

## Aula 1

### Trabalho Autónomo

#### Exercício 1

Indique o que é escrito na consola decorrente da execução do seguinte programa (Dica: elabore o diagrama do tempo).

```
public class MinhaThread extends Thread {  
    public void run() {  
        System.out.println("Thread  
started::"+Thread.currentThread().getName());  
        try {  
            Thread.sleep(3000);  
        } catch (InterruptedException e) {  
            e.printStackTrace();  
        }  
        System.out.println("Thread  
ended::"+Thread.currentThread().getName());  
    }  
    public static void main(String[] args) throws InterruptedException{  
        Thread t1 = new MinhaThread();  
        Thread t2 = new MinhaThread();  
        Thread t3 = new MinhaThread();  
        Thread t4 = new MinhaThread();  
        Thread t5 = new MinhaThread();  
  
        t1.start();
```

EM PARCERIA COM



COM O APOIO



XXII GOVERNO



```

        t1.join(2000);
        t2.start();
        t1.join();
        t3.start();
        t2.join(1000);
        t4.start();
        t3.join(4000);
        t5.start();
        t1.join();
        t2.join();
        t3.join();
        t4.join();
        t5.join();

        System.out.println("All threads are dead, exiting main thread");
    }
}

```

## Exercício 2

Indique o que é escrito na consola decorrente da execução do seguinte programa (Dica: elabore o diagrama do tempo).

```

public class AMinhaThread extends Thread {
    public void run() {
        System.out.println("Thread
started::"+Thread.currentThread().getName());
        try {
            Thread.sleep(3000);
        } catch (InterruptedException e) {
            e.printStackTrace();
        }
        System.out.println("Thread
ended::"+Thread.currentThread().getName());
    }
    public static void main(String[] args) throws InterruptedException
{
    Thread t1 = new AMinhaThread();

```

```
Thread t2 = new AminhaThread();
Thread t3 = new AminhaThread();
Thread t4 = new AminhaThread();
Thread t5 = new AminhaThread();
Thread t6 = new AminhaThread();
Thread t7 = new AminhaThread();
Thread t8 = new AminhaThread();

t1.start();
t1.join(1000);
t2.start();
t1.join(3000);
t3.start();
t2.join(1000);
t4.start();
t3.join(4000);
t5.start();
t1.join(1000);
t6.start();
t4.join(4000);
t7.start();
t8.start();

t2.join();
t3.join();
t4.join();
t5.join();

    System.out.println("All threads are dead, exiting main
thread");
}
}
```

### Exercício 3

Pretende-se desenvolver uma aplicação que permite determinar qual o dígito que existe em maior número num conjunto de *Strings*. A aplicação deve contar o número de vezes que cada dígito aparece na *String*.

Exemplo de Strings: "4 3 7 3 5 8 7 3 4 5 7 4 6 3 1 2 1 2 4 5 3 5 8 7 4 3 2 6 7 8", "4 6 3 6 4 5 3 7 4 5 7 3 4 5 2 1 3 4 2 4", "5 7 8 9 4 2 5 7 4 6 4 3 3 6 5 3 6 7 5"

Por questões de eficiência, a contagem dos dígitos deve ser implementada através de *threads* (uma *thread* para cada *String*). Cada *thread* representa uma tarefa de contagem. Assim, se existirem 3 *String*, vão existir 3 *threads*. No final das pesquisas, o método *main()* deverá devolver o dígito que existe em maior número. Dica: crie métodos adicionais nas tarefas (*threads*).

### Exercício 4

Descreva o ciclo de vida de uma Thread.

### Exercício 5

Descreva o que fazem os métodos *interrupt()*, *join()* e *sleep()* e qual a sua utilidade.