CDI Examen Teoría **Dr. Arno Formella, Mayo de 2016**

Apellidos: _____ **Nombre:** _____

D.N.I: _____ **Firma:** _____

1	2	3	4	5	6	7	8	9	10	11	12	Suma
(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(5)	(30/60)

Los/as estudiantes **con** evaluación continua deben contestar solamente las primeras **seis** (6) preguntas (Parte I). Tiempo: 1 hora y media.

Los/as estudiantes **sin** evaluación continua deben contestar todas las **doce** (12) preguntas (Parte I+II+III). Tiempo: 3 horas y media.

Una vez empezado el examen, el/la estudiante asume que cualquier uso de un dispositivo móvil sin previo aviso, resulta en un suspenso inmediato del examen con puntuación 0, informe a los órganos competentes de la Universidad de un posible intento de fraude, y posibilidad a un futuro examen en este curso solamente mediante orden escrito del órgano superior.

Parte I

Pregunta 1: [5 Puntos] La *región crítica* es un concepto abstracto útil para la programación concurrente. ¿Está disponible este concepto en el lenguaje de programación Java? Razona críticamente sobre posibles ventajas y desventajas. ¿Cuál es el desafio principal cuando se traslada el concepto a un entorno distribuido?

Pregunta 2: [5 Puntos] Durante la fase de análisis para una nueva aplicación que se debe realizar con un programa concurrente ¿cuáles serían aspectos a los cuales hay que prestar atención especial para el diseño del programa concurrente?

Pregunta 3: [5 Puntos] ¿Cuáles son las condiciones que se tienen que cumplir para que se produzca un bloqueo entre procesos? ¿Cuál es el peligro principal de un bloqueo entre procesos? ¿Cómo se puede prevenir la aparencia de un bloqueo entre procesos en muchas ocaciones ya durante el diseño de una aplicación? Razona críticamente sobre sus ventajas y limitaciones.

Pregunta 4: [5 Puntos] Ha aparecido en el marcado un nuevo protocolo de entrada y salida para garantizar la exclusión mutua de una sección crítica involucrando dos procesos. Tienes acceso al código fuente en lenguaje ensemblador y sabes que todas las instrucciones básicas se ejecutan de forma atómica. ¿Cuáles son las propiedades de los protocolos que deberías analizar? ¿Entre ellas, cómo verificas si el protocolo es correcto (en el sentido que garantice la exclusión mutua) antes de comprarlo?

Pregunta 5: [5 Puntos] ¿Qué se entiende bajo una *condición de carrera*? Explica un caso de cierta importancia en el uso de los monitores de Java al cual hay que presentar atención especial durante el diseño de una aplicación concurrente. ¿Cómo implementarías (en código de Java) una solución al problema?

Pregunta 6: [5 Puntos] ¿En la quinta práctica, existe en la propuesta de tu solución la posibilidad de un bloqueo entre hilos? En caso que no, ¿cuál es el argumento? ¿Existe la posiblidad que ocurra una espera infinita (a una respuesta del cliente) en el servidor? En caso que no, ¿cómo lo has logrado (o cómo lo lograrías)?

Parte II

Pregunta 7: [5 Puntos] Explica la semántica del modificador `volatile` de Java y su uso en programas concurrentes, entre otras, ¿qué tiene que ver con una relación *ha-pasado-antes*, es decir, qué garantías tiene el/a programador/a con el uso de `volatile` escribiendo y leyendo variables en su código?

Pregunta 8: [5 Puntos] Describe brevemente cinco (5) características disponibles en los paquetes `java.util.conconcurrent` y `java.util.conconcurrent.lock`. Destaca en cada caso su semántica princial y su especial relevancia para su uso en programas concurrentes.

Pregunta 9: [5 Puntos] Describe/implementa con la instrucción hardware CAS un protocolo de entrada y salida a una sección crítica para un número fijo de procesos con garantía de espera finita.

Pregunta 10: [5 Puntos] Durante la fase de depuración de un programa concurrente, al/a responsable de realizar las pruebas ocurre la idea de producir diferentes versiones del código introduciendo simples comandos de `sleep` de cierta duración al azar entre líneas sin cambiar nada más. ¿Crees que es buena idea? En caso que sí, ¿qué tipo de fallos se pueden detectar? ¿Te ocurre algún inconveniente de tal prueba (a parte que las pruebas siempre consumen tiempo en la producción de una aplicación software)?

Parte III

Pregunta 11: [5 Puntos] Describe como has implementado la medición de tiempo de ejecución de tu programa concurrente. ¿Qué tipos de tiempo hay que distinguir cuya combinación resulta en el tiempo global? ¿Cúales fueron tus tiempos de ejecución medidos (a grandes rascos) para el problema con la operación sobre matrices grandes y para el problema del ping-pong aumentando gradualmente el número de hilos participantes en ambos casos?

Pregunta 12: [5 Puntos] ¿Cómo implementaste el contador concurrente en una de las prácticas? ¿Te ocurre ahora otra posible solución?