

ANTES DE COMENZAR A REALIZAR LA PRUEBA DEBE LEER LAS SIGUIENTES NORMAS
(Solamente debe entregar al tribunal, una hoja de lectura óptica con sus datos y respuestas)

Material: NINGUNO

IMPORTANTE

Si encuentra alguna anomalía en el enunciado, indique ésta en el reverso de la hoja de lectura óptica (o si fuera estrictamente necesario en una hoja adjunta) y argumente la solución adoptada al efecto. Estos comentarios serán de gran importancia ante posibles reclamaciones en la revisión de exámenes. Sólo el Equipo Docente podrá anular preguntas del examen.

1. Deberá entregar ÚNICAMENTE la hoja de lectura óptica con sus datos y respuestas.
2. La prueba consta de un test de 20 preguntas a contestar en una hoja de lectura óptica. Lea atentamente las instrucciones que figuran en la hoja de lectura óptica (no olvide poner su DNI, Código de Carrera, Código de Asignatura y Tipo de Examen).
3. Para superar la prueba se deberá obtener una puntuación mínima de 5 puntos. En cada pregunta del test se proponen cuatro respuestas de las cuales sólo una es correcta. Únicamente puntuarán las preguntas contestadas. Si la respuesta es correcta la puntuación será de 0.5 puntos y si es incorrecta restará 0.25 puntos.

PREGUNTAS

1. ¿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.

2. 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        }
18    }
19 }
```

```

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.

Comentario [A1]: Falsa. Java 2, H. Schildt, pág. 478.

Comentario [A2]: Correcta. El ámbito de definición de `fr` es el bloque `try` de las líneas 5-7.

Comentario [A3]: Falta, puesto que B) es verdadera.

3. 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:

Comentario [A4]: Respuesta correcta: B)

AB: texto
AB: TEXTO

D) El resultado de la ejecución del método *main* de la clase Ejemplo4 es el siguiente:

A: texto
AB: TEXTO

4. ¿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 error. No se pueden añadir dos elementos iguales a un objeto TreeSet.

5. ¿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 error. Es necesario realizar un *casting* de lectura.

6. ¿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* solo funciona con objetos *InputStream*

7. 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

8. ¿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) null3456

9. ¿Cuál es el resultado de la ejecución del método *main* del siguiente código ?

```
class Prueba1{
    int x;
    char y;
}

class Prueba2 extends Prueba1{
    int z;
    int x=3;
}

public class Prueba3 extends Prueba2{
    public Prueba3() {
        int x=5;
    }
    public static void main(String[] args) {
        Prueba1 AAA = new Prueba3();
        AAA.x=1;
        System.out.println(AAA.x);
    }
}
```

- A) Imprime en consola el entero 1
- B) Ninguna. No se puede asignar un manejador de una clase a un objeto de otra
- C) Imprime en consola el entero 5
- D) Imprime en consola el entero 3

10. ¿Cuál es el resultado de la ejecución del método *main* del siguiente 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	B)Imprime en consola	C)Imprime en consola	D)Imprime en consola
null	0	null	0
null	0	null null	0 0
null	0	null null null	0 0 0
4	4	4	4

11. La clase *Escribe* se define de la manera siguiente (se han numerado las sentencias para facilitar su localización):

```
1 public class Escribe {
2     public String Saludo() {
3         return "Hola";
4     }
5     public static void main(String arg[]) {
6         System.out.println(Saludo());
7     }
8 }
```

Indique cuál de las siguientes afirmaciones es correcta:

- A) Se obtiene un error de compilación en la línea 2: la declaración del método no es válida puesto que no se indica el tipo de retorno.
- B) Se obtiene un error de compilación en la línea 6: el método no estático *Saludo()* no puede ser referenciado desde un contexto estático.

- C) Las dos afirmaciones anteriores son correctas.
- D) La clase es correcta y se compila sin errores.

