Como interromper a execução de uma thread

<http://www.arquivodecodigos.net/dicas/java-como-interromper-a-execucao-de-uma-thread-2766.html>

A classe Thread fornece um método chamado interrupt() que serve para interrompermos a execução de uma determinada thread. Os métodos sleep() e yield() transferem a thread sendo executada, do estado "executando" para o estado "pronta". O método interrupt() , por sua vez, faz com que uma thread vá para o estado "finalizada", ou seja, ela não voltará mais a ser executada.

O problema é que, nem sempre, o método interrupt() parece surtir o efeito desejado, ou seja, fazer com que a thread pare de executar imediatamente. Por esta razão, a forma mais fácil de "matar" uma thread é forçar a saída de seu método run(). Veja um exemplo:

// criamos uma classe que servirá como thread

class MinhaThread extends Thread{

 private String nome;

 public MinhaThread(String nome){

 this.nome = nome;

 }

 public void run(){

 for(int i = 1; i <= 20; i++){

 System.out.println(nome + ": " + i);

 if((i >= 3) && (nome.equals("Thread 1"))){

 try{

 System.out.println(nome + " foi eliminada");

 return; // interrompe a execução do método run()

 }

 catch(SecurityException e){

 System.out.println(e.getMessage());

 }

 }

 }

 }

}

public class Estudos{

 public static void main(String[] args){

 // vamos criar duas threads

 MinhaThread t1 = new MinhaThread("Thread 1");

 t1.start();

 MinhaThread t2 = new MinhaThread("Thread 2");

 t2.start();

 }

}

Ao usar a instrução return nós forçamos a saída do método run() e consequentemente, o fim da tarefa sendo realizada pela thread.