

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, 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.

Pregunta 2: Según el texto de la bibliografía básica de la asignatura, un prototipo es ☐

- a. Una versión de la aplicación en la que se simula una parte de ella, en vías a experimentar con las restantes partes.
- b. Una versión de la aplicación en la que se simulan varias partes, en vías a experimentar con una de sus partes.
- c. Una versión de la aplicación en la que se simulan varias partes, en vías a experimentar con las restantes partes.
- d. Ninguna de las anteriores

Pregunta 3: Respecto a las excepciones en Java, podemos afirmar ☐

- a. Todas las subclases de la clase estándar de Java `RuntimeException` son excepciones comprobadas.
- b. Todas las subclases de la clase estándar de Java `Exception` son excepciones comprobadas.
- c. `Error` es una subclase directa de `Throwable`, mientras que `Exception` es una subclase directa de `Error`.
- d. Tanto `Error` como `Exception` son subclases directas de `Throwable`.

Pregunta 4: ¿Qué mecanismo usa Java para implementar herencia múltiple?

- a. En Java no se permite la herencia múltiple de clases, ni tampoco la implementación múltiple de interfaces.
- b. En Java no se permite la herencia múltiple de clases, pero sí la implementación múltiple de interfaces.
- c. En Java se permite la herencia múltiple de clases, pero no la implementación múltiple de interfaces.
- d. En Java se permite la herencia múltiple de clases, y también la implementación múltiple de interfaces.

Pregunta 5: ¿Cuál es el valor de la variable `d` después de ejecutar la siguiente línea de código?

```
double d = Math.round (2.5 + Math.random());
```

- a. 2
- b. 3
- c. 4
- d. 2.5

Pregunta 6: Sea el siguiente código:

```
switch (x) {  
    default:  
        System.out.println("Hola");  
}
```

¿Qué dos posibles tipos son aceptables para `x`?

1: byte 2: long 3: char 4: float 5: Short 6: Long

- a. 1 y 3.
- b. 2 y 4.
- c. 3 y 5.
- d. 4 y 6.

Pregunta 7: ¿Qué interfaz proporciona la capacidad de almacenar objetos usando un valor llave?

- a. Java.util.Map.
- b. Java.util.Set.
- c. Java.util.List.
- d. Java.util.Collection.

Pregunta 8: ¿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.

Pregunta 9: ¿Cuál de las siguientes sentencias declara legalmente, construye e inicializa un array?

- a. `int [] miLista = {"1", "2", "3"};`
- b. `int [] miLista = (5, 8, 2);`
- c. `int miLista [] [] = {4,9,7,0};`
- d. `int miLista [] = {4, 3, 7};`

Pregunta 10: ¿Cuál de las siguientes listas contiene sólo palabras clave de Java?

- a. `class, if, void, long, Int, continue.`
- b. `goto, instanceof, native, finally, default, throws.`
- c. `try, virtual, throw, final, volatile, transient.`
- d. `byte, break, assert, switch, include.`

Pregunta 11: ¿Cuál es la salida que produce el siguiente programa?

```
public class Test
{
    public static void leftshift(int i, int j)
    {
        i <<= j;
    }
    public static void main(String args[])
    {
        int i = 4, j = 2;
        leftshift(i, j);
        System.out.println(i);
    }
}
```

- a. 2
- b. 4
- c. 8
- d. 16

Pregunta 12: ¿Cuál es la salida del siguiente programa?

```
class Test {
    public static void main(String [] args) {
        Test p = new Test();
        p.start();
    }
}
```

```

void start() {
    boolean b1 = false;
    boolean b2 = fix(b1);
    System.out.println(b1 + " " + b2);
}
boolean fix(boolean b1) {
    b1 = true;
    return b1;
}
}

```

- a. true true
- b. false true
- c. true false
- d. false false

Pregunta 13: ¿Cuál es la salida que se obtiene al ejecutar este programa?

```

1. public class Test {
2.     public int aMethod() {
3.         static int i = 0;
4.         i++;
5.         return i;
6.     }
7.     public static void main(String args[]) {
8.         Test test = new Test();
9.         int i = test.aMethod();
10.        int j = test.aMethod();
11.        System.out.println(j);
12.    }
13. }

```

- a. 0
- b. 1
- c. 2
- d. La compilación falla.

Pregunta 14: ¿Cuál es la salida del siguiente programa?

```

for (int i = 0; i <= 4; i += 2) {
    System.out.print(i + " ");
}
System.out.println(i);

```

- a. 0 2 4
- b. 0 2 4 5
- c. 0 1 2 3 4
- d. La compilación falla.

Pregunta 15: ¿Qué código hay que añadir en la posición indicada en el código para que compile?

```

public class ExceptionTest {
    class TestException extends Exception {}
    public void runTest() throws TestException {}
    public void test() /* Código a añadir */ {
        runTest();
    }
}

```

- a. No hay que añadir código alguno.
- b. throws Exception
- c. catch (Exception e)
- d. throws RuntimeException

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 volvería 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

- a) **[1,5 puntos]** Proponga y explique, utilizando diagramas de clase, cómo modelaría este nuevo tipo de nave alienígena aprovechando el diseño que ha realizado en la práctica. Debe hacerse uso de herencia siempre que sea posible. Se valorará un buen diseño que favorezca la reutilización de código y facilite su mantenimiento.
- b) **[1,5 puntos]** Implemente la nueva clase especificando sus atributos y métodos, justificando las decisiones de implementación que considere relevantes.
- c) **[2,0 puntos]** Implemente la regla del juego: **“La nave alienígena Tipo C se comporta como una 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)”**.
- a) **[1,5 puntos]** Explique razonadamente qué cambios serían necesarios en el diseño que ha realizado en los apartados anteriores para que la nave alienígena Tipo C pudiera realizar un número N de idas y venidas (de derecha a izquierda y viceversa) y que cuando pasaran ese número de veces sin ser destruida, entonces se autodestruyera.