



Programación Avanzada
– Septiembre de 2002 –

NOTA

Examen de Teoría

Universidade de Vigo

ESCOLA SUPERIOR DE
ENXEÑERÍA INFORMÁTICA

Apellidos

Nombre

DNI

ACLARACIONES:

- El tiempo para la realización del examen es de 2 horas 30 minutos.
- Se deben cubrir los datos del encabezado (*apellidos, nombre y DNI*) y firmar la parte de atrás 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 PRÁCTICA:

Miércoles (25 de Septiembre) de 10:00 h a 13:30 h

REVISIÓN DE EXÁMENES:

Miércoles (25 de Septiembre) de 16:00 h a 18:00 h

Jueves (26 de Septiembre) de 11:00 h a 13:30 h

☐ 1.- Variables, métodos, referencias y clases finales en Java: ¿qué ocurre cuando se aplica el modificador **final** a cada uno de estos elementos?.

☐ 2.- Paquetes en Java. Si queremos crear nuestro propio paquete de clases: ¿qué pasos se deben seguir hasta que está disponible para su utilización?. ¿Para que podría servir la utilidad **jar** de Java?. ¿Cómo se modificaría la variable CLASSPATH si se utiliza la herramienta **jar** en la elaboración de un paquete de clases?.

☐ 3.- Responder a los siguientes subapartados:

3.1.- Atributos con el mismo nombre: ¿De qué dos formas se podría acceder desde el método *muestraValor()* de la clase **B** al atributo *numElementos* de la clase **A**?

```
class A
{
    public int numElementos = 7;
    // ...
}
class B extends A
{
    protected int numElementos = 3;
    // ...
    public int muestraValor()
    {
        return numElementos;
    }
}
```

3.2.- Redefinición de métodos de la superclase: Suponiendo que el método *muestraValor()* estuviera definido en la clase **A** y fuera redefinido en la clase **B**, ¿a qué clase pertenece el método *muestraValor()* que se ejecuta cuando se llama al método *prueba()*?

//...

```

class B extends A
{
    // ...
    public int muestraValor()
    {
        return numElementos;
    }
    public void prueba()
    {
        // ...
        ((A) this).muestraValor();
        // ...
    }
}

```

3.3.- Conversión de objetos: ¿A qué clase pertenece la variable *x* referenciada en cada una de las expresiones señadas en el método *muestraValor()* de la clase *C*?

```

class A
{
    public int x;
    // ...
}
class B extends A
{
    public int x;
    // ...
}
class C extends B
{
    public int x;
    public void muestraValor()
    {
        x
        this.x
        super.x
        ((B) this).x
        ((A) this).x
    }
}

```

☐ 4.- Herencia en Interfaces.

☐ 5.- Clases internas en Java: Enumerar los diferentes tipos que existen en Java. ¿Cómo trata la JVM a las clases internas?. ¿Qué hace el compilador de Java?.

☐ 6.- Gestión de excepciones en Java: ¿para qué se utilizan las palabras reservadas **throw**, **throws** y **finally**?. Poner un ejemplo de cada una de ellas.

☐ 7.- Programación multihilo: ¿para qué se utiliza la palabra reservada **synchronized**?. Explicarlo con un ejemplo. ¿Tiene sentido declarar un constructor como **synchronized**?. ¿Cuáles son los 3 métodos que se utilizan en Java para la comunicación entre hilos?.

☐ 8.- Gestión de Entrada/Salida: ¿en qué consiste la serialización?. ¿Cómo ha de declararse una clase para que pueda utilizar la serialización?.

☐ 9.- Servlets, JSP y Applets: ¿dónde se ejecutan las páginas JSP, los Servlets y los Applets, en el cliente o en el servidor?. La especificación JSP requiere que todas las implementaciones de JSP soporten salida con buffer, ¿por qué?. Enumerar los 4 tipos de etiquetas que soporta JSP y la finalidad de cada grupo de etiquetas.

☐ 10.- Manejo de eventos: existen dos formas en Java de gestionar eventos (JDK 1.02 y JDK 1.1). Citar el nombre de cada una de ellas y comentar brevemente su esquema de programación.

Fdo.: