



PROGRAMACIÓN AVANZADA

[SEPTIEMBRE DE 2005]

ESCOLA SUPERIOR DE
ENXEÑERÍA INFORMÁTICA

UNIVERSIDADE
DE VIGO

NOTA

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 septiembre de 2005 (miércoles) de [9:00h - 13:00h] y [16:00h - 19:00h]
- Tiempo de realización: 30 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. ¿Cuáles son los componentes básicos de la plataforma Java?

- A) La JVM y el API Java.
- B) CPU, memoria y entrada/salida.
- C) Java Base Platform y Embedded Java Platform.
- D) Todas las afirmaciones anteriores son falsas.

2. ¿Qué se podría decir acerca de la siguiente declaración?

```
final static double Pi = 3.141592654;
```

- A) Se está declarando la variable *Pi* con una única copia para todos los objetos.
- B) Se está declarando la constante *Pi* con una copia para cada objeto.
- C) No se pueden utilizar conjuntamente los modificadores *final* y *static*.
- D) Todas las afirmaciones anteriores son falsas.

3. ¿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*.
- C) El tipo de dato entero en Java no tiene connotaciones booleanas.
- D) Las afirmaciones B y C son verdaderas.

4. ¿Qué se puede afirmar sobre los constructores en Java?

- A) Los constructores no se heredan.
- B) Los constructores siempre devuelven el valor *void*.
- C) Los constructores siempre deben ser declarados con el modificador *public*.
- D) Las afirmaciones A y C son falsas.

5. ¿Qué se puede afirmar sobre los destructores en Java?

- A) En una clase se pueden definir varios destructores.
- B) Cuando en una clase no se especifica un destructor, el compilador proporciona uno a través de la clase *Object*.
- C) El destructor de una clase no puede ser invocado directamente por el programador.
- D) Todas las afirmaciones anteriores son verdaderas.

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

- A) Un paquete puede contener a otros subpaquetes y así sucesivamente.
- B) Cuando el compilador de Java necesita localizar una clase de un determinado paquete, busca primero en el directorio donde se encuentra el fichero que se está compilando.
- C) Una clase con nivel de protección de paquete no está disponible para otros paquetes, ni siquiera para los subpaquetes.
- D) El programador de Java puede crear sus propios paquetes de clases utilizando la palabra reservada *package*.

7. 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.

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

- A) Una subclase hereda todos los miembros de su superclase.
- B) En Java existe herencia múltiple de interfaces.
- C) Una subclase no tiene acceso directo a los miembros privados de su superclase.
- D) Las afirmaciones B y C son verdaderas.

