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

Septiembre 2005
Tipo de Examen: A
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 la HOJA DE LECTURA ÓPTICA, no olvidando marcar que su tipo de examen es A.

1. En Java, un objeto JavaBean se caracteriza por:

   A) Poseer un estado interno y unos métodos para consultar y cambiar ese estado.
   B) Ser una interfaz del paquete `java.beans` con las propiedades y métodos que hay que implementar para definir un estado y un comportamiento particular.
   C) Implementar todos los métodos presentes en la interfaz `java.beans.BeanDescriptor`.
   D) Ninguna de las anteriores.

2. Para que un servidor desarrollado en Java sea capaz de trabajar simultáneamente con varios clientes conectados con él mediante una conexión TCP, es necesario que:

   A) Se recurra a la clase `ServerSocket`, la cual permanecerá a la escucha en todos y cada uno de los puertos que se hayan asociado previamente a los clientes para establecer la conexión. Para ello habrá que instanciar tantos objetos de la clase ServerSocket como clientes quieran conectarse al servidor.
   B) Por cada conexión que acepte el servidor, éste instancie un objeto que permita atender la conexión con el cliente de forma exclusiva. El objeto instanciado deberá heredar de la clase `Thread` y recibir como parámetro el socket TCP creado con el método `accept()`.
   C) Se utilice un objeto de la clase `MultipleServerSocket()` para crear un socket TCP que sea capaz de permanecer simultáneamente a la escucha en varios puertos. Por cada conexión entrante se generará un nuevo thread que recibirá como parámetro de entrada el socket creado.
   D) Ninguna de las anteriores.

3. Si se dispone de un computador con dos tarjetas de red y, por lo tanto, con la posibilidad de utilizar dos números IPs diferentes para atender conexiones de red, ¿qué constructor hay que utilizar para que el programa que hace de servidor permanezca a la escucha de las conexiones entrantes por una dirección IP concreta?

   A) Hay que utilizar el constructor `ServerSocket(int port, int backlog, InetAddress ip)`, especificándose en el parámetro `ip` el número IP en el que el programa permanecerá a la escucha.
   B) Hay que recurrir al constructor `MultipleServerSocket (int[] port, InetAdress[] ip)`, utilizando un array del tipo `InetAdress` con tantos elementos como direcciones IP diferentes estén disponibles, es decir, en este supuesto, dos elementos, uno por cada tarjeta de red.
   C) Se debe utilizar el constructor `ServerSocket(int port, int backlog)` y mediante la información proporcionada en `backlog` se conoce qué tarjeta de red es la que recibe la petición de conexión y, por tanto, es posible crear un socket particular para cada dirección IP y puerto.
   D) Ninguna de las anteriores.

4. Para establecer una conexión con una base de datos utilizando nombre de usuario y contraseña hay que recurrir a:

   A)
   ```
   DriverManager driverDB = DriverManager.getDriver(dbURL);
   Connection bd = driverDB.getConnection(usuario, contrasenia);
   ```

   B)
   ```
   Connection bd = DriverManager.getLoginConnection(dbURL,usuario,contrasenia);
   ```

   C)
   ```
   Connection bd = DriverManager.getConnection(dbURL,usuario,contrasenia);
   ```

   D) Ninguna de las anteriores

5. Existen diferentes razones por las que un thread en Java puede quedar bloqueado. ¿Cuál de las siguientes razones enumeradas a continuación es falsa, es decir, no provoca el bloqueo del thread?

   A) El thread se ha colocado en estado "durmiente" mediante una llamada a la instrucción `sleep(milisegundos)`, con lo que quedará detenido durante el tiempo que se haya especificado.
   B) Se ha suspendido la ejecución del thread con `resume()`. No volverá a ejecutarse de nuevo hasta que el thread reciba el mensaje `notify()` o `notifyAll()`.
   C) Se ha suspendido la ejecución del thread con `wait()` y no se despertará hasta la recepción del mensaje `notify()` o `notifyAll()`.
   D) Ninguna de las anteriores.

