



PROGRAMACIÓN AVANZADA

[FEBRERO]

2007

ESCOLA SUPERIOR DE
ENXEÑERÍA INFORMÁTICA

NOTA

UNIVERSIDADE
DE VIGO

APELLIDOS _____ NOMBRE _____

DNI _____ FIRMA _____

IMPORTANTE:

- Cada pregunta tiene una única respuesta correcta.
- Se deberá firmar la hoja del examen y una hoja de control de firmas 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:

1 de febrero de 2007 (jueves)	de [16:00h - 19:00h]
2 de febrero de 2007 (viernes)	de [09:00h - 13:00h]
- 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. Características del lenguaje Java, ¿qué afirmación es la correcta?

- A) Java no dispone de macros (`#define`) pero permite declarar registros mediante la palabra reservada `struct`.
- B) El `byte-code` generado con un JDK estándar se puede interpretar en cualquier plataforma sin importar en modo alguno la máquina donde fue generado.
- C) Los ficheros de `byte-code` no se pueden decompilar.
- D) Las afirmaciones A y B son verdaderas.

2. Estructura del lenguaje Java, ¿qué afirmación es la incorrecta?

- A) Java no proporciona mecanismos de sobrecarga de operadores aritméticos.
- B) Los operadores `+` y `+=` son sobrecargados automáticamente para concatenar cadenas o para convertir otros tipos de datos a cadenas.
- C) Al contrario que C++, Java trata a los arrays como objetos reales.
- D) Java permite la utilización de aritmética de punteros con referencias de tipo `const`.

3. Componentes de la plataforma Java, ¿qué afirmación es la correcta?

- A) La plataforma Java consta de dos partes: la máquina virtual Java y el API Java.
- B) Aunque cada plataforma implementa su propia máquina virtual, no todas disponen del mismo API.
- C) La máquina virtual Java se estructura en tres componentes básicos: CPU, memoria y entrada/salida.
- D) Todas las afirmaciones anteriores son verdaderas.

4. Tipos de datos en Java, ¿qué afirmación es la correcta?

- A) El tipo de dato `int` en Java ocupa 32 bits de longitud en cualquier implementación de la máquina virtual.
- B) El tipo de dato `short` en Java sólo puede almacenar números positivos (≥ 0).
- C) El tipo de dato `char` en Java ocupa un byte.
- D) El tipo de dato `float` en Java permite representar un dato en como flotante de 64 bits en el formato IEEE 754.

5. Palabras reservadas del lenguaje Java, ¿qué afirmación es la correcta?

- A) `final`, `continue`, `extends` y `endif` son palabras reservadas de Java.
- B) `default`, `break`, `import` y `package` son palabras reservadas de Java.
- C) `super`, `finally`, `implemented` e `inherits` son palabras reservadas de Java.
- D) Todas las afirmaciones anteriores son falsas.

6. Conversión entre tipos de datos en Java, ¿qué afirmación es la correcta?

- A) La conversión de un `double` a un `float` en Java es una conversión implícita.
- B) La conversión de un `char` a un `int` en Java es una conversión implícita.
- C) La conversión de un `boolean` a un `int` en Java es una conversión implícita.
- D) En Java todas las conversiones deben hacerse de forma explícita.

7. Operadores en Java, ¿qué afirmación es la correcta?

- A) El resultado de una operación de relación (`<`, `>`, `<=`, `>=`, `!=`, `==`) es un valor entero que puede ser utilizado como condición en una sentencia `if`.
- B) Java no dispone de operadores de cortocircuito al estilo de Pascal.
- C) En Java existen operadores unitarios como por ejemplo el operador de complemento a 1 (`~`).
- D) El operador `new` de Java tiene más prioridad que el operador `++`.

8. Sentencias de control en Java, ¿qué afirmación es la incorrecta?

- A) La sentencia `switch` puede incluir cualquier número de cláusulas `case`.
- B) La cláusula `default` de la sentencia `switch` puede estar en cualquier parte del bloque `switch` y no necesariamente al final.
- C) El compilador de Java detecta los bucles `for` infinitos y genera un error en tiempo de compilación.
- D) Se pueden anidar sentencias `switch` unas dentro de otras.

