



## PROGRAMACION ORIENTADA A OBJETOS

### *Guía para la preparación del examen final*

Este documento presenta algunas observaciones a tener en cuenta para quienes deban rendir el final de la materia Programación Orientada a Objetos, en función de las dudas que la mayoría tiene al momento de preparar el examen.

- 1) **No estudiar directamente de las copias de las transparencias.** Las mismas son construídas y usadas para el dictado de las clases teóricas, y no como material de estudio. Se recomienda utilizarlas como referencia, pero estudiar de la bibliografía de la materia. Algunas transparencias tienen errores o descuidos y esto no puede ser usado como pretexto ante la corrección del examen.
- 2) Es indispensable estudiar detalladamente los siguientes capítulos del libro "Object Oriented Software Construction", de Bertrand Meyer:

- Capítulo 1 [*Calidad del Software*]
- Capítulo 3 [*Modularidad*]
- Capítulo 7 [*Clases*]
- Capítulo 8 [*Objetos*]
- Capítulo 9 [*Administración de memoria*]
- Capítulo 10 [*Genericidad*]
- Capítulo 11 [*Diseño por Contrato*]
- Capítulo 12 [*Excepciones*]
- Capítulo 14 [*Introducción a la herencia*]
- Capítulo 15 [*Herencia Múltiple*]
- Capítulo 16 [*Técnicas de Herencia*]  
Secciones 16.1, 16.4 y 16.6
- Capítulo 22 [*Cómo encontrar las clases*]
- Capítulo 24 [*Usando herencia correctamente*]

El lenguaje base de este libro es Eiffel. El mismo **no** forma parte de los requerimientos del final, esto es, no se evalúa Eiffel, salvo el tratamiento de excepciones (`rescue..retry`), el cual es utilizado en nuestro lenguaje de diseño. En algunos casos conviene examinar los ejemplos de Eiffel dados en el libro, pues muestran algún aspecto particular de lo que se está explicando.

- 3) Se espera un conocimiento adecuado del lenguaje Java básico: definición de clases, manipulación de objetos, construcción de jerarquías de herencia, visibilidad de entidades de una clase, applets, utilización de streams, etc. El nivel de conocimiento esperado es el de aquél que ha podido construir un sistema de mediana complejidad, tal cual ocurre en el proyecto final. Conviene repasar los tutoriales, los manuales y el proyecto desarrollado cuando la cursaron. Repasar las clases finales sobre Java Avanzado. **Abstenerse de presentarse al examen si no se ha practicado con el lenguaje. Si en el examen que rinde se falla en todos los ejercicios que involucren Java no puede estar aprobado.**
- 4) El examen final **usualmente** consta de diez ejercicios o preguntas puntuales. El número puede disminuir si se incluye un ejercicio de diseño de clases. Los ejercicios pueden ser de programación (en Java o en lenguaje de diseño) o de desarrollo teórico. La nota que se obtiene es proporcional a los ejercicios respondidos correctamente. Cualquier detalle que no corresponda a lo solicitado en el enunciado no se tendrá en cuenta.
- 5) El examen libre consta de dos exámenes separados, donde se evalúa teoría y práctica correspondiente a cada examen parcial de la materia. Deben desarrollar durante un período de tiempo limitado (usualmente una semana o diez días) un proyecto de software cuyo enunciado es entregado por la cátedra. La mesa de examen libre debe ser solicitada con quince días de anticipación. La nota que se obtiene surge de las notas de los exámenes y del proyecto realizado. El permiso de examen debe ser entregado al momento de comenzar el proceso de evaluación. No puede rendirse un examen, o evaluarse el proyecto, sin haber aprobado la instancia anterior.