

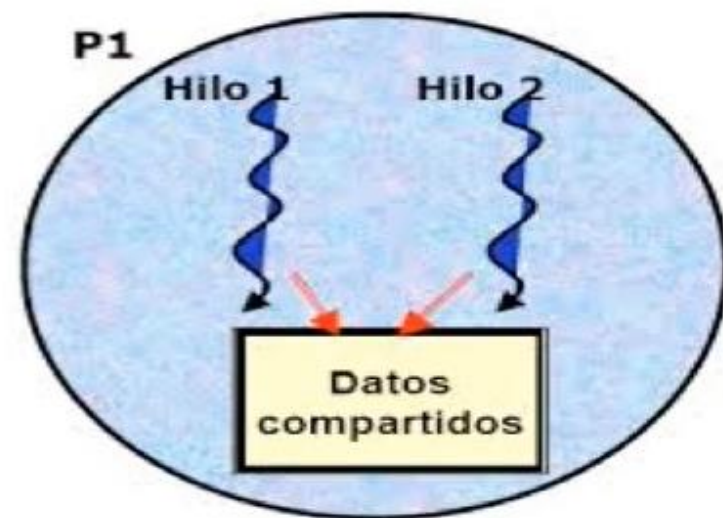
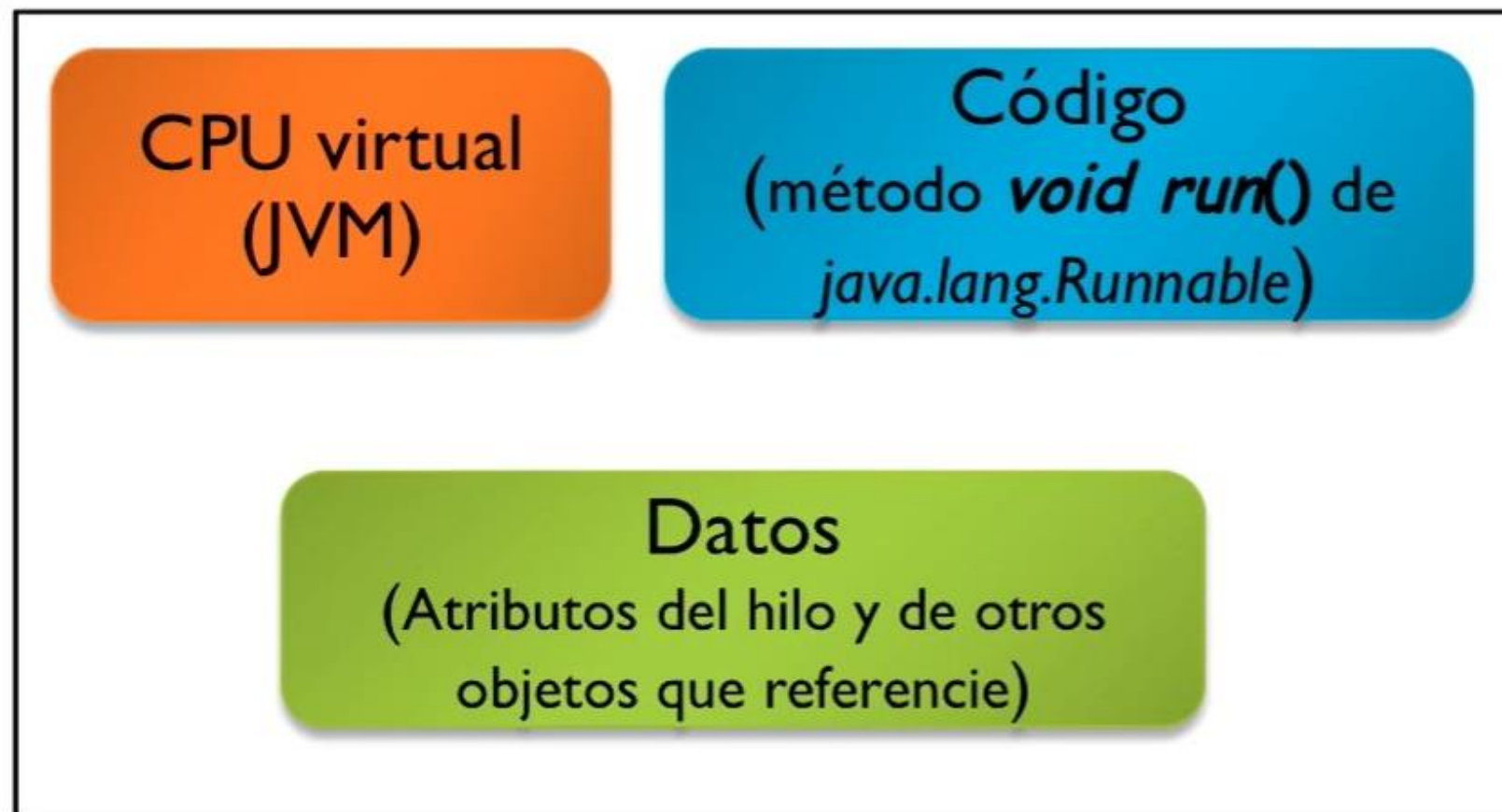
# CREACIÓN DE HILOS EN JAVA

