SISTEMAS INFORMÁTICOS I — **Junio 2005 – 2ª Semana**
INGENIERÍA EN INFORMÁTICA - Código Carrera 55 - Código Asignatura 077 — **Tipo de Examen: G**
NO SE PERMITE EL USO DE NINGÚN MATERIAL — **Tiempo: 2 horas**

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

Centro Asociado en el que está MATRICULADO: _____

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 G.

1. A continuación se muestra el código de dos clases: *Ejemplo1* y *Ejemplo1b*. La clase *Ejemplo1* está dentro del paquete *p1*. La clase *Ejemplo1b* está dentro del paquete *p2*. Las líneas de la clase *Ejemplo1b* se han numerado con el propósito de facilitar la referencia a las mismas.

Clase *Ejemplo1*, guardada en el archivo *p1/Ejemplo1.java*:

```
package p1;
public class Ejemplo1 {
      int n=1;
      private   int n_pri = 2;
      protected int n_pro = 3;
      public    int n_pub = 4;

      public Ejemplo1() {
      }
}
```

Clase *Ejemplo1b*, guardada en el archivo *p2/Ejemplo1b.java*:

```
1     package p2;
2     import p1.Ejemplo1;
3     public class Ejemplo1b extends Ejemplo1 {
4           Ejemplo1b() {
5                 System.out.print("n = " + n);
6                 System.out.print(", n_pri = " + n_pri);
7                 System.out.print(", n_pro = " + n_pro);
8                 System.out.print(", n_pub = " + n_pub);
9           }
10          public static void main(String args[]) {
11                Ejemplo1b ej1b = new Ejemplo1b();
12          }
13    }
```

Señale cuál de las siguientes afirmaciones es correcta:

A) Error en la línea 5 de la clase *Ejemplo1b*.
B) Error en la línea 5 y en la línea 6 de la clase *Ejemplo1b*.
C) Error en la línea 5, en la línea 6 y en la línea 7 de la clase *Ejemplo1b*.
D) Todas las afirmaciones anteriores son falsas.

2. Indique cuál es el resultado de la ejecución del método *main* de la clase *Ejemplo3*. El código de dicha clase, que no contiene errores, se muestra a continuación.

```
package p1;
import java.io.*;
public class Ejemplo3 {
   static int cuentaCaracteres(Reader fr, char c) {
         int numChar = 0;
         try {
               int i;
               while ((i = fr.read()) != -1) {
                     if ((char)i == c) numChar++;
               }
         } catch (IOException e) { return -1; }
         return numChar;
```

```
    }
    public static void main(String args[]) {
          FileReader fr;
          try {
                fr = new FileReader("ejemplo3.txt");
          } catch (FileNotFoundException e) {
                System.out.println("Fichero no encontrado");
                return;
          }
          System.out.println("Núm. 'a': " + cuentaCaracteres(fr,'a'));
          System.out.println("Núm. 'a': " + cuentaCaracteres(fr,'a'));
    }
}
```

El contenido del archivo *ejemplo3.txt*, que se encuentra en el directorio actual (y por tanto no se genera una excepción de fichero no encontrado), es el siguiente:

```
Este fichero contiene
tres caracteres 'a'
```

A)
```
Núm. 'a': 3
Núm. 'a': 3
```

B)
```
Núm. 'a': 3
Núm. 'a': 0
```

C)
```
Núm. 'a': 3
Núm. 'a': 6
```

D)
```
Núm. 'a': 3
Núm. 'a': -1.
```

3. Indique cuál de las afirmaciones acerca de la clase siguiente es correcta. Las líneas se han numerado para facilitar la referencia a las mismas.

```
1  package p4;
2  import javax.swing.*;
3  import java.awt.*;
4  import java.awt.event.*;
5  public class BotonEvento3 {
6        public static void main(String args[]) {
7              JButton jb = new JButton("Botón");
8              jb.setBackground(Color.white);
9
10             jb.addActionListener(new ActionListener() {
11                   public void actionPerformed(ActionEvent ev) {
12                         System.out.println(
13                                     "Se ha pulsado el botón");
14                   }
15             });
16             JFrame  f  = new JFrame("Ventana con botón");
17             f.setSize(250,100);
18             f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
19             f.getContentPane().add(jb);
20             f.setVisible(true);
21       }
22 }
```

A) Si se elimina la línea 4, se produce error.
B) Al ejecutar su método *main*, se muestra una ventana. El título de la ventana es "Ventana con botón". La ventana contiene un botón. Al hacer clic sobre el botón, se imprime en el flujo de salida estándar el mensaje: "Se ha pulsado el botón".
C) Las dos respuestas anteriores son correctas.
D) Ninguna de las respuestas anteriores es correcta.