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

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

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

7. ¿Cuál sería la salida del siguiente fragmento de código?

```
public class ejer1 {
      public static void main(String args[]) {
            {
                  int cuenta=120;
                  { int saldo=23; }
                  System.out.print("Cuenta=" + cuenta);
                  System.out.print(", Saldo=" + saldo);
            }
      }
}
```

   A) `Cuenta=120, Saldo=23`
   B) `Cuenta=120, Saldo=0`
   C) `, Saldo=143`
   D) Ninguna de las anteriores.

8. ¿Cuales de las siguientes líneas de código son correctas?

```
1. int [42] x;
2. int x [42];
3. int [] x = (1,2,3);
```

A) Son todas incorrectas.
B) Sólo la 1 es correcta.
C) Sólo la 2 es correcta.
D) Ninguna de las respuestas anteriores es correcta.

9. Complete el código marcado como XXXXXX. Esta clase debe calcular la media de los valores almacenados en el array llamado `resultado`.

```
public class ejer1 {
public static void main(String args[]) {
        double resultado[] = { 76.0, 84.5, 92.5, 88.0, 96.0 };
        double sum= 0;
        double media=0;
        for (int i = 0; XXXXXXXXXXXXX; i ++) {
            sum += resultado[i];
        }
        media = sum / resultado.length;
        System.out.println("La media de los resultados es " + media);
    }
}
```

A) `i<resultado.length`
B) `i<7`
C) `i<resultado.media`
D) Ninguna de las respuestas anteriores produce el resultado deseado.

10. Dado el siguiente código, determinar qué línea 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) línea 7, `dos`
B) línea 5, `uno`
C) línea 10, `dos`
D) línea 8, `final`

11. En el siguiente fragmento de programa, ¿cuántas veces se ejecuta la sentencia `System.out.println`?

```
for (int i = 1; i <= 20; i = i+2) {
     a = 0;
     do {
          System.out.println(i + " " +a);
          a = a + 1;
     } while (a < 10);
}
```

A) 90
B) 100
C) 81
D) 110

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

```
public class matriz1 {

  public matriz1() {
  }

  public static void main(String[] args) {
    Integer[] arrayInteger = new Integer[5];
    arrayInteger[0]= new Integer(3);
    arrayInteger[1]= new Integer(4);
    arrayInteger[2]= new Integer(5);
    arrayInteger[3]= new Integer(6);
    for ( int i = 0; i < arrayInteger.length; i++) {
        System.out.print(arrayInteger[i]);
    }
  }
}
```

A) 01234
B) null34567
C) 3456null
D) Ninguno de los anteriores.

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

```
import java.util.*;
public class Iterador {

  public static void main(String[] args) {
    TreeSet conjunto = new TreeSet();
    conjunto.add("Esto ");
    conjunto.add("es ");
    conjunto.add("una ");
    conjunto.add("prueba ");
    conjunto.add("prueba ");

    Iterator itr = conjunto.iterator();

    while(itr.hasNext()){
      System.out.print(itr.next());
    }
  }
}
```

A) Esto es una prueba prueba
B) Esto es una prueba
C) Esto es prueba una
D) Al compilar da un error. No se pueden añadir dos elementos iguales a un objeto `TreeSet`.

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

```
import java.io.*;
public class Valores {
  public Valores() {
  }
  public static void main(String[] args) {
   byte[] byteArray={0,1,2,3,4,5,6,7,8,9};
   ByteArrayInputStream flujoArrayByte =
                new ByteArrayInputStream(byteArray);
   while(flujoArrayByte.available() !=0){
         byte leido = (byte)flujoArrayByte.read();
         System.out.print(leido);
   }
  try {
    flujoArrayByte.close();
  }
  catch (IOException ex) {
  }
  }
}
```

A) `0101010101`
B) `0123456789`
C) `9876543210`
D) Al compilar da un error. Es necesario realizar un *casting* en la lectura.

15. ¿Cuál de las siguientes afirmaciones es cierta?

A) La clase `StreamTokenizer` deriva de `InputStream`.
B) La clase `StreamTokenizer` deriva de `OutputStream`.
C) La clase `StreamTokenizer` funciona con cualquier tipo de objetos.
D) La clase `StreamTokenizer` sólo funciona con objetos `InputStream`.

