.

A Figura 2 possui um código de teste da classe da Figura 1, os comentários mostram a situação da fila, a situação do array pode também ser visualizada na representação da Figura 3. Os elementos com fundo amarelo estão em uso e os demais não.

```
public class Fila {
   private int[] vetor;
   private int total; /* número de elementos na fila*/
   /* posição do próximo elemento a ser retirado da fila */
   private int inicio;

   public Fila(){
      vetor = new int[5];
      inicio = 0;
      total = 0;
   }
```

```
public class Principal {
   public static void main(String[] args) {
      Fila fila = new Fila();
      fila.imprimir();  //Fila vazia
      fila.inserir(2);
      fila.imprimir();  //2
      fila.inserir(4);
      fila.imprimir();  //2 4
      fila.inserir(6);
      fila.imprimir();  //2 4 6
      fila.inserir(8);
```

```
    /* informa se a fila está vazia */
    public boolean isVazia(){
       return total == 0;
    }

    /* informa se a fila está cheia */
    public boolean isCheia(){
       return total == vetor.length;
    }

    /* insere um elemento no final da fila */
    public void inserir(int nro){
       if( isCheia() ){
          System.out.println("Fila cheia");
       }
       else{
          /* descobre onde será inserido o próximo elemento */
          int fim = (inicio + total) % vetor.length;
          vetor[fim] = nro;
          total++;
       }
    }

    /* retira o elemento do início da fila */
    public int remover(){
       if( isVazia() ){
          System.out.println("Fila vazia");
          return -1;
       }
       else{
          int nro = vetor[inicio];
          /* atualiza a posição de início */
          inicio = (inicio + 1) % vetor.length;
          total--;
          return nro;
       }
    }

    /* remove todos os elementos da fila */
    public void limpar(){
       inicio = 0;
       total = 0;
    }

    public void imprimir(){
       if( isVazia() ){
          System.out.println("Fila vazia");
       }
       else{
          for( int i = 0, pos = inicio; i < total; i++ ){
             pos = pos % vetor.length;
             System.out.print( vetor[pos] + " ");
             pos++;
          }
          System.out.println();
       }
    }
}
```

Figura 1 – Código da classe Fila usando array.

```
        fila.imprimir();  //2 4 6 8
        fila.inserir(9);
        fila.imprimir();  //2 4 6 8 9
        fila.inserir(11); //Fila cheia
        fila.imprimir();  //2 4 6 8 9
        fila.remover();   //remove o 2
        fila.remover();   //remove o 4
        fila.remover();   //remove o 6
        fila.imprimir();  //8 9
        fila.inserir(3);
        fila.imprimir();  //8 9 3
        fila.inserir(5);
        fila.imprimir();  //8 9 3 5
    }
}
```

Figura 2 – Código da classe Principal para testar a classe Fila da Figura 1.

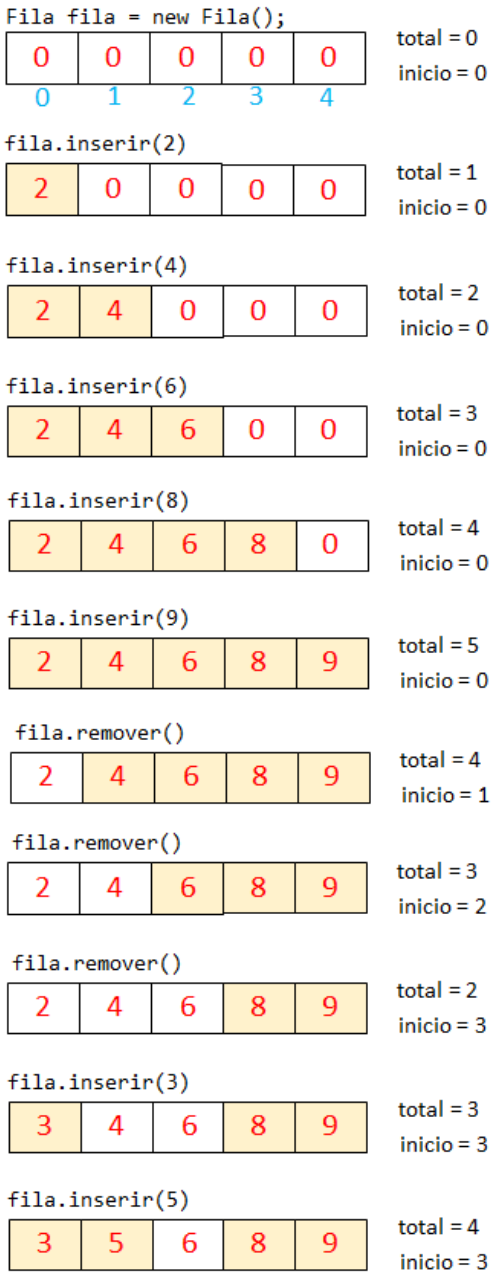

Figura 3 – Representação do código da Figura 2 no array do objeto Fila.

