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 B.

1. A continuación se muestra el código de la clase *ListaDeArrays1*. Se han numerado las líneas para facilitar la referencia a las mismas.

```
1 package p5;
2 import java.util.*;
3 public class ListaDeArrays1 {
4     public static void main(String args[]) {
5         ArrayList al = new ArrayList();
6         int suma = 0;
7         for (int i=0; i<3; i++)
8             al.add(i);
9         for (int i=0; i<al.size(); i++)
10            suma += al.get(i);
11        System.out.print("Suma: " + suma);
12    }
13 }
```

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 10.
- C) Las dos afirmaciones anteriores son verdaderas.
- 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:

Suma: 3

2. A continuación se muestra el código de dos clases, *Ejemplo1* y *Ejemplo1a*. Ambas clases están dentro del paquete *p1*. Las líneas de la clase *Ejemplo1a* 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 *Ejemplo1a*, guardada en el archivo *p1/Ejemplo1a.java*:

```
1 package p1;
2 public class Ejemplo1a extends Ejemplo1 {
3     Ejemplo1a() {
4         System.out.print("n = " + n);
5         System.out.print("; n_pri = " + n_pri);
6         System.out.print("; n_pro = " + n_pro);
    }
}
```

```

7         System.out.print("; n_pub = " + n_pub);
8     }
9     public static void main(String args[]) {
10        Ejemplola ejla = new Ejemplola();
11    }
12 }

```

Señale cuál de las siguientes afirmaciones es correcta:

- A) Error en la línea 5 de la clase *Ejemplola*.
- B) Error en la línea 5 y en la línea 6 de la clase *Ejemplola*.
- C) El resultado de la ejecución del método *main* de la clase *Ejemplola* es el siguiente:
 $n = 1; n_pri = 2; n_pro = 3; n_pub = 4$
- D) Todas las afirmaciones anteriores son falsas.

3. Señale cuál de las siguientes afirmaciones acerca de la clase *Ejemplo2* es correcta. Las líneas se han numerado con el fin de facilitar la referencia a las mismas:

```

1 package p1;
2 import java.io.*;
3 class Ejemplo2 {
4     public static void main(String args[]) {
5         try {
6             FileReader fr = new FileReader("prueba.txt");
7         } catch (FileNotFoundException e) {
8             System.out.println("Excepción: fichero no encontrado");
9             return;
10        }
11        BufferedReader br = new BufferedReader(fr);
12        String s;
13        try {
14            while ((s=br.readLine())!=null) {
15                System.out.println(s);
16            }
17            fr.close();
18        } catch (IOException e) {
19        }
20    }
21 }

```

- A) Se produce error en la línea 7, ya que el constructor `FileReader(String dirFichero)` no puede producir una excepción `FileNotFoundException`.
- B) Se produce error en la línea 11 y en la línea 17, debido a que se desconoce qué es `fr` (*fr cannot be resolved*).
- C) El programa se ejecuta sin producir errores. Si el fichero *prueba.txt* existe en el directorio actual, el programa lee las líneas del mismo y las imprime al flujo de salida estándar. Si el fichero *prueba.txt* no existe, el programa imprime en el flujo de salida estándar el mensaje "Excepción: fichero no encontrado".
- D) Todas las afirmaciones anteriores son falsas.

4. Indique cuál es el resultado de ejecutar el método *main* de la clase siguiente. Las líneas se han numerado para facilitar la referencia a ellas:

```

1     import javax.swing.*;
2     import java.awt.*;
3     public class Ventana3 extends JFrame {
4         public Ventana3(String texto) {
5             super("Ventana con letrero");
6             setBounds(100,100,300,300);
7             getContentPane().setBackground(Color.white);
8             setDefaultCloseOperation(EXIT_ON_CLOSE);
9             JLabel letrero = new JLabel(texto);
10            getContentPane().add(letrero);

```

```

11         setVisible(true);
12     }
13     public static void main(String args[]) {
14         Ventana3 v = new Ventana3("TEXTO");
15     }
16 }

```

- A) Se obtiene error en la línea 1, ya que no es necesario importar el paquete *javax.swing*
- B) Se obtiene error en la línea 11, ya que el método *setVisible* no es aplicable para argumentos del tipo booleano.
- C) Se muestra una ventana. El título de la ventana es: *Ventana con letrero*. En el interior de la ventana, sobre fondo blanco, se muestra el texto: TEXTO.
- D) Ninguna de las anteriores es correcta.