9. Teniendo en cuenta el siguiente fragmento de código, ¿cómo se puede acceder al atributo *numElementos* de la clase *A* desde el método *muestraValor()* de la clase *B*?

```
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 clase *B*.
- B) Se puede acceder utilizando *super.numElementos*.
- C) Se puede acceder utilizando *((claseA) this).numElementos*.
- D) Las afirmaciones B y C son verdaderas.

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. Respecto a las clases abstractas y las interfaces en Java, ¿qué afirmación es la correcta?

- A) Una clase no puede heredar de dos clases abstractas, pero sí puede heredar de una clase abstracta e implementar una interfaz, o bien implementar dos o más interfaces.
- B) De cara al polimorfismo, las referencias de un tipo interfaz se pueden utilizar de modo similar a las clases abstractas.
- C) Las interfaces tienen una jerarquía propia, independiente y más flexible que la de las clases, ya que tienen permitida la herencia múltiple.
- D) Todas las afirmaciones anteriores son verdaderas.

12. Respecto a las clases internas en Java, ¿qué afirmación es la correcta?

- A) Existen cuatro tipos de clases internas: clases internas static, clases internas miembro, clases internas locales y clases anónimas.
- B) La JVM trata explícitamente las clases internas como clases especiales de Java.
- C) Las clases internas miembro sólo pueden ser creadas dentro de otra clase al máximo nivel, es decir, en el bloque de definición de la clase contenedora.
- D) De las clases anónimas sólo es posible tener un único objeto.

13. Sobre las excepciones en Java, ¿qué afirmación es la correcta?

- A) La gestión de excepciones en Java se realiza mediante cuatro palabras clave: *try*, *catch*, *throws* y *finally*.
- B) Todos los tipos de excepción son subclases de la clase *Exception*.
- C) En un programa Java es posible tener sentencias *try* anidadas y cláusulas *catch* múltiples.
- D) Las afirmaciones A y C son verdaderas.

14. Respecto a los ficheros *jar* de Java, ¿qué afirmación es la correcta?

- A) Los ficheros *jar* son ficheros comprimidos que contienen exclusivamente archivos de clases Java.
- B) La herramienta *jar* de Java se puede utilizar para firmar ficheros *jar*.
- C) Para poder utilizar las clases contenidas en un fichero *jar* desde un programa Java, se debe incluir en la variable CLASSPATH la ruta completa del directorio que contiene al fichero *jar*.
- D) Todas las afirmaciones anteriores son falsas.

15. Respecto a la generación de documentación de forma automática en Java, ¿qué afirmación es la correcta?

- A) La herramienta *javadoc* se incluye de serie en el JRE de Java.
- B) Los *doclets* son programas de Java que especifican el contenido y el formato de la salida de *javadoc*.
- C) La etiqueta *@throws* tiene un comportamiento diferente a la etiqueta *@exception*.
- D) La etiqueta *@deprecated* no afecta al comportamiento del compilador de Java puesto que se incluye dentro de un comentario *doc*.

16. Sobre los *applets* en Java, ¿qué afirmación es la correcta?

- A) Cuando un navegador carga un *applet*, siempre se llama automáticamente a su método *init*.
- B) Los métodos *start* y *stop* de un *applet* no pueden ser invocados automáticamente.
- C) El método *destroy* de un *applet* debe ser redefinido siempre para garantizar su correcto funcionamiento.
- D) Un *applet* se puede ejecutar correctamente tanto en el servidor web como en el cliente que realiza la petición.

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

- A) Cuando un hilo sale del estado *dormido* retoma automáticamente la CPU.
- B) Un hilo termina su ejecución cuando completa su método *run*.
- C) Utilizando métodos *synchronized* se puede programar correctamente el problema del productor-consumidor.
- D) El entorno en tiempo de ejecución de Java es capaz de detectar, aunque no solucionar, el problema del interbloqueo.

18. 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*.
- A) La interfaz *FilenameFilter* se utiliza para limitar el número de archivos devueltos por el método *list* de la clase *File*.

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

- A) Las clases que implementan la interfaz *Serializable* deben definir los métodos *WriteObject* y *ReadObject*.
- B) Las variables y objetos *static* no son serializados.
- C) La palabra clave *transient* permite indicar que un objeto o variable miembro no sea serializado con el resto del objeto.
- D) Las afirmaciones B y C son verdaderas.

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

- A) El método *ExecuteUpdate* ejecuta sentencias SQL de tipo SELECT devolviendo un entero que indica el número de filas afectadas por el comando (denominado cuenta de actualización).
- B) Para la ejecución de sentencias SQL complejas, donde una sentencia puede proporcionar un conjunto de resultados múltiple, se utiliza el método *Execute*.
- C) Los métodos *ExecuteUpdate* y *Execute* devuelven un objeto de tipo *ResultSet*.
- D) Todas las afirmaciones anteriores son verdaderas.

22. Sobre el desarrollo de aplicaciones distribuidas con Java, ¿qué afirmación es incorrecta?

- A) En RMI, la capa de transporte utiliza por defecto sockets TCP para comunicarse con el esqueleto del servidor, aunque pueden utilizarse también otros protocolos como SSL y UDP.
- B) La herramienta *rmiregistry* está presente en el JRE de Java.
- C) RMI permite comunicar programas escritos en Java con otros programas escritos en otros lenguajes de programación (como C++) siempre y cuando el fragmento adaptador esté escrito en el lenguaje de definición de interfaz (IDL, *Interface Definition Language*).
- D) Las afirmaciones A y B son verdaderas.

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

- A) Los *servlets* pueden incorporar interfaz de usuario al igual que los *applets*.
- B) Los *servlets* se pueden ejecutar tanto en el cliente (navegador web) como en el servidor http.
- C) Mediante el uso de *servlets* se pueden establecer y enviar al cliente objetos de tipo *Cookie* de Java.
- D) Todas las afirmaciones anteriores son verdaderas.

24. 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 la afirmaciones anteriores son correctas.

25. Sobre el uso de diferentes tecnologías de programación web, ¿qué afirmación es la correcta?

- A) ColdFusion proporciona un conjunto de etiquetas similares a las de HTML.
- B) Las páginas ASP soportan múltiples lenguajes de script entre los que se incluyen: PerlScript, JScript y VBScript..
- C) En contraposición con otros lenguajes de script, SSJS es un lenguaje interpretado.
- D) Las afirmaciones A y B son verdaderas.