



UNIVERSIDADE
DE VIGO

NOTA

ESCOLA SUPERIOR DE
ENXEÑERÍA INFORMÁTICA

PROGRAMACIÓN AVANZADA [EXAMEN DE TEORÍA]

18 de diciembre de 2004

CONVOCATORIAS: NOVIEMBRE / FIN DE CARRERA 2005

(MODELO A)

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.

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. ¿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. 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.

3. 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) ^ (y=44)
- B) (x=43) ^ (y=44)
- C) (x=43) ^ (y=43)
- D) (x=44) ^ (y=44)

4. ¿Qué restricción se aplica a los métodos declarados en Java con el modificador static?

- A) Sólo pueden acceder a métodos declarados como *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.

5. ¿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.

6. 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.

7. Si se declara una clase en Java con los modificadores abstract y final, ¿qué afirmación es correcta?

- A) La clase contiene métodos finales.
- B) La clase contiene métodos abstractos.
- C) No tiene sentido declarar una clase con los modificadores *abstract* y *final*.
- D) Todas las afirmaciones anteriores son falsas.

8. Sobre los destructores en Java, ¿qué afirmación es correcta?

- A) Java no dispone de destructores, la destrucción de objetos se lleva a cabo por el recolector de basura.
- B) Los destructores en Java se codifican redefiniendo el método *finalize*.
- C) Se pueden codificar varios destructores por clase.
- D) El método *finalize* no puede ser llamado explícitamente.

9. 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.

10. Respecto a interfaces en Java, ¿qué se puede afirmar?

- A) Las interfaces tienen una jerarquía propia, independiente y más flexible que la de las clases.
- B) No pueden definir constantes.
- C) Las interfaces pueden codificar métodos declarados con el modificador `final`.
- D) Las afirmaciones A y C son verdaderas.

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. 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.

13. ¿Qué tipos de comentarios permite la herramienta `javadoc` de Java?

- A) Comentarios `doc`.
- B) Etiquetas `javadoc` y etiquetas HTML.
- C) Comentarios `doc` y etiquetas `javadoc`.
- D) Las afirmaciones A y B son verdaderas.

14. ¿Qué se puede afirmar sobre el modelo de seguridad del JDK 1.2 de Java?

- A) Es menos flexible que los modelos anteriores a la hora de especificar permisos.
- B) No existe la noción de código en el que por defecto se confía (*trusted code*).
- C) No dispone de *sandbox*.
- D) No se puede discriminar entre código firmado digitalmente y lugar de procedencia del código a la hora de conceder permisos.

15. En el fichero `java.policy` ...

- A) Las entradas garantizadas conceden un permiso sobre un determinado recurso con su correspondiente lista de acciones.
- B) Las entradas garantizadas contienen entradas de permiso.
- C) En una entrada garantizada, las cláusulas `signedBy` y `CodeBase` son opcionales, pero al menos una de ellas debe aparecer.
- D) Todas las afirmaciones anteriores son falsas.

16. Para la invocación remota de métodos (RMI) Java, ¿qué ficheros necesita tener el cliente para poder compilar correctamente?

- A) El fragmento adaptador.
- B) El esqueleto y el fichero de bytecodes de la clase remota.
- C) El fragmento adaptador y el esqueleto.
- D) El fragmento adaptador y el fichero de bytecodes de la interfaz remota.

17. ¿Qué se puede afirmar acerca de los servlets de Java?

- A) Los servlets pueden invocar a métodos de otros servlets.
- B) Son independientes del servidor utilizado y de su sistema operativo.
- C) Son aplicaciones que se ejecutan dentro de un servidor web, pero no tienen interfaz gráfica.
- D) Todas las afirmaciones anteriores son verdaderas.

18. Referente a la utilización de páginas JSP de Java, ¿qué se puede afirmar?

- A) Las directivas de JSP permiten generar salidas visibles para el cliente.
- B) Las páginas JSP dan soporte para el uso de componentes reutilizables (*JavaBeans*).
- C) Para que una página JSP esté disponible, el fichero compilado deben subirse previamente al servidor web.
- D) Todas las afirmaciones anteriores son verdaderas.

19. 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.

20. En los applets de Java, ¿qué métodos son invocados automáticamente por el navegador?

- A) El método `paint`.
- B) Los métodos `init` y `destroy`.
- C) Los métodos `start` y `stop`.
- 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 a la utilización de Streams en Java, ¿qué afirmación es correcta?

- A) El método `flush` se utiliza en clases derivadas de `OutputStream`.
- B) Las clases que derivan de `Reader` y `Writer` permiten realizar entrada/salida de bytes en Java.
- C) Los métodos `mark` y `reset` se utilizan en clases derivadas de `OutputStream`.
- D) Todas las afirmaciones anteriores son falsas.

23. ¿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` y `static`.
- C) Por defecto no se serializa ningún atributo.
- D) Sólo se serializan los atributos declarados con el modificador `static`.

24. Cuando se accede a una base de datos desde Java utilizando JDBC ...

- A) El controlador JDBC debe ser cargado antes de establecer la conexión con la base de datos.
- B) Se deben utilizar obligatoriamente los tipos de datos definidos por la clase `Types` de Java.
- C) Las afirmaciones A y D son verdaderas.
- D) Es necesario que el controlador JDBC esté disponible en la variable `CLASSPATH`.

25. Referente al paquete `java.net` que permite trabajar con la red desde Java, ¿qué afirmación es falsa?

- A) Las clases `Socket` y `ServerSocket` se utilizan para crear conexiones basadas en flujo, punto a punto, bidireccionales y fiables entre nodos de la red.
- B) Java implementa los datagramas sobre el protocolo UDP utilizando las clases `DatagramPacket` y `DatagramSocket`.
- C) El método `accept` de la clase `ServerSocket` es bloqueante.
- D) El método `receive` de la clase `DatagramSocket` no es bloqueante.