9. Acerca del método `main` de la clase principal de un programa Java, ¿qué afirmación es la correcta?

- A) El método `main` es un método estático.
- B) El método `main` devuelve un `int`.
- C) Desde el método `main` se puede acceder a las variables miembro de los objetos del programa.
- D) Las afirmaciones A y C son verdaderas.

10. Sobre los paquetes en Java, ¿qué afirmación es la incorrecta?

- A) Las clases de usuario que se guardan en un fichero cuando se escribe un programa, pertenecen automáticamente al paquete `java.lang`.
- B) Todas las clases del paquete `java.lang` son importadas de manera automática por todos los programas Java.
- C) Todas las clases de usuario definidas en el mismo fichero deberán obligatoriamente pertenecer al mismo paquete.
- D) Un paquete puede contener subpaquetes.

11. Sobre los métodos declarados como `static` en Java, ¿qué afirmación es la correcta?

- A) Sólo pueden llamar a otros métodos `static`.
- B) Sólo pueden acceder a atributos declarados como `static`.
- C) No pueden hacer uso de `this` ni `super`.
- D) Todas las afirmaciones anteriores son verdaderas.

12. Sobre el control de acceso a los miembros de una clase, ¿qué afirmación es la incorrecta?

- A) Un miembro declarado sin modificadores de acceso puede ser accedido por cualquier clase perteneciente al mismo paquete y subpaquetes.
- B) Un miembro declarado con el modificador `protected` puede ser accedido desde una subclase de diferente paquete.
- C) Un miembro declarado como `private` sólo puede ser accedido desde los métodos de su propia clase.
- D) Un miembro declarado como `public` puede ser accedido desde cualquier lugar.

13. Variables, métodos y clases finales en Java, ¿qué afirmación es la incorrecta?

- A) Cuando se declara una variable como `final`, se transforma en una constante.
- B) Declarar una referencia `final` a un objeto supone que esa referencia sólo pueda utilizarse para referenciar a ese objeto.
- C) Un método declarado como `final` no puede ser redefinido en ninguna subclase.
- D) Cuando una clase se declara como `final`, se impide que de esa clase se puedan crear instancias con el operador `new`.

14. La variable `CLASSPATH`, ¿qué afirmación es la incorrecta?

- A) Se utiliza por el compilador y el intérprete de Java para localizar las clases necesarias en tiempo de compilación y ejecución.
- B) Es una variable de entorno del Sistema Operativo.
- C) Existe en Windows y Linux y tiene la misma función.
- D) El directorio actual siempre está incluido por defecto en la variable `CLASSPATH`.

15. Sobre la práctica `SERVICE ERROR NOTIFIER`, ¿qué afirmación es la incorrecta?

- A) El servidor de SERENO (sSERENO) debe escribir las entradas en el fichero de logs mediante RMI.
- B) El fichero de logs debe registrar los comandos ejecutados por cada usuario.
- C) El controlador JDBC debe ser cargado por el servidor de autenticación y logs (sAUTLOG).
- D) El cliente de SERENO (cSERENO) no utiliza RMI para realizar llevar a cabo la autenticación de usuarios.

16. Variables, métodos y clases abstractas en Java, ¿qué afirmación es la incorrecta?

- A) Una clase abstracta puede contener el mismo tipo de miembros que una clase normal y además puede contener métodos abstractos.
- B) Cuando una clase contiene algún método abstracto, debe ser definida obligatoriamente como abstracta.
- C) No se puede aplicar el modificador `abstract` a la definición de variables de tipos básicos de Java.
- D) Los métodos abstractos pueden ser codificados en la misma clase abstracta en la que residen.

