

# POOII – Ex Threads III

**Aluno:** Mateus de Oliveira Lopes

1) Saída do programa:

Tique Tique Tique Tique Tique Taque  
Taque  
Taque  
Taque  
Taque

Process finished with exit code 0

Isso ocorre pois sem o uso do wait() e notify() não existe comunicação entre as threads para fazer com que uma fique suspensa enquanto a outra não for executada, fazendo com que a ordem de execução não seja pré-definida.

2)

MyThread.java

```
public class MyThread implements Runnable {
    Thread thrd;
    boolean suspender;
    boolean encerrar;
    MyThread(String name) {
        thrd = new Thread(this, name);
        suspender = false;
        encerrar = false;
        thrd.start();
    }

    // Este é o ponto de entrada para a Thread.
    public void run() {
        System.out.println(thrd.getName() + " iniciando.");
        try {
            for (int i = 1; i < 1000; i++) {
                System.out.print(i + " ");
                if ((i % 10) == 0) {
                    System.out.println();
                    Thread.sleep(250);
                }
                // Usa um bloco sincronizado para verificar suspender e encerrar.
                synchronized (this) {
                    while (suspender) {
                        wait();
                    }
                    if (encerrar)
                        break;
                }
            }
        } catch (InterruptedException exc) {
            System.out.println(thrd.getName() + " interrompido.");
        }
        System.out.println(thrd.getName() + " finalizando.");
    }
    // Finalizar uma thread.
    synchronized void myStop() {
```

```
suspender = false;
encerrar = true;
notify();
}
// Suspend a thread.
synchronized void mySuspend() {
    suspender = true;
    notify();
}
// Retomar a thread.
synchronized void myResume() {
    suspender = false;
    notify();
}
}
```