

Mãos à obra: wait() e notify() na lista

É preciso coordenar as threads. Só devemos imprimir os elementos quando a lista realmente está preenchida.

1) Quando a lista chegar no último elemento, notifique a outra thread:

```
public class Lista {

    //atributos omitidos

    public synchronized void adicionaElementos(String elemento) {
        this.elementos[indice] = elemento;
        this.indice++;

        try{
            Thread.sleep(10);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }

        if (this.indice == this.tamanho()) { //novo if
            System.out.println("lista tá cheia, notificando");
            this.notify();
        }
    }

    //outros métodos omitidos
}
```

2) Deixe a outra thread **esperar**. Só imprima os elementos quando for notificado:

```
public class TarefaImprimir implements Runnable {

    //atributo e construtor omitido

    @Override
    public void run() {

        synchronized (lista) { //novo, obtendo a chave da lista
            try { //novo try-catch
                System.out.println("esperando, aguardando notificacao");
                lista.wait();
            } catch (InterruptedException e) {
                e.printStackTrace();
            }

            for (int i = 0; i < lista.tamanho(); i++) {
                System.out.println(i + " - " + lista.pegarElemento(i));
            }
        }
    }
}
```

Execute a classe `Principal` e fique de olho no console:

```
esperando, aguardando notificacao  
lista tá cheia, notificando  
0 - Thread 0 - 0  
1 - Thread 9 - 0  
2 - Thread 7 - 1  
3 - Thread 5 - 1  
... *
```