4. A continuación se muestra el código de dos clases: *Ejemplo5_A* y *Ejemplo5_AB*.

Clase Ejemplo5_A, guardada en el fichero p7/Ejemplo5_A.java:

```
package p7;
public class Ejemplo5_A {
   public Ejemplo5_A() {
         System.out.println("A:");
   }
   public Ejemplo5_A(String texto) {
         System.out.println("A: "+texto);
   }
}
```

Clase Ejemplo5_AB, guardada en el fichero p7/Ejemplo5_AB.java:

```
package p7;
public class Ejemplo5_AB extends Ejemplo5_A {
   Ejemplo5_AB (String texto) {
         System.out.println("AB: "+texto);
   }
   public static void main(String[] args) {
         Ejemplo5_AB ej = new Ejemplo5_AB("texto");
   }
}
```

Señale cuál es el resultado de la ejecución del método *main* de la clase *Ejemplo5_AB*:

A)
```
AB: texto
```

B)
```
A:
AB: texto
```

C)
```
A: texto
AB: texto
```

D)
```
A:
A: texto
AB: texto
```

5. La especificación de la URL que hay que componer para acceder a una base de datos denominada notas mediante JDBC utilizando el subprotocolo ODBC es:

A) jdbc:odbc: notas.
B) jdbc-odbc:notas.
C) jdbc.odbc:notas.
D) Ninguna de las anteriores.

6. Tras realizar una consulta SQL a una base de datos, el resultado:

A) Se almacena en un objeto ResultSet, recorriéndose este objeto con el método next() y utilizando métodos get***() para recuperar la información de los campos que componen los registros recuperados.

B) Se almacena en un objeto StringTokenizer y se recorre el contenido del objeto utilizando, por ejemplo, el método hasMoreTokens(). Los registros recuperados se extraen utilizando el método nextToken().

C) Se almacena en un objeto Vector. Los registros recuperados de la base de datos se extraen de ese objeto Vector mediante el método elementAt() que devuelve un objeto genérico Object por cada registro recuperado de la base de datos.

D) Ninguna de las anteriores.

7. **Cuál de los siguientes fragmentos de código es el correcto si se desea proceder a la carga de un fichero:**

A)
```
URL url = new URL(getDocumentBase(), "datos.txt");
InputStream is = url.openStream();
BufferedReader in = new BufferedReader(new
InputStreamReader(is));
```
B)
```
InputStream is = URL.openStream(getDocumentBase(),
"datos.txt");
BufferedReader in = new BufferedReader(new
InputStreamReader(is));
```
C)
```
BufferedReader in = new BufferedReader(new
InputStreamReader(getDocumentBase(), "datos.txt"));
```
D) Ninguna de las anteriores.

8. **Se dispone de un paquete formado por una clase Joystick que contiene un método para recuperar una de las coordenadas de la palanca, int getX(), y una interfaz JoystickListener que contiene el método para detectar el movimiento de la palanca, void joystickAxisEvent (Joystick j). Si se escribe código para hacer uso del joystick en la aplicación, ¿cuál de los siguientes fragmentos de código es correcto?**

A)
```
juguete = new Joystick ();
juguete.addJoystickListener (new JoystickListener() {
          public void joystickAxisEvent (Joystick  j) {};
});
int coordenada = juguete.getX();
```
B)
```
juguete = new Joystick ();
public int joystickAxisEvent (Joystick  j) {
          return  j.getX();
};
```
C)
```
juguete = new Joystick();
juguete.      addJoystickListener      (joystickAxisEvent
(juguete));
int coordenada = juguete.getX();
```
D) Ninguna de las anteriores.

