**ACTIVIDAD 4**

**Enunciado**

Realiza el siguiente ejercicio utilizando una única clase Java (Posicion) que sustituya a las clases Primero y Segundo.

El nombre de la clase principal será Actividad4 y el número de veces que se sacará por la consola las cadenas de caracteres “Primero” y “Segundo” será 15.

// Hilos2.java

**public** **class** Hilos2 {

**public** **static** **void** main(String arg[]) {

Thread p = **new** Thread(**new** Primero());

Thread s = **new** Thread(**new** Segundo());

p.start();

s.start();

System.***out***.println( "Fin programa ");

}

}

// Primero.java

**public** **class** Primero **implements** Runnable {

**public** **void** run() {

**for** (**int** i=1;i<100;i++)

System.***out***.println( "Primero " + i );

}

}

// Segundo.java

**public** **class** Segundo **implements** Runnable {

**public** **void** run() {

**for** (**int** i=1;i<100;i++)

System.***out***.println( "Segundo " + i );

}

}

**Solución**

package ejercicios;

public class Actividad4 implements Runnable {

// Guardamos el nombre del hilo en una variable

private String hilo;

public Actividad4(String hilo) {

this.hilo = hilo;

}

public static void main(String[] args) {

//Instanciamos ambos hilos

Thread hilo1 = new Thread(new Actividad4("Primero"));

Thread hilo2 = new Thread(new Actividad4("Segundo"));

// Ejecutamos los dos hilos

hilo1.start();

hilo2.start();

try {

// Bloqueamos el hilo principal hasta que el hilo1 termine su ejecucion

hilo1.join();

// Ahora lo mismo con el hilo2

hilo2.join();

} catch (InterruptedException e) {

e.printStackTrace();

}

System.out.println("Fin programa");

}

@Override

public void run() {

//Bucle para 15 repeticiones

for (int i = 1; i <= 15; i++) {

System.out.println(hilo+" "+i);

}

}

}

**Resolución de problemas**

**ACTIVIDAD 5**

**Enunciado**

Utiliza el método sleep de la clase Thread en las clases Primero y Segundo del siguiente programa, de acuerdo a los siguientes requerimientos:

1. En la clase Primero, modifica el bucle para que durante 15 repeticiones detenga la ejecución del hilo durante 100 milisegundos.
2. En la clase Segundo, modifica el bucle para que durante 15 repeticiones detenga la ejecución del hilo durante 0,2 segundos.

El nombre de la clase principal será Actividad5.

//Actividadx.java

**public** **class** Actividadx {

**public** **static** **void** main(String arg[]) {

Primero p = **new** Primero();

Segundo s = **new** Segundo();

p.start();

s.start();

System.***out***.println("Fin programa");

}

}

//Primero. java

**class** Primero **extends** Thread {

**public** **void** run() {

**for** (**int** i=1;i<10;i++)

{

System.***out***.println( "Primero " + i );

}

}

}

//Segundo. java

**class** Segundo **extends** Thread {

**public** **void** run() {

**for** (**int** i=1;i<=10;i++)

{

System.***out***.println( "Segundo " + i );

}

}

}

**Solución**

package ejercicios;

public class Actividad5 {

public static void main(String[] args) {

//Instanciamos ambos hilos

Primero p = new Primero();

Segundo s = new Segundo();

// Lanzamos ambos hilos

p.start();

s.start();

System.out.println("Fin programa");

}

}

class Primero extends Thread {

public void run() {

//Bucle de 15 repeticiones

for (int i = 1; i <= 15; i++) {

System.out.println("Primero " + i);

try {

// Detiene el hilo durante 100 milisegundos

Thread.sleep(100);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

class Segundo extends Thread {

public void run() {

// Bucle de 15 repeticiones

for (int i = 1; i <= 15; i++) {

System.out.println("Segundo " + i);

try {

// Detiene el hilo durante 200 milisegundos

Thread.sleep(200);

} catch (InterruptedException e) {

e.printStackTrace();

}

}

}

}

**Resolución de problemas**

**CRITERIOS DE CORRECCIÓN**

1. Nivel de ajuste a la nomenclatura de objetos definida en el enunciado de la actividad y nivel de ajuste al almacenamiento de la información requerida en las subcarpetas definidas para la actividad y unidad didáctica correspondientes (15%).
2. Hacer lo que se indica en el enunciado (55%).
3. Claridad del código Java (15%).
4. Documentación del código Java (15%).
5. Explicación detallada de cómo se han resuelto problemas que hayan podido surgir en la realización de la actividad (hasta un 15% adicional sobre la nota obtenida). Dicha explicación se incluirá al principio de los archivos de código fuente Java mediante líneas de comentarios. La primera línea de comentarios sólo incluirá lo siguiente: // Resolución de problemas.
6. Variables no usadas (-15%).
7. Librerías no usadas (-15%).