

PARTE TEÓRICA - TEST [2,5 PUNTOS]:

Sólo una de las respuestas es válida. Las respuestas correctas se puntuarán con +1.0, mientras que las respondidas de manera incorrecta se puntuarán con -0.25. Las no contestadas no tendrán influencia ni positiva ni negativa en la nota.

Pregunta 1: Sobre los objetos, se puede decir (indica la respuesta **falsa**)

- a. Los objetos son especificados por las clases.
- b. Los objetos representan instancias de las clases.
- c. Los objetos se comunican con las clases con métodos.
- d. Una clase puede manejar objetos.

Pregunta 2: Dado este fragmento de código, ¿cuál sería el resultado de compilar/ejecutar el código?

Número de Línea	Código
4	public static void main(String[] args) {
5	int valor = 5;
6	cambiarValor(valor);
7	System.out.println(valor);
8	}
9	public static void cambiarValor(int valor) {
10	private int valor = valor * 2;
11	}

- a. 5
- b. 10
- c. Error en la línea 10
- d. Error en la línea 6

Pregunta 3: Dado el siguiente código, ¿cuál será su salida?

Número de Línea	Código
4	class Cantante { public static String cantar() { return "la"; } }
5	public class Tenor extends Cantante {
6	public static String cantar() { return "fa"; }
7	public static void main(String[] args) {
8	Tenor t = new Tenor();
9	Cantante s = new Tenor();
10	System.out.println(t.cantar() + " " + s.cantar());
11	}
12	}

- a. fa fa
- b. fa la
- c. la la
- d. la fa

Pregunta 4: ¿Cuál de los siguientes condicionales compilaría sin errores?

```
int[] array = new int[15];  
    // EL CÓDIGO IRÍA AQUÍ  
array[j] = j;
```

- a. for (int j=0; j<array.length;j++)
- b. for (int j=0; j<array.length();j++)
- c. for (int j=0; j<array.size;j++)
- d. for (int j=0; j<array.size();j++)

Pregunta 5: Tienes que hacer una clase que almacena objetos únicos. No es necesario que estén ordenados. ¿Qué interfaz sería la más apropiada implementar en esta clase?

- a. Set
- b. List
- c. Map
- d. Vector

Pregunta 6: Cuando varios componentes de un software colaboran para completar una misma tarea se dice que entre ellos hay ...

- a. una clase clara y bien definida.
- b. una instancia clara y bien definida.
- c. una interfaz clara y bien definida.
- d. un proceso claro y bien definido.

Pregunta 7: Dado el siguiente código, ¿cuál será su salida?

Número de Línea	Código
4	class Vehiculo {
5	public void imprimirSonido() {
6	System.out.print("Vehiculo");
7	}
8	}
9	class Coche extends Vehiculo {
10	public void imprimirSonido() {
11	System.out.print("Coche");
12	}
13	}
14	class Bicicleta extends Vehiculo {
15	public void imprimirSonido() {
16	System.out.print("Bicicleta");
17	}
18	}
19	public class Test {
20	public static void main(String[] args) {
21	Vehiculo v = new Coche();
22	Bicicleta b = (Bicicleta) v;
23	v.imprimirSonido();
24	b.imprimirSonido();
25	}
26	}

- a. Fallo de compilación.
- b. Lanza una excepción en tiempo de ejecución.
- c. Imprime "VehiculoCoche".
- d. Imprime "BicicletaBicicleta".

Pregunta 8: ¿Qué pasará si se compila / ejecuta este código?

Número de Línea	Código
4	class Padre {}
5	class Hijo extends Padre {}
6	class Hijo2 extends Padre {}
7	public class CEx{
8	public static void main(String[] args){
9	Padre p=new Padre();
10	Hijo h=(Hijo) p;
11	}
12	}

- a. El código compilará y se ejecutará sin errores.
- b. El código daría un error a compilar.
- c. El código daría un error a ejecutar.
- d. El código no daría ningún error; sin embargo, h no tendría el tipo deseado.

Pregunta 9: ¿Cuál sería el resultado de ejecutar el método goo () ?

Número de Línea	Código
4	public void goo() {
5	foo f = new foo();
6	System.out.println(f);
7	}
8	public class foo {
9	String f = "22";
10	public String toString(){
11	return("44");
12	}
13	public foo(){}
14	}

- a. null
- b. 22
- c. 44
- d. Un error de ejecución

Pregunta 10: ¿Cuál de las siguientes definiciones es correcta para una clase abstracta?

