

# PROGRAMACIÓN AVANZADA

PEC1T – 23 de marzo de 2017

## Normas de Examen:

No se recogerá el enunciado del examen.

En cada pregunta se indica el valor de la misma sobre la nota final.

La duración del examen es de dos horas

Se numerarán todas las hojas con el número de hoja y el total (ejemplo: 1/4, 2/4, 3/4, 4/4).

1) (1,5 puntos) Elegir la respuesta correcta (cada pregunta correcta puntúa 0,25, las no respondidas no puntúan y las erróneas puntúan -0,25).

1. Los paradigmas complementarios...
  - a) implican que el nivel de cumplimiento de uno se complementa con el otro
  - b) sirven para completar aspectos pendientes de un paradigma general
  - c) se caracterizan porque no cumplir uno implica cumplir el otro
  - d) complementan las metodologías asociadas
2. El tipo de HW MIMD engloba a los dispositivos:
  - a) multihilo, multitarea y multiproceso
  - b) multiprocesadores, multitareas y multidistribuidos
  - c) multicomputadores, multiprocesadores y sistemas distribuidos
  - d) multiprocesadores, sistemas distribuidos y multihilo
3. El nivel de compromiso de Java con la concurrencia es...
  - a) completo en las versiones enterprise
  - b) dependiente del rendimiento de la máquina virtual y la implementación
  - c) completo y se usa incluso para el funcionamiento base del sistema
  - d) inferior que el de la mayoría de lenguajes existentes
4. Los diagramas de precedencia y redes de petri son herramientas para...
  - a) representar en diagramas las situaciones de concurrencia que plantean los algoritmos
  - b) optimizar el rendimiento de los algoritmos en general
  - c) mostrar el rendimiento de programas y mejorar los detalles de implementación de los mismos
  - d) comparar algoritmos mediante herramientas y algoritmos de comparación de grafos
5. Una sección crítica es:
  - a) una sección de código que compromete la corrección por usar variables compartidas
  - b) una sección con exclusión combinada pero no mutua
  - c) un mecanismo de distribución de la coordinación
  - d) una sección de código asegurada con exclusión mutua mediante mecanismos o algoritmos
6. En un modelo de sincronización pura:
  - a) los procesos sincronizan cada una de sus actividades con el resto de los procesos
  - b) los hilos sincronizan sus actividades con todos los hilos en su mismo o en otros procesos
  - c) los procesos comparten información pero se coordinan previamente
  - d) los procesos no comparten información y sólo coordinan su ejecución en el tiempo

2) (1,5 puntos) Definir el concepto Thread-Safe en contraposición al concepto Reentrant. Explicar mediante ejemplos Java (comentando las instrucciones).

3) (1 punto) Explicar las diferencias entre la ejecución de tareas en pools de hilos o mediante la creación directa de hilos (utilizar como máximo una cara).



4) (3 puntos) Un museo de obras de arte con capacidad máxima de 20 personas tiene una salida y dos entradas (Este y Oeste). Las personas pueden entrar si hay plazas libres. Si está lleno, no podrá entrar una persona hasta que otra haya salido. En este caso, si hay personas esperando en las dos entradas, hay que alternar la entrada seleccionada (es decir, una vez entrará una persona del Este y la siguiente vez una del Oeste).

Se debe desarrollar un programa completo que simule este funcionamiento, utilizando una clase compartida que implemente este protocolo de gestión. En la clase compartida deben existir al menos las siguientes tres operaciones:

- “entrar\_museo\_este”, “entrar\_museo\_oeste”: los procesos llaman a esta operación cuando quieren entrar por la entrada Este u Oeste, respectivamente. Cuando el museo está lleno, los procesos se deben quedar bloqueados hasta que se les permita entrar.
- “salir\_museo”: los procesos llaman a esta operación cuando quieren salir del museo.

Resolver este problema utilizando únicamente cerrojos (y sus variantes).

*Se pide: desarrollar el código completo del programa utilizando exclusivamente cerrojos (y sus variantes). El único elemento static del programa será el método main. El programa deberá funcionar a excepción de posibles errores menores “de compilación”.*

5) (3 puntos) Se desea modelar el comportamiento de un conjunto de coches que van a hacer una carrera y los coches de seguridad (safety-car) que, además, son los responsables de la seguridad en la carrera.

Se cuenta con 20 coches de carreras y 2 coches de seguridad. Todos ellos van llegando al circuito de forma escalonada y en cualquier orden (tiempo aleatorio entre 4-16s). Una vez reunidos los 22 coches en el circuito se pasa a la fase de revisión del circuito. Los 2 coches de seguridad inician una vuelta de revisión del circuito para comprobar que todo está en orden (tardan de 20s a 30s) mientras los 20 coches de carreras esperan a que terminen. 1

A continuación comienzan la carrera los 20 coches de carreras (duración 120s). Los 2 coches de seguridad esperan a que todos los corredores terminen y los 2 coches de seguridad finalizan entonces con un mensaje indicando el final de la carrera.

Se mostrará en pantalla cada cambio de estado, incluyendo la finalización de cada estado y el comienzo del siguiente.

El programa se deberá modelar sin utilizar monitores, semáforos ni cerrojos.

*Se pide: desarrollar el código completo del programa. El único elemento static del programa será el método main. El programa deberá funcionar a excepción de posibles errores menores “de compilación”.*