## Programación Concurrente

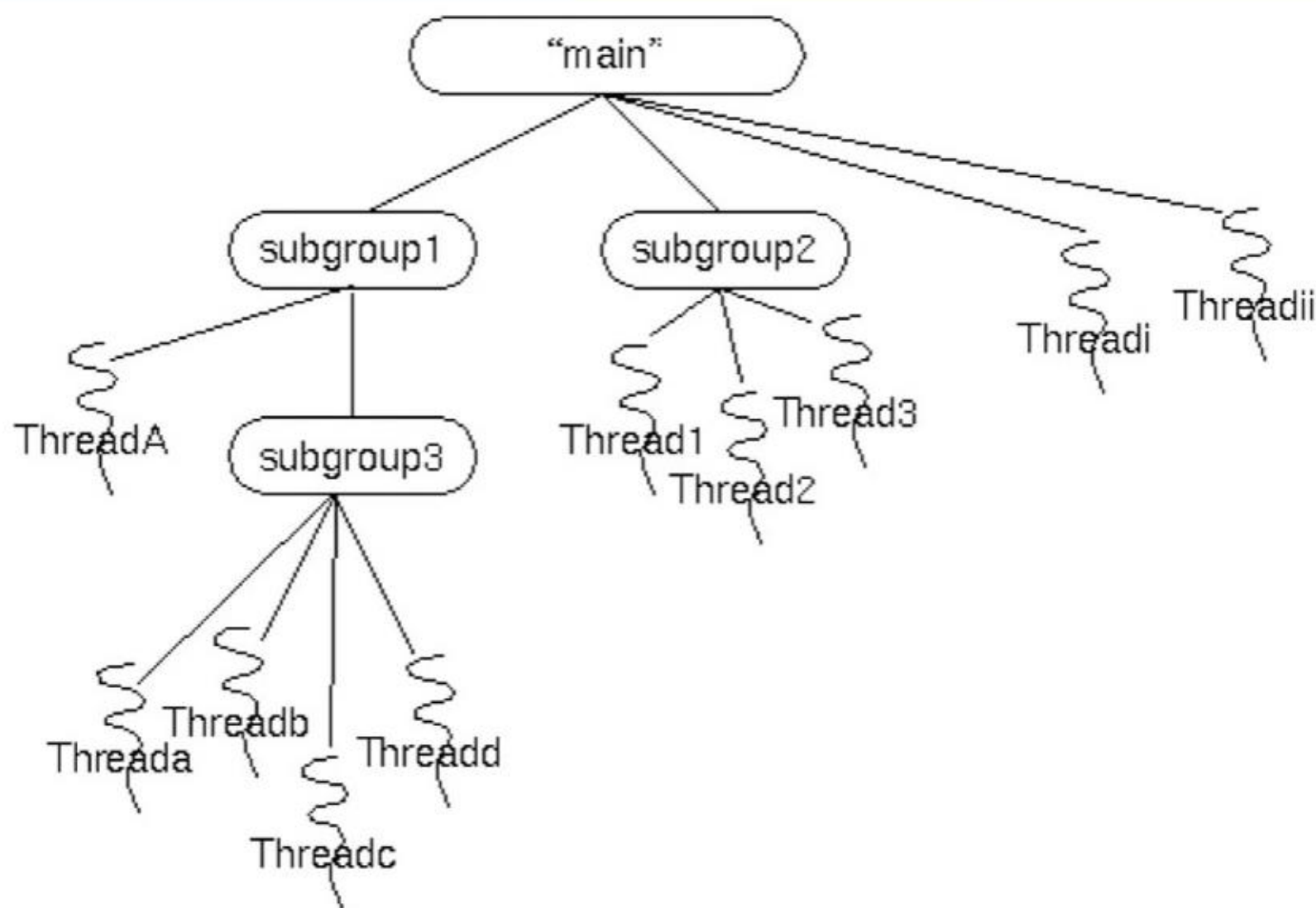
- ❖ Qué es un hilo en Java
- ❖ Creación de hilos
- ❖ Ejecución de hilos
- ❖ Identificación de hilos

# ¿Qué es un hilo (en Java)?

- ▶ Thread (hilo) = contexto de ejecución, formado por:



# ¿Qué es un hilo (en Java)?



# ¿Cómo crear hilos en Java?

## ▶ Alternativas:

- ▶ Implementando la interfaz **Runnable**

```
public class HolaRunnable implements
Runnable {
    public void run()
    { System.out.println("Hola mundo!");    }

    public static void main(String args[])
    { (new Thread(new HolaRunnable())).start();
    }
}
```

- ▶ Extendiendo la clase **Thread**

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

    public static void main(String args[])
    { (new HolaThread()).start();
    }
}
```



# ¿Cómo crear hilos en Java? → **Opción 1**

	Clase con nombre
Implementando Runnable	<pre>public class H implements Runnable {     public void run() {         System.out.println("ejecuta hilo");     } } Thread t= new Thread(new H()); t.start();</pre>
Extendiendo Thread	<pre>public class H extends Thread {     public void run() {         System.out.println("ejecuta hilo");     } } H t= new H();    t.start();</pre>

# ¿Cómo crear hilos en Java? → **Opción 1**

## Clase con nombre

### Implementando Runnable

