

# vale , como no puedes darme las 50 preguntas seguidas, no se porque, dame una primera respuesta con 26 preguntas

y una segunda respuesta de la pregunta 27 hasta el final con la solución.

Entendido. Aquí tienes las primeras 26 preguntas del cuestionario:

**1. Pregunta:** Indica cuál de las siguientes afirmaciones es incorrecta:

- a. Las clases definen tipos.
- b. Los programas de Java muestran las clases de una aplicación y la relación entre ellos.
- c. Las clases crean objetos.
- d. Las clases definen métodos.

**2. Pregunta:** Teniendo en cuenta Mayúsculas/Minúsculas, ¿Cuál de las siguientes es una palabra reservada en Java?:

- a. NULL
- b. New
- c. instanceof
- d. wend

**3. Pregunta:** ¿Cuál de las siguientes instrucciones compila sin provocar un warning o un error?

- a. `char c= "a";`
- b. `byte b=257;`
- c. `boolean b=null;`
- d. `int i=10;`

**4. Pregunta:** ¿Cuál de los siguientes no es un identificador válido en Java?

- a. `#variable`
- b. `$variable`
- c. `_variable`
- d. `variable`

**5. Pregunta:** Dada la declaración de las siguientes variables, indica cuáles de ellas son correctas.

```
1. float foo = -1;
2. float fool = 1.0;
```

```
3. float foo3 = 1.02f;  
4. float foo4 = 3.03d;
```

- a. 1 y 2
- b. 1 y 4
- c. 1, 3
- d. 3 y 4

**6. Pregunta:** Respecto a los bucles, indique cuál de las siguientes afirmaciones es falsa:

- a. El cuerpo de un bucle for-each puede repetirse 0 o más veces.
- b. Un bucle for-each puede aplicarse sobre cualquier clase que implemente la interfaz Iterable.
- c. El cuerpo de un bucle while siempre se ejecute que implemente la interfaz iterable
- d. Un bucle for-each puede aplicarse sobre arreglos (arrays).

**7. Pregunta:**

```
return j;  
}  
  
public static void main (String []args) {  
    System.out.println (new Ejemplo().i);  
}
```

- a. Da un error de compilación debido a las restricciones de acceso a las variables privadas de Ejemplo.
- b. Da un error de compilación debido a la referencia que se hace a métodos declarados con posterioridad.
- c. No da ningún error de compilación y produce como salida el valor 0.
- d. No da ningún error de compilación y produce como salida el valor 10.

**8. Pregunta:** ¿Qué elementos crees que definen a un objeto?

- a. Sus cardinalidad y su tipo.
- b. Sus atributos y sus métodos
- c. La forma en que establece comunicación e intercambia mensajes
- d. Su interfaz y los eventos asociados

**9. Pregunta:** ¿Qué significa instanciar una clase?

- a. Duplicar una clase
- b. Eliminar una clase
- c. Crear un objeto a partir de la clase
- d. Conectar dos clases entre sí

**10. Pregunta:** Indica cuál de las siguientes afirmaciones es correcta:

- a. La firma es el encabezado de un método y proporciona la información necesaria para invocarlo.
- b. La firma está formada por los parámetros de un método y proporciona la información

necesaria para invocarlo.

c. La firma es el nombre de un método y puede tener parámetros para proporcionar la información necesaria para realizar una tarea.

d. La firma es el encabezado del método y puede tener parámetros para proporcionar información adicional para realizar una tarea.

**11. Pregunta:** ¿Cuál de las siguientes sentencias son correctas?

```
(1) int w = (int) 888.8;  
(2) byte x = (byte) 1000L;  
(3) long y = (byte) 100;  
(4) byte z = (byte) 100L;
```

a. 1 y 2.

b. 2 y 3.

c. 3 y 4.

d. Todas son correctas.

