



**Programación Avanzada**  
**– Convocatoria de Diciembre de 2001 –**

NOTA

**Examen de Teoría**

Universidade de Vigo

ESCOLA SUPERIOR DE  
ENXEÑERÍA INFORMÁTICA

Apellidos

Nombre

DNI

1.- Distinguir entre: *plataforma java*, *máquina virtual java* y *API de java*. Esquema de la *plataforma java*.

2.- Antes de ejecutar el *byte-code* correspondiente a un programa Java, se verifica para asegurar que ese código cumple con las especificaciones del lenguaje. En concreto ¿qué verificaciones hace el verificador de *byte-code* de java?.

3.- Responder a los siguientes sub-apartados:

3.1.- El siguiente código genera un error, ¿de qué error se trata?. ¿Cuándo surge el problema: en tiempo de compilación o en tiempo de ejecución?

```
byte b = 50;  
b = b * 2;
```

3.2.- ¿Qué valor toman las variables **x** e **y** al finalizar la ejecución del siguiente código?:

```
int x, y;  
x = 42;  
y = ++x;
```

3.3.- ¿Existe algún problema con la siguiente expresión?. En caso afirmativo explicarlo:

```
int done = 0;  
boolean valor;  
  
if (!done)  
    valor = true;  
else  
    valor = false;
```

4.- ¿A qué entidades del lenguaje se les puede aplicar el modificador **abstract**?, y ¿**final**?. Explicar su finalidad.

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

**6.-** ¿Qué es una excepción de Java?. ¿Cuál es la forma general de un bloque de gestión de excepciones?. Explicar cada una de sus partes.

**7.-** ¿Para que se utiliza la herramienta **javadoc**?. ¿Qué tres tipos de comentarios soporta?. Explica su sintaxis general.

**8.-** ¿Para qué se utiliza el fichero *java.policy*?, ¿qué es una *entrada garantizada* y una *entrada de permiso*?.

**9.-** Explica para que sirven las siguientes palabras reservadas de Java: *static*, *this*, *extends*, *implements*, *break*.

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