Uma alternativa a implementação circular no array é redimensionar o array a cada vez que a fila estiver cheia, ou seja, é necessário criar um novo array e copiar os elementos atuais para ele. Para evitar redimensionamentos frequentes, convém que o novo array tenha o dobro de elementos do atual.

3 - Implementação de Fila usando Lista Encadeada

Ao contrário do array que possui um tamanho fixo, a lista encadeada é uma estrutura dinâmica que inicia vazia e pode comportar qualquer quantidade de elementos. A implementação tradicional de lista encadeada mantém apenas uma referência para o nó de início (cabeça da lista). Ao usar uma lista para implementar uma fila será necessário manter também uma referência para o último nó da lista.

Como a fila é uma estrutura que aceita novos elementos somente no final e retiradas somente no início, então para ter uma fila usando lista encadeada é necessário ter:

- Uma referência para o inicio e fim da lista encadeada.

Uma fila deve ter as seguintes operações:

- Criar a fila: deve iniciar os atributos de inicio e fim da fila;
- Checar se a fila está vazia: a fila está vazia quando o inicio ou fim são nulos;
- Limpar a fila: para liberar a fila basta setar nulo nos atributos inicio e o total. No caso do Java, os nós (objetos do tipo No) não referenciados serão liberados pelo Garbage Collector;
- Inserir um elemento na fila: adiciona o elemento após o fim da fila;
- Retirar um elemento da fila: move a posição de inicio para a próxima posição.

Veja que não é necessário a operação para verificar se a fila está cheia, já que a lista encadeada não possui limite.

A classe Fila da Figura 4 mostra a implementação de uma fila usando lista encadeada. Como a fila utiliza uma lista então é necessário que cada elemento seja um objeto do tipo No (Figura 5).

A Figura 6 possui um código de teste da classe da Figura 4, os comentários mostram a situação da fila. Veja que o resultado é difere daquele da Figura 2 apenas ao aceitar o valor 11, já que a lista não fica cheia.

```
public class Fila {
   private No inicio, fim;

   public Fila(){
      inicio = null;
      fim = null;
   }

   /* informa se a fila está vazia */
   public boolean isVazia(){
      return fim == null;
   }

   public void inserir(int nro){
      /* criar um nó */
      No no = new No();
      no.conteudo = nro;
      no.proximo = null; /* este será o último nó da fila */
      /* checa se a lista está vazia */
      if( fim == null ){
         inicio = no;
      }
      else{
         /* alterar o próximo do último para o endereço do no */
```

```
public class No {
  int conteudo;
  No proximo;
}
```

Figura 5 – Código da classe No usada em cada elemento da Fila da Figura 4.

```
public class Principal {
   public static void main(String[] args) {
      Fila fila = new Fila();
      fila.imprimir();  //Fila vazia
      fila.inserir(2);
      fila.imprimir();  //2
      fila.inserir(4);
      fila.imprimir();  //2 4
      fila.inserir(6);
      fila.imprimir();  //2 4 6
      fila.inserir(8);
      fila.imprimir();  //2 4 6 8
      fila.inserir(9);
      fila.imprimir();  //2 4 6 8 9
```

```
            fim.proximo = no;
        }
        fim = no;
    }

    /* retira o elemento do início da fila */
    public int remover(){
        if( isVazia() ){
            System.out.println("Fila vazia");
            return -1;
        }
        else{
            int nro = inicio.conteudo;
            /* atualiza a posição de início */
            inicio = inicio.proximo;
            if( inicio == null ){
                fim = null;
            }
            return nro;
        }
    }

    /* remove todos os elementos da fila */
    public void limpar(){
        inicio = null;
        fim = null;
    }


    public void imprimir(){
        /* checa se a fila está vazia */
        if( isVazia() ){
            System.out.println("Lista vazia");
        }
        else{
            /* percorrer a fila até encontrar o último nó */
            No no = inicio;
            while( no != null ){
                System.out.print( no.conteudo +" ");
                no = no.proximo;
            }
            System.out.println(); /* quebra de linha na tela */
        }
    }
}
```

Figura 4 – Código da classe Fila usando lista encadeada.

```
        fila.inserir(11);
        fila.imprimir();  //2 4 6 8 9 11
        fila.remover();   //remove o 2
        fila.remover();   //remove o 4
        fila.remover();   //remove o 6
        fila.imprimir();  //8 9 11
        fila.inserir(3);
        fila.imprimir();  //8 9 11 3
        fila.inserir(5);
        fila.imprimir();  //8 9 11 3 5
    }
}
```

Figura 6 – Código da classe Principal para testar a classe Fila da Figura 4.

4 - Exercícios

1 – Programar uma classe Fila onde os elementos são mantidos em um array. Use a implementação não circular, ou seja, a fila estará cheia quando a última posição for preenchida.

2 – Programar uma classe Fila onde os elementos são mantidos em um array. O array deverá ser redimensionado quando a fila estiver cheia. Use a implementação não circular.

3 – Programar uma classe Fila onde os elementos são mantidos em uma lista circular. Neste caso, a fila deverá manter apenas a referência para o início da fila.