16. Señale cuál de las siguientes afirmaciones acerca del código siguiente es correcta. Las líneas se han numerado con el propósito de facilitar la referencia a las mismas.

```
1     public class NumeroPI {
2           public static void main(String args[]) {
3                 double numeroPI = Math.PI;
4                 String[] pi = new String[3];
5                 pi[0] = numeroPI.toString();
6                 pi[1] = Double.toString(numeroPI);
7                 pi[2] = "" + numeroPI;
8                 for (int i=0; i<pi.length; i++)
9                       System.out.println("Número pi: " + pi[i]);
10          }
11    }
```

A) Se obtiene error en la línea 5.
B) Se obtiene error en las línea 6.
C) Se obtiene error en las líneas 6 y 7.
D) Ninguna de las anteriores.

17. Señale cuál es el resultado de la ejecución del método `main` de la clase Ejemplo1. Las líneas se han numerado con el propósito de facilitar la referencia a las mismas.

```
1     class Ejemplo1_A {
2           String mensaje = "A";
3           void mensajeA() { System.out.print(mensaje); }
4     }
5
6     class Ejemplo1_B extends Ejemplo1_A {
7           String mensaje = "B";
8           void mensajeA() {System.out.print(super.mensaje); }
9           void mensajeB() {System.out.print(mensaje); }
10    }
11
12    public class Ejemplo1 {
13          public static void main(String[] args) {
14                Ejemplo1_A e1 = new Ejemplo1_B();
15                e1.mensajeA();
16                e1.mensajeB();
17          }
18    }
```

A) AB
B) Se obtiene error en la línea número 14.
C) Se obtiene error en la línea número 16.
D) Ninguna de las anteriores.

18. Señale cuál es el resultado de la ejecución del código siguiente. Las líneas se han numerado con el propósito de facilitar la referencia a las mismas.

```
1     public class Ejemplo2 {
2
3           int        a = 1;
4           static int b = 2;
5
6           static int producto(int p1, int p2) {
7                 return p1*p2;
8           }
9
10          public static void main(String[] args) {
11                Ejemplo2 e = new Ejemplo2();
12                int axb_1 = producto(a,b);
13                int axb_2 = producto(e.a,b);
14                System.out.println(axb_1+axb_2);
15          }
16    }
```

A) Se obtiene error en la línea 12.
B) Se obtiene error en la línea 13.
C) 4
D) Ninguno de las anteriores.

19. La clase siguiente no contiene errores. Indique cuál es el resultado de la ejecución de su método `main`.

```
public class Ejemplo3 {
      public static void main(String[] args) {
            int[] a = {1,2,3};
            try {
                  for (int i=0; i<=a.length; i++)
                        System.out.print(a[i]);
                  System.out.print("OK");
            } catch (ArrayIndexOutOfBoundsException e1) {
                  System.out.print("e1");
            } catch (Exception e2) {
                  System.out.print("e2");
            } finally {
                  System.out.print("F");
            }
      }
}
```

A) `123OK`
B) `123OKF`
C) `123e1F`
D) Ninguno de los anteriores.

20. A continuación se muestra el código de la clase Ejemplo4. Se han numerado las líneas para facilitar la referencia a las mismas.

```
1     import java.util.*;
2     public class Ejemplo4 {
3           public static void main(String[] args) {
4                 ArrayList sumandos  = new ArrayList();
5                 for (int i=0; i<3; i++)
6                       sumandos.add(new Integer(i));
7                 for (int i=0; i<sumandos.size(); i++)
8                       sumandos.remove(i);
9                 for (int i=0; i<sumandos.size();i++) {
10                      System.out.println("Elemento " + i + ": " +
11                            (Integer)sumandos.get(i));
12                }
13                System.out.println("FIN");
14          }
15    }
```

Señale cuál de las siguientes afirmaciones es correcta:

A) Se produce un error en la línea 6.
B) 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:
```
FIN
```
C) 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:
```
Elemento 0: 1
FIN
```
D) Ninguna de las anteriores.