Práctica 2

Hilos

Hilos

- Hay dos posibilidades para crear hilos en Java:
  - Crear un objeto cuya clase **herede de la clase Thread**
  - Crear un objeto cuya clase **implemente el interfaz Runnable**
- En ambos casos, la clase debe implementar el método

  public void run()

  Que contendrá el código que ejecutará el hilo

```java
class HiloThread extends Thread {
    @Override
    public void run() {
        // entry point for thread
    }
}
```

```java
class HiloRunnable implements Runnable {
    @Override
    public void run() {
        // entry point for thread
        }
    }
```

Hilos

- Para lanzar un hilo, cuya clase hereda de Thread es suficiente con crear un objeto de dicha clase, e invocar a su método *start()*. Cuidado, no invocar a *run()*, que provocaría una ejecución secuencial, no concurrente.

```
HiloThread h1 = new HiloThread();

h1.start();
```

- Para lanzar un hilo, cuya clase implemente *Runnable* necesitamos crear un objeto *Thread* a cuyo constructor pasemos como parámetro el objeto de la clase que implementa Runnable.

```
HiloRunnable r = new HiloRunnable();

Thread h1 = new Thread( task: r);

h1.start();
```

Hilos

- Hay dos operaciones que podemos realizar para controlar la ejecución de un hilo:
  - *join*(), Hace que el invocador se detenga, hasta que el hilo finalice.
  - *interrupt*(). Provoca que el hilo reciba *InterruptedException*()

```
HiloThread h1 = new HiloThread();

h1.start();

if( v==0) h1.interrupt();

h1.join();
```

```
HiloRunnable r = new HiloRunnable();

Thread h1 = new Thread( task: r);

h1.start();

if( v==0) h1.interrupt();

h1.join();
```

Hilos

- Algunas operaciones sobre hilos:
  - `setPriority`
  - `getPriority`
  - `setName`
  - `getName`
  - `getState`
  - `getId`
- Estos métodos son invocables sobre cualquier objeto `Thread`, por tanto lo son desde dentro de la propia clase que hereda de `Thread` (mediante `this`), pero no en las clases que implementan `Runnable`.
- Para acceder al hilo que está ejecutando el código se usa:
  - `Thread.currentThread()`

Hilos

```
class HiloThread extends Thread {

    @Override
    public void run() { // entry point for thread
        for (int i = 1; i <= 10; i++) {
            setPriority( newPriority:i);
            System.out.println("Soy el hilo tipo A " + getName() +
                    " con prioridad " + getPriority() + " en estado "
                    + getState());
        }
    }
}

class HiloRunnable implements Runnable {

    @Override
    public void run() { // entry point for thread
        Thread running = Thread.currentThread();
        for (int i = 1; i < 10; i++) {
            running.setPriority( newPriority:i);
            System.out.println("Soy el hilo tipo B " + running.getName() +
                    " con prioridad " + running.getPriority() +
                    " en estado " + running.getState());
        }
    }
}
```

Hilos

```
public static void main(String[] args) {

    Thread.currentThread().setName( name:"HiloMain");
    Thread.currentThread().setPriority( newPriority:9);
    System.out.println("Soy el hilo " + Thread.currentThread().getName() +
                       " con prioridad " + Thread.currentThread().getPriority() +
                   " en estado "         + Thread.currentThread().getState());


    HiloThread h1 = new HiloThread();
    HiloRunnable r = new HiloRunnable();
    Thread h2 = new Thread( task:r);

    h1.setName( name:"Hilo1");
    h1.setPriority( newPriority:4);

    System.out.println("Soy el hilo " + h1.getName() +
                       " con prioridad " + h1.getPriority() + " en estado "
                       + h1.getState());

    h2.setName( name:"Hilo2");
    h2.setPriority( newPriority:8);

    System.out.println("Soy el hilo " + h2.getName() +
                       " con prioridad " + h2.getPriority() + " en estado "
                       + h2.getState());
```