

28 de Octubre de 2013

Apellidos:

Nombre:

DNI:

Normativa del examen

- La duración máxima del examen será de 1:50 horas.
- La hoja del examen debe contener claramente el nombre, los apellidos y el DNI del alumno.
- Esta prueba está compuesta por una prueba sobre la Práctica 1 y el Examen 1.

Práctica 1

Contesta a las siguientes preguntas relacionadas con la Práctica 1. Si lo consideras necesario para explicar tu respuesta, indica la parte del código a la que te refieres con el número de línea.

1) Describe de forma general, sin entrar en detalles, cómo funciona el programa implementado.

2) Explica qué tipos de sincronización utilizas en el programa.

3) Explica para qué sirven cada uno de los semáforos que has usado. Justifica el valor de inicialización.

4) Explica para qué sirve cada una de las variables de control que usas (contadores y variables booleanas)

Examen 1

Ejercicio 1) Preguntas cortas (6p)

1.a - ¿Qué es un programa secuencial? ¿Y un proceso? ¿En qué se parecen y en qué se diferencian los procesos de los hilos de ejecución? ¿En qué casos se usa cada uno de ellos? (1.5p)

1.c - Describe qué son la sincronización condicional y la exclusión mutua. (1.5p)

1.b - ¿Qué es un semáforo? ¿Para qué se utiliza? ¿Qué operaciones tiene y cómo se comportan? ¿Cómo se usan para implementar la sincronización condicional y la exclusión mutua? (3p)

Ejercicio 2) Programa concurrente (4p)

Se pide implementar completamente un programa concurrente en Java con SimpleConcurrent que tenga los siguientes requisitos:

- 1) El programa debe tener 3 procesos iguales (que ejecuten el mismo método)
- 2) El método recibirá como parámetro un número que indicará el número de proceso.
- 3) Los procesos se crearán con un bucle for para pasar como parámetro el número de proceso que corresponda a cada uno de ellos.
- 4) Cada proceso deberá escribir en pantalla la letra 'X' seguida del número de proceso y quedará bloqueado a la espera de que los demás procesos también hayan escrito la letra 'X' con su número de proceso para poder continuar.
- 5) Los procesos escribirán la salida por pantalla de forma concurrente con los demás procesos (simulando una tarea que se puede realizar en paralelo).
- 6) Uno de los procesos debe escribir un espacio después de que todos hayan escrito la X con el número de proceso y antes de que vuelvan a realizar esa tarea. De esa forma, la salida de cada "fase" se separará con un espacio.
- 7) Las tareas que realizan los procesos se deben repetir indefinidamente
- 8) Para la implementación deben utilizarse semáforos
- 9) Una de las posibles salidas del programa es la que se muestra a continuación (aquí se ha cortado la ejecución, pero el programa deberá mostrar este patrón indefinidamente):

x1x3x2 x2x1x3 x1x2x3 x3x2x1 x3x2x1 x3x1x2
--

- 10) Además de escribir todo el código fuente, el alumno tiene explicar razonadamente la finalidad de cada uno de los semáforos que utilice.