# Programación Avanzada
# – Diciembre de 2003 –

## Examen de Teoría

| NOTA |
| --- |
| |

Universidade de Vigo

ESCOLA SUPERIOR DE
ENXEÑERÍA INFORMÁTICA

**Apellidos** _____

**Nombre** _____

**PLAN:** • **Nuevo** • **Viejo** **DNI** _____

---

**IMPORTANTE**:

- El tiempo para la realización del examen es de 2 horas.
- Se deben cubrir los datos del encabezado y firmar la parte de atrás de esta hoja, que será grapada junto con las hojas de respuestas que deberán ir numeradas.

REVISIÓN DE EXÁMENES:
**Viernes (5 de diciembre) [9:00-11:00] y [17:30-18:30]**.

---

**1.-** *Modificadores de acceso de los miembros (métodos y variables) en java. Explicar el alcance de cada uno de ellos.*

**2.-** ¿Una clase puede ser a la vez *final* y *abstract*?. Razonar la respuesta.

**3.-** *Generalidades sobre Java:*
3.1.- ¿Qué diferencia existe entre un JDK y un JRE?.
3.2.- Si se utilizan desde un programa java las clases contenidas en el fichero "c:\temp\mysql-connector-java-3.0.8-stable-bin.jar" para cargar un controlador JDBC, ¿cuál debe ser el contenido de la variable CLASSPATH para que las clases sean encontradas por el intérprete de java?.
3.3.- ¿Qué significado tiene el incluir un punto (.) en el contenido de la variable CLASSPATH?.

**4.-** Miembros (atributos y métodos) de clase e instancia en Java:
4.1.- ¿Cuál es su diferencia?.
4.2.- ¿Por qué el método *main* es un método de clase?.
4.3.- ¿Para qué sirve un *iniciador estático*?.

**5.-** Atributos, métodos, referencias, clases y parámetros finales. ¿Qué ocurre cuando aplicamos el modificador *final* en la declaración de estas entidades en Java?.

**6.-** Programación multihilo en Java:

6.1.- ¿De qué dos formas se puede crear un hilo en Java?.

6.2.- ¿Cuáles son los dos niveles de bloqueo de un recurso en Java?.

6.3.- ¿Cómo podemos garantizar el acceso exclusivo a objetos de una clase que no fue diseñada para acceso multihilo?.

**7.-** Gestión de Entrada / Salida en Java:

7.1.- Diferencia entre las clases Reader/Writer e InputStream/OutputStream.

7.2.- ¿Qué es la serialización?.

7.3.- ¿Para qué sirve declarar un atributo con el modificador *transient*?.

**8.-** *Herencia de clases y tratamiento de excepciones*. Si un método de la clase B redefine (no sobrecarga) un método de su superclase A que utiliza *throws*:

8.1.- ¿Qué excepciones puede lanzar el método de la subclase?.

8.2.- ¿Qué ventajas tiene esto de cara al polimorfismo?.

**9.-** *Gestión de Bases de Datos con JDBC*:

9.1.- ¿En qué situaciones es imprescindible utilizar el puente JDBC-ODBC?.

9.2.- ¿Por qué es necesario disponer en Java de la clase *Types*, que define nuevos tipos de datos para las variables Java?.

**10.-** *Gestión de Eventos en Java.* ¿Cuál es la principal diferencia en la forma de gestionar los eventos de usuario mediante los 2 enfoques proporcionados por Java: *modelo heredado* y *modelo de delegación de eventos*?.

En Ourense, a 4 de diciembre de 2003

Fdo.: