

# Cuestionario.

**Instrucciones:** Cada pregunta vale 1 punto , en caso de respuesta múltiple sólo se considerará correcta si se seleccionan todas y sólo las opciones correctas.

Las respuestas incorrectas no restan puntos.

La nota del cuestionario supondrá un **20%** de la nota final.

La nota de la parte práctica supondrá un **80%** de la nota final.

Tiempo para la realización : **40 minutos**.

---

1. Indica cuál o cuáles de las siguientes afirmaciones es falsa:

- ☐ Las variables numéricas de instancia se inicializan a cero por defecto.
- ☐ Las variables numéricas locales (interior de métodos) no se inicializan a ningún valor.
- ☐ **Las variables estáticas no pueden ser privadas.**
- ☐ Para llamar a un método estático no es necesario crear un objeto de la clase en la que está declarado

2. Dada la siguiente clase:

```
public Class Serie{  
  
    private String titulo;  
    private float valoracion;  
  
    public Serie(String titulo , float valoracion){  
  
        this.titulo = titulo;  
        this.valoracion = valoracion;  
    }  
}
```

¿Cuál o cuáles de las siguientes sentencias son correctas para crear objetos de tipo Serie ?

- ☐ `Serie s = new Serie();`
- ☐ `new Serie("Stranger things",9);`
- ☐ `new Serie();`
- ☐ `Serie s = new Serie("Los Soprano",9.5f);`
- ☐ Todas las anteriores son correctas.

3. Teniendo en cuenta las reglas de conversión numérica de variables en Java de qué tipo sería la variable **suma** en el siguiente fragmento de código:

```
byte x = 10;  
byte y = 9;  
? suma = x + y;
```

- ☐ long
- ☐ short
- ☐ int
- ☐ double
- ☐ byte

4. ¿Cuál o cuáles de las siguientes líneas daría error de compilación?

- ☐ `int x= 10/7;`
- ☐ `double x = 5.6;`
- ☐ `float x = 5.6;`
- ☐ `byte x = 300;`

5. ¿Cuál de los siguientes paquetes no tiene que ser importado para utilizar sus clases?

- ☐ java.util
- ☒ **java.lang**
- ☐ java.io
- ☐ javax.swing
- ☐ java.math

6. Completa el código para poder almacenar correctamente el valor de la variable String en la variable int:

```
String cadena = "45";  
int numero = _____ (cadena);
```

- ☐ String.parseInt(cadena);
- ☒ **Integer.parseInt(cadena);**
- ☐ Integer.toString(cadena);
- ☐ cadena.parseInt();
- ☐ Integer.parseInt(cadena);

7. Indica cuál es la salida del siguiente programa:

```
String cadena = "1er Curso DAW";
```

```
System.out.println(cadena.substring(cadena.indexOf("C"),cadena.indexOf("o"));
```

- ☐ Curso
- ☒ **Curs**
- ☐ urs
- ☐ urso
- ☐ Error de compilación.

8. ¿Cuál es la salida del siguiente programa ?

```
String cad = "hola caracola";  
String cad2 = "adios caracola";  
cad2 = cad;  
System.out.println(cad.equals(cad2) + " , " + cadena.indexOf(" "));
```

☐ false , 4

☒ **true , 4**

☐ false , 5

☐ true , 5

9. ¿Cuál de los siguientes métodos debemos ejecutar antes de realizar una búsqueda en una matriz con el método `Arrays.binarySearch()`

☐ `Arrays.order();`

☒ **`Arrays.sort();`**

☐ `Arrays.equals();`

☐ `Arrays.copyOf();`

☐ No es necesario llamar a ningún método previamente.

10. Dada la siguiente matriz 2D :

```
int matriz[][] = {{1,2,3},{1,2,3}};
```

¿Qué valor me devuelve `matriz.length` ?

☐ 6

☐ 3

☒ **2**

☐ Error de compilación

11. Dada las siguientes clases:

<pre><b>class A</b>{     private String valor;     private int numero;     public A(String valor){         this.valor = valor;     } }</pre>	<pre><b>Class B extends A</b>{     private float puntos;     } }</pre>
--	--

Indica cual o cuales de los siguientes es un constructor válido para la clase B

- ☐ **public B() { }**
- ☐ **public B(String valor , float puntos) {  
 super(valor);  
  
 this.puntos = puntos; }**
- ☐ **public void B(String valor , float puntos) {  
 super(valor);  
  
 this.puntos = puntos; }**
- ☐ **public B(String valor , float puntos) {  
 this.puntos = puntos;  
  
 super(valor);  
  
 }**

12. ¿Que valor almacenan las variables booleanas c y d ?

*int j = 4 , k = 5;*

*boolean c = !(j==4 && k > 9 );*

*boolean d = !c || ( ! k == 5 ) ;*

- ☐ true , true
- ☐ false , false
- ☐ false , true
- ☐ **true , false**

13. ¿Que se verá por pantalla ?

```
int x = 2;           int i = 3;
x++;                while(i < y) {
int y = x*4;         if( i % 2 == 0){
                     break;
                     }
                     System.out.print(i + " ");
                     }
```

- ☐ 3, 5 , 7 , 9 , 11
- ☐ 4 , 6 , 8 , 10 , 12
- ☒ 3 3 3 3 3 3 3 3 ... bucle infinito
- ☐ 4 4 4 4 4 4 4 4 ... bucle infinito

14. ¿Cuál es el valor de la variable x?

```
float n = 20.5f;
int n2 = (int) Math.floor(n);
int x = 20;
if(n2%3==0){
    x++; }
else if(n2%2 == 0) {
    x-=10;
}
else{
    x = 0;
}
```

- ☐ 20
- ☒ 10
- ☐ 0
- ☐ -10

15. Dadas las siguientes clases :

```
public interface Saltarin {}  
  
public class Rana implements Saltarin { }  
  
public class RanaPortuguesa extends Rana {}  
  
public class RanaBrasilenha extends Rana {}
```

Cuál de las siguientes sentencias pueden ser insertadas en la línea para que el código compile:

\_\_\_\_\_ rana = new RanaPortuguesa();

☐ Object

☐ Saltarin

☐ RanaPortuguesa

☐ Rana

☐ RanaBrasilenha

16. ¿Cuál de los siguientes operadores puede ser utilizado para consultar de que tipo es una referencia?:

☐ ==

☐ ?

☐ instanceof

☐ type

☐ referenceof

17. ¿Cuál de las siguientes interfaces define un criterio de ordenación para las clases?

☐ Sortable

☐ **Comparable**

☐ Compare

☐ Ordenator

☐ Rastreator

18.Cuál de las siguientes sentencias es correcta para indicar que la clase X hereda de la clase A e implementas la interfaces Z e Y :

☐ class X implements A extends Z , Y

☐ class X extends Z , Y implements A

☐ **class X extends A implements Z , Y**

☐ No se puede implementar más que una interfaz

19.Cuál de las siguientes afirmaciones son ciertas

☐ **Una clase no puede ser abstracta y final al mismo tiempo.**

☐ Una clase abstracta no puede contener métodos no abstractos.

☐ **Una clase no abstracta no puede contener métodos abstractos.**

☐ Una clase final puede ser extendida por otra clase final.

20. Necesito un algoritmo que me permita asociar un DNI a los datos de un cliente de la empresa y me permita realizar búsquedas muy rápidas a partir del DNI . ¿Qué tipo de colección sería más útil para este caso?

☐ HashMap<Cliente , String>

☐ Set<Cliente>

☐ ArrayList<String>

☐ ArrayList<Cliente>

☐ **HashMap<String , Cliente>**