



Programación de servicios y procesos

DAM2: Examen 1 Evaluación

Nombre y Apellidos

FECHA

La puntuación máxima será de 10 puntos. La duración del examen 2 horas.

SUPUESTO PRÁCTICO 1.

Crea el programa en Java que simule el siguiente comportamiento:

Cierta **ITV** quiere gestionar las revisiones de sus clientes de manera concurrente.

Los miembros del sistema, serán los **ITV**, **Vehículo** y **Puesto**.

Cada día llegarán una serie de coches y éstos no podrán acceder de forma simultánea a la ITV.

Tanto **Vehículo** como **Puesto** contendrán un identificador numérico. **Puesto** además tendrá el tiempo total que se ha trabajado (necesitamos comprobar que los funcionarios cumplen con su jornada laboral en su totalidad).

La **ITV** contendrá un ArrayList de números para llevar la cuenta de los coches que quedan pendiente y, además, realizará dos acciones: **añadirCoche(int numeroCoche)** y **terminarCoche(int tiempoRevisión)**.

- El primero añadirá el coche que le toque a la cola de espera hasta que sea atendido para ser revisado.
- El segundo se ejecutará cuando el coche haya sido revisado, añadirá el tiempo que ha tardado en revisar el coche al total del trabajo realizado por la ITV en el día de hoy, liberará el primer elemento de la lista y comprobará si hay coches pendientes de tramitar.

El **Vehículo** deberá poder realizar la siguientes acciones:

- `System.out.println("Soy coche" +id+" quiero entrar");`
- `/* posibles acciones para sincronizar */`
- `System.out.println("Soy coche" +id+" quiero que me atiendan");`
- `/* posibles acciones para sincronizar */`
- `System.out.println("Soy coche" +id+" estoy añadido");`
- `/* posibles acciones para sincronizar */`

El **Puesto** deberá poder realizar las siguientes acciones:



- `System.out.println("Soy puesto "+id+" comienzo a atender");`
- `/* posibles acciones para sincronizar */`
- `System.out.println("He terminado el vehículo, me duermo hasta que me toque currar"+n);`
- `/* posibles acciones para sincronizar */`
- `System.out.println("Soy puesto "+id+" termino mi jornada");`

Cuando el puesto abra, trabajará mientras que haya coches pendientes.

La revisión realizada a cada **Vehículo** conllevará un tiempo aleatorio entre 1 y 10 segundos.

Tras terminar de revisar el **Vehículo** la ITV modificará su ArrayList para que tenga un elemento menos, el **Puesto** tendrá que prepararse para la siguiente revisión, por lo que se "dormirá" un tiempo aleatorio entre 1 y 2 segundos, tiempo dedicado a revisar dicho coche. Cuando el Puesto verifique que no hay más puestos pendientes en ITV, mostrará un mensaje indicando su identificador junto al tiempo trabajado.

El programa **Principal** creará una **ITV** más un conjunto de **Vehículos** de 7, y exactamente 3 puestos. Tras su inicialización y ejecución, los **Puestos** no podrán abandonar el recinto hasta que todos los Vehículo hayan sido revisados: esperarán a que todos sus afines Puesto terminen.

Al final de la jornada, cuando ya no haya ningún coche que revisar, se mostrará el tiempo total trabajado por la **ITV** en el día de hoy.

Criterios de calificación:

- Si el programa funciona perfectamente: 5 puntos máximo.
- El programa tiene errores leves de sincronización (retorna o muestra los datos correctamente) o faltas leves en la funcionalidad: no realiza alguna de las acciones solicitadas: 3 puntos máximo.
- El programa tiene errores graves de sincronización (se produce *interbloqueo* en algunos procesos): 1 punto máximo.
- El programa no cumple con la funcionalidad solicitada: 0 puntos.