**12. Pregunta:** ¿Cuál es el resultado de ejecutar el siguiente código?

```
public class Ejemplo{  
    private int i=dameJ();  
    private int j=10;
```

**13. Pregunta:** Dado el siguiente fragmento de código, indica cuál de las siguientes afirmaciones es correcta en relación al valor de la variable `foo`.

```
int index = 1;  
boolean[] test = new boolean[3];  
boolean foo = test [index];
```

a. `foo` tiene el valor 0

b. `foo` tiene el valor `null`

c. `foo` tiene el valor `false`

d. Se produce una excepción y `foo` no posee ningún valor.

**14. Pregunta:** Indique el resultado de ejecutar el siguiente código que se muestra a continuación:

```
public class Test {  
    public static void add3 (Integer i) {  
        int val = i.intValue();  
        val += 3;  
        i = new Integer (val);  
    }  
  
    public static void main (String args[]) {  
        Integer i = new Integer (0);  
        add3 (i);
```

```
        System.out.println (i.intValue());  
    }  
}
```

- a. El programa indicará un fallo en tiempo de compilación.
- b. El programa imprime por pantalla el valor "0".
- c. El programa imprime por pantalla el valor "3".
- d. El programa lanzará una excepción en la línea: `int val = i.intValue();`.

**15. Pregunta:** Indica cuál de las siguientes afirmaciones es correcta:

- a. Los métodos pueden devolver información de algún objeto mediante un valor de retorno.
- b. Los métodos siempre tienen parámetros con los que obtener la información necesaria.
- c. A partir de una clase tan solo se puede crear un solo objeto.
- d. El estado de los objetos se representa mediante los parámetros de su constructor.

**16. Pregunta:** Indica cuál de las siguientes afirmaciones es correcta:

- a. Los atributos se conocen como variables de objeto.
- b. El alcance de una variable define la sección de código desde donde la variable puede ser declarada.

**17. Pregunta:** Indica cuáles de las siguientes expresiones resultan verdaderas:

1. `(1 < 4 && 5)`
2. `(2 > 2 || (4 == 4) && !(3 < 0) )`
3. `!(0 > 1 || (4 != 4) && (3 < 0) )`
4. `(2 > 2 || !(4 == 4) && !(1 < 0) )`
5. `(34 != 33) && !false`

- a. Las expresiones 3 y 4.
- b. Las expresiones 2 y 4.
- c. Las expresiones 3 y 5.
- d. Las expresiones 4 y 5.

**18. Pregunta:** Dadas las siguientes expresiones, indica cuál de las opciones es la correcta.

1. `(1 < 1) && (1 >= 1) == (1 > 1) == false`
2. `(1 == 1) || (10 > 1) == true || true == true`

- a. La expresión 1 es evaluada como falsa y la expresión 2 como falsa.
- b. La expresión 1 es evaluada como falsa y la expresión 2 como verdadera.
- c. La expresión 1 es evaluada como verdadera y la expresión 2 como falsa.
- d. La expresión 1 es evaluada como verdadera y la expresión 2 como verdadera.

**19. Pregunta:** el alcance de una variable...

- a. Define la forma en la que la variable puede ser accedida.
- b. Define el conjunto de métodos que puede acceder a la variable.

- c. Define la sección de código en la que la variable puede ser accedida.
- d. Ninguna de las anteriores.

**20. Pregunta:** ¿Cuál es la descripción que crees que define mejor el concepto 'clase' en la programación orientada a objetos?

- a. Es un concepto similar al de 'array'
- b. Es un tipo particular de variable
- c. Es un modelo o plantilla a partir de la cual creamos objetos
- d. Es una categoría de datos ordenada secuencialmente

**21. Pregunta:** Indica cuál de las siguientes afirmaciones es correcta:

- a. Los atributos o campos se definen dentro de los constructores y de los métodos.
- b. Los atributos se usan para almacenar datos que nunca persisten durante la vida del objeto.
- c. Los atributos tienen un tiempo de vida que perdura después de terminar el objeto.
- d. La accesibilidad de los atributos se extiende a toda clase y por este motivo pueden usarse dentro de cualquier constructor o método de clase en la que estén definidos.

