



## PROGRAMACIÓN AVANZADA

[DICIEMBRE / FC]  
2005

ESCOLA SUPERIOR DE  
ENXEÑERÍA INFORMÁTICA

NOTA

UNIVERSIDADE  
DE VIGO

APELLIDOS \_\_\_\_\_ NOMBRE \_\_\_\_\_

DNI \_\_\_\_\_ FIRMA \_\_\_\_\_

### IMPORTANTE:

- Cada pregunta tiene una única respuesta correcta.
- Se deberá firmar esta hoja del examen en el momento de la entrega.
- La aportación a la puntuación por el tipo de respuesta es la siguiente:

Correcta	+ 0,4	puntos
No contestada	0	puntos
Incorrecta	- 0,2	puntos
- La revisión de TEORÍA y PRÁCTICAS se llevará a cabo en el siguiente horario:  
21 de diciembre de 2005 (miércoles) de [9:00h - 13:00h] y [16:00h - 19:00h]
- Tiempo de realización: 40 minutos.
- Sólo se corregirán las respuestas que aparezcan marcadas en la tabla que sigue a continuación:

Indicar en la siguiente tabla las respuestas que se consideran correctas al test que se adjunta:

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14	C15	C16	C17	C18	C19	C20	C21	C22	C23	C24	C25
A																									
B																									
C																									
D																									

### 1. ¿Cuál es el comportamiento por defecto del mecanismo de serialización en Java para una clase que implemente la interfaz Serializable?

- A) Se serializan todos los atributos declarados con el modificador `transient`.  
B) Por defecto se serializan todos los atributos excepto los declarados con `transient` o con `static`.  
C) Por defecto no se serializa ningún atributo.  
D) Sólo se serializan los atributos declarados con el modificador `static`.

### 2. Respecto al manejo de eventos en Java, ¿qué afirmación es correcta?

- A) El modelo de delegación de eventos dispone de clases para representar conjuntos de eventos concretos.  
B) Las clases adaptadoras forman parte del modelo heredado de gestión de eventos.  
C) El modelo heredado es más flexible que el modelo de delegación de eventos.  
D) Las clases internas se utilizan en el modelo heredado de gestión de eventos.

### 3. Referente a la gestión de excepciones en Java, ¿cuál de las siguientes afirmaciones es falsa?

- A) El bloque `finally` se ejecuta siempre, incluso si no se produce ninguna excepción.  
B) El bloque `finally` es opcional, pero cada sentencia `try` necesita al menos un bloque `catch` o `finally`.  
C) Si un método redefine a otro método de una superclase que utiliza `throws`, el método de la subclase debe lanzar las mismas excepciones que el método original.  
D) La misma secuencia de código contenida en un bloque `try` puede activar más de un tipo de excepción.

### 4. Hablando de herencia en Java, ¿qué se puede afirmar?

- A) Java soporta herencia múltiple en interfaces y herencia simple en clases.  
B) Una subclase hereda todos los miembros de su superclase, excepto los constructores.  
C) Una subclase no tiene acceso directo a los miembros privados de su superclase.  
D) Todas las afirmaciones anteriores son verdaderas.

### 5. Si se declara una variable en Java con los modificadores `final` y `static`, ¿qué afirmación es correcta?

- A) Se debe asignar a la variable un valor en el momento de la declaración.  
B) No tiene sentido declarar una variables con los modificadores `final` y `static`.  
C) Se crea una constante de clase.  
D) Las afirmaciones A y C son verdaderas.

### 6. ¿Desde dónde se puede acceder a un método declarado sin ningún modificador de acceso en Java?

- A) Desde la misma clase y desde cualquier clase declarada en el mismo paquete.  
B) Desde la misma clase y desde cualquier subclase (declara o no en el mismo paquete).  
C) Desde la misma clase y las subclases declaradas en el mismo paquete.  
D) Todas las afirmaciones anteriores son falsas.

### 7. Considerando el siguiente fragmento de código, ¿qué valores toman las variables `x` e `y`?

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

- A)  $(x=42) \wedge (y=44)$     B)  $(x=43) \wedge (y=44)$     C)  $(x=43) \wedge (y=43)$     D)  $(x=44) \wedge (y=44)$

### 8. Respecto a la asignación “`float pi = 3.14;`”, ¿cuál es la afirmación correcta?

- A) El compilador realiza un *casting* implícito.  
B) Las afirmaciones A y C son correctas  
C) El literal “3.14” es de tipo `double`.  
D) Las afirmaciones A y C son falsas.

