**ACTIVIDAD 1**

**Enunciado**

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

El nombre de la clase principal será Actividad1 y el número de veces que se sacará por la consola las cadenas de caracteres “Primero” y “Segundo” será 20.

**//Hilos1.java**

**public** **class** Hilos1 {

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

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

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

p.start();

s.start();

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

}

}

**//Primero.java**

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

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

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

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

}

}

**//Segundo.java**

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

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

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

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

}

}

**Solución**

package ejercicios;

public class Actividad1 extends Thread {

public static void main(String[] args) {

Actividad1 hilo = new Actividad1();

hilo.start();

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

}

public void run() {

// Blucle para mostrar 20 veces el hilo 1

for (int i = 1; i <= 20; i++) {

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

}

// Blucle para mostrar 20 veces el hilo 2

for (int i = 1; i <= 20; i++) {

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

}

}

}

**Resolución de problemas**

**ACTIVIDAD 2**

**Enunciado**

Realiza un programa que cree n hilos, donde n se pasará como argumento.

Por cada hilo creado, se sacará por consola 20 veces un mensaje del tipo: “Hilo n”, donde n será el número de hilo que se está ejecutando.

Después de haber creado los n hilos, sacar por consola un mensaje que diga “Final Programa”.

El nombre de la clase principal será Actividad2.

**Solución**

package ejercicios;

public class Actividad2 {

public static void main(int n) {

// Recibimos n como la cantidad de hilos que vamos a crear

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

Thread hilo = new HiloPersonalizado(i);

hilo.start();

}

System.out.println("Final Programa");

}

}

class HiloPersonalizado extends Thread {

private final int numeroHilo;

public HiloPersonalizado(int numeroHilo) {

this.numeroHilo = numeroHilo;

}

public void run() {

// Sacamos 20 veces por pantalla el mensaje por cada Hilo

for (int i = 1; i <= 20; i++) {

System.out.println("Hilo " + numeroHilo);

}

}

}

**Resolución de problemas**

**ACTIVIDAD 3**

**Enunciado**

Realiza un programa que use los métodos setName() y setPriority() de la clase Thread para cambiar el nombre y prioridad iniciales del hilo. Además, será necesario utilizar los métodos getName() y getPriority() de la clase Thread para visualizar en consola, tanto los valores iniciales del nombre y prioridad del hilo como los valores posteriormente modificados.

El nombre de la clase principal será Actividad3.

El resultado esperado será el siguiente:

El nombre del hilo es Thread-0 y tiene la prioridad 5

Ahora el nombre del hilo es SUPER-HILO-DM2 y tiene la prioridad 6

Final programa

**Solución**

package ejercicios;

public class Actividad3 {

public static void main(String[] args) {

Thread hilo = Thread.currentThread();

// Mostramos el hilo que acabamos de crear con sus valores iniciales

System.out.println("El nombre del hilo es " + hilo.getName() + " y tiene la prioridad " + hilo.getPriority());

// Cambiar nombre

hilo.setName("SUPER-HILO-DM2");

// Le sumamos uno a su prioridad

hilo.setPriority(Thread.NORM\_PRIORITY + 1);

// Mostrar el hilo con sus nuevos valores

System.out.println("Ahora el nombre del hilo es " + hilo.getName() + " y tiene la prioridad " + hilo.getPriority());

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

}

}

**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. Correcto funcionamiento de los programas desarrollados en Java (35%).
3. Ajuste a los resultados esperados (35%).
4. Claridad 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%).