

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: ¿Qué incluye como mínimo la descripción de un patrón?

- a. Un nombre, una descripción del problema, una descripción de la solución y las consecuencias del uso del patrón.
- b. Un nombre, una clase, una descripción del problema y las consecuencias del uso del patrón.
- c. Una clase, una descripción del problema, una descripción de la solución y las consecuencias del uso del patrón.
- d. Un nombre, una clase, una descripción de la solución y las consecuencias del uso del patrón.

Pregunta 2: ¿Cuál es el resultado de ejecutar el siguiente código?

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

    private int dameJ ()
    {
        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.

Pregunta 3: ¿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;

Pregunta 4: ¿Java permite herencia múltiple?

- a. Sí, es una característica del lenguaje.
- b. No, por definición no admite herencia múltiple y no puede implementarse de modo alguno.
- c. No, pero puede implementarse mediante combinación de "extends" y de "implements interface".
- d. No, pero puede implementarse mediante combinación de "implements" y de "extends interface".

Pregunta 5: ¿Qué ocurre si se compila y ejecuta el siguiente código?

```
public class Q {
    public static void main(String argv[]){
        int anar[]=new int[5];
        System.out.println(anar[0]);
    }
}
```

- a. Error: anar is referenced before it is initialized
- b. null
- c. 0
- d. 5

Pregunta 6: ¿Cuál de las siguientes afirmaciones es falsa?

- a. `System.out.println(-1 >>> 2)`; producirá un resultado mayor que 10.
- b. `System.out.println(-1 >>> 2)`; producirá un resultado mayor que cero (positivo).
- c. `System.out.println(2 >> 2)`; producirá el resultado 0.
- d. `System.out.println(1 <<< 2)`; producirá un resultado 4.

Pregunta 7: ¿Qué se mostrará por pantalla cuando se intenta compilar y ejecutar el siguiente código?

```
import java.awt.*;
public class Butt extends Frame{
    public static void main(String argv[]){
        Butt MyBut=new Butt();
    }

    Butt(){
        Button HelloBut=new Button("Hola");
        Button ByeBut=new Button("Adios");
        add(HelloBut);
        add(ByeBut);
        setSize(300,300);
        setVisible(true);
    }
}
```

- a. Dos botones juntos ocupando todo el frame. Hola en la izquierda y Adios en la derecha.
- b. Un botón ocupando todo el frame diciendo Hola.
- c. Un botón ocupando todo el frame diciendo Adios.
- d. Dos botones en la parte superior del frame diciendo uno Hola y el otro Adios.

Pregunta 8: Dado el siguiente fragmento, ¿podemos instanciar un objeto de esta clase?

Número de Línea	Código
4	<code>public abstract class ClaseAbstracta {</code>
5	<code> abstract void MetodoAbstracto(int a);</code>
6	<code>}</code>

- a. Sí, creando una nueva clase que extienda a `ClaseAbstracta`.
- b. Sí, creando una nueva clase que extienda a `ClaseAbstracta` e implemente el método `MetodoAbstracto`.
- c. Si, las clases abstractas se pueden instanciar como cualquier otra clase si necesidad de extenderlas si se redefine el método `MetodoAbstracto`.
- d. Ninguna respuesta anterior es correcta.

Pregunta 9: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

- a. Un encapsulamiento apropiado en las clases reduce el acoplamiento.
- b. El término acoplamiento describe cuánto se ajusta una unidad de código a una tarea lógica o a una entidad.
- c. El acoplamiento describe la conectividad de los propios objetos de una clase.
- d. Un sistema débilmente acoplado se caracteriza por la imposibilidad de modificar una de sus clases sin tener que realizar cambios en ninguna otra.

Pregunta 10: Si se ejecuta el siguiente código, ¿qué se imprime por pantalla?

Número de Línea	Código
4	<code>String s = new String ("Bicycle");</code>
5	<code>int iBegin = 1;</code>
6	<code>char iEnd = 3;</code>
7	<code>System.out.println(s.substring(iBegin,iEnd));</code>

- a. Bic.
- b. ic.
- c. icy.
- d. error: no method matching substring(int, char).

