

MODULO 9

Threads

Exemplo 1

Classe Tarefa

```
public class Tarefa implements Runnable{
    private int tempoADormir; // random
    private String nomeThread;
    private static Random generator = new Random();

    public Tarefa(String name){
        nomeThread = name;
        // tempo randomico entre 0 e 5 segundos
        tempoADormir = generator.nextInt( 5000 );
    }

    // codigo que será executado pelo thread
    public void run(){
        try {
            System.out.println( nomeThread+" vai dormir por "+tempoADormir+"
milisegundos.");
            Thread.sleep( tempoADormir );
        }
        // catch será chamado se o thread é interrompido enquanto dorme
        catch ( InterruptedException exception ){
            exception.printStackTrace();
        }
        System.out.println(nomeThread+ " acordou");
    }
}
```

Classe CriadorTarefas

```
public class CriadorTarefas{
    public static void main( String[] args ){
        System.out.println( "Criando as threads" );

        // Criamos threads com Runnable Tarefa
        Thread thread1 = new Thread( new Tarefa( "tarefa1" ) );
        Thread thread2 = new Thread( new Tarefa( "tarefa2" ) );
        Thread thread3 = new Thread( new Tarefa( "tarefa3" ) );

        System.out.println( "Threads criadas, start()." );

        // Rodamos os threads
        thread1.start(); // invoca metodo run da tarefa1
        thread2.start(); // invoca metodo run da tarefa2
        thread3.start(); // invoca metodo run da tarefa3

        System.out.println( "Tarefas rodando, fim do método main.\n" );
    }
}
```

Saída ao rodar a classe CriadorTarefas:

```

Criando as threads
Threads criadas, start().
tarefa1 vai dormir por 2120 milisegundos.
Tarefas rodando, fim do método main.

tarefa2 vai dormir por 1857 milisegundos.
tarefa3 vai dormir por 3878 milisegundos.
tarefa2 acordou
tarefa1 acordou
tarefa3 acordou

```

Exemplo 2**Classe ImprimidorNumeros**

```

public class ImprimidorNumeros implements Runnable {
    private int id;
    public ImprimidorNumeros(int id) {
        this.id=id;
    }

    public void run() {
        for (int i = 0; i < 1000; i++) {
            System.out.println("Imprimidor " + this.id + " valor: " + i);
        }
    }
}

```

Classe TesteImprimidorNumeros

```

public class TesteImprimidorNumeros {

    public static void main(String[] args) {
        ImprimidorNumeros p1 = new ImprimidorNumeros(1);
        Thread t1 = new Thread(p1);
        t1.start();

        ImprimidorNumeros p2 = new ImprimidorNumeros(2);
        Thread t2 = new Thread(p2);
        t2.start();
    }
}

```

Saída ao rodar a classe TesteImprimidorNumeros:

```

Imprimidor 1 valor: 0
Imprimidor 1 valor: 1
...
Imprimidor 1 valor: 13
Imprimidor 1 valor: 14
Imprimidor 2 valor: 0
Imprimidor 2 valor: 1

```

```
Imprimidor 1 valor: 15
Imprimidor 1 valor: 16
...
Imprimidor 1 valor: 30
Imprimidor 1 valor: 31
Imprimidor 2 valor: 2
Imprimidor 2 valor: 3
...
Imprimidor 2 valor: 99
Imprimidor 2 valor: 100
Imprimidor 2 valor: 101
Imprimidor 1 valor: 32
Imprimidor 1 valor: 33
Imprimidor 1 valor: 34
...
```

Lista de Exercícios 9

1. Crie duas threads onde:
 - a) uma imprime notícias a cada 1 segundos (noticia1, noticia2, ...),
 - b) a outra imprime a hora a cada 2 segundos.Para criar as threads deverá usar duas classes (NotificadorNoticia, NotificadorHora) que implementem Runnable.
Crie a classe TesteNotificadores contendo um método main onde faça as threads rodarem simultaneamente. A execução finalizará quando a hora tiver sido informada 5 vezes e a noticia 10 vezes.
2. Faça um programa que crie 3 threads. A primeira escreve na tela “A”, a segunda “B” e a terceira “C”. Faça que seja sempre escrito na tela “ABC”.