- a. abstract Animal {abstract void ladrar();}
- b. class abstract Animal {abstract void ladrar();}
- c. abstract class Animal {abstract void ladrar();}
- d. abstract class Animal {abstract void ladrar() {System.out.println("RRRRRRR");}}

Pregunta 11: ¿Cómo podemos detectar que el usuario ha hecho click en un botón en una interfaz Swing?

- a. Implementando public void actionPerformed(ActionEvent e) de la interfaz ActionListener
- b. Implementando public void actionPerformed(ActionEvent e) de la interfaz ActionListener
- c. Implementando public boolean actionPerformed(ActionEvent e) de la interfaz ActionListener
- d. Implementando public void actionPerformed(Event e) de la interfaz ActionListener

Pregunta 12: Las sentencias de código que podrían lanzar una excepción se protegen de la siguiente manera:

```
try {  
    // aquí se protege una o más sentencias  
}  
_____  
    // aquí se informa y se recupera de la excepción  
}
```

¿Qué habría que colocar en el hueco correspondiente?

- a. catch (exception Exception)
- b. catch (Event exception)
- c. catch (Exception exception)
- d. catch (event Exception)

Pregunta 13: Las clases de un sistema corresponden a las X y los métodos a las Y. Donde X y Y son ...

- a. X = verbos, Y = sustantivos
- b. X = sustantivos, Y = verbos
- c. X = sustantivos, Y = sustantivos
- d. X = verbos, Y = verbos

Pregunta 14: ¿Cómo se puede crear una nueva instancia de la clase Vector y añadir un elemento? (indica la respuesta falsa)

- a. `Vector<Integer> v = new Vector<Integer>(); v.add(99);`
- b. `Vector<Integer> v = new Vector<Integer>(99); v.add(99);`
- c. `Vector<Integer> v = new Vector<Integer>(99, 99); v.add(99);`
- d. `Vector<Integer> v = new Vector<Integer>(99); v.add(99, 99);`

Pregunta 15: Sobre una variable local que se declara dentro del bloque "try", se puede decir que ...

- a. es visible dentro de los bloques "catch" y "finally".
- b. es visible dentro del bloque "catch" pero no del bloque "finally".
- c. es visible dentro del bloque "finally" pero no del bloque "catch".
- d. no es visible dentro de los bloques "catch" y "finally".

PARTE PRÁCTICA [6,5 PUNTOS]:

Una empresa de alquiler de automóviles tiene a su disposición un conjunto de vehículos indicados en la siguiente tabla. Se quiere diseñar e implementar un programa que almacene y gestione la información relacionada con estos vehículos.

Tipo de vehículo	Características
Motos	marca, matrícula, número de identificación, número de kilómetros, estado actual de depósito de gasolina.
Coches (turismos)	marca, matrícula, número de identificación, tipo (normal / familiar), número de puertas, número de kilómetros, tipo de motor (gasolina / gasoil), estado actual del depósito.
Coches (deportivos)	marca, matrícula, número de identificación, capacidad de motor, número de kilómetros, turbo o no, número de puertas, número de asientos, estado actual del depósito de gasolina.
Coches (4x4)	marca, matrícula, número de identificación, número de kilómetros, tipo de motor (gasolina / gasoil), número de asientos, estado actual de depósito.
Monovolúmenes	marca, matrícula, número de identificación, número de kilómetros, número de puertas, puertas laterales, número de asientos, tipo de motor (gasolina / gasoil), capacidad de carga, estado actual del depósito.
Furgonetas	marca, matrícula, número de identificación, número de kilómetros, capacidad de carga, altura, estado actual del depósito de gasoil.

Se pide:

- 1) **[1,5 puntos]** Identificar la estructura y las relaciones de herencia y de uso de las clases necesarias para almacenar y gestionar esta información.
- 2) **[1,5 puntos]** Dibujar un esquema de la organización de estas clases en el diseño global.
- 3) **[2,0 puntos]** Implementar la especificación de las clases.
- 4) **[1,5 puntos]** Se quiere declarar un array de objetos para almacenar todos los vehículos, independientemente del tipo.
 - a) ¿Cómo declararías el array?
 - b) ¿Qué métodos se necesitan para acceder a un vehículo concreto?
 - c) ¿Cómo se almacenan las diferencias entre los distintos tipos de vehículos?