UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA – ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

71901072 – PROGRAMACIÓN ORIENTADA A OBJETOS (GRADO EN INGENIERÍA INFORMÁTICA / TECNOLOGÍAS DE LA INFORMACIÓN)

JUNIO 2012 – MODELO B – NO ESTÁ PERMITIDO EL USO DE MATERIAL ADICIONAL

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 cual de las siguientes afirmaciones es falsa:

a. Únicamente las clases que implementan la interfaz List permiten el uso de iteradores.
b. Un iterador es un objeto que proporciona funcionalidad para recorrer todos los elementos de una colección.
c. Un iterador permite recorrer cualquier tipo de colección hacia adelante utilizando el método next() combinado con el método hasNext() para comprobar si se ha alcanzado el final de la colección.
d. Una colección puede recorrerse tanto con un iterador como con un ciclo for-each. Ambas formas son equivalentes.

Pregunta 2: Respecto a los bucles, indique cual 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 ejecuta, como mínimo, una vez.
d. Un bucle for-each puede aplicarse sobre arreglos (arrays).

Pregunta 3: Dado el siguiente código

```
String c1=new String(”Hola”);
String c2=new String(”Mundo”);
if (…….)
      System.out.println(“Ambas cadenas son iguales”);
else
      System.out.println(“Ambas cadenas no son iguales”);
```

¿Cuál de las siguientes opciones debería ponerse en la línea de puntos para llevar a cabo la comparación de las cadenas c1 y c2 en función de la salida proporcionada por el programa?

a. c1==c2
b. c1.equals(c2)
c. c1.compareTo(c2)>=0
d. c1=c2

Pregunta 4: Indique cual de las siguientes afirmaciones es verdadera:

a. Para definir una variable de instancia es necesario utilizar la palabra reservada **static**.
b. Un método estático puede acceder a cualquier componente (método o variable) no estático de su clase.
c. Los métodos estáticos pueden ser sobreescritos.
d. Una variable de clase puede ser modificada sin necesidad de haber instanciado objeto alguno de dicha clase.

Pregunta 5: Indique cual 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.

Pregunta 6: Dada la siguiente definición de clase

```
public class TV {
	private String marca;
	private String modelo;
	public TV(String marca, String modelo) {
		this.marca = marca;
		this.modelo = modelo;
	}
	public boolean equals(Object t) {
		TV television=(TV)t;
		return marca.equals(television.marca)&&
modelo.equals(television.modelo);
	}
	 public int hashCode() {
		 return marca.length() * 10 + modelo.length();
	}
}
```

¿Cual sería el resultado visualizado al ejecutar el siguiente código en un método main?

```
		TV tv1 = new TV("Sony", "Bravia");
		TV tv2 = new TV("Sony", "aivarB");

		if(tv1.equals(tv2)) {
		System.out.println("los televisores son iguales");
		} else {
		System.out.println("los televisores no son iguales");
		}
```

a. Los televisores son iguales.
b. Los televisores no son iguales.
c. Error de compilación
d. Error en tiempo de ejecución

Pregunta 7: Dado el siguiente código, indique cual de las siguientes afirmaciones es cierta:

```
	public class MiClase {
		int x = 2;
		float y = 4.3f;
		public static void main (String [] args) {
		for (int z = 1; z < x; z++ )
			System.out.println("Valor de y="+y);
		}
	}
```

a. Se produce un error en tiempo de ejecución.
b. El código no compila.
c. Se imprime en pantalla “Valor de y=4.3”
d. Se imprime en pantalla “Valor de y=4.3000”

Pregunta 8: 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.

Pregunta 9: Dado el siguiente código, ¿Cuál de las siguientes afirmaciones es correcta?

```
Set < Object > objetos = new HashSet<Object>();
String obj1 = "JAVA";
int obj2 = 5;
Boolean obj3 = new Boolean(true);
objetos.add(obj3);
objetos.add(obj1);
objetos.add(obj2);
objetos.add(obj3);
for(Object object : objetos) {
System.out.print(object);
```

a. Error en tiempo de ejecución.
b. Se muestran por pantalla *JAVA 5* y *true* en un orden no determinado.
c. Se muestran por pantalla *JAVA 5* y *true* en el orden exacto en el que fueron insertadas en la colección.
d. Se muestran por pantalla *JAVA 5* y *true* en un orden no determinado y, además, "true" se muestra dos veces.

