

# RA2 EXAMEN PRÁCTICO

## TABLA DE CONTENIDO

<b>NOTAS Y CONSEJOS .....</b>	<b>2</b>
<b>ENTREGAS .....</b>	<b>2</b>
<b>PROBLEMA .....</b>	<b>3</b>
CONCEPTO .....	3
ESTRUCTURA .....	4
<i>Clases</i> .....	4
<i>Documentación</i> .....	5
QUE IMPRIME EL PROGRAMA POR PANTALLA.....	5
<i>Lo que se tiene que imprimir</i> .....	5
<i>captura de pantalla de ejemplo</i> .....	6
<b>CORRECCIÓN .....</b>	<b>7</b>

## NOTAS Y CONSEJOS

- 1) Antes de hacer nada, **lea atentamente el contenido**.
- 2) Es una **prueba individual**, no pidas ayuda a tus compañeros.
- 3) **Revisar el programa es tu responsabilidad**, si no funciona correctamente cuando lo pruebas, no funcionará cuando tu maestro lo revise.
- 4) Nombre del proyecto: *NombreApellido1Apellido2-Ex-RA2-E-20231123*
- 5) Los proyectos que no compilen o no se ejecuten y no incluyan el fichero .jar no se corregirán.
- 6) Se penalizará el código que esté mal estructurado y no esté refactorizado.
- 7) Es **obligatorio** poner comentarios en el código significativos que expliquen de manera clara el código programado.

### Recuerda que tienes que implementar:

- ☐ Creación de múltiples hilos.
- ☐ Estados de los hilos
- ☐ Prioridad.
- ☐ Datos compartidos.
- ☐ Sincronización.
- ☐ Modelo productor-consumidor.
- ☐ Documentación.

## ENTREGAS

- ☐ El proyecto **maven** en formato zip.
- ☐ Fichero .jar de la aplicación.

## PROBLEMA

### CONCEPTO

Se está librando una horrible batalla en las afueras de la Galaxia entre las flotas rebeldes e imperiales.

**Beslu Palug, Maza Aho y Daor Ban** trabajan juntos en un bombardero espacial de la Rebelión que ataca una fragata imperial.

**Daor Ban y Maza Aho** son oficiales especialistas manejando Drones Asesinos que son disparados hacia la fragata imperial. **Beslu Palug** por su parte, es el ingeniero que configura y prepara los drones en un dispositivo al que **Daor** y **Maza** acceden para bombardear la fragata imperial.

El dispositivo está formado por tres tubos en el que se colocan los drones para ser disparados. Por tanto, tiene una capacidad máxima de 3 drones (cada uno en un tubo) e inicialmente está vacío.

Solo una persona, tanto si es ingeniero u oficial, podrá acceder simultáneamente al dispositivo de drones.

**Beslu Palug** empezará a configurar un dron cada 600 ms.

**Daor Ban** tiene más **prioridad** que **Maza Aho** a la hora de tomar el control del dispositivo porque tiene un rango más alto. En un rango del 1 al 10, **Daor Ban** tiene prioridad 10 y **Maza Aho** 5.

**Daor Ban y Maza Aho** se dedicarán a:

- ☐ Tomar el control del dispositivo.
- ☐ Esperar un tiempo para simular la cadencia de disparo.
- ☐ Hacer un disparo del dron (imprimir un mensaje por pantalla)

**Daor Ban** intentará disparar 1 dron cada 5 segundos.

**Maza Aho** intentará disparar un dron cada 8 segundos.

El proceso de configurar el dron y dispararlo debe ser absolutamente preciso y sincronizado:

- ☐ Si **Daor** o **Maza** obtener un dron mientras no hay drones en el dispositivo, puede ocurrir un accidente.
- ☐ Si **Beslu Palug** intenta introducir un dron cuando el dispositivo está lleno, puede ocurrir un accidente.

La batalla termina después de que la fragata imperial sea golpeada **6** veces. Cuando la fragata imperial es destruida, los estoicos combatientes rebeldes dicen “¡Sí! ¡La batalla ha terminado!” y se muestra un recuento de cuántos drones han disparado **Daor** y **Maza**.