5. A continuación se muestra el código de tres clases: *Ejemplo4_A*, *Ejemplo4_AB* y *Ejemplo4*. Las líneas de la clase *Ejemplo4* se han numerado para facilitar la referencia a las mismas.

Clase *Ejemplo4_A*, guardada en el fichero p6/Ejemplo4_A.java:

```

package p6;
public class Ejemplo4_A {
    String s;
    public Ejemplo4_A(String s) {
        this.s = s;
    }
    public void Escribe_s() {
        System.out.println("A: " + s);
    }
}

```

Clase *Ejemplo4_AB*, guardada en el fichero p6/Ejemplo4_AB.java:

```

package p6;
public class Ejemplo4_AB extends Ejemplo4_A {
    Ejemplo4_AB(String s) {
        super(s);
    }
    public void Escribe_s() {
        System.out.println("AB: " + s);
    }
    public void Escribe_s_enMayusculas() {
        System.out.println("AB: " + s.toUpperCase());
    }
}

```

Clase *Ejemplo4*, guardada en el fichero p6/Ejemplo4.java:

```

1 package p6;
2 public class Ejemplo4 {
3     public static void main(String[] args) {
4         Ejemplo4_A ej = new Ejemplo4_AB("texto");
5         ej.Escribe_s();
6         ej.Escribe_s_enMayusculas();
7     }
8 }

```

Indique cuál de las siguientes afirmaciones es correcta:

- A) Se produce error en la línea 4 de la clase *Ejemplo4*.
- B) Se produce error en la línea 6 de la clase *Ejemplo4*.
- C) El resultado de la ejecución del método *main* de la clase *Ejemplo4* es el siguiente:
AB: texto
AB: TEXTO
- D) El resultado de la ejecución del método *main* de la clase *Ejemplo4* es el siguiente:
A: texto
AB: TEXTO

6. ¿Cuál de las siguientes afirmaciones es correcta?

- A) El resultado de una consulta SQL a la base de datos mediante el método `executeQuery` se almacena en un objeto `ResultSet`.
- B) Los resultados de una consulta SQL mediante el método `getConnection` se almacenan en un objeto `ResultSet`.
- C) El resultado de una consulta SQL con el método `executeQuery` se almacena en un objeto `String[]`.
- D) Ninguna de las anteriores.

7. La especificación de la URL que hay que elaborar para acceder a una base de datos mediante JDBC debe constar de los siguientes campos:

- A) `jdbc:<subprotocolo>:<nombre_base_de_datos>`.
- B) `jdbc:odbc:<nombre_base_de_datos>`.
- C) `jdbc-odbc:<subprotocolo>:<nombre_base_de_datos>`.
- D) Ninguna de las anteriores.

8.Cuál de los siguientes fragmentos de código es el correcto si se desea realizar la carga y visualización de una página HTML en un applet:

A)	<code>URL u = new URL (getCodeBase(), "pagina.html"); getAppletContext().showDocument (u, " blank");</code>
B)	<code>URI servidor = new URI (getDocumentBase()); URL u = new URL (servidor, "pagina.html"); getAppletContext().showDocument (u, " blank");</code>
C)	<code>getAppletContext().showDocument (getDocumentBase()+ "pagina.html", " blank");</code>
D)	Ninguna de las anteriores.

9. En Swing algunas interfaces de listeners (WindowListener, MouseListener, KeyListener, etc.) contienen muchos métodos lo que obliga a su implementación al crear los correspondientes listeners incluso aunque no sean necesarios. ¿Qué solución existe para evitar esto?

- A) La utilización de las clases adapters. Por ejemplo, para crear un listener de un componente `JFrame` se puede evitar la implementación de todos los métodos incluidos en la interfaz `WindowListener` creando un listener que herede de la clase `WindowAdapter`.
- B) Crear una interfaz adapter, es decir, escribiendo la propia interfaz del listener con los métodos que se consideren necesarios para la aplicación que se está creando. Por ello, si se desea crear un listener para un `JFrame` con un único método que detecte el cierre de la ventana entonces basta con programar una nueva interfaz `WindowAdapter` que contenga un método `WindowClosing` y utilizarla para crear el listener.
- C) No existe solución posible salvo la inclusión de todos los métodos considerados en la interfaz aunque su contenido se deje vacío. Por lo tanto, si se desea crear una clase listener para un objeto del tipo `JFrame` habrá que incluir los siete métodos que contiene la interfaz.
- D) Ninguna de las anteriores.