12. Indique si son correctas las siguientes afirmaciones acerca de las clases abstractas:

- I. En una clase abstracta pueden declararse métodos pero no definirse.
- II. En una clase abstracta, la declaración de los métodos abstractos puede no contener el tipo de retorno (return type). Por ejemplo, sería posible definir el método `perimetro()` de la forma siguiente `public abstract perimetro()`. Esto facilita la sobrecarga de los métodos.

- A) I: Si ; II: Si
- B) I: Si ; II: No
- C) I: No ; II: Si
- D) I: No ; II: No**

13. Indicar cuál de las siguientes respuestas corresponde con el resultado de la ejecución del método `main` de la clase `ClaseB`:

<pre> abstract class ClaseA { String v; void metodo() { v = "Padre"; } } public class ClaseB extends ClaseA { String v; void metodo() { v = "Hija"; super.metodo(); System.out.print(v+" "); System.out.println(super.v); } public static void main(String args[]) { ClaseB b = new ClaseB(); System.out.print(b.v+" "); b.metodo(); } } </pre>	<ul style="list-style-type: none"> A) null Hija Padre B) null Padre Padre C) Padre Padre D) Hija Padre Padre
--	---

14. Indique cuál de las siguientes afirmaciones, relacionadas con el ámbito (scope) de las variables, es correcta:

- A) Las funciones miembro de una clase B derivada de otra A (es decir, A es superclase de B), tienen acceso a todas las variables miembro de A declaradas como `public` o `private` pero no a las declaradas como `protected`.
- B) Es posible declarar una variable dentro de un bloque (es decir, dentro de unas llaves `{}`) con el mismo nombre que una variable miembro. La variable declarada dentro del bloque oculta a la variable miembro en ese bloque. Para acceder a la variable miembro oculta será preciso utilizar el operador `super`.
- C) Las dos afirmaciones son correctas
- D) Las dos afirmaciones son falsas.**

15. El contenido de un bloque `finally`:

- A) Solo se ejecuta cuando la excepción ha sido previamente capturada y procesada.
- B) Se ejecuta únicamente cuando la excepción lanzada no ha sido procesada por un bloque `catch`
- C) No se ejecuta si existe un bloque `catch` que capture la excepción que se ha producido

D) Se ejecuta independientemente de que las excepciones sean o no capturadas.

16. Considérese el siguiente fragmento de código:

```
ClaseA var1 = new ClaseA();  
ClaseA var2 = var1;
```

Indique cuál de las siguientes respuestas es correcta:

- A) var1 y var2 son dos instancias de la clase ClaseA
- B) var1 y var2 son el mismo objeto de la clase A
- C) var1 y var 2 son dos referencias a un mismo objeto de la ClaseA
- D) var1 es un objeto de la clase ClaseA. Var2 es una referencia al objeto var1.

17. Supóngase que se tiene definida una clase denominada `perro`. ¿Qué efecto produce la siguiente sentencia?

```
new perro()
```

- A) Inicializa un elemento de la clase `perro`
- B) Llama al constructor por defecto de la clase `perro`
- C) Reserva espacio de memoria para un objeto de la clase `perro`
- D) Todas las anteriores

18. Dentro de una unidad de compilación:

- A) Debe haber una clase pública
- B) Pueden existir varias clases públicas
- C) Puede haber una única clase pública
- D) Ninguna de las anteriores

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

```
class perro{  
    perro(int i) {  
        System.out.println(i);  
    }  
}  
public class mastin extends perro{}  
  
    public static void main(String[] args){  
        mastin chucho=new mastin(1);  
    }  
}
```

- A) null
- B) 1
- C) 0
- D) Ninguna, el código es erróneo.

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

```
public class Suma {  
    public static void main(String[] args) {  
        int x=12;  
        {int x=100;
```

```
        x=x+1;  
    }  
    System.out.println(x);  
    }  
}
```

- A) 13
- B) 101
- C) 12
- D) Ninguna, el código es erróneo.