## QUE LA FUERZA TE ACOMPAÑE

### ESTRUCTURA

#### CLASES

El programa debe contener las siguientes clases:

- ☐ **Clase Principal.**
  - Nombre: App.java
  - Es la clase que crea y lanza todos los hilos a ejecución.
- ☐ **Clase del objeto compartido (se implementará un monitor).**
  - Nombre: DispositivoDrones.java
  - El constructor debe permitir especificar la capacidad del dispositivo (número de tubos) y el número tubos libres (sin drones) en el dispositivo.
  - Esta clase debe guardar: el **número de tubos del dispositivo (3 drones)**, el **número de tubos libres (inicialmente 3 porque está vacío)** y el **número de golpes a la flota** que se han hecho.
  - Tendrá los métodos:
    - *setDron*: donde el ingeniero intentará tomar control del dispositivo e introducir un nuevo dron.
    - *getDron*: donde los oficiales intentarán tomar el control del dispositivo para obtener un dron y dispararlo.
- ☐ **Clase Productor.**
  - Ingeniero.java
  - Debe intentar producir un dron mientras no termine la batalla.
- ☐ **Clase Consumidor.**
  - Oficial.java
  - Debe intentar obtener un dron mientras no termine la batalla.

## DOCUMENTACIÓN

Debes crear un comentario identificador y explicativo encima de cada bloque, como por ejemplo los atributos, el constructor y cada uno de los métodos.

### QUE IMPRIME EL PROGRAMA POR PANTALLA

#### LO QUE SE TIENE QUE IMPRIMIR DE MANERA SINCRONIZADA

##### 1) **Principal.**

- a. Imprime *"Hola, el usuario <nombreusuario> está haciendo la aplicación para el RA2!"*.
- b. Imprime *"Hace mucho tiempo en una galaxia muy muy lejana..."*
- c. Imprime *"Se está librando una horrible batalla en las afueras de la Galaxia entre las flotas rebeldes e imperiales."*
- d. Cuando **termine** la ejecución de todos los hilos se deberán imprimir los siguientes mensajes: *"<nombre del hilo> ha disparado: X drones"* y *"FIN"*.

##### 2) **Productor.**

- a. Imprime cada vez que el ingeniero toma el control del objeto compartido *"Beep Beep, crrrk, beep, ingeniero <nombre del hilo> he tomado el control"*.
- b. Imprime cada vez que comienza a configurar un dron *"Soy el ingeniero <nombre del hilo> y estoy configurando un nuevo dron asesino!"*.
- c. Esperar **100 ms** para simular el tiempo de configurar un dron.
- d. *"Soy el ingeniero <nombre del hilo>! He configurado un nuevo dron!"*.
- e. Imprime *"Tubos libres:X"*. Donde X es el número de tubos sin drones en el dispositivo.
- f. Imprime cuando la batalla ha terminado *"Si! La batalla ha terminado!"*

##### 3) **Consumidor.**

- a. Imprime cada vez que uno de los artilleros coge el control del objeto compartido *"Beep Beep, crrrk, beep, oficial <nombre del hilo> he tomado el control"*.
- b. Imprime que va a coger un dron *"Soy el oficial <nombre del hilo>! Voy a coger un nuevo dron asesino!"*.
- c. Esperar **20 ms** antes de disparar para simular la cadencia de disparo.
- d. Imprime cuando dispara el dron *"Soy el oficial <nombre del hilo>!He disparado el dron asesino!"*.
- e. Imprime *"Tubos libres:X"*. Donde X es el número de tubos sin drones en el dispositivo.
- f. Imprime cuando la batalla ha terminado *"<nombre hilo> Si! La batalla ha terminado!"*

