

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

- a. La signatura es el encabezado de un método y proporciona la información necesaria para invocarlo.
- b. La signatura está formada por los parámetros de un método y proporciona la información necesaria para invocarlo.
- c. La signatura es el nombre de un método y puede tener parámetros para proporcionar información adicional para realizar una tarea.
- d. La signatura es el encabezado de un método y puede tener parámetros para proporcionar información adicional para realizar una tarea.

Pregunta 2: Dado el siguiente fragmento de código:

```
int indice = 1;  
boolean[] examen = new boolean[8];  
boolean poo = examen [indice];
```

Indica cual de las siguientes afirmaciones es correcta en relación al valor de la variable poo.

- a. poo tiene el valor 0
- b. poo tiene el valor null
- c. poo tiene el valor false
- d. Se produce una excepción y poo no posee ningún valor

Pregunta 3: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

- a. Los campos se conocen como variables de objeto.
- b. El alcance de una variable define la sección de código desde donde la variable puede ser declarada.
- c. Los constructores permiten que cada objeto sea preparado adecuadamente cuando es creado.
- d. El tiempo de vida de una variable describe el número de veces que es utilizada en un método.

Pregunta 4: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
if(" String ".trim() == "String")  
    System.out.println("Igual");  
else  
    System.out.println("No Igual");
```

- a. El código compilará e imprimirá "Igual".
- b. El código compilará e imprimirá "No Igual".
- c. El código provocará un error de compilación.
- d. El código provocará un error en tiempo de ejecución.

Pregunta 5: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

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

Pregunta 6: Según el texto de la bibliografía básica de la asignatura, indique cuales de las siguientes expresiones resultan verdaderas:

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

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

Pregunta 7: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes afirmaciones es correcta:

- a. Un objeto de tipo String puede ser modificado una vez que está creado, por tanto no es un ejemplo de objeto inmutable
- b. La clase String tiene un método de nombre trim que permite modificar caracteres en cualquier posición de una cadena
- c. Como regla general, las cadenas de texto de tipo String se suelen comparar mediante el operador `==`
- d. Un objeto es inmutable si su contenido o su estado no puede ser cambiado una vez que se ha creado

Pregunta 8: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes afirmaciones es correcta:

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

Pregunta 9: Dado el siguiente fragmento de código que pretende mostrar un ejemplo de sobrescritura:

```
class Examen {
    private float pregunta = 1.0f ;
    protected float getNota () {return pregunta;}
}

class Test extends Examen {
    private float nota = 2.0f;
    //Insertar código aquí
}
```

Indique cual de las siguientes opciones completaría el código anterior para dar lugar a un ejemplo correcto de sobrescritura:

- a. public float getNota (float valor) { return valor;}
- b. public float getNota () { return nota;}
- c. float getNota () { return nota;}
- d. float double getNota () { return nota;}

Pregunta 10: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes afirmaciones es correcta:

- a. Un mapa es una colección que almacena pares llave/valor como entradas.
- b. Un mapa es una colección que almacena tríos llave/índice/valor como entradas.
- c. Un mapa es una colección que almacena pares índice/valor como entradas.
- d. Un mapa es una colección que almacena tríos índice/posición/valor como entradas.

Pregunta 11: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
public class Examen {

    public static void main(String args[])
    {
        char nota = -1;
        System.out.println(nota);
    }

}
```

- a. No habrá error de compilación, la salida será -1.
- b. La expresión "char nota = -1;" provocará un error de compilación debido a que el rango de la clase "char" es $0-2^{(128-1)}$.
- c. La expresión "char nota = -1;" provocará un error de compilación debido a que el rango de la clase "char" es $0-2^{(16-1)}$.
- d. No habrá error de compilación, la salida será un carácter Unicode.

Pregunta 12: Según el texto de la bibliografía básica de la asignatura, indique cual de las siguientes opciones declarará un método en una clase que fuerza a una subclase a implementarlo:

- a. protected void metodoPI (double d1){}
- b. abstract public void metodoPI ();
- c. static void metodoPI (double d1) {}
- d. public native double metodoPI ();

Pregunta 13: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

- a. Una superclase es una clase que es implementada por otra.
- b. Una subclase es una clase que implementa a otra clase.
- c. La herencia nos permite heredar pero no reutilizar en un nuevo contexto clases que fueron escritas previamente.
- d. Las clases que están vinculadas mediante una relación de herencia forman una jerarquía de herencia.

Pregunta 14: En el siguiente fragmento de código hemos definido la ejecución de cinco bloques. Estos bloques se ejecutarán dependiendo de las excepciones que se produzcan en cada caso.

```
// Bloque1
try{
    // Bloque2
} catch (ArithmeticException e) {
    // Bloque3
} finally{
    // Bloque4
}
// Bloque5
```

Indique cual de las siguientes afirmaciones es correcta:

- a. El Bloque4 no se ejecutará si se produce una excepción de tipo aritmético en el Bloque2
- b. El Bloque4 se ejecutará antes de que la excepción producida por un acceso a un objeto *null* en el Bloque2 se propague hacia arriba
- c. El Bloque4 no se ejecutará si se produce un acceso a un objeto *null* en el Bloque2
- d. El Bloque4 se ejecutará antes que el Bloque3 si se produce una excepción de tipo aritmético en el Bloque2

Pregunta 15: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

- a. En Javadoc la etiqueta @param indica el número de parámetros del método
- b. En Javadoc la etiqueta @deprecated indica que el valor devuelto por el método puede contener errores
- c. En Javadoc la etiqueta @see indica una referencia cruzada
- d. En Javadoc la etiqueta @throws indica el modo en que debe ser lanzado un método

PARTE PRÁCTICA [6,5 PUNTOS]:

La práctica del presente curso ha sido una versión del juego “R-Type”. A continuación se presentan las reglas del juego tal y como se solicitaba para la práctica del curso:

1. El juego comenzará con una pantalla de bienvenida a partir de la cual se podrá seleccionar el modo de juego (FÁCIL, NORMAL, COMPLICADO, IMPOSIBLE) y comenzar a jugar.
2. El juego constará de un único nivel donde el jugador deberá acabar con una horda de naves alienígenas. El número de alienígenas con los que acabar dependerá del modo de juego seleccionado. Fácil=10, Normal=15, Complicado=20, Imposible=30.
3. El jugador controlará la nave aliada y dispondrá de 1 sola vida.
4. Las naves alienígenas serán controladas por el ordenador.
5. Las naves alienígenas no disparan.
6. No hay que implementar relieve. Es decir, no hay que mostrar ningún tipo de suelo o techo como en el juego original.
7. La nave aliada podrá moverse arriba (Tecla Q), abajo (Tecla A), izquierda (Tecla O) y derecha (Tecla P). Así mismo podrá disparar su laser utilizando la tecla ESPACIO.
8. El área de movimiento permitido para la nave será toda la pantalla, aunque habrá que comprobar que la nave no salga de estos límites.
9. El disparo que realiza la nave aliada es continuo, es decir, no es necesario esperar a que el misil disparado abandone la pantalla para que la nave aliada pueda volver a disparar.
10. La nave aliada sólo puede realizar un tipo de disparo que se desplazará horizontalmente hacia la derecha de la pantalla, sin variar su trayectoria y a velocidad constante.
11. Las naves alienígenas se mueven a velocidad constante y podrán ser de dos tipos:
 - a. **Nave Alienígena Tipo A.** Aparecen por la parte derecha de la pantalla y se mueven horizontalmente hacia la izquierda a velocidad constante sin variar su trayectoria, es decir, su coordenada “y” no varía en todo el desplazamiento.
 - b. **Nave Alienígena Tipo B.** Aparecen por la parte derecha de la pantalla y se mueven horizontalmente hacia la izquierda a velocidad constante. La principal diferencia con las Naves de Tipo A es que éstas pueden variar su trayectoria, es decir, en su desplazamiento horizontal pueden variar su coordenada “y” de manera aleatoria.
12. La velocidad a la que se mueven las naves alienígenas dependerá del modo de juego seleccionado. Todas las naves se mueven a la misma velocidad.
13. Cuando las naves alienígenas alcancen la parte izquierda de la pantalla volverán a aparecer por la parte derecha de ésta.
14. Se deberán de detectar dos tipos de colisiones.
 - a. Las colisiones entre la nave aliada y las naves alienígenas, lo que supondrá el final del juego.
 - b. Las colisiones entre los misiles disparados por la nave aliada y las naves alienígenas, lo que supondrá la destrucción de la nave alienígena contra la que ha chocado el misil.
15. Si el jugador finaliza el nivel del juego deberá aparecer un mensaje de felicitación y se volverá a mostrar el menú inicial.

Se pide diseñar utilizando una aproximación orientada a objetos una ampliación a la práctica realizada a lo largo del curso que permite la existencia de un nuevo tipo de nave alienígena (Tipo C).

Este nuevo tipo de nave es una modificación de la nave Tipo A, con la particularidad de que, cuando alcanza la parte izquierda, en lugar de aparecer por la parte derecha, lo que hace es desplazarse de izquierda a derecha, hasta que alcanza nuevamente el límite derecho, volviendo entonces a desplazarse hacia la izquierda (y así sucesivamente hasta ser destruida).

En un momento determinado, sólo puede haber una nave de Tipo C en el juego, y no se puede crear otra mientras la anterior de ese tipo no haya sido destruida. Además, como máximo, se podrán generar 2 naves de Tipo C en el modo Fácil; 3 en el modo Normal; 4 en el modo Complicado y 5 en el modo Imposible.

Se pide:

- a) **[1,5 puntos]** Diseñar utilizando un paradigma orientado a objetos, los elementos necesarios para la aplicación explicada de la práctica durante el curso. Es necesario identificar la estructura y las relaciones de herencia y de uso de las clases necesarias para almacenar y gestionar esta información. Debe hacerse uso de los mecanismos de herencia siempre que sea posible. Se valorará un buen diseño que favorezca la reutilización de código y facilite su mantenimiento.
- a) **[1,5 puntos]** Implementa la clase `NaveAlienigenaTipoB`. Especifica sus atributos y métodos y justifica las decisiones de implementación que creas importantes.
- b) **[1,5 puntos]** Implementa un método que gestione las colisiones entre los misiles disparados por la nave aliada y las naves alienígenas.
- c) **[2 puntos]** Indique los cambios que serían necesarios en el diseño y programa para permitir que las naves alienígenas de Tipo B pudieran variar su velocidad de desplazamiento a medida que se incrementa el número de naves alienígenas destruidas. La velocidad se determina en el momento de aparición de la nave en la pantalla y por tanto podría darse el caso de que en la pantalla co-existiesen naves alienígenas con velocidades diferentes.