17. Métodos redefinidos en Java, ¿qué afirmación es la correcta?

- A) Redefinir un método heredado significa volver a codificarlo en la subclase con el mismo nombre, la misma lista de parámetros y el mismo tipo del valor de retorno que tenía en la superclase.
- B) Cuando en una subclase se redefine un método de una superclase, se oculta el método de la superclase, pero no las sobrecargas que existan del mismo en dicha superclase.
- C) No se puede redefinir un método en una subclase y hacer que su control de acceso sea más restrictivo que el original.
- D) Todas las afirmaciones anteriores son verdaderas.

18. Conversión de objetos en Java, ¿qué afirmación es la incorrecta?

- A) Para la conversión de objetos de distintas clases, Java exige que dichas clases estén relacionadas por herencia.
- B) El compilador realiza una conversión implícita de una subclase a una superclase siempre que se necesite.
- C) No se puede acceder a las variables exclusivas de la subclase a través de una referencia de la superclase.
- D) Todas las afirmaciones anteriores son falsas.

19. Sobre clases abstractas e interfaces en Java, ¿qué afirmación es la incorrecta?

- A) Una clase no puede heredar de dos clases abstractas, pero sí puede heredar una clase abstracta e implementar una o más interfaces.
- B) Las interfaces tiene una jerarquía propia, independiente y más flexible que la de las clases, ya que tienen permitida la herencia múltiple.
- C) El nombre de una interfaz no se pueden utilizar como una referencia.
- D) Las constantes definidas en una interfaz se pueden utilizar en cualquier clase (aunque no implemente la interfaz) precediéndolas del nombre de la interfaz.

20. Clases internas en Java, ¿qué afirmación es la incorrecta?

- A) Se pueden declarar clases e interfaces internas estáticas.
- B) En Java existen clases internas miembro, pero no existen interfaces internas miembro.
- C) Las clases internas locales se pueden crear en un iniciador `static`.
- D) Todas las afirmaciones anteriores son falsas.

21. Gestión de excepciones en Java, ¿qué afirmación es la incorrecta?

- A) Todas las clases que implementan excepciones en Java heredan (directa o indirectamente) de la clase `Exception`.
- B) Las clases derivadas de `Exception` pueden pertenecer a distintos paquetes de Java.
- C) La misma secuencia de código contenida en una sentencia `try` puede activar más de un tipo de excepción.
- D) La cláusula `throws` en la declaración de un método permite listar los tipos de excepciones que ese método puede lanzar.

22. Seguridad en Java, ¿qué afirmación es la correcta?

- A) No es posible especificar un fichero de seguridad (*java.policy*) específico para ejecutar un determinado programa utilizando la línea de comandos.
- B) En el modelo de seguridad del JDK 1.2 no existe la noción de código seguro.
- C) La cláusula `CodeBase` de una entrada garantizada contiene una lista separada por comas con los firmantes del código a los que se aplica dicha entrada.
- D) La herramienta *jar* permite firmar digitalmente un fichero de clases comprimido.

23. Generación de archivos de documentación en Java, ¿qué afirmación es la incorrecta?

- A) El programa *javadoc* reconoce tipos especiales de comentarios que se insertan en el código fuente generado.
- B) La etiqueta `@throws` es similar a la etiqueta `@exception`.
- C) Los *doclets* son programas Java que especifican el contenido y el formato de la salida de *javadoc*.
- D) La etiqueta `@serial` se utiliza para indicar la versión del programa o método sobre el que se aplica.

24. Manejo de eventos en Java, ¿qué afirmación es la correcta?

- A) En el modelo de delegación de eventos todos los eventos se representan mediante la clase `Event`.
- B) Las clases adaptadoras pertenecen al modelo de delegación de eventos.
- C) El modelo de delegación de eventos es anterior al modelo heredado.
- D) Todas las afirmaciones anteriores son falsas.

25. Programación multihilo en Java, ¿qué afirmación es la correcta?

- A) El método `isAlive` devuelve `true` cuando el hilo sobre el que se invoca está en estado suspendido.
- B) El método `join` permite que dos hilos hijo terminen al mismo tiempo.
- C) En Java existen hilos que no pertenecen a ningún grupo de hilos.
- D) El modificador `synchronized` aplicado sobre un método, posibilita que dos hilos accedan concurrentemente a dicho método.