9. **Para fijar el layout a null en un objeto elemento1 de tipo java.awt.Frame y en un objeto elemento2 del tipo java.awt.Panel, ¿cuál de las siguientes secuencias de código es la correcta?**

A)
```
elemento1. setLayout (null);
elemento2.setLayout (null);
```
B)
```
elemento1.getContentPane().setLayout (null);
elemento2. getContentPane().setLayout (null);
```
C)
```
elemento1. getContentPane().setLayout (null);
elemento2. setLayout (null);
```
D) Ninguna de las anteriores.

10. **Dado el siguiente código, ¿cuál es el resultado esperado?.**

```
public class Test {
  static boolean b;
  public static void main(String [] args) {
     int x=0;
     if (b ) {
      x=1;}
     else if (b = false) {
      x=2;}
    else if (b) {
     x=3;}
    else {
     x=4;}
    System.out.println("x = " + x);
```

```
    }
  }
```

A) x = 4
B) x = 3
C) x = 2
D) Es un programa erróneo y no se ejecuta.

11. Dado el código siguiente, ¿cuál es el resultado esperado?

```
1. class prueba {
2.   prueba(String s) {System.out.print(s); }
3. }
4. class examen extends prueba { }
5.  public class ejer1 {
6.   public static void main(String [] args) {
7.    new prueba("Clase ");
8.    new examen();
9.    System.out.print("FIN");
10.   }
11. }
```

A) El código es incorrecto, da un error de compilación
B) Escribe FIN
C) Escribe Clase
D) Ninguna de las anteriores

12. Dado el código siguiente cuál es resultado esperado

```
1. public class ejer1 {
2.   static int count = 0;
3.   public static void main(String [] args) {
4       int x = 1;
5.      for ( test('1'); test('2') && (x<= 2); test('3') ) {
6.        x++;
7.        test('4');
8.      }
9.      System.out.println(" Cuenta = " + count);
10.   }
11.   static boolean test(char num) {
12.     System.out.print(" "+num);
13.     count++;
14.     return true;
15.   }
16. }
```

A) 1 2 4 3 2 4 3 2 Cuenta = 8
B) 1 4 2 3 1 4 2 3 Cuenta = 8
C) 1 2 3 4 1 2 3 4 Cuenta = 8
D) Ninguna de las anteriores.

13. Dado el código

```
1.   public class X {
2.     public void main(String [] args) {
3.       System.out.println("Parámetro " + args[0]);
4.     }
5.   }
```

Cuando se invoca al programa en la línea de comandos de la forma

```
java X Y
```

¿Cuál es el resultado?

A) Se produce una excepción en tiempo de ejecución
B) Parámetro X
C) Parámetro Y
D) Ninguna de las anteriores

14. Dada la siguiente orden de ejecución en línea de comandos

```
Java Clases para1 para2 para3
```

¿Donde se almacenan inicialmente los valores de los parámetros?

A) Todos los parámetros se almacenan en una única cadena denominada arg
B) Se almacena en un array indexado entre 0 y 2
C) Se almacena en un array indexado entre 1 y 3
D) Ninguna de las anteriores.

15. ¿Cuál es el resultado de la ejecución del siguiente fragmento de código?

```
class A{
  void callme(){
    System.out.println("Llama al metodo callme dentro de A");
  }
}
class B extends A{
  //Sobreescribo callme()
  void callme(){
    System.out.println("Llama al metodo callme desde B");
  }
}
class C extends A{
  void callme(){
    System.out.println("Llama el metodo callme dentro de C");
  }
}
public class Dispatch {
  public Dispatch() {
  }
  public static void main(String[] args) {
    A a=new A();
    B b=new B();
    C c=new C();
    A r;
    r=a;
    r.callme();
    r=b;
    r.callme();
    r=c;
    r.callme();
  }

}
```

A) Llama al metodo callme dentro de A
   Llama al metodo callme desde B
   Llama el metodo callme dentro de C
B) Llama al metodo callme dentro de B
   Llama al metodo callme desde A
   Llama el metodo callme dentro de C
C) Llama al metodo callme dentro de C
   Llama al metodo callme desde B
   Llama el metodo callme dentro de C
D) Llama al metodo callme dentro de A
   Llama al metodo callme desde A
   Llama el metodo callme dentro de A

