

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 A.

1. Se desea obtener la dirección especial IP conocida como *localhost* y que corresponde al bucle local del computador en el que se ejecuta el programa. Una de las formas genéricas para obtenerla es recurrir a la expresión:
 - A) `InetAddress localAddress = InetAddress.getLocalLoop()`
 - B) `InetAddress localAddress = InetAddress.getByName("localhost")`
 - C) `InetAddress localAddress = InetAddress.getByName(null)`
 - D) Ninguna de las anteriores.
2. Para obtener el número IP que se encuentra asociado a un nombre almacenado en un DNS, se utiliza el método:
 - A) `getByName(String host)` de la clase `IPAddress`.
 - B) `getByName(String host)` de la clase `InetAddress`.
 - C) `getIP(String name)` de la clase `InetAddress`.
 - D) Ninguna de las anteriores.
3. ¿Es posible que un programa utilice simultáneamente varias instancias de la clase `ServerSocket`, estando cada una de ellas a la espera de la llegada de conexiones en un número de puerto diferente?
 - A) Sí. Basta con utilizar diferentes threads y en cada uno de ellos instanciar un objeto del tipo `ServerSocket` con un número de puerto diferente.
 - B) Sí. Se utiliza el constructor `ServerSocket` introduciendo como argumentos un array de objetos `InetAddress` de forma que cada objeto del tipo `InetAddress` contenga un número de puerto diferente.
 - C) No. Un programa servidor sólo puede permanecer a la escucha en un único puerto ya que la máquina virtual de Java no permite la creación de varios `serverSocket` simultáneos en un mismo programa por razones de seguridad.
 - D) Ninguna de las anteriores.
4. Un objeto `ServerSocket` crea un tipo especial de socket, ya que:
 - A) Únicamente permite aceptar conexiones entrantes pero no enviar/recibir paquetes de datos de los clientes.
 - B) Además de aceptar conexiones entrantes se puede utilizar para intercambiar información con los clientes, es decir, paquetes de datos.
 - C) No necesita que se le indique la dirección IP y el puerto en el que permanecer a la espera de conexiones entrantes.
 - D) Ninguna de las anteriores.
5. Una vez establecida la conexión entre un cliente y un servidor, es necesario:
 - A) Que cada parte implicada cree dos streams: uno para el envío de datos mediante el método estático `Socket.getInputStream()` y otro para la recepción mediante el método estático `Socket.getOutputStream()`.
 - B) Que cada parte implicada cree dos streams: uno para la recepción de datos se crea mediante `getInputStream()` y otro para el envío se crea mediante `getOutputStream()`.
 - C) Crear dos streams. El programa que realiza la recepción de los datos tiene que crear un stream de salida mediante el método `getOutputStream()`, mientras que el programa que efectúa el envío de la información recurre al método `getInputStream()`.
 - D) Ninguna de las anteriores.

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

```
{int x=12;
    {int x=100;
        x=x+1;
    }
    System.out.println(x);
}
```

- A) 13
- B) 101
- C) 12
- D) Ninguna de las anteriores

7. ¿Cuál sería el resultado de la ejecución del siguiente fragmento de código?

```
public class Figura {
    public Figura() {
    }
    static int edad;
    String s1[]=new String[4];
    String s2[][]=new String[4][3];
    String s3[][][]=new String[4][5][6];
    String s4[][]=new String[3][];
    static void setEdad(int x){
        edad=x;
    }
    public static void main(String[] args) {
        Figura f = new Figura();
        System.out.println(f.s1[1]);
        System.out.println(f.s2[2][1]);
        System.out.println(f.s3[1][2][3]);
        setEdad(4);
        System.out.println(edad);
    }
}
```

- A) Imprime en consola
null
null
null
4
- B) Imprime en consola
0
0
0
4
- C) Imprime en consola
null
null null
null null null
4
- D) Imprime en consola
0
0 0
0 0 0
4

8. ¿Cuál sería el resultado de la ejecución del siguiente fragmento de código?

```
public class Exc2 {  
    public Exc2() {  
    }  
    public static void main(String[] args) {  
        int a=10;  
        int b=1;  
  
        for(int i=0; i<10; i++){  
            try{  
                b=a/(i);  
            }  
            catch(ArithmeticException e){  
                a=0;  
            }  
        }  
        System.out.println("b:= " +b);  
    }  
}
```

- A) Imprime en consola b:=1
- B) Imprime en consola b:=10
- C) Imprime en consola b:=0
- D) Imprime en consola b:=0.1

9. ¿Cuál sería el resultado de la ejecución del siguiente fragmento de 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) Imprime en consola abc
- B) Imprime en consola cba
- C) Imprime en consola bcd
- D) Imprime en consola cde

10. ¿Cuál sería el resultado de la ejecución del siguiente fragmento de código?

```
import java.io.*;

public class FicheroInOut {

    public static void main(String[] args) throws IOException{
        int i,j;
        String nombreArchivoEntrada="";
        String nombreArchivoSalida="";

        BufferedReader br= new BufferedReader(new InputStreamReader(System.in));

        nombreArchivoEntrada=br.readLine();
        nombreArchivoSalida=br.readLine();

        FileReader entrada;
        FileWriter salida;

        try{
            entrada = new FileReader(nombreArchivoEntrada);
            salida = new FileWriter(nombreArchivoSalida);
            do{
                i=entrada.read();
                if(i!=-1){
                    salida.write((char) i);
                }
            }
            while(i!=-1);
            entrada.close();
            salida.close();
        }
        catch(FileNotFoundException e){
            System.out.println("Archivo no encontrado");
        }
    }
}
```

- A) Copia en el fichero cuyo nombre se encuentra en la variable nombreArchivoSalida, el fichero cuyo nombre se encuentra en la variable nombreArchivoEntrada.
- B) Imprime en consola el fichero cuyo nombre se encuentra en la variable nombreArchivoSalida.
- C) No hace nada debido a un error en la programación de las excepciones.
- D) No hace nada debido a un error en los flujos.

