

			
	1ª Semana		
			Nacional

SISTEMAS INFORMÁTICOS I
INGENIERÍA EN INFORMÁTICA - Código Carrera 55 - Código Asignatura 077
NO SE PERMITE EL USO DE NINGÚN MATERIAL

Junio 2007
Tipo de Examen: C
Tiempo: 2 horas

Apellidos: _____ **Nombre:** _____ **DNI:** _____

Centro Asociado en el que está MATRICULADO: _____ **Especialidad:** _____

INSTRUCCIONES: Complete sus datos personales en la cabecera de esta hoja, y **ENTRÉGUELA OBLIGATORIAMENTE** con el resto de hojas de su examen. Cualquier examen que no venga acompañado de esta hoja de enunciados no será corregido. Complete **TODOS** los datos que se piden en la hoja de lectura óptica o en caso contrario su examen no será corregido. El examen consta de 20 preguntas. Cada respuesta acertada tiene un valor de 0.5 puntos y cada respuesta incorrecta de -0.25 puntos. Para superar el examen es necesario obtener una puntuación de al menos 5 puntos.

Test : Conteste exclusivamente en una HOJA DE LECTURA ÓPTICA, no olvidando marcar que su tipo de examen es C.

1.- ¿Qué hay que hacer para empaquetar un Java bean denominado grafica situado en el paquete utilidades?

a) Se empaqueta mediante la orden `jarbean cfm grafica.jar grafica.mf grafica` utilizando el fichero de manifiesto:

```
Manifest-Version: 1.0
Name: utilidades/grafica.class
Java-Bean: True
```

b) Se empaqueta mediante la orden `jar cfmp grafica.jar utilidades.`

c) Se empaqueta mediante la orden `jar cfm grafica.jar grafica.mf utilidades` utilizando el fichero de manifiesto:

```
Manifest-Version: 1.0
Name: utilidades/grafica.class
Java-Bean: True
```

d) Ninguna de las anteriores.

2.- ¿Cómo se comprueba en un programa Java que la conexión JDBC con una base de datos es correcta?

a) Si la expresión `Connection c = DriverManager.getConnection(dbUrl, user, password)` devuelve una excepción entonces la conexión no es correcta.

b) Si la sentencia `Connection c = DriverManager.getConnection(dbUrl, user, password)` devuelve una excepción entonces la conexión es correcta.

c) Si la sentencia `Connection c = DriverManager.getConnection(dbUrl, user, password)` devuelve una excepción entonces la conexión no es correcta.

d) Ninguna de las anteriores.

3.- ¿Cuál de las siguientes respuestas es una solución válida para que un programa servidor sea capaz de atender múltiples clientes simultáneamente a través de sockets TCP?

- a) La solución más frecuente es crear un único `ServerSocket` en el servidor e invocar al método `accept()` para esperar la llegada de conexiones. Cuando la conexión se produce, se toma el socket que devuelve como resultado `accept()` y se utiliza para crear un *thread* cuyo trabajo será atender al cliente en exclusiva. Creado el *thread* se vuelve a llamar al método `accept` para que espere por un nuevo cliente.
- b) Una solución es recurrir a la utilización de un `MultipleServerSocket` en el servidor e invocar al método `accept()` para esperar la llegada de conexiones. En cada conexión que se produce, el método `accept()` crea un nuevo *thread* y produce un desdoblamiento del código para atender al cliente en exclusiva.
- c) Una solución muy utilizada es recurrir a un bucle `while` en cuyo interior se coloca el mismo código que se utilizaría para atender a un único cliente pero recurriendo a un `ThreadServerSocket` cuyo método `accept()` se ocupa de crear los *threads* según se vayan aceptando las conexiones con los clientes y con el código que sea necesario ejecutar según el tipo de cliente.
- d) Ninguna de las anteriores.

4.- En el método `init` de un applet se realiza la asignación de un `BorderLayout` para la disposición de los elementos gráficos en su interior mediante la sentencia `getContentPane().setLayout(new BorderLayout())`. Dentro del mismo método `init` ¿qué sentencias habría que escribir para ubicar una etiqueta denominada `etiqueta` en la parte superior y un botón denominado `salir` en la parte inferior del `BorderLayout`:

- a) `getContentPane().add(etiqueta, BorderLayout.NORTH);`
`getContentPane().add(salir, BorderLayout.SOUTH);`
- b) `getBorderLayout().add(etiqueta, BorderLayout.NORTH);`
`getBorderLayout().add(salir, BorderLayout.SOUTH);`
- c) `getContentPane().add(etiqueta, new BorderLayout(NORTH));`
`getContentPane().add(salir, new BorderLayout(SOUTH));`
- d) Ninguna de las anteriores.

5.- Cuando las interfaces del tipo `listener` constan de un único método, su implementación es directa ya que sólo hay que programar ese método para disponer del manejador de eventos. Sin embargo, ¿qué hay que hacer para no tener que programar todos los métodos de una interfaz de tipo `listener` cuando ésta se compone de muchos?

