

02

Wait antes de Notify!

Transcrição

Há ainda um risco no nosso código. Imagine que os threads preencham a lista tão rápido que a notificação acontece antes que o outro thread chame `lista.wait()`. Ou seja, na verdade não tinha a necessidade de esperar, pois a lista já foi preenchida.

Como não sabemos qual será a ordem exata de execução, isso pode acontecer. Nesse caso, o thread que chama `lista.wait()` ficará esperando para sempre.

É preciso se proteger e ajustar o código para isso não ocorrer. Para simplificar a implementação, vamos colocar um método auxiliar na classe `Lista` que devolve `true` ou `false` se a lista "está cheia" ou não.

```
public class Lista {  
  
    //outros atributos e métodos omitidos  
  
    public boolean estaCheia() {  
        return this.indice == this.tamanho();  
    }  
  
}
```

Agora vamos verificar na `TarefaImprimir` se realmente é preciso esperar:

```
public class TarefaImprimir implements Runnable {  
  
    //atributo e construtor omitido  
  
    @Override  
    public void run() {  
  
        synchronized (lista) {  
  
            //só espera se a lista não estiver cheia  
            if(!lista.estaCheia()) {  
                try {  
                    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.pegaElemento(i));  
            }  
        }  
    }  
}
```

```
    }  
}
```

Dessa maneira, não há o risco que o thread fique esperando sem ser notificado!