



PSP

Práctica 3

Mario Sánchez Barroso

2º CFGS DAM

IES Profesor Juan Antonio Carrillo Salcedo

Índice

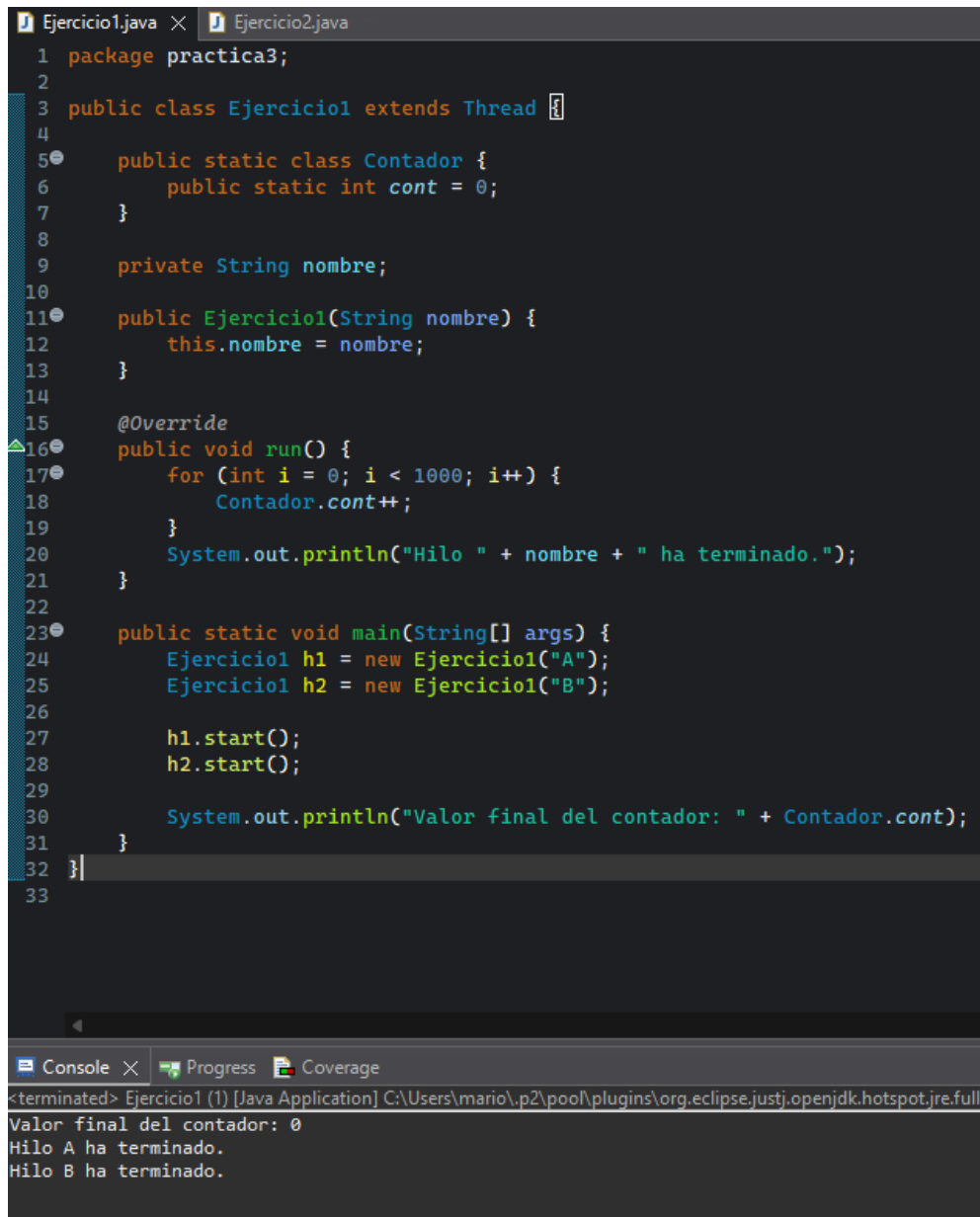
Práctica 3	3
Ejercicio 1.....	3
Ejercicio 2.....	4

Práctica 3

Ejercicio 1

Crea una clase con un atributo público y estático de tipo entero. Dicho atributo lo incrementará una clase principal, así creará dos 2 hilos, para dos procesos, que incrementarán también este atributo estático.

Explica el funcionamiento de una ejecución del main. Hubo algo raro.



```
1 package practica3;
2
3 public class Ejercicio1 extends Thread {
4
5     public static class Contador {
6         public static int cont = 0;
7     }
8
9     private String nombre;
10
11     public Ejercicio1(String nombre) {
12         this.nombre = nombre;
13     }
14
15     @Override
16     public void run() {
17         for (int i = 0; i < 1000; i++) {
18             Contador.cont++;
19         }
20         System.out.println("Hilo " + nombre + " ha terminado.");
21     }
22
23     public static void main(String[] args) {
24         Ejercicio1 h1 = new Ejercicio1("A");
25         Ejercicio1 h2 = new Ejercicio1("B");
26
27         h1.start();
28         h2.start();
29
30         System.out.println("Valor final del contador: " + Contador.cont);
31     }
32 }
33
```

Console Output:

```
<terminated> Ejercicio1 (1) [Java Application] C:\Users\mario\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full...
Valor final del contador: 0
Hilo A ha terminado.
Hilo B ha terminado.
```

En este código el error principal viene dado a que los hilos no esperan a que acaben la ejecución, por eso nuestro valor final del contador pone que es 0. La ejecución del main es la siguiente:

