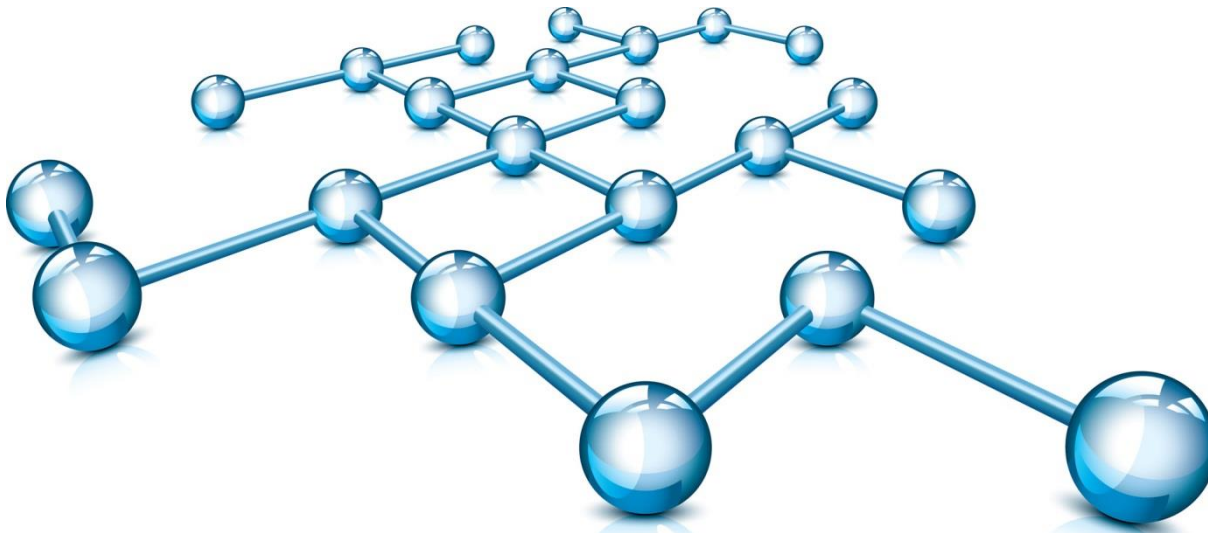


Breve tutorial de Hilos en Java

Profesor:

Dr. J. Octavio Gutiérrez García

octavio.gutierrez@itam.mx



Maneras de definir un Hilo

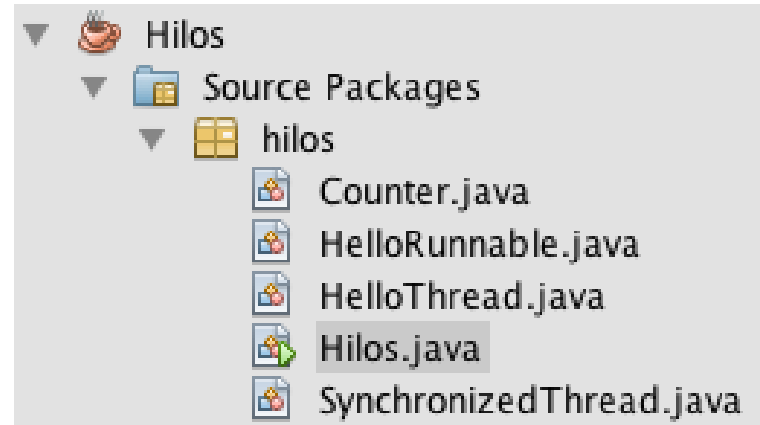
implements Runnable



extends Thread



Tejiendo con mis primeros hilos



Implementando Runnable



```
public class HelloRunnable implements Runnable {
```

```
    public void run() {  
        System.out.println("Hola");  
    }
```

```
}
```

Se ejecuta con método **start()**

Extendiendo Thread



```
public class HelloThread extends Thread {  
  
    public void run() {  
        System.out.println("Hola");  
    }  
  
}
```

También se ejecuta con método **start()**

Creando los hilos



```
HelloThread hilo1 = new HelloThread();
```

```
Thread hilo2 = new Thread(new HelloRunnable());
```

```
hilo1.start();
```

```
hilo2.start();
```

Un poco de manejo de hilos

- Detener hilos `Thread.sleep(milisegundos)`
- Esperar hilos `join()` o `join(milisegundos)`
- Nombre `Thread.currentThread().getName()`
- Sincronizar `synchronized`

Creando hilos con “regiones críticas”



```
Counter aCounter;
```

```
aCounter = new Counter(0);
```

```
SynchronizedThread hilo3 = new SynchronizedThread(aCounter);
```

```
SynchronizedThread hilo4 = new SynchronizedThread(aCounter);
```

```
hilo3.start();
```

```
hilo4.start();
```


Código ejemplo de concurrencia sincronizada

```
public class SynchronizedThread extends Thread {  
    private Counter aCounter;  
  
    public SynchronizedThread (Counter aCounter){  
        this.aCounter=aCounter;  
    }  
  
    public void run(){  
        aCounter.aMethod();  
    }  
}
```

synchronized

```
public class Counter {  
    private int n = 0;  
  
    public Counter(int n){  
        this.n=n;  
    }  
  
    public void aMethod(){  
        for (int i = 0; i<10; i++){  
            n++;  
            System.out.println(n + " ");  
        }  
    }  
}
```