**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 );

}

}

**Solucion:**

**//Posicion.java**

public class Posicion implements Runnable{

@Override

public void run() {

// TODO Auto-generated method stub

for (int i=1; i<=15; i++) { //Bucle de 15 repeticiones que saca el nombre del hilo + el numero de repeticion

System.out.println(Thread.currentThread().getName() + " " + i);

}

}

}

**//Actividad4.java**

public class Actividad4 {

public static void main(String[] args) {

//Creacion de los 2 hilos

Thread p = new Thread(new Posicion());

Thread s = new Thread(new Posicion());

//Les asignamos un nombre

p.setName("Primero");

s.setName("Segundo");

//Los ejecutamos

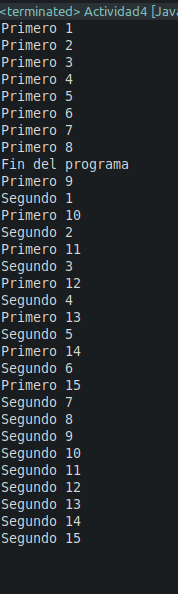
p.start();

s.start();

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

}

}



**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 );

}

}

}

**Solucion:**

**//Actividad5.java**

public class Actividad5 {

public static void main(String[] args) {

//Creamos los hilos

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

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

//Los ejecutamos

p.start();

s.start();

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

}

}

**//Primero.java**

public class Primero extends Thread {

@Override

public void run() {

for (int i=1; i<15; i++) { //Bucle de 15 repeticiones que saca por pantalla "Primero" + el numero de repeticion

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

try {

sleep(100); //Detenemos la ejecucion por 100 milisegundos (0.1s)

} catch (InterruptedException e) {}

}

}

}

**//Segundo.java**

public class Segundo extends Thread{

@Override

public void run() {

for (int i=1;i<=10;i++) { //Bucle de 15 repeticiones que saca por pantalla "Segundo" + el numero de repeticion

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

try {

sleep(200); //Detenemos la ejecucion por 200 milisegundos (0.2s)

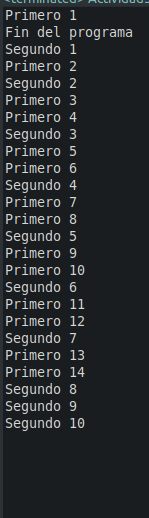
} catch (InterruptedException e) {

}

}

}

}



**REQUISITOS**

1. Entregar las soluciones de las actividades en un único archivo comprimido donde tendremos:
   1. Un archivo de extensión .odt o .docx que contendrá
      1. Una copia del código del programa
      2. Una captura de pantalla del resultado de la ejecución del programa
   2. Los archivos \*.java empleados
2. El único archivo con la soluciones contendrá el nombre original de la actividad, seguido de un guión bajo y del primer apellido del alumno.

**CRITERIOS DE CORRECCIÓN**

1. Cumplimiento de los requisitos (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%).