Pregunta 11: ¿Cuál es el resultado de la siguiente operación: `System.out.println (4 | 3);` ?

- a. 6.
- b. 0.
- c. 1.
- d. 7.

Pregunta 12: Necesita crear una clase que almacene como elemento base de la misma objetos únicos. No se necesita que guarden orden alguno, pero sí que no se repitan. ¿Qué interfaz sería la más apropiada para este fin?

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

Pregunta 13: Dado el siguiente código, indique qué ocurriría al llamar al método `ejemplo()`;

Número de Línea	Código
4	<code>class Examen {</code>
5	<code> private int i;</code>
6	<code> public void ejemplo () {</code>
7	<code> for (int i=0; i<5; i++)</code>
8	<code> System.out.print (this.i++);</code>
9	<code> }</code>
10	<code>}</code>

- a. Imprime 00000.
- b. Imprime 01234.
- c. Imprime infinitos ceros.
- d. Se producirá un error en tiempo de compilación por no estar inicializada la propiedad i.

Pregunta 14: Dado el siguiente fragmento de código:

Número de Línea	Código
4	<code>String cadena1 = "Halo";</code>
5	<code>String cadena2 = "HALO";</code>
6	<code>cadena1.toUpperCase();</code>

¿Cuál sería el resultado de evaluar: `if (cadena1.equals(cadena2))?`

- a. true.
- b. false.
- c. Error en la expresión.
- d. Ninguna de las anteriores es correcta.

Pregunta 15: En lo que se refiere a las clases internas anónimas, se puede afirmar que:

- a. Suelen emplearse en los lugares en los que se requiere la implementación de una sola instancia.
- b. Siempre se hará referencia la instancia mediante su supertipo.
- c. Permiten definir una clase y crear una instancia de ella, todo en un solo paso.
- d. Todas las anteriores son correctas.

PARTE PRÁCTICA [6,5 PUNTOS]:

Considere para su estudio una versión del conocido juego Space Invaders. Este juego consiste en que varias filas de naves alienígenas o UFOs avanzan hacia la base defensora, con movimientos oscilatorios de izquierda a derecha, bajando poco a poco. Así, una nave guardián defiende la base y trata de evitar los misiles lanzados esporádicamente por las naves invasoras. La nave guardián lanza disparos de uno en uno. El juego finaliza cuando todos los invasores han sido alcanzados o cuando los invasores llegan a la base.

- a) **[2,5 puntos]** Suponga que la implementación del juego se hace en base a la existencia de una clase `Nave`, de tipo abstracto, que sirve de clase de referencia para otras posibles subclases (tanto las naves alienígenas como la nave guardiana). A partir de ésta, se genera una nueva clase denominada `NaveGuardian` que sirve para modelar la nave guardián que se encuentra en la parte inferior de la pantalla del juego. Proporcione la estructura de ambas clases, así como los diferentes atributos que considere imprescindibles para cada una de ellas y los principales métodos accesores y modificadores. Si se necesita del uso de alguna otra clase auxiliar, debe definirse también en este apartado.

- b) **[2,5 puntos]** Proporcione los métodos `desplazarIzquierda` y `desplazarDerecha` que simular el movimiento de la nave a derecha e izquierda. El movimiento se realizará cuando el usuario pulse las teclas 'P' en el caso de la derecha, u 'O' en el caso de la izquierda. Se deja a su elección el prototipo que tienen que tener estos métodos, pero han de ser coherentes con lo expuesto en el apartado anterior. Si se necesita del uso de alguna otra clase auxiliar, debe definirse también en este apartado.

- c) **[1,5 puntos]** Proporcione el método `disparaMisil`, que simula el disparo de un proyectil ascendente por parte de la nave guardiana al pulsar la tecla Barra Espaciadora. Se recuerda la restricción de que en un momento determinado sólo puede haber activo un único misil. Se deja a su elección el prototipo que tienen que tener estos métodos, pero han de ser coherentes con lo expuesto en el primer apartado. Si se necesita del uso de alguna otra clase auxiliar, debe definirse también en este apartado.