Pregunta 10: Dados las siguientes definiciones de clases:

```
public 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) {
        Disparo dn=new DisparoNave();
        new TestUFO().inicio(dn);
    }
    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.

Pregunta 11: Según el texto de la bibliografía básica de la asignatura, cuando un objeto permite realizar un conjunto de tareas muy relacionadas entre sí, podemos afirmar que:

a. El objeto presenta una alta cohesión.
b. El objeto está muy acoplado.
c. El objeto está poco encapsulado.
d. El objeto presenta una baja cohesión.

Pregunta 12: Dado el siguiente código:

```
public class ClaseUno {
      ClaseUno obj;
      ClaseUno() { }
      ClaseUno(ClaseUno m) { obj = m; }

      void inicializar() { System.out.print("Inicializando. ");}
}

public class Test {
public static void main(String[] args) {
      ClaseUno obj1 = new ClaseUno();
      ClaseUno obj2 = new ClaseUno(obj1);
      obj2.inicializar();
      ClaseUno obj3 = obj2.obj;
      obj3.inicializar();
      ClaseUno obj4 = obj1.obj;
      obj4.inicializar();
      }
}
```

Podemos afirmar que:

a. Se mostrará el mensaje "Inicializando. Inicializando. Inicializando."
b. Se mostrará el mensaje "Inicializando. Inicializando".
c. Se mostrará el mensaje "Inicializando. Inicializando" seguido de una excepción.
d. Se mostrará el mensaje "Inicializando. Inicializando. Inicializando." seguido de una excepción

Pregunta 13: Indique cual de las siguientes afirmaciones es cierta:

a. Una interfaz puede implementar alguno de los métodos que declara.
b. Una interfaz puede declarar variables de instancia o de clase.
c. Cuando una clase implementa una interfaz específica no hace falta que implemente todos los métodos que ésta declara.
d. Una clase puede implementar más de una interfaz al mismo tiempo.

Pregunta 14: Dada la siguiente clase Prueba:

```
public class Prueba {
       public static void main(String[] args) {
              ArrayList < Integer > valores = new ArrayList < Integer > ();
              valores.add(4);
              valores.add(5);
```

```
                valores.set(1, 6);
                valores.remove(0);
                for(Integer v : valores) {
                        System.out.print(v);
                }
        }

}
```

Al ejecutar el código obtendremos:

a. Un error en tiempo de ejecución
b. Se mostrará 4
c. Se mostrará 5
d. Se mostrará 6

Pregunta 15: Si quisiera organizar los componentes de una interfaz gráfica de acuerdo a una tabla utilizaría como gestor de contenido:

a. GridLayout
b. BoxLayout
c. FlowLayout
d. BorderLayout

PARTE PRÁCTICA [6,5 PUNTOS]:

La práctica del presente curso ha sido una versión del legendario arcade "Space Invaders". A continuación se muestra la propuesta del juego tal y como se solicitaba para la práctica del curso. En el juego aparecen cuatro clases de elementos (Ver Figura):

1) Naves alienígenas o UFOs, que se mueven de izda. a dcha. y van bajando hacia abajo poco a poco. Esporádicamente lanzan misiles.
2) La nave guardián es controlada por el jugador.
3) El láser disparado por la nave guardián (trayectoria ascendente). Cuando el láser de la nave alcanza una nave enemiga, ésta desaparece del juego.
4) Los misiles disparados por los UFOs (trayectoria descendente). Cuando un misil alcanza a la nave, finaliza el juego.

Se pide diseñar utilizando una aproximación orientada a objetos una ampliación a la práctica realizada a lo largo del curso que permita la existencia de un nuevo tipo de nave alienígena (NaveDeReconocimiento) que se desplace horizontalmente por la parte superior de la pantalla (es decir, por encima del conjunto de UFOs que se van desplazando hacia la nave guardián) y que al llegar a uno de los extremos de la pantalla desaparezca por este y vuelva a aparecer por el extremo opuesto. Para ello el alumno deberá responder de manera razonada a los siguientes apartados:

a) **[X puntos]** Proponga, utilizando diagramas de clase, y explique como modelaría este nuevo tipo 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) **[X puntos]** Implemente la nueva clase NaveDeReconocimiento especificando sus atributos y métodos, justificando las decisiones de implementación que considere relevantes.
c) **[X puntos]** Implemente la regla de juego: "La nave de reconocimiento aparecerá en uno de los extremos de la pantalla y se desplazará hasta el extremo contrario de manera horizontal. Una vez alcanzado el extremo opuesto desaparecerá por éste y volverá a aparecer en el extremo inicial de manera iterativa hasta que sea destruida por un disparo". En caso de que lo considere necesario puede apoyarse en aquellas clases que ha utilizado en su práctica explicando claramente sus funcionalidades.
d) **[X puntos]** Explique razonadamente qué cambios serían necesarios en el diseño que ha realizado en los apartados anteriores para que la NaveDeReconocimiento también pudiera realizar disparos.