16. ¿Cuál es el resultado de la ejecución del siguiente fragmento de código?

```
public class ThreadEscritor implements Runnable{
  int interno;

  public ThreadEscritor(int msg) {
    super();
    interno=msg;

  }

   synchronized  public void run(){
    while(true){
      System.out.println(interno);
      Thread.currentThread().yield();
    }
  }
  public static void main(String[] args) {
    for(int i=0;i<5;i++){
      Thread t=new Thread(new ThreadEscritor(i));
      t.start();}


    try{
      Thread.currentThread().sleep(1000);
    }
    catch(InterruptedException e){

    }
    finally{
      System.exit(0);
    }
  }

}
```

A) La secuencia ordenada de salida 01234 durante un cierto tiempo
B) La secuencia ordenada de salida 1234 durante un cierto tiempo
C) La secuencia ordenada de salida 0123 durante un cierto tiempo
D) No se puede saber, el orden con el que la CPU atiende los hilos es indeterminado.

17. ¿Cuál de las siguientes afirmaciones es cierta?

A) Un *Applet* depende de la plataforma en la que se ejecute.
B) Un *Applet* depende del servidor que lo ejecute.
C) Un *Applet* tiene independencia completa de la plataforma.
D) Ninguna es cierta

18. ¿Cuál es el resultado de la ejecución del siguiente fragmento de código suponiendo que se ejecuta "java Detergente"?

```
class ProductoLimpieza{
  private String s = new String("Producto Limpieza");
  public void aniadir(String a){s +=a;}
  public void diluir() {aniadir("diluir()");}
  public void aplicar() {aniadir("aplicar");}
  public void escribir() {System.out.println(s);}

  public static void main(String[] args){
    ProductoLimpieza x= new ProductoLimpieza();
    x.diluir();
    x.aplicar();
    x.escribir();
  }
}
```

```
public class Detergente extends ProductoLimpieza{

  public void aplicar(){
    aniadir("Detergente.frotar()");
    super.aplicar();
  }
  public static void main(String[] args) {
    Detergente detergente1 = new Detergente();
    detergente1.aniadir("a");
    detergente1.diluir();
    detergente1.aplicar();
    detergente1.escribir();

    System.out.println("Probando la clase base");
    ProductoLimpieza.main(args);
  }

}
```

A) Probando la clase base
Producto Limpiezaadiluir()Detergente.frotar()aplicar
Producto Limpiezadiluir()aplicar

B) Producto Limpiezaadiluir()Detergente.frotar()aplicar
Probando la clase base
Producto Limpiezadiluir()aplicar

C) Producto Limpiezadiluir()aplicar
Producto Limpiezaadiluir()Detergente.frotar()aplicar
Probando la clase base

D) Ninguna de ellas, la clase Detergente no puede heredar de ProductoLimpieza

19. En JAVA, para poder guardar un objeto de manera persistente es necesario que implemente la interfaz:

A) Runnable
B) Serializable
C) Localizable
D) Ninguna de ellas

20. A continuación se muestra el código de la clase *ListaDeArrays2*. Se han numerado las líneas para facilitar la referencia a las mismas.

```
1       package p5;
2       import java.util.*;
3       public class ListaDeArrays2 {
4            public static void main(String args[]) {
5                 ArrayList a1   = new ArrayList();
6                 int       suma = 0;
7                 for (int i=0; i<3; i++)
8                      a1.add(new Integer(i));
9                 a1.add("a");
10                a1.add("b");
11                for (int i=0; i<a1.size(); i++)
12                     System.out.print(a1.get(i));
13
14           }
15      }
```

Señale cuál de las siguientes afirmaciones es correcta:

A) Se produce un error en la línea 8.

B) Se produce un error en la línea 9 y un error en la línea 10.
C) Las dos afirmaciones anteriores son correctas.
D) El código de la clase no contiene errores. El resultado de la ejecución de su método *main* es la impresión en el flujo de salida estándar de: `012ab`