10. Para fijar el layout a null en un objeto elemento1 de tipo `javax.swing.JFrame` y un objeto elemento2 del tipo `javax.swing.JPanel` elemento2, ¿cuál de las siguientes secuencias de código es la correcta?

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

11. Dado el siguiente código

```
public class Ejer1 {
    public static void main(String args[]) {
        for (int i=0;i<5; i++){
            System.out.print(" Numero i= " + i + ":");
            for(int j=0;j<12;j++){ if (j==10) break;
                System.out.print("Numero j="+j+":");
            }
            System.out.println();
        }
    }
}
```

¿Cuál de las afirmaciones siguientes es correcta?

- A) Cuando se ejecuta **break** por primera vez se sale del segundo bucle y finaliza el programa independientemente del valor de *j*.
- B) Se ejecuta **break** si la variable *i* es igual a 10 y finaliza el programa
- C) Cada vez que se ejecuta **break** se sale del segundo bucle y continúa la ejecución del primer bucle.
- D) Todas las afirmaciones son falsas.

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

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

- A) Saldo=120
- B) Saldo=23
- C) Saldo =143
- D) Ninguna de las anteriores

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

```
class prueba {
    int i;
    public prueba() {
        i=0;} }

public class ejer1 {
    public static void main(String args[]) {
        prueba uno = new prueba();
        prueba dos = new prueba();
        uno = dos;
        uno.i= 20;
        dos.i= 30;
        System.out.println(uno.i);
    }
}
```

- A) 50
- B) 20
- C) 30
- D) Ninguna de ellas.

14. Dado el siguiente código. Determinar qué 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(" uno ");
6.         } catch(Exception e) {
7.             System.out.print(" dos ");
8.             System.exit(0);
9.         } finally {
10.            System.out.print(" final ");
11.            throw new Exception();
12.        }
13.    }
14. }
```

- A) línea 5, dos
- B) línea 7, uno
- C) línea 10, dos
- D) línea 8, final

15. ¿A qué se denomina en Java *Constante Blanca*?

- A) A una constante tipo Array.
- B) A un campo declarado como **final** pero que no se inicializa.
- C) A campos declarados como **final**
- D) A una constante **static**

16. Complete el código marcado como XXXXXX de esta clase que calcula la media del 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; XXXXXXXXXXXXXXX; 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) j=10

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

```
public class Class1 {
public static void main(String[] args) {
int[] a1 = { 1, 2, 3, 4, 5 };
int[] a2={0,0,0,0,0};
a2 = a1;
for(int i = 0; i < a2.length; i++)
++a2[i];
for(int i = 0; i < a1.length; i++)
{
System.out.print(
"a1[" + i + "] = ");
System.out.println(a1[i]);
}
```

```

    }

    try{ System.in.read();
    }
    catch(Exception e){}

}

```

- A)
 - al[0]=2
 - al[1]=3
 - al[2]=4
 - al[3]=5
 - al[4]=6
- B)
 - al[0]=1
 - al[1]=2
 - al[2]=3
 - al[3]=4
 - al[4]=5
- C)
 - al[0]=0
 - al[1]=0
 - al[2]=0
 - al[3]=0
 - al[4]=0
- D) Ninguna de ellas. Existe un error de compilación

18. ¿A qué se denomina en Java *Clase interna*?

- A) A una clase que se define dentro de la definición de otra.
- B) A una clase cuyos parámetros son internos
- C) A una clase cuyos métodos no están sobrecargados.
- D) A una clase cuya definición contiene métodos vacíos.

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

```

public class Caracteres {
    public Caracteres() {
    }
    public static void main(String[] args) {
        char x='a';
        for(int i=0;i<3;i++)
            System.out.print(++x);
    }
}

```

- A) dcb
- B) cbd
- C) bcd
- D) aaa

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

```

class Balance{
    String nombre;
    double bal;

    Balance(String n, double b){
        nombre=n;
        bal=b;
    }
}

```

```

void show(){
    if(bal<0)
        System.out.print("-->");
        System.out.println(nombre + ": &" + bal);
    }
}
public class CuentaBalance {
    public CuentaBalance() {
    }
    public static void main(String[] args) {
        Balance actual[] = new Balance[3];

        actual[0]=new Balance("Pepe",100);
        actual[1]=new Balance("Luis",-100);

        for(int i=0;i<2;i++) actual[i].show();
    }
}

```

- A) Pepe: &100.0
-->Luis: &-100.0
- B) Luis: &100.0
-->Pepe: &-100.0
- C) Pepe: &-100.0
-->Luis: &-100.0
- D) Pepe: &100.0
-->Luis: &100.0