

### **PARTE TEÓRICA - TEST [2,5 PUNTOS]:**

Solo 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:** Indica cual de las siguientes afirmaciones es incorrecta:

- a. Todos los métodos de una interfaz son implícitamente públicos y abstractos.
- b. Todas las variables definidas en una interfaz deben ser públicos, estáticos y finales.
- c. Una interfaz puede declarar constantes y variables de instancia.
- d. Una interfaz puede extender una o más interfaces.

**Pregunta 2:** Dado el siguiente fragmento de código ¿Cuál será el resultado?:

```
3. class ClaseA {
4.     static String s = " ";
5.     protected ClaseA() { s += "ClaseA "; }
6. }
7. class ClaseB extends ClaseA {
8.     private ClaseB() { s += "ClaseB "; }
9. }
10. public class ClaseC extends ClaseA {
11.     private ClaseC() { s += "ClaseC "; }
12.     public static void main(String[] args) {
13.         new ClaseC();
14.         System.out.println(s);
15.     }
16. }
```

- a. ClaseA ClaseC
- b. ClaseA ClaseB ClaseC
- c. Error de compilación
- d. Error en tiempo de ejecución. Se lanza una excepción

**Pregunta 3:** Dado el siguiente código. ¿Cuál será el resultado?

```
1. class ClaseA {
2.     public static void main(String[] args) {
3.         int[][] a = {{1,2},{3,4}};
4.         int[] b = (int[]) a[1];
5.         Object o1 = a;
6.         int[][] a2 = (int[][]) o1;
7.         int[] b2 = (int[]) o1;
8.         System.out.println(b[1]);
9.     } }
```

- a. 2
- b. Error en tiempo de ejecución se lanza una excepción

- c. 4
- d. Error de compilación en la línea 4

**Pregunta 4:** Dado el siguiente código. ¿Cuál será el resultado?

```
3