



UNIVERSIDADE
DE VIGO

NOTA

ESCOLA SUPERIOR DE
ENXEÑERÍA INFORMÁTICA

PROGRAMACIÓN AVANZADA [EXAMEN DE TEORÍA]

3 de febrero de 2005

CONVOCATORIA: FEBRERO DE 2005

(MODELO B)

APELLIDOS _____ NOMBRE _____

DNI _____

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
- Tiempo de realización: 40 minutos.
- La revisión de teoría y prácticas se llevará a cabo en el siguiente horario:

REVISIÓN DE TEORÍA Y PRÁCTICAS:
7 de febrero de 2005 (lunes) de [8:30h - 13:30h]

- 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																									

FDO.: _____

1. 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.

2. 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.

3. ¿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.

4. Sobre los niveles de protección de una clase en Java, ¿qué afirmación es la correcta?

- A) Una clase con nivel de protección de paquete está disponible para sus subpaquetes.
- B) Sólo existen dos niveles de protección de una clase: de paquete (*package*) y público (*public*).
- C) Existen los niveles de protección: *public*, *protected*, *private* y *package*.
- D) Todas las afirmaciones anteriores son falsas.

5. ¿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.

6. Sobre el modificador *final* en Java, ¿qué afirmación es incorrecta?

- A) Cuando se declara una variable utilizando *final*, se transforma en una constante.
- B) Un método *final* puede ser redefinido.
- C) Se puede utilizar el modificador *final* en la declaración de referencias.
- D) Una clase declarada con el modificador *final* puede ser instanciada.

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

- A) De una clase declarada con el modificador *abstract* no se pueden tener instancias.
- B) El modificador *abstract* puede ser aplicado a los atributos de una clase.
- C) Todos los métodos de una clase declarada con el modificador *abstract* deben ser abstractos.
- D) No se pueden utilizar referencias a clases abstractas.

8. Hablando de herencia en Java, ¿qué afirmación es la correcta?

- A) Java sólo dispone de herencia simple en clases e interfaces.
- B) Las interfaces en Java sólo pueden ser implementadas, no existe herencia de interfaces.
- C) La herencia de interfaces en Java puede ser múltiple.
- D) Una clase en Java puede heredar de varias clases pero implementar una única interfaz.

9. 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.

10. Respecto a las excepciones en Java, ¿qué afirmación es la correcta?

- A) Todas las excepciones que define Java pueden surgir únicamente en tiempo de ejecución.
- B) Las excepciones no comprobadas en Java requieren la utilización de *throws* en la declaración del método.
- C) Se puede utilizar *throw* para que un método lance excepciones definidas por el programador.
- D) No es posible utilizar una clase de excepción como clase base de la cual heredar para crear una nueva clase.

11. En cuanto al bloque *finally* en Java, ¿qué afirmación es la correcta?

- A) El bloque *finally* siempre debe ser el último, situándose después de los bloques *try* y *catch*.
- B) Pueden existir varios bloques *finally* para un bloque *try*.
- C) Si se utiliza el bloque *catch*, no se puede utilizar el bloque *finally*.
- D) Todas las afirmaciones anteriores son falsas.

12. Respecto al fichero *java.policy*, ¿qué afirmación es la correcta?

- A) Las entradas de permiso permiten especificar entradas garantizadas.
- B) Las entradas de permiso pueden incorporar las cláusulas *SignedBy* y *CodeBase* sin importar su orden.
- C) Una entrada garantizada permite especificar un permiso sobre un recurso utilizando una lista de acciones.
- D) Todas las afirmaciones anteriores son falsas.

13. Sobre los ficheros *.jar* de Java, ¿qué afirmación es la correcta?

- A) Son ficheros de clases Java comprimidos mediante la herramienta *jarsigner*.
- B) Para que sus clases estén disponibles debe aparecer su ruta completa (incluyendo el nombre del fichero *.jar*) en la variable *PATH*.
- C) Es posible firmar digitalmente un fichero *.jar* por varias identidades digitales.
- D) Las afirmaciones A y B son verdaderas.

14. 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.

15. 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.

16. Respecto al manejo de eventos en Java, ¿qué afirmación es la correcta?

- A) Actualmente se puede utilizar tanto el *modelo de delegación de eventos* como el *modelo heredado*.
- B) En el *modelo heredado* de manejo de eventos sólo existe una clase para representar a todos los eventos de Java.
- C) El modelo de delegación de eventos se ha quedado *deprecated*.
- D) Las afirmaciones A y B son verdaderas.

17. Sobre la programación multihilo en Java, ¿qué afirmación es la correcta?

- A) El hilo principal puede terminar y dejar a sus hilos hijos en ejecución.
- B) Cuando un hilo sale de los estados: *en espera*, *dormido*, *suspendido* o *bloqueado* siempre pasa por el estado *listo para ejecutarse* antes de volver a obtener la CPU.
- C) El método *isAlive* permite conocer si un hilo está ocupando la CPU en un momento determinado.
- D) Las prioridades de los hilos no tienen ningún efecto sobre el tiempo que éstos pasan ejecutándose en la CPU.

18. 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*.

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

- A) La interfaz *FilenameFilter* se utiliza para limitar el número de archivos devueltos por el método *list* de la clase *File*.
- B) Las clases que derivan de *InputStream* permiten realizar entrada de caracteres.
- C) La clase *File* no deriva de ninguna de la siguiente clases: *InputStream*, *OutputStream*, *Reader*, *Writer*.
- D) La clase *RandomAccessFile* no deriva de *InputStream* ni de *OutputStream*.

20. Respecto al mecanismo de serialización en Java, ¿qué afirmación es la correcta?

- A) Para que una clase pueda ser serializada debe implementar la interfaz *Serializable* y codificar los métodos *writeObject* y *readObject*.
- B) Los atributos declarados como *transient* o *final* no son serializados con el resto del objeto.
- C) Los métodos *writeObject* y *readObject* permiten escribir directamente un objeto a disco.
- D) Todas las afirmaciones anteriores son falsas.

21. Sobre la programación en la red, ¿qué afirmación es la correcta?

- A) La clase *Socket* se utiliza para crear conexiones basadas en flujo, punto a punto, bidireccionables y fiables.
- B) Mediante la clase *DatagramPacket* se envían objetos de la clase *DatagramSocket*.
- C) Mediante un objeto de la clase *Socket* únicamente se pueden enviar y recibir bytes, no caracteres.
- D) Todas las afirmaciones anteriores son verdaderas.

22. 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 clase *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.

23. 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.

24. Respecto a la programación de *servlets* en Java, ¿qué afirmación es incorrecta?

- A) Para la ejecución de *servlets* se necesita un servidor web que de soporte esta tecnología.
- B) Para la compilación de *servlets* se necesita un API a mayores que no viene incluido en la distribución estándar del JDK.
- C) La herramienta *javadoc* no puede ser utilizada para generar documentación técnica de los *servlets*.
- D) Los *servlets* no poseen interfaz gráfica y se ejecutan en el servidor.

25. Sobre la programación de páginas JSP, ¿qué afirmación es la correcta?

- A) Las páginas JSP se suben sin compilar al servidor web.
- B) Los scripts CGI utilizan un mecanismo de ejecución claramente diferente a las páginas JSP.
- C) El conjunto de etiquetas propias de JSP son: *directivas*, *elementos de script* y *acciones*.
- D) Las afirmaciones A y B son verdaderas.