- a) No hay otra solución que implementar toda la interfaz, es decir, programar en el manejador de eventos todos los métodos de que consta la interfaz de tipo `listener` que se utilice.
- b) Se recurre a la utilización de las clases de tipo `adapter` asociadas a la interfaz `listener` que se utilice. Por ejemplo, para no tener que programar todos los métodos del manejador de una ventana se recurriría a `class MyWindowListener extends WindowAdapter {.....};`
- c) Se recurre a la utilización de las interfaces de tipo `adapter` en lugar de las `listener`. Por ejemplo, para escribir el manejador de la ventana de una aplicación se recurriría a `class MyWindowListener implements WindowAdapter {.....};`
- d) Ninguna de las anteriores.

6.-A continuación, se muestra el código de dos clases: **Prueba2a** y **Prueba2b**. El método **main** de la clase **Prueba2b** se ejecuta sin errores. Indique cuál es el resultado de su ejecución.

```
public class Prueba2a {
    static int i = 47;
    public void incr() { i++; }
    public int geti() { return i; }
}
```

```
public class Prueba2b {
    public static void main(String[] args) {
        Prueba2a p2a = new Prueba2a();
        Prueba2a.i++;
        p2a.incr();
        System.out.println("i = " + Prueba2a.i + ", " +
            p2a.geti());
    }
}
```

a) i = 48, 48

b) i = 49, 49

c) i = 49, 47

d) Ninguna de las anteriores.

7.- Indique qué fragmento de código produce error de “división por cero” en tiempo de ejecución:

- a) `int i0 = 0, i1 = 1, i2 = 2;`
`boolean b = i1 > i2 && i2/i0 > i1;`
- b) `int i0 = 0, i1 = 1, i2 = 2;`
`boolean b = i1 > i2 & i2/i0 > i1;`
- c) Los fragmentos de las dos respuestas anteriores producen error.
- d) Ninguno de los fragmentos produce error.

8.-Indique cuál de las siguientes afirmaciones acerca de la clase **Exterior** es correcta. Obsérvese que esta clase tiene una clase interna. Las líneas se han numerado para facilitar la referencia a las mismas.

```
1 public class Exterior {
2     private int a = 2;
3     public int b;
4     public Exterior (int b) { this.b = b; }
5     public Interior getInterior() { return new Interior(); }
6     class Interior {
7         void display() {
8             System.out.print( "a = " + a );
9             System.out.println( ", b = " + b);
10        }
11    }
12 }
```

- a) Se produce error de compilación en la línea 8, ya que la clase interna **Interior** no tiene acceso a las variables privadas de la clase **Exterior**.
- b) Se produce error de compilación en la línea 9, ya que la clase interna **Interior** no tiene acceso a las variables públicas de la clase **Exterior**.
- c) Las dos afirmaciones anteriores son correctas.
- d) Ninguna de las afirmaciones anteriores es correcta.

9.- A continuación se muestra el código de la interfaz **Interf1** y de la clase **Ejem_Interf1**, que se encuentran en el mismo paquete. Indique cuál de las siguientes afirmaciones es correcta. Se han numerado las líneas de la clase **Ejem_Interf1** para facilitar la referencia a las mismas.

```
public interface Interf1 {
    void display();
    void display2();
    int n = 2;
}

1 public class Ejem_Interf1 implements Interf1 {
2     public void display() {
```

```

3           System.out.println("n = " + n);
4       }
5       public static void main(String[] args) {
6           Ejem_Interf1 e = new Ejem_Interf1();
7           e.n = 4;
8           e.display();
9       }
10  }

```

- a) La clase Ejem_Interf1 produce error de compilación, ya que no implementa el método Interf1.display2()
- b) Se produce error de compilación en la línea 7, ya que la variable n, declarada en la interfaz, es final y static, (es decir, constante y estática).
- c) Las dos anteriores son correctas.
- d) Ninguna de las anteriores.

10.- Indique cuál de las siguientes afirmaciones acerca de la palabra clave final es correcta.

- a) Los métodos que se declaran como final no pueden ser sobrescritos.
- b) Las clases que se declaran como final no pueden ser heredadas.
- c) Las dos afirmaciones anteriores son correctas.
- d) Ninguna de las afirmaciones anteriores es correcta.

11.- ¿Cuál es el resultado de la ejecución del siguiente código?

```

public class PruebaContador {
    public static void f(int contador){
        ++contador;
    }
    public static void g(int contador){
        f(contador);
        f(contador);
    }
    public static void h(int contador){
        f(contador);
        g(contador);
        f(contador);
    }
    public static void main(String[] args){
        int contador=0;
        f(contador);
        System.out.print(contador);
        g(contador);
        System.out.print(contador);
        h(contador);
        System.out.print(contador);
    }
}

```

- a) 126
- b) 015
- c) 000
- d) Ninguna de las anteriores. Existe un error.

