

Práctica MultiHilo I

“Una máquina puede hacer el trabajo de cincuenta hombres ordinarios. Ninguna máquina puede hacer el trabajo de un hombre extraordinario.”

Ejercicios

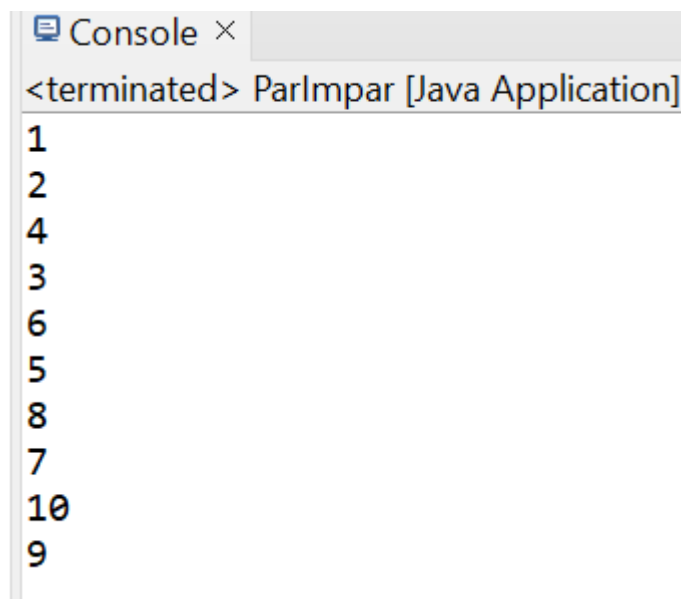
En al menos 2 de los ejercicios debes implementar la lógica heredando de la clase `Thread` y en al menos otros 2 debes hacerlo implementando la interfaz `Runnable`.

Ejercicio 1 - PAR / IMPAR

Escribe una clase con un método que en función de un dato saque por consola los impares entre 1 y 10 o los pares entre 1 y 10.

Genera 2 hilos de la clase, uno para que genere los pares y otro para generar los impares.

Prueba y documenta varias respuestas



```
<terminated> ParImpar [Java Application]
1
2
4
3
6
5
8
7
10
9
```

Ejercicio 2 - IRON MAN

Escribe una clase que tenga 3 métodos para simular un IronMan: bici (180 km), marathon (42 km) y nadar (4.000 m). Luego escribe otro método va enlazando las pruebas.

En la clase tendremos 4 variables: velocidad a la que corres, velocidad en la bici y velocidad nadando, además del nombre del participante.

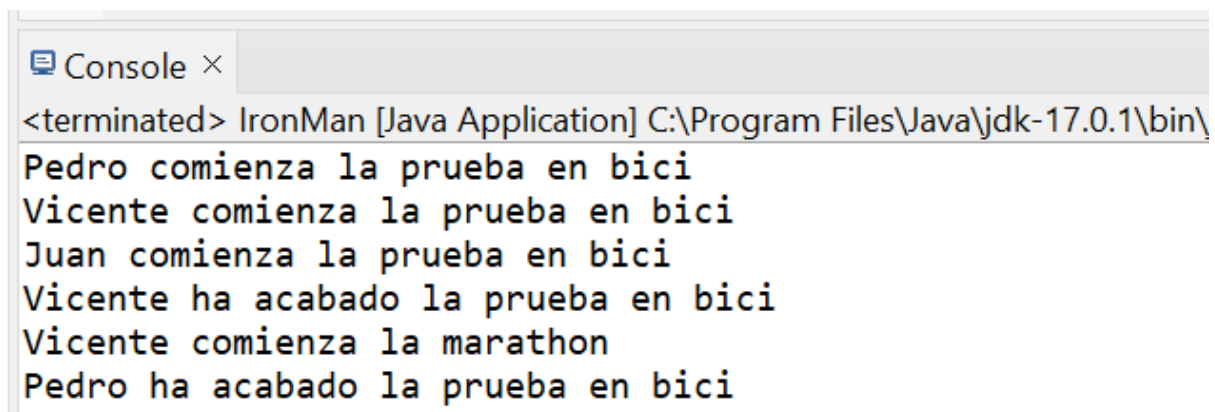
En cada prueba sacaremos por consola cuando un participante empieza la prueba y cuando la acaba.

Vamos a suponer que 1 hora son 1000 milisegundos en nuestro programa.

