



PROGRAMACIÓN AVANZADA
[FEBRERO]
2008
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:

12 de febrero de 2008	(martes)	de	[16:00h - 19:00h]
13 de febrero de 2008	(miércoles)	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:

	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. ¿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.

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 CPU definida en la máquina virtual contiene 5 registros de 4 bytes: PC, vars, optop, frame, stack.
B) Cada plataforma implementa su propia máquina virtual, que interpreta y ejecuta los ficheros *byte-code*.
C) Cada plataforma emplea la misma definición de máquina virtual y todas disponen del mismo API.
D) El área de memoria direccionable por Java es de 4Gb, ya que los registros son de 64 bits.

4. Sentencia *import* en Java, ¿qué afirmación es la correcta?

- A) Mediante una única sentencia *import* se pueden importar clases pertenecientes a distintos paquetes.
B) En un programa Java pueden aparecer cualquier número de sentencias *import*.
C) Al importar un paquete en Java se importan automáticamente todos los subpaquetes que este contenga.
D) Las afirmaciones B y C 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 variable con los modificadores *final* y *static*.
C) Se crea una constante de clase.
D) Las afirmaciones A y C son verdaderas.

6. ¿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.

7. Hablando de constructores en Java, ¿qué afirmación es la correcta?

- A) Los constructores deben declararse siempre haciendo uso del modificador PUBLIC.
B) La llamada a un constructor sólo puede realizarse desde dentro de otro constructor.
C) Un constructor puede ser invocado directamente a través de su nombre como cualquier otro método.
D) Los constructores no pueden ser sobrecargados.

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

- A) Las clases internas miembro pueden utilizar los modificadores de acceso: PUBLIC, PRIVATE, PROTECTED y *de paquete*.
B) Un objeto de una clase interna local sólo puede existir en relación con un objeto de la clase contenedora, que debe existir previamente.
C) Las clases anónimas deben definir al menos un constructor.
D) Las clases internas STATIC pueden utilizar los miembros STATIC de la clase contenedora.

10. 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.

11. En cuanto a las clases internas de Java, ¿qué afirmación es correcta?

- A) La JVM no maneja, a nivel de intérprete, el concepto de clase interna.
- B) Existen clases internas `static`, clases internas `miembro`, clases internas anónimas y clases internas `final`.
- C) Se pueden crear varias instancias de una clase anónima en el momento de su definición.
- D) Las clases internas `final`, sólo pueden ser declaradas en clases `final`.

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. Subclases en Java, ¿qué afirmación es la correcta?

- A) Si en la definición de una clase no se especifica la cláusula `EXTENDS`, se entiende que la superclase de la clase que está siendo definida es la clase `SYSTEM`.
- B) Una subclase no hereda los constructores de su superclase.
- C) Cuando en una subclase se redefine un método de una superclase, se oculta el método de la superclase y todas sus sobrecargas.
- D) No es posible acceder a un método de una superclase que ha sido redefinido en la subclase.

14. Clases e interfaces internas en Java, ¿qué afirmación es la correcta?

- A) Las clases internas estáticas pueden acceder a todos los atributos de la clase contenedora.
- B) Las clases internas miembro puede ser declaradas utilizando los modificadores de acceso `private` y `protected`.
- C) Las clases internas locales sólo tienen acceso a las variables y parámetros pertenecientes al método donde han sido declaradas, no pudiendo acceder a los atributos de la clase contenedora.
- D) Todas las afirmaciones anteriores son falsas.

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. Para ejecutar una aplicación Java *standalone* en nuestro PC, ¿qué necesitamos tener instalado?

- A) Java Runtime Environment (JRE).
- B) Java Development Kit (JDK).
- C) JDK + JRE.
- D) Cualquiera de las opciones anteriores permite ejecutar una aplicación *standalone* Java.

17. Sobre la generación de archivos de documentación de forma automática en Java, ¿qué afirmación es la correcta?

- A) Existen tres tipos de comentarios especiales reconocidos por la utilidad `JAVADOC` del JDK: *comentarios doc*, *etiquetas javadoc* y *etiquetas HTML*.
- B) La etiqueta `DEPRECATED` representa la única ocasión en la que el contenido de un comentario afecta al comportamiento del compilador.
- C) Los `DOCLETS` son programas Java que especifican el contenido y el formato de la salida de `JAVADOC`.
- 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. Respecto a la utilización de los protocolos TCP/IP y UDP, ¿qué afirmación es la correcta?

- A) Mediante la clase `DatagramSocket` se pueden enviar objetos de tipo `DatagramPacket`.
- B) Cuando se crea un objeto `Socket`, implícitamente se establece una conexión entre el cliente y el servidor.
- C) El constructor de la clase `ServerSocket` permite especificar la IP sobre la que escuchará el objeto creado.
- D) Todas las afirmaciones anteriores son verdaderas.

21. En programación multihilo de Java, ¿qué afirmación es falsa?

- A) Un hilo en Java se crea heredando de la clase `Thread` o implementando la interfaz `Runnable`.
- B) Un hilo en Java termina cuando finaliza su método `run`.
- C) La prioridad de los hilos puede utilizarse para establecer el orden de ejecución de estos en la CPU.
- D) La sincronización fuerza a que la ejecución de los hilos sobre un objeto sea mutuamente exclusiva.

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. Gestión de Entrada/Salida, ¿qué afirmación es la incorrecta?

- A) La clase `BufferedInputStream` permite encapsular cualquier `InputStream` en un flujo con búfer.
- B) La clase `PushbackInputStream` proporciona el método `unread` que permite devolver al flujo de entrada un carácter especificado.
- C) La *serialización* es un proceso por el que un objeto cualquiera se puede convertir en una secuencia de caracteres con la que más tarde se podrá reconstruir dicho objeto manteniendo el valor de sus variables.
- D) La clase `STREAMTokenizer` se utiliza para dividir el flujo de entrada en símbolos (*tokens*) que están delimitados por conjuntos de caracteres.

24. Sobre `CLASSPATH` y `PATH`, ¿qué afirmación es la correcta?

- A) Ambas son variables de entorno del Sistema Operativo.
- B) Para poder cargar el controlador JDBC desde una aplicación Java, se debe establecer su ruta correctamente en la variable `PATH`.
- C) El JRE (entorno de ejecución de Java) a diferencia del JDK (kit de desarrollo de Java) no hace uso de la variable `CLASSPATH`.
- D) Todas las afirmaciones anteriores son falsas.

25. 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 `++`.