**22. Pregunta:** Dado el siguiente código, el resultado será:

```
class MiClase { public int valor;}

class Test {
    public static void main(String[] args){
        MiClase a1 = new MiClase();
        MiClase a2 = new MiClase();
        MiClase a3 = new MiClase();
        a1.valor=150;
        a2.valor=150;
        a1 = a2;
        if (a1 == a2){ System.out.println ("UNO");};
        if (a1 == a3){ System.out.println ("DOS");};
        if (a2 == a3){ System.out.println ("TRES");};
    }
}
```

- a. UNO
- b. UNO DOS
- c. UNO DOS TRES

**23. Pregunta:** Indique cuál de las siguientes afirmaciones es falsa:

- a. El objetivo de la sobrecarga de métodos es facilitar la invocación de un mismo método pasándole un conjunto de parámetros de entrada diferentes.
- b. Se puede sobrecargar un método variando el tipo de retorno de éste sin variar los parámetros de entrada.
- c. Un método puede ser sobrecargado en la misma clase o en una subclase.
- d. Los métodos sobrecargados pueden cambiar el modificador de acceso del método original.

**24. Pregunta:** Sea la siguiente definición de clase:

```
public class ClaseA {  
    public ClaseA(String s) { System.out.print ("Construyendo Clase A."); } }
```

y la siguiente definición de una subclase:

```
public class ClaseB extends ClaseA {  
    public ClaseB(String s) {  
        System.out.print ("Construyendo Clase B.");  
        super(s);  
    }  
  
    public static void main(String[] args) {  
        new ClaseB("Objeto Clase B");  
        System.out.println (" ");  
    }  
}
```

¿Cuál de las siguientes afirmaciones es cierta al ejecutar el código?

- a. Se produce un error en tiempo de ejecución.
- b. Se muestra el mensaje "Construyendo Clase B. Construyendo Clase A.".
- c. Se muestra el mensaje "Construyendo Clase A. Construyendo Clase B.".
- d. Error de compilación.

**25. Pregunta:** Dados las siguientes definiciones de clases:

```
abstract class Disparo {  
    protected int velocidad=10;  
    abstract public void disparar();  
}  
  
public class DisparoUFO extends Disparo {  
    public void disparar() {  
        this.velocidad=20;  
        System.out.println("Dispara la nave");  
    }  
}  
  
public class DisparoNave extends Disparo {  
    public void disparar() {  
        this.velocidad=10;  
        System.out.println("Dispara la nave");  
    }  
}  
  
public class TestUFO {  
    public static void main(String[] args) {  
        DisparoNave disparoNave = new DisparoNave();  
        new TestUFO().inicio(disparoNave);  
    }  
  
    public void inicio(Disparo d) {  
        d.disparar();  
    }  
}
```

```
}  
}
```

Podemos afirmar:

- a. El método disparar está sobrecargado.
- b. Muestra por pantalla el mensaje "Dispara la nave".
- c. No se muestra por pantalla ningún mensaje.
- d. Obtenemos un error en tiempo de ejecución.

**26. Pregunta:** La siguiente figura muestra una captura de pantalla del editor Eclipse con una línea de código marcada. Indica cuál de las siguientes afirmaciones es correcta en relación a la línea recuadrada:

```
public class Main {  
  
    public void showInstanceOf() {  
        Vehicle vehicle;  
        vehicle = new Vehicle(2, "azul");  
        System.out.println(vehicle instanceof Vehicle);//true  
    }  
  
    public static void main(String[] args) {  
        new Main().showInstanceOf();  
    }  
}
```

- a. Muestra un error en tiempo de ejecución.
- b. Muestra un error de compilación.
- c. Muestra un punto de interrupción.
- d. Muestra una el lanzamiento de una excepción.