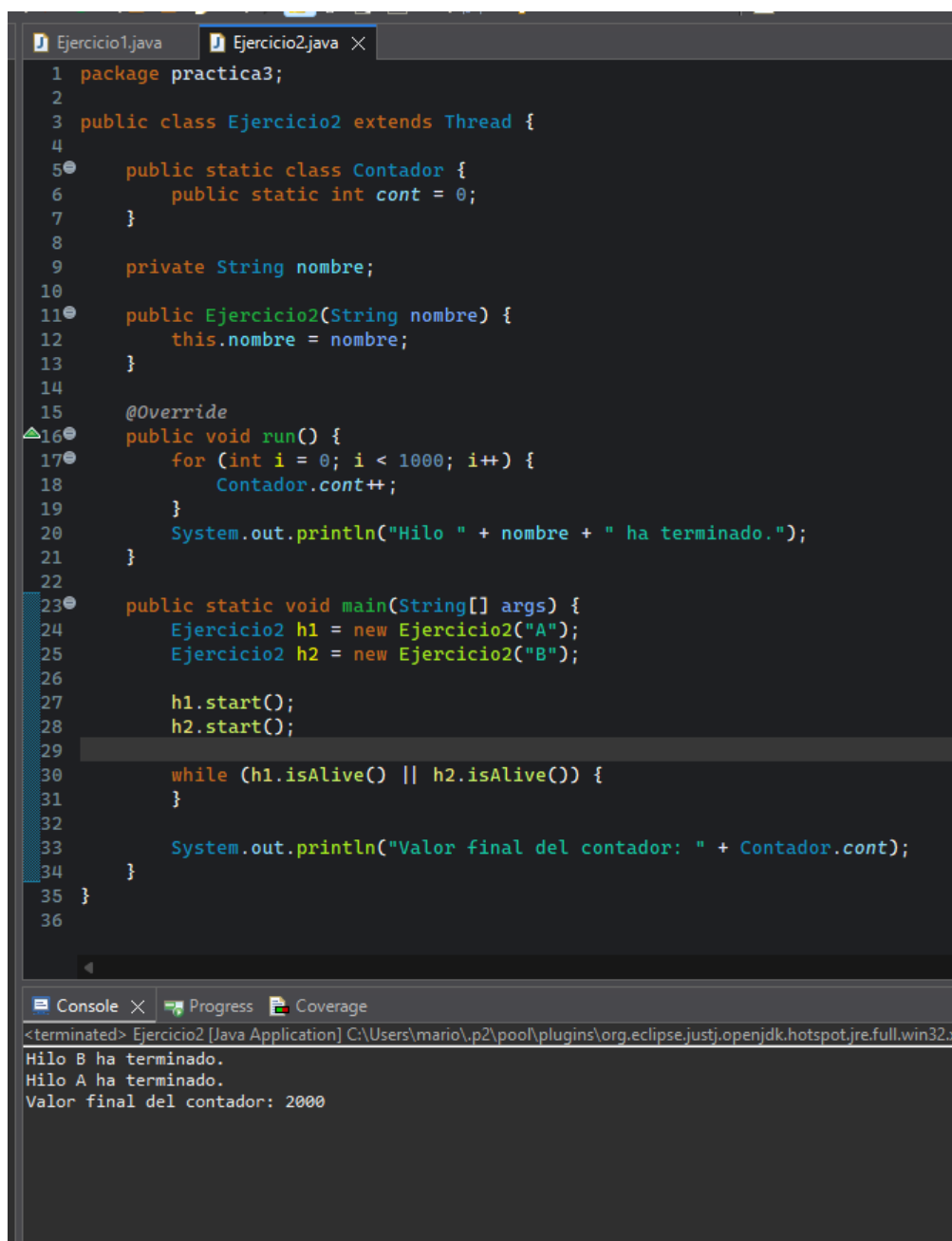
Crea 2 hilos, en este caso h1 y h2, los inicia con el método `start()` y justo después imprime el valor de contador.

Lo que hace el hilo se establece en el método `run` que hereda de `Thread`

Ejercicio 2

Haciendo uso del método `isAlive()` de la clase `Thread`, intenta solucionar este problema de concurrencia en el acceso al atributo

Explica ahora el nuevo funcionamiento de una ejecución del `main`. Hubo algo raro ahora.



```
1 package practica3;
2
3 public class Ejercicio2 extends Thread {
4
5     public static class Contador {
6         public static int cont = 0;
7     }
8
9     private String nombre;
10
11     public Ejercicio2(String nombre) {
12         this.nombre = nombre;
13     }
14
15     @Override
16     public void run() {
17         for (int i = 0; i < 1000; i++) {
18             Contador.cont++;
19         }
20         System.out.println("Hilo " + nombre + " ha terminado.");
21     }
22
23     public static void main(String[] args) {
24         Ejercicio2 h1 = new Ejercicio2("A");
25         Ejercicio2 h2 = new Ejercicio2("B");
26
27         h1.start();
28         h2.start();
29
30         while (h1.isAlive() || h2.isAlive()) {
31             ;
32         }
33
34         System.out.println("Valor final del contador: " + Contador.cont);
35     }
36 }
```

Console

```
<terminated> Ejercicio2 [Java Application] C:\Users\mario\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
Hilo B ha terminado.
Hilo A ha terminado.
Valor final del contador: 2000
```

En este caso lo que he hecho es poner un bucle while con la condición de isAlive() para cada uno de los hilos, en cuanto acabe ya si pone el valor del contadors