12.- ¿Cuál de las siguientes afirmaciones es cierta?

- a) Una clase declarada como **final** permite herencia múltiple.
- b) Una clase declarada como **final** solamente puede ser heredada de manera simple.
- c) Una clase declarada como **final** no puede ser heredada.
- d) Una clase declarada como **final** admite herencia tanto simple como múltiple.

13.- ¿Cuál de las siguientes afirmaciones es cierta?

- a) Dos clases que forman parte del mismo paquete pueden acceder a los miembros públicos, protegidos y con acceso por defecto.
- b) Dos clases que forman parte del mismo paquete NO pueden acceder a los miembros públicos, protegidos y con acceso por defecto.
- c) Dos clases que forman parte del mismo paquete solo pueden acceder a los miembros públicos.
- d) Ninguna de las anteriores.

14.- ¿Cuál de las siguientes afirmaciones NO es cierta?

- a) Si en una clase no se definen constructores, Java crea uno por defecto.
- b) Todas las clases heredan los métodos **toString()**, **equals()** y **clone()**
- c) Un método de instancia NO puede acceder a las variables de clase
- d) Los métodos que tienen el modificador **static** se denominan métodos de clase.

15.- ¿Cuál de las siguientes afirmaciones NO es cierta?

- a) Una subclase puede añadir nuevos atributos y métodos.
- b) Todas las clases heredan de manera explícita o implícita de la clase **Object**.
- c) Una clase solo puede heredar de una clase pero puede implementar varias interfaces.
- d) Si una subclase reescribe un método, NO puede acceder al método de la clase de la que hereda.

16.- Dado el siguiente código. Determinar que línea de código impide la correcta ejecución del mismo. Eliminada esta línea cuál sería el resultado obtenido

```
1. public class Test {
2.     public static void main (String [] args) throws Exception {
3.         try {
4.             throw new Exception();
5.             System.out.print(" dos ");
6.         } catch(Exception e) {
7.             System.out.print(" uno ");
8.             System.exit(0);
9.         } finally {
10.            System.out.print(" final ");
11.            throw new Exception();
12.        }
13.    }
14. }
```

- A) Error en la línea 7. Resultado: dos
- B) Error en la línea 5. Resultado: uno
- C) Error en la línea 10. Resultado: dos
- D) Error en la línea 8. Resultado final

17.- Dado el siguiente código ¿cuál es el resultado esperado?.

```
1. public class Test {
2.     static boolean b;
3.     public static void main(String [] args) {
4.         int x=0;
5.         if (b) {
6.             x=1;
7.         }
8.         else if (b) {
9.             x=2;
10.        }
11.        else if (b) {
12.            x=4;
13.        }
14.        else {
15.            x=3;
```

```
16. }
17. System.out.println("x = " + x);
18. }
19. }
```

- A) x = 4
- B) x = 3
- C) x = 2
- D) Ninguna de las anteriores.

18.- Dado el código siguiente ¿cuál es resultado esperado?.

```
1. public class Test {
2.     public static void main(String [] args) {
3.         int x = 0;
4.         boolean b [] = new boolean[3];
5.         b[1] = true;
6.         for (x= 0;x<3; x++) {
7.             System.out.print(" " + b[x]);
8.         }
9.     }
10. }
```

- A) El código es incorrecto, da un error de compilación
- B) false false false
- C) false true false
- D) Ninguna de las anteriores

19.- Dado el código

```
public class ejer1 {
    public static void main(String [] args) {
        int i=3, j=0, result=1;
        result += i-- * --j ;
        System.out.println( result );
    }
}
```

¿Cuál sería el resultado final obtenido?

- A) 1
- B) -2
- C) 2
- D) Ninguna de las anteriores.

20.-Dado el siguiente fragmento de código obtenido de un programa perfectamente creado, por tanto, se entiende que existe definición de todas las clases y elementos necesarios para su ejecución, se pide decir que ocurrirá al ejecutarse correctamente.

```
public Conexión() {
    String driver = new String("com.mysql.jdbc.Driver");
    String url = new String("jdbc:mysql://localhost:3306/supermercado");
    try{
        Class.forName(driver);
        conexion = DriverManager.getConnection(url,"root", " ");
    }
    catch (Exception e){
        System.out.println(e.getLocalizedMessage());
    }
}
```

- A) Se establecerá una conexión con una base de datos de MySQL llamada supermercado
- B) Se establece una conexión con el usuario root del sistema operativo.
- C) Se establece una conexión al servidor llamado com.mysql.jdbc y devuelve su dirección IP
- D) Ninguna de las anteriores

	TIPO B		
1	C		
2	C		
3	A		
4	A		
5	B		
6	B		
7	B		
8	D		
9	C		
10	C		
11	C		
12	C		
13	A		
14	C		
15	D		
16	B		
17	B		
18	C		
19	B		
20	A		