



Universidade de Vigo

ESCOLA SUPERIOR DE
ENXEÑERÍA INFORMÁTICA

Programación Avanzada – Febrero de 2004 –

Examen de Teoría

NOTA

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 (incluyendo PLAN: []Nuevo, []Viejo) de esta hoja, que será grapada junto con las hojas de respuestas que deberán ir numeradas.
- En el momento de la entrega del examen se deberá firmar una hoja de entrega, que estará a disposición del alumno.

REVISIÓN DE EXÁMENES:

Martes (3 de Febrero) de [9:00-11:00] y [17:30-19:30].

☐ **1.- Tipos de datos y literales en Java:**

1.1.- ¿Qué diferencia existe entre las siguientes declaraciones?:

```
final static double PI1 = 3.1415926539;  
static double PI2 = 3.1415926539;  
final double PI3 = 3.1415926539;
```

1.2.- ¿De qué tipo son los siguientes literales?:

```
4326  
4593L  
2085000000  
15.34  
2.2e+5  
3
```

1.3.- Enumera las conversiones implícitas permitidas en Java.

1.4.- ¿Qué operación realizan los siguientes operadores?: **&&**, **^**, **~**, **>>>**.

☐ **2.- Métodos estáticos en Java:**

2.1.- ¿Cuáles son sus restricciones?.

2.2.- ¿Tiene sentido declarar un método estático y final?, ¿por qué?.

2.3.- ¿En qué se diferencian de los iniciadores estáticos?.

☐ **3.- Clases y métodos abstractos en Java:**

3.1.- ¿Cuál es la finalidad de las clases abstractas de Java?.

3.2.- ¿Puede una clase abstracta no contener ningún método abstracto?.

3.3.- ¿Para que sirve una referencia a una clase abstracta si no podemos hacer un **new()** de esa clase?.

☐ 4.- *Métodos redefinidos en Java:*

4.1.- ¿Cómo se puede acceder a un método de una superclase que ha sido redefinido en una subclase?.

4.2.- Suponiendo que el método `getWidth()` estuviera definido en la clase *X*, ¿qué método se ejecutaría en el siguiente ejemplo: el de la clase *X* o el de la clase *Z*?, ¿por qué?.

```
class Z extends X
{
    // ...

    public float getWidth()
    {
        return width;
    }

    public double calculateArea()
    {
        // ...
        ((X) this).getWidth();
    }
}
```

4.3.- ¿Qué diferencia existe entre redefinir y sobrecargar métodos?.

☐ 5.- *Clases internas en Java:*

5.1.- Enumera los tipos de clases internas que soporta Java.

5.2.- ¿Cómo trata la JVM las clases internas?.

5.3.- ¿Para qué se utilizan las clases internas de Java en el modelo de delegación de eventos?.

☐ 6.- *Gestión de excepciones en Java:*

6.1.- ¿Qué son las excepciones no comprobadas de Java?.

6.2.- ¿Para qué se utilizan las palabras reservadas **throw** y **throws**?.

6.3.- ¿Cuándo se puede introducir la cláusula **finally** en un bloque **try/catch** de manejo de excepciones?, ¿cuál es su misión?.

☐ 7.- *Trabajo con la red:*

7.1.- ¿Cuál es la principal diferencia a nivel funcional (para un programador Java), de utilizar sockets o datagramas para conseguir conectar dos programas a través de la red?.

7.2.- Enumera las 4 clases principales que se utilizan cuando se programan sockets y datagramas. Explica brevemente (1 o 2 líneas) para que sirve cada una de ellas).

7.3.- En la práctica del cliente/servidor de FTP no bloqueante, ¿qué tipo de comunicación se utiliza: sockets o datagramas?, razona el por qué.

☐ 8.- *Seguridad en Java:* Sandbox del JDK 1.0 y 1.2. Esquema y explicación

☐ 9.- *Gestión de entrada/salida en Java:*

9.1.- ¿Cuál es la diferencia que existe entre las clases *FileOutputStream* y *FileWriter* de Java?, ¿en qué paquete se encuentran definidas?.

9.2.- ¿Por qué es necesaria la *serialización* para que funcione RMI (*Invocación Remota de Métodos*)?.

☐ 10.- *Gestión de bases de datos con JDBC:*

10.1.- ¿Qué método de la clase *Statement* se utiliza en Java para la ejecución de sentencias de tipo query?, ¿y para sentencias de tipo update?, ¿qué tipo de dato retorna cada uno de los métodos anteriores?.

10.2.- ¿Qué problema puede existir en Java con los tipos de datos que definen los sistemas gestores de bases de datos?.