Hay una carrera entre:

- Pedro: corre a 6 km / hora, en bici va a 20 km / hora y nada a 1.000 m / hora
- Juan: corre a 7 km / hora, en bici va a 18 km / hora y nada a 2.000 m / hora
- Vicente: corre a 6 km / hora, en bici va a 30 km / hora y nada a 1.000 m / hora

Todos evidentemente empiezan a competir a la vez. ¿Quién gana? Prueba y documenta



```
<terminated> IronMan [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\
Pedro comienza la prueba en bici
Vicente comienza la prueba en bici
Juan comienza la prueba en bici
Vicente ha acabado la prueba en bici
Vicente comienza la marathon
Pedro ha acabado la prueba en bici
```

Ejercicio 3 - RELEVOS

Crea una clase java con un método que simule que un corredor está corriendo 100 metros. La velocidad con la que corre 10 metros será un número aleatorio entre 950 y 1.050 milisegundos.

El método nos indicará cuando un corredor empieza el relevo y cuando lo acaba, además de al equipo al que pertenece.

Un corredor no puede empezar a correr hasta que le hayan pasado el relevo (ha acabado el corredor anterior). ¿Cómo podemos controlar que un hilo no empiece hasta haber acabado el anterior? “join”

```
Console ×
<terminated> Correr [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (3 o
España:Pepe comienza su relevo
España:Pepe ha acabado su relevo - Ha tardado: 9.81 segundos
España:Maria comienza su relevo
España:Maria ha acabado su relevo - Ha tardado: 10.36 segundos
España:Juan comienza su relevo
España:Juan ha acabado su relevo - Ha tardado: 10.26 segundos
España:Marta comienza su relevo
España:Marta ha acabado su relevo - Ha tardado: 9.92 segundos
FINAL DE PROGRAMA
```

Ejercicio 4 - CARRERA DE RELEVOS

Utilizando la clase anterior, o una copia ya que posiblemente necesites hacerle alguna modificación, haz un proyecto y añade otra clase con la que los equipos de España y EEUU van a hacer una carrera de relevos:

```
Console ×
<terminated> Equipo [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (3 oct 21
EEUU:John comienza su relevo
España:Pepe comienza su relevo
España:Pepe ha acabado su relevo - Ha tardado: 9.71 segundos
España:Maria comienza su relevo
EEUU:John ha acabado su relevo - Ha tardado: 10.3 segundos
EEUU:Kim comienza su relevo
España:Maria ha acabado su relevo - Ha tardado: 10.32 segundos
España:Juan comienza su relevo
EEUU:Kim ha acabado su relevo - Ha tardado: 10.21 segundos
EEUU:Mike comienza su relevo
España:Juan ha acabado su relevo - Ha tardado: 9.56 segundos
España:Marta comienza su relevo
EEUU:Mike ha acabado su relevo - Ha tardado: 10.0 segundos
EEUU:Sarah comienza su relevo
España:Marta ha acabado su relevo - Ha tardado: 9.92 segundos
FINAL DE PROGRAMA - España
EEUU:Sarah ha acabado su relevo - Ha tardado: 10.45 segundos
FINAL DE PROGRAMA - EEUU
```

Ejercicio 5 - El artificiero y la bomba

Haz una clase java en la que tengamos 2 métodos:

- Uno simplemente arranca una cuenta atrás desde el 9 a 0. Cuando llega a 0, la bomba explota.
- Otro es un artificiero que intenta desactivar la bomba. Tarda en desactivarla un número aleatorio entre 9000 y 10000 milisegundos.
 - El método indicará cuándo empieza y cuando acaba su tiempo.
 - En este método controlaremos la excepción de que se interrumpa el hilo (InterruptedException) e indicaremos que el artificiero ha muerto.

Estos métodos se arrancarán uno u otro en función de un parámetro.

Arranca 2 hilos, uno con el artificiero y otro con el contador.

Controla el final del hilo de la bomba para que cuando acabe, interrumpa el hilo del artificiero produciendo una salida como está

```
Console x
<terminated> Bomba [Java Application] C:\Program Files\Java\jdk-17.0.1\bin
9
El artificiero comienza a desactivar la bomba
8
7
6
5
4
3
2
1
0
PUUUUUUMMMMMMMMMM
El artificiero ha muerto
```

Si el artificiero acaba desactivando la bomba, la salida sería de este estilo:

<terminated> Bomba [Java Application] C:\Program Files\Java\jdk-17.0.1\bin\javaw.exe (

9

El artificiero comienza a desactivar la bomba

8

7

6

5

4

3

2

1

0

El artificiero ha desactivado la bomba

La bomba ha sido desactivada