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

```
1      public class Ejemplo1 {
2          public static void main(String[] args) {
3              int precision=2;
4              double pi;
5              if (precision==2) { pi = 3.14; }
6              if (precision==4) { pi = 3.1415; }
7              System.out.println(pi);
8          }
9      }
```

- A) 3.14
- B) 3.1415
- C) Se obtiene error en la línea 7, ya que la variable pi puede no haber sido inicializada.
- D) Se obtiene error en las líneas 5 y 6 por utilizar la variable pi, ya que dentro del bloque if no se tiene acceso a las variables declaradas en un ámbito exterior.

12. Señale cuál es el resultado de la ejecución del método main de la clase Ejemplo3. Las líneas se han numerado con el propósito de facilitar la referencia a las mismas.

```
1      public class Ejemplo3_A {
2          int i = 1;
3          void letreroA() { System.out.print(" A"+i); }
4      }
5
6      public class Ejemplo3_B extends Ejemplo3_A {
7          int i = 2;
8          void letreroA() {System.out.print(" AA"+i); }
9          void letreroB() {System.out.print(" B"+i); }
10     }
11
12     public class Ejemplo3 {
13         public static void main(String[] args) {
14             Ejemplo3_A e1 = new Ejemplo3_B();
15             e1.letreroA();
16             e1.letreroB();
17         }
18     }
```

- A) A1 B2
- B) AA1 B2
- C) Se obtiene error en la línea número 14.
- D) Se obtiene error en la línea número 16.

13. Indique cuál es el resultado de la ejecución del método main de la clase siguiente.

```
public class Ejemplo4 {
    static int a = 10;
    static { a += 1; }
    static int duplica(int a) { return a *= 2; }
    public static void main(String[] args) {
        System.out.println(duplica(a));
    }
}
```

- A) 20
- B) 21
- C) 22
- D) Ninguno de las anteriores.

14. Indique cuál es el resultado de la ejecución del método main de la clase siguiente.

```
public class Ejemplo5 {
    public static void main(String[] args) {
        int x[] = { 1, 2, 3}, y=0; double z;
        try {
            z = x[x.length] / y; System.out.print(" P1");
        } catch (ArrayIndexOutOfBoundsException e1) {
            System.out.print(" P2"); y = 1;
        } catch (Exception e2) {
            System.out.print(" P3");
        }
        System.out.print(" P4");
    }
}
```

- A) P2 P4
- B) P3 P4
- C) P2 P1 P4
- D) Ninguna de las anteriores.

15. Indique cuál es el resultado de la ejecución del método main de la clase Ejemplo7. Las líneas se han numerado con el propósito de facilitar la referencia a las mismas.

```
1      public class Pila {
2          int n=10;
3          int pila[] = new int[n];
4          int puntero;
5          Pila(int n) { puntero = -1; this.n = n; }
6          void push( int elemento ) {
7              if (puntero==(n-1))
8                  System.out.println("Pila llena");
9              else
10                 pila[++puntero]=elemento;
11          }
12      public String toString() { return puntero + 1 + " elementos"; }
13  }
14
15  public class Ejemplo7 {
16      public static void main(String[] args) {
17          Pila p = new Pila(7);
18          for (int i=1; i<4; i++) p.push(i);
19          System.out.println(p);
20      }
21  }
```

- A) Se obtiene error en la línea 12, ya que la clase Pila no puede sobrescribir el método toString().
- B) Se obtiene error en la línea 19, ya que el objeto p no es String.
- C) Las dos respuestas anteriores son verdaderas.
- D) 3 elementos
16. Indicar qué afirmación es correcta:
- A) La variable de entorno CLASSPATH indica dónde se encuentra los ficheros ejecutables del SDK
- B) La variable de entorno CLASSPATH indica dónde se encuentran los ficheros objeto (.class)
- C) La variable de entorno CLASSPATH indica dónde se encuentra los ficheros fuente (.java)
- D) Ninguna de las anteriores
17. Indicar cuál de las siguientes expresiones da como resultado el valor TRUE
- A) (27>=6)|| (5<3)
- B) (27>=6)&& (5<3)
- C) (27>=6)== (5<3)
- D) Ninguna de las anteriores.
18. Una instrucción SWITCH
- A) Puede contener una cláusula default
- B) Debe contener una cláusula default
- C) Puede contener varias cláusulas default
- D) Ninguna de las anteriores.
19. ¿Cuál de las siguientes afirmaciones es falsa?
- A) Las clases pueden contener propiedades y métodos.
- B) Las clases deben contener propiedades y métodos.
- C) Las clases pueden contener sólo propiedades.
- D) Las clases pueden contener sólo métodos.
20. Indicar qué afirmación es correcta:
- A) El nombre de una variable puede empezar con cualquier carácter.
- B) El nombre de una variable está limitado a 18 caracteres.
- C) El nombre de una variable no debe tener espacios en blanco.
- D) Ninguna de las anteriores.