---

## CAPTURA DE PANTALLA DE EJEMPLO

```
Hola, el usuario Joaquín Franco Ros está haciendo la aplicación para el RA2!
----- STAR WARS -----

Hace mucho tiempo en una galaxia muy muy lejana...

Se está librando una horrible batalla en las afueras de la Galaxia

entre las flotas rebeldes e imperiales ...
-----

Beep Beep, crrrk, beep, ingeniero Beslu Palug he tomado el control
    Soy el ingeniero Beslu Palug y estoy configurando un nuevo dron asesino
    Soy el ingeniero Beslu Palug he configurado un nuevo dron!
Tubos libres 2
Beep Beep, crrrk, beep, oficial Daor Ban he tomado el control
    Soy el oficial Daor Ban y voy a coger un nuevo dron asesino!
    Soy el oficial Daor Ban he disparado un dron asesino!. Número de golpes: 1
Tubos libres 3
Beep Beep, crrrk, beep, ingeniero Beslu Palug he tomado el control
    Soy el ingeniero Beslu Palug y estoy configurando un nuevo dron asesino
    Soy el ingeniero Beslu Palug he configurado un nuevo dron!
Tubos libres 2
Beep Beep, crrrk, beep, oficial Maza Aho he tomado el control
    Soy el oficial Maza Aho y voy a coger un nuevo dron asesino!
    Soy el oficial Maza Aho he disparado un dron asesino!. Número de golpes: 2
Tubos libres 3
Beep Beep, crrrk, beep, ingeniero Beslu Palug he tomado el control
    Soy el ingeniero Beslu Palug y estoy configurando un nuevo dron asesino
    Soy el ingeniero Beslu Palug he configurado un nuevo dron!
Tubos libres 2
Beep Beep, crrrk, beep, ingeniero Beslu Palug he tomado el control
    Soy el ingeniero Beslu Palug y estoy configurando un nuevo dron asesino
    Soy el ingeniero Beslu Palug he configurado un nuevo dron!
Tubos libres 1
Beep Beep, crrrk, beep, ingeniero Beslu Palug he tomado el control
    Soy el ingeniero Beslu Palug y estoy configurando un nuevo dron asesino
    Soy el ingeniero Beslu Palug he configurado un nuevo dron!
Tubos libres 0
Beep Beep, crrrk, beep, oficial Daor Ban he tomado el control
    Soy el oficial Daor Ban y voy a coger un nuevo dron asesino!
    Soy el oficial Daor Ban he disparado un dron asesino!. Número de golpes: 3
Tubos libres 1

.....

Tubos libres 1
Beep Beep, crrrk, beep, ingeniero Beslu Palug he tomado el control
    Soy el ingeniero Beslu Palug y estoy configurando un nuevo dron asesino
    Soy el ingeniero Beslu Palug he configurado un nuevo dron!
Tubos libres 0
Beslu Palug:Si! La batalla ha terminado
Maza Aho:Si! La batalla ha terminado
Daor Ban:Si! La batalla ha terminado
-----
Daor Ban ha disparado: 4 drones
Maza Aho ha disparado: 2 drones
FIN
-----
BUILD SUCCESS
```

## CORRECCIÓN

RA2-CE-b) Se han reconocido los mecanismos para crear, iniciar y finalizar hilos.		
Logro	Peso %	Valoración
Se crean los hilos correctamente utilizando el constructor adecuado para introducir todos los datos necesarios.	30	Sí/No
Se han inicializado los hilos correctamente.	30	Sí/No
Los hilos finalizan correctamente se han disparados los 6 drones	40	Sí/No

RA2-CE-c) Se han programado aplicaciones que implementen varios hilos.		
Logro	Peso %	Valoración
Se crean los hilos necesarios para la simulación del problema correctamente.	20	Sí/No
Los hilos hacen la tarea correcta descrita en el enunciado.	80	Sí/No

RA2-CE-d) Se han identificado los posibles estados de ejecución de un hilo y programado aplicaciones que los gestionen.		
Logro	Peso %	Valoración
Los hilos simulan el tiempo que tardan en hacer una tarea pausándose el tiempo correcto y en el momento correcto.	80	Sí/No
El hilo principal se bloquea correctamente para imprimir los mensajes finales.	20	Sí/No

RA2-CE-e) Se han utilizado mecanismos para compartir información entre varios hilos de un mismo proceso.		
Logro	Peso %	Valoración
Se utiliza el mecanismo correcto para compartir información entre los hilos.	50	Sí/No
Se protege la información compartida.	50	Sí/No

RA2-CE-f) Se han desarrollado programas formados por varios hilos sincronizados mediante técnicas específicas.		
Logro	Peso %	Valoración
Se lleva a cabo a cabo la sincronización correcta de los hilos para el acceso al monitor.	50	Sí/No
La salida del programa es correcta.	50	Sí/No

RA2-CE-g) Se ha establecido y controlado la prioridad de cada uno de los hilos de ejecución.		
Logro	Peso %	Valoración
Se ha establecido la prioridad de todos los hilos afectados.	100	Sí/No

RA2-CE-h) Se han depurado y documentado los programas desarrollados		
Logro	Peso %	Valoración
El programa es robusto y funciona perfectamente.	50	Sí/No
El programa contiene comentarios explicativos claros y completos.	50	Sí/No