```
public class H implements Runnable {  
    public void run() {  
        System.out.println("ejecuta hilo");  
    }  
}  
Thread t= new Thread(new H());  
t.start();
```

### Extendiendo Thread

```
public class H extends Thread {  
    public void run() {  
        System.out.println("ejecuta hilo");  
    }  
}  
H t= new H();    t.start();
```

# ¿Cómo crear hilos en Java? → **Opción 2**

	Clase anónima
Implementando Runnable	<pre><b>new Thread(new Runnable() {</b>     <b>public void</b> run() {         System.out.println("ejecuta hilo");     } }).start();</pre>
Extendiendo Thread	<pre><b>new Thread() {</b>     <b>public void</b> run() {         System.out.println("ejecuta hilo");     } }.start();</pre>



# ¿Cómo ejecutar hilos en Java?

- ▶ La ejecución del hilo se arranca con **start()**
  - ▶ Lanza la ejecución del hilo.
  - ▶ Ese hilo ejecuta su método *run()*
  - ▶ **Error típico:** invocar *run()* en lugar de *start()*

```
public class T extends Thread {  
    protected int n;  
    public T(int n) {this.n = n;}  
  
    public static void main(String[] argv) {  
        for (int i=0; i<3; ++i)  
            new T(i).start();  
    }  
    public void run() {  
        for (int i=0; i<5; ++i) {  
            System.out.println("Hilo "+n+" iteración "+i);  
            retardo((n+1)*1000);  
            System.out.println ("Fin iteración hilo "+n);  
        }  
    }  
}
```



# ¿Cómo identificar hilos en Java?

---

- ▶ Al crear un hilo le podemos asociar un nombre:
  - ▶ El constructor de *Thread* admite un nombre para el hilo

```
new T(i).start();
```

- ▶ En cualquier momento podemos asociarle un nombre:
  - ▶ Utilizando método ***setName()***

```
t.setName("thread" + i);
```

- ▶ Identificador accesible con ***getName()*** sobre cualquier objeto *Thread*

```
t.getName();
```

# ¿Cómo identificar hilos en Java?

```
public class ExThread {  
    public static void main (String[] args) {  
        System.out.println(Thread.currentThread().getName());  
        for (int i=0; i<10; i++) {  
            new Thread("Hilo "+i) {  
                public void run() {  
                    System.out.println ("ejecutado por "+  
                        Thread.currentThread().getName());  
                }  
            }.start();  
        }  
    }  
}
```

# A modo de ejemplo....

```
public class ThreadName extends Thread {  
    public void run() {  
        for (int i = 0; i < 3; i++)  
            printMsg();  
    }  
    public void printMsg() {  
        System.out.println ("nombre=" +  
            Thread.currentThread().getName());  
    }  
    public static void main(String[] args) {  
        for ( int i = 0; i < 10; i++ ) {  
            ThreadName tt = new ThreadName();  
            tt.setName("hilo" + i);  
            if (i<5) tt.start();  
        }  
    }  
}
```



# A modo de ejemplo....

```
public class ThreadName extends Thread {  
    public void run() {  
        for (int i = 0; i < 3; i++)  
            printMsg();  
    }  
    public void printMsg() {  
        System.out.println ("nombre=" +  
            Thread.currentThread().getName());  
    }  
    public static void main(String[] args) {  
        for ( int i = 0; i < 10; i++ ) {  
            ThreadName tt = new ThreadName();  
            tt.setName("hilo" + i);  
            if (i<5) tt.start();  
        }  
    }  
}
```

¿Cuántos hilos  
se crean?

¿Cuántos hilos  
se ejecutan?

¿Cómo se  
identifica cada  
hilo?

# A modo de ejemplo....

```
public class ThreadName extends Thread {  
    public void run() {  
        for (int i = 0; i < 3; i++)  
            printMsg();  
    }  
    public void printMsg() {  
        System.out.println ("nombre=" +  
            Thread.currentThread().getName());  
    }  
    public static void main(String[] args) {  
        for ( int i = 0; i < 10; i++ ) {  
            ThreadName tt = new ThreadName();  
            tt.setName("hilo" + i);  
            if (i<5) tt.start();  
        }  
    }  
}
```

```
nombre=hilo0  
nombre=hilo0  
nombre=hilo0  
nombre=hilo1  
nombre=hilo1  
nombre=hilo1  
nombre=hilo2  
nombre=hilo2  
nombre=hilo2  
nombre=hilo2  
nombre=hilo3  
nombre=hilo3  
nombre=hilo3  
nombre=hilo4  
nombre=hilo4  
nombre=hilo4
```

# Resumen

---

▶ Creación de hilos en Java

Thread

Runnable

▶ Ejecución de hilos en Java

start()

run()

▶ Identificación de hilos en Java

setName()

getName()