### 9. Sobre el desarrollo de aplicaciones distribuidas con Java, ¿qué afirmación es la correcta?

- A) RMI utiliza internamente el mecanismo de la serialización.  
B) El compilador de RMI de java (*rmic*) se aplica sobre el fichero que implementa la interfaz remota.  
C) El fichero compilado correspondiente a la interfaz remota debe estar disponible para el cliente y el servidor.  
D) Todas las afirmaciones anteriores son verdaderas.

### 10. Respecto al acceso a bases de datos desde Java, ¿qué afirmación es la correcta?

- A) El controlador JDBC se carga mediante el método `forName` de la clase `DriverManager`.  
B) El método `execute` de la interfaz `Statement` permite ejecutar sentencias de tipo DDL y DML.  
C) La clase `Types` redefine los tipos básicos existentes en Java para que puedan ser utilizados en el acceso a datos de un SGBD específico.  
D) Todas las afirmaciones anteriores son verdaderas.

### 11. Respecto a la sincronización de hilos en Java, ¿qué afirmación es la correcta?

- A) Un bloque sincronizado permite que varios hilos ejecuten un bloque de código concurrentemente.  
B) Un método declarado como `synchronized` bloquea el acceso a todos los demás métodos de esa clase u objeto.  
C) En Java la sincronización puede hacerse a nivel de objeto y de clase.  
D) En un entorno multihilo, los constructores pueden ser declarados como `synchronized`.

**12. Sobre la programación de *applets* en Java, ¿qué afirmación es la correcta?**

- A) Los *applets* pueden recibir parámetros a través de la etiqueta <PARAM> de HTML.
- B) El método `init` de un *applet* se invoca automáticamente por el navegador después de que se haya ejecutado el método `start`.
- C) Por defecto los *applets* se ejecutan en la JVM sin ninguna restricción (a partir del JDK 1.2).
- D) Las afirmaciones A y C son falsas.

**13. ¿Qué se puede afirmar sobre los *iniciadores estáticos* de Java?**

- A) Se utilizan para inicializar los atributos de instancia de una clase.
- B) Son métodos anónimos que pueden recibir parámetros.
- C) Se ejecutan justo antes que el constructor de la clase.
- D) Todas las afirmaciones anteriores son falsas.

**14. ¿Qué se podría decir acerca de los siguientes fragmentos de código?**

```
int i = 0;
salir:
for (i=0; i<10; i++)
{
    if (i == 5)
        break salir;
}
System.out.println(i);
```

```
int i = 0;
for (i=0; i<10; i++)
{
    if (i == 5)
        continue;
}
System.out.println(i);
```

- A) El resultado de imprimir la variable *i* genera la misma salida en los dos casos.
- B) El código de la izquierda realiza menos iteraciones en el bucle `for` que el código de la derecha.
- C) El código de la derecha imprime un valor de *i* igual a 10.
- D) Las afirmaciones B y C son verdaderas.

**15. Respecto a los tipos de datos básicos, ¿qué afirmación es la correcta?**

- A) Un `char` en Java ocupa 2 bytes.
- B) Todos los tipos de datos básicos ocupan lo mismo en cualquier JVM independientemente del sistema operativo.
- C) Un `long` ocupa lo mismo que un `double`.
- D) Todas las afirmaciones anteriores son correctas.

**16. ¿Qué se podría decir acerca del siguiente fragmento de código?**

```
int num = 0;

if (!num)
    num = 7;

System.out.println(num);
```

- A) El fragmento de código imprime el valor “7”.
- B) En Java, una variable de tipo entero (*num*) no se puede utilizar como condición en una sentencia `if` como la del ejemplo.
- C) El tipo de dato entero en Java no tiene connotaciones booleanas.
- D) Las afirmaciones B y C son verdaderas.

**17. Sobre el modificador *abstract* en Java, ¿qué afirmación es la correcta?**

- A) Un método abstracto no posee cuerpo y deberá ser codificado obligatoriamente si una clase normal hereda de la clase abstracta.
- B) Cuando una clase contiene algún método abstracto, dicha clase deberá obligatoriamente ser definida como abstracta.
- C) Una clase abstracta podría llegar a contener el mismo tipo de miembros que una clase normal.
- D) Todas las afirmaciones anteriores son verdaderas.

**18. Teniendo en cuenta el siguiente fragmento de código, ¿cómo se puede acceder al atributo *numElementos* de la *claseA* desde el método *muestraValor()* de la *claseB*?**

```
class claseA
{
    public int numElementos = 7;
}

class claseB extends claseA
{
    protected int numElementos = 3;

    public int muestraValor()
    {
        ¿acceso a numElementos de claseA?
    }
}
```

- A) No se puede acceder, el atributo ha quedado oculto para la *claseB*.
- B) Se puede acceder utilizando `super.numElementos`.
- C) Se puede acceder utilizando `((claseA) this).numElementos`.
- D) Las afirmaciones B y C son verdaderas.

**19. Sobre la gestión de entrada/salida en Java, ¿qué afirmación es incorrecta?**

- A) Dentro del paquete `java.io` existen dos familias de jerarquías distintas para la E/S de información: las clases que heredan de `InputStream` (para bytes y caracteres) y las clases que heredan de `OutputStream` (para bytes y caracteres).
- B) Dentro de cualquier programa Java existe una clase llamada `System` que contiene tres variables con flujos predefinidos: `in`, `out` y `err`, que pueden ser utilizados sin tener una referencia a un objeto `System` específico.
- C) Un directorio en Java se trata igual que un archivo con una propiedad adicional: una lista de nombres de archivo que se puede examinar utilizando el método `list`.
- D) La interfaz `FilenameFilter` se utiliza para limitar el número de archivos devueltos por el método `list` de la clase `File`.

**20. Respecto a la programación de páginas JSP, ¿qué afirmación es la correcta?**

- A) JSP proporciona cuatro tipos principales de etiquetas: *directivas*, *elementos de script*, *comentarios* y *acciones*.
- B) Cuando se ejecuta una página JSP el servidor http lleva a cabo una compilación interna de la página JSP para generar un *servlet* equivalente.
- C) Las páginas JSP necesitan que el servidor http que les da soporte implemente obligatoriamente salida con buffer.
- D) Todas las afirmaciones anteriores son correctas.

**21. ¿Cuáles son los componentes de la plataforma Java?**

- A) La JVM (*Java Virtual Machine*) y el API (*Application Program Interface*) Java.
- B) La CPU, la memoria y la Entrada/Salida de la JVM.
- C) Los registros PC (*Program Counter*), `vars`, `optop` y `frame`.
- D) Todas las afirmaciones anteriores son falsas.

**22. Respecto a la instanciación de tipos básicos y clases en Java, ¿qué afirmación es la correcta?**

- A) La instanciación de tipos básicos se realiza en tiempo de compilación y la instanciación de clases en tiempo de ejecución.
- B) Ambas se realizan en tiempo de compilación.
- C) Ambas se realizan en tiempo de ejecución.
- D) Todas las afirmaciones anteriores son falsas.

**23. Respecto a las interfaces en Java, ¿qué afirmación es la correcta?**

- A) Las interfaces pueden definir constantes.
- B) Los nombres de las interfaces se pueden utilizar como un nuevo tipo a la hora de declarar referencias.
- C) Las interfaces se pueden utilizar como valor de retorno de un método.
- D) Todas las afirmaciones anteriores son verdaderas.

**24. Respecto a la herramienta *javadoc* de Java, ¿qué afirmación es la correcta?**

- A) Los *doclets* son programas de Java que especifican el contenido y el formato de la salida de *javadoc*.
- B) Las etiquetas *javadoc* se insertan en los comentarios *doc*.
- C) Los comentarios *doc* soportan la incrustación de código HTML estándar.
- D) Todas las afirmaciones anteriores son verdaderas.

**25. Respecto a la conversión de objetos en Java, ¿qué afirmación es la correcta?**

- A) Java no permite conversiones implícitas de una subclase a su superclase.
- B) La utilización de una referencia de una superclase para acceder a un objeto de la subclase puede hacerse de modo implícito.
- C) Para realizar una operación de casting entre objetos de dos clases distintas, Java exige que una debe ser subclase de la otra.
- D) Todas las afirmaciones anteriores son falsas.