

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 ocurrirá al compilar y ejecutar el siguiente código?

```
class Padre {}
class ClaseHija extends Padre {}
class ClaseHija2 extends Padre {}
public class Test{
    public static void main(String argv[]){
        Padre b=new Padre ();
        ClaseHija s=(ClaseHija) b;
        System.out.print("Ejecutando Aplicación");
    }
}
```

- a. Compilará y se ejecutará sin problemas
- b. Error de Compilación
- c. Excepción en tiempo de ejecución
- d. Excepción en tiempo de ejecución y luego mostrará el mensaje "Ejecutando Aplicación"

Pregunta 2: Según la bibliografía básica, ¿Qué elementos cree 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

Pregunta 3: Dada la siguiente definición de clase, ¿Cuál sería el contenido más coherente a implementar en el constructor?

```
class Test {
    int var;
    Test(int var) { CONTENIDO CONSTRUCTOR }
}
```

- a. var=var;
- b. int var=var;
- c. this.var=var;
- d. No se puede llamar igual el parámetro del constructor que el atributo de la clase

Pregunta 4: De acuerdo a la bibliografía básica ¿Qué es el bytecode en Java?

- a. Un formato de intercambio de datos
- b. El formato que obtenemos tras compilar una clase .java
- c. Un tipo de variable
- d. Un depurador de código

Pregunta 5: Dado el siguiente fragmento de programa, indique que afirmación es cierta:

```
int cont;
for (cont=5 ; cont>0 ; cont--)
System.out.print(cont);
System.out.print(cont);
```

- a. Se imprime en pantalla 543210
- b. Se imprime en pantalla 5432100
- c. Se imprime en pantalla 554433221100
- d. Se imprime en pantalla 543210-1

Pregunta 6: Dados los siguientes fragmentos de código ¿Cuál de ellos asociaría a una Interfaz en Java?

- a. public class Componente interface Product
- b. Componente cp = new Componente (interfaz)
- c. public class Componente implements Printable
- d. Componente cp = new Componente.interfaz

Pregunta 7: De acuerdo a la bibliografía básica ¿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

Pregunta 8: 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;
        a3 = 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 TRES
- c. UNO DOS TRES
- d. TRES

Pregunta 9: Dadas las siguientes definiciones de clases:

```
class ClasePadre{}
class ClaseHija1 extends ClasePadre {}
class ClaseHija2 extends ClasePadre {}
```

y las siguientes instanciaciones:

```
ClasePadre var0 = new ClasePadre();
ClaseHija1 var1 = new ClaseHija1();
ClaseHija2 var2 = new ClaseHija2();
ClasePadre var3 = new ClaseHija1();
ClasePadre var4 = new ClaseHija2();
```

¿Cuál de las asignaciones es válida?

- a. var0 = var1;
- b. var2 = (ClaseHija2)var1;
- c. var2 = var4;
- d. var1 = var2;

Pregunta 10: ¿Qué ocurrirá al compilar y ejecutar el siguiente código?

```
public class MiClase{
    static int variableEstatica;
    public static void main(String argv[]){
        System.out.println(variableEstatica);
    }
}
```

- a. Error en tiempo de ejecución. La variable “variableEstatica” no ha sido inicializada
- b. Se mostrará en pantalla null
- c. Se mostrará en pantalla 1
- d. Se mostrará en pantalla 0

Pregunta 11: De acuerdo a la bibliografía básica ¿Qué significa sobrecargar un método?

- a. Editarlo para modificar su comportamiento
- b. Cambiarle el nombre dejándolo con la misma funcionalidad
- c. Crear un método con el mismo nombre pero diferentes argumentos
- d. Añadirle funcionalidades a un método

Pregunta 12: ¿Qué se mostrará en pantalla al ejecutar el siguiente código?

```
import java.awt.*;
import javax.swing.JFrame;

public class AppBoton extends JFrame{

    public static void main(String argv[]){
        AppBoton MiAppBoton=new AppBoton ();
    }

    public AppBoton(){
        Button boton1=new Button("BOTON 1");
        Button boton2=new Button("BOTON 2");
        add(boton1);
        add(boton2);
        setSize(100,100);
        setVisible(true);
    }
}
```

- Dos botones, uno junto a otro ocupando todo el frame. En el botón de la izquierda aparecerá BOTON 1 y en el de la derecha aparecerá BOTON 2.
- Un botón ocupando todo el frame con la etiqueta BOTON 1.
- Un botón ocupando todo el frame con la etiqueta BOTON 2.
- Dos botones en la parte superior del frame, uno de ellos con la etiqueta BOTON 1 y otro de ellos con la etiqueta BOTON 2.

Pregunta 13: De acuerdo a la bibliografía básica, ¿Cuál es la descripción que crees que define mejor el concepto 'clase' en la programación orientada a objetos?

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

Pregunta 14: De acuerdo a la bibliografía básica, el que una variable en una clase sea estática implica

- Hace falta crear un objeto para usarla.
- Cualquier objeto de esa clase puede modificar su valor.
- Todos los objetos tienen una copia de la variable.
- Que es una variable global y se puede usar en cualquier parte de la aplicación.

Pregunta 15: Según el código siguiente ¿Qué se visualizará en pantalla?

```
class ClaseA{
    public ClaseA ( int x ){
        System.out.print("ClaseA-" + x);}
}

class ClaseB extends ClaseA{
    public ClaseB( ){
        super(6);
        System.out.print(" ClaseB-");}
}

public class ClasePrincipal{
    public static void main(String[] args) {
        ClaseB objB1=new ClaseB();
        ClaseB objB2;
        System.out.println(" FIN"); }
}
```

- ClaseA-6 ClaseB- FIN
- ClaseB- ClaseA-6 FIN
- Hay un error en la clase B. La sentencia "super(6);" no puede ser la primera en el constructor
- Hay un error en la clase ClasePrincipal. Falta el new en "ClaseB objB2;"

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:

- 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.
- 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.

Tomando como punto de partida la práctica realizada, se pide implementar algunas mejoras para que la nave aliada disponga de un escudo protector. Este escudo protector podrá activarse pulsando la tecla "E" y, en ese momento, hará que la nave sea inmortal durante 5 segundos. El jugador dispondrá inicialmente de 3 escudos protectores que irán disminuyendo conforme éste los vaya activando.

- a) **[1,5 puntos]** Muestre el diagrama de clases general que ha diseñado en la práctica, explicando claramente la funcionalidad de cada una de las clases y sus métodos asociados.
- b) **[1,5 puntos]** Explique qué clases de la práctica habría que modificar para materializar los cambios solicitados. Muestre claramente las variables de instancia nuevas que habría que definir y los métodos que habría que implementar, indicando en cada caso a qué clases afectarían.
- c) **[2 puntos]** Implemente las modificaciones propuestas.
- d) **[1,5 puntos]** Explique razonadamente qué cambios serían necesarios en el diseño que ha realizado en los apartados anteriores para que aparezcan aleatoriamente **Bonus de Escudos Protectores** que la nave aliada pueda recoger al pasar por encima de ellos. El efecto de recoger un bonus de escudos protectores será el aumento del número de escudos disponibles para su uso.