

EXAMEN EXTRAORDINARIO DE PROGRAMACIÓN CONCURRENTE

Grado en Ingeniería de Computadores, URJC

11 de julio de 2020

Este examen cuenta 8 puntos en la nota final extraordinaria de la asignatura

Por favor, **es muy importante que primero leas las siguientes instrucciones** para realizar este examen. Cualquier incumplimiento de cualquiera de ellas será un 0 en todo examen.

- La resolución del examen es totalmente individual y deberá hacerse sin ayuda de ningún tipo: no se pueden mirar las diapositivas, ni ejercicios resueltos, ni Internet, ni consultar con otros compañeros, ni consultar con otras personas, etc. Las soluciones enviadas se pasarán por varios software anticopia, y además los profesores examinaremos personalmente todo el código con nuestros ojos. **Si hay fraude, tanto el alumno copiado como el alumno que copia suspenderán inmediatamente toda la asignatura y además se les abrirá un expediente disciplinario que puede conducir a la expulsión de la universidad.**
- La hora de comienzo del examen son las 9:00 de la mañana, y la hora de finalización será a las 12:30. No obstante, se podrán entregar exámenes hasta las 13:00. A las 13:00 se cerrará automáticamente la entrega y no se admitirán bajo ningún concepto más entregas por cualquier otro medio de envío.
- Debido a que a las 13:00 ya no se admitirán más entregas, tienes que terminar tu examen a las 12:30 para que tengas tiempo de sobra para enviarlo, en previsión de que pueda ocurrir algún percance (caída de la conexión a Internet o fallo en el aula virtual, por ejemplo). No se admitirá ninguna excusa si el examen no ha sido enviado a tiempo, ya que es responsabilidad del alumno asegurarse de que su conexión funciona y terminar a las 12:30 por si ocurre cualquier imprevisto.
- El único medio de envío de la solución de este examen es el Aula Virtual, en la categoría "Evaluación". No se admitirán envíos por otros medios (correo, mensajes privados, etc.)
- Se deberá enviar un único archivo .zip con todo el código fuente del examen. Los archivos fuentes de cada ejercicio del examen se pondrán en una carpeta distinta, y luego todas las carpetas se comprimirán en un único .zip.
- Se permite 1 sólo envío de la solución, por lo que, una vez enviado, no se permitirán más modificaciones. El alumno tendrá que asegurarse bien antes de enviar su examen.

Es **muy importante que también leas lo siguiente**, a tener en cuenta por cada ejercicio del examen:

- Hay que respetar totalmente los requisitos de cada ejercicio, incluyendo la forma de hacerlo o la complejidad temporal que deba respetar. En caso de que no se respete alguno, se tendrá un 0 en ese ejercicio.
- Se restará nota en cada método si los nombres de variables, parámetros, métodos, etc. no son adecuados.
- Se restará nota en cada método si el código es confuso, lioso, difícil de entender, repetitivo, innecesariamente complicado, sin utilidad, etc.
- Se valorará el orden, la claridad, la buena estructuración en métodos/clases, y la buena nomenclatura en el código
- Es necesario utilizar únicamente Java estándar en todos los ejercicios.
- En cada ejercicio, recomendamos utilizar sólo lo aprendido durante la asignatura. En caso de duda, mejor preguntar.
- A no ser que se indique otra cosa, no se podrá reutilizar código de los ejemplos/ejercicios desarrollados durante la asignatura, ni de las diapositivas del curso, ni de otros exámenes, ni de otras personas, ni de libros, ni de Internet, etc.

- Un ejercicio tendrá un 0 si no compila o si da error en tiempo de ejecución (ej: violación de memoria, excepción en tiempo de ejecución).
- En cada método, cualquier otro error restará una cantidad de puntos proporcional a su importancia, según el criterio del profesor.
- Se valorará fuertemente el grado de concurrencia alcanzado.

1) (4 puntos) Utilizando únicamente semáforos de Java como herramientas de sincronización y/o concurrencia (en caso contrario, este ejercicio tendrá un 0), implementa un programa que simule una carrera de coches de tipo "Scalextric":

- El circuito está formado por N piezas, puestas una detrás de otra y sin formar un circuito cerrado. Es decir: el primer coche que llegue a la última pieza gana, la última pieza no está conectada con la primera.
- El circuito se genera aleatoriamente al inicio del programa. Se imprime por consola en ese momento la composición del circuito: cada una de las piezas en orden secuencial y el número de pistas que tiene cada pieza.
- Tenemos M coches compitiendo.
- La primera pieza siempre tiene M pistas y cada coche se sitúa en una pista.
- El resto de cada una de las piezas tienen de 1 a M pistas. El número de pistas de cada una de las piezas (cuando se crea el circuito) se selecciona aleatoriamente de forma equiprobable.
- No puede haber dos o más coches simultáneamente en la misma pista de una misma pieza.
- Cada pista de cada pieza está conectada con todas las pistas de la siguiente pieza, de tal modo que un coche que entre en una pieza puede colocarse en cualquiera de sus pistas libres (selecciona una de las pistas aleatoriamente, de forma equiprobable).
- Cada coche, en cada pieza, tarda una cantidad aleatoria de segundos en recorrerla, entre 1 y 3 segundos.
- Si una pieza tiene todas sus pistas ocupadas y un coche de la anterior pieza quiere entrar, el coche deberá esperar a que una de las pistas esté libre. El coche entrará inmediatamente en la primera pista que se quede libre en esa pieza. Si hay varios coches esperando, entrará uno de ellos, según la política del semáforo (no hay que preocuparse por la política del semáforo).
- Los coches nunca se chocan ni se salen de la pista o el circuito.
- El programa imprimirá una línea por consola cada vez que un coche entre en una nueva pieza, indicando el número de coche, el número de pieza, y el número de pista en la que ha entrado (empezando a numerar desde la 0). La impresión del mensaje tiene que ser inmediata al entrar el coche a la pieza, de tal modo que podamos saber, únicamente viendo qué mensaje aparece antes, en qué orden van llegando a cada pieza y pista los coches.
- El programa también imprimirá por consola cada vez que un coche llega a la meta (cuando se acaba de recorrer la última pieza), indicando el número de coche. La impresión del mensaje tiene que ser inmediata al llegar el coche a la meta, de tal modo que podamos saber, únicamente viendo qué mensaje aparece antes, en qué orden van llegando a la meta los coches.

2) (4 puntos) Utilizando el esquema y las herramientas de "Java Fork Join" (en caso contrario, el ejercicio se evaluará con cero), implementa una búsqueda recursiva de un número entero en un array de N números enteros. El array se llenará al principio con números enteros aleatorios (sin límite ni por la parte positiva ni por la negativa). La búsqueda debe devolver finalmente la posición del array en donde se encuentra el elemento buscado, o -1 si no se encuentra. La búsqueda recursiva debe realizarse de la siguiente manera:

- Si el array a buscar es suficientemente pequeño, se busca en él secuencialmente.
- Si el array es más grande, se divide en dos partes y se realiza recursivamente la búsqueda en ambas partes al mismo tiempo.