

03

Visualizando a primeira Thread

Transcrição

E vou provar para vocês a existência de threads através de um pequeno programa. Vamos criar um novo projeto Java dentro do Eclipse, chamado **threads**, e criar uma nova classe **Principal** (no pacote `br.com.alura.threads`) e dentro dela um método `main`. Nós sabemos que qualquer programa Java sempre começa com o método `main`. Quando rodamos, automaticamente a máquina virtual irá criar um novo thread ou uma nova linha de execução. E essa linha de execução pode ser executada em paralelo!

O código é bem simples:

```
public class Principal {  
  
    public static void main(String[] args) {  
  
        System.out.println("Thread MAIN");  
    }  
}
```

No método `main`, definimos apenas o que queremos executar, como a máquina virtual realmente faz esse trabalho, não sabemos. Como esse programinha acaba muito rápido, vamos pedir para a máquina virtual mandar aquele thread dormir! Como? Muito fácil, através do método estático `Thread.sleep()`:

```
public class Principal {  
  
    public static void main(String[] args) throws InterruptedException {  
  
        System.out.println("Thread MAIN");  
  
        Thread.sleep(50000);  
    }  
}
```

Ao executar, esse programa vai demorar no mínimo 50 segundos para ser encerrado. Tempo suficiente para visualizar o Thread que a JVM criou. Como? Executando a ferramenta `jconsole` na linha de comando.

Se você não tem instalada essa ferramenta não se preocupe, ela não é necessária para continuação do treinamento.

Através dela podemos nos conectar como nossa aplicação. A ferramenta `jconsole` mostra informações sobre a máquina virtual que está rodando o nosso programa.

Reparem que existe uma aba com o nome *Threads* e nessa aba podemos ver que já existem vários threads, inclusive um que se chama `main`. Esse é aquele que rodou o nosso método `main`! Então a máquina virtual já cria vários Threads por padrão (por exemplo para atender conexões remotas ou fazer a coleta de lixo). Isso já está embutido na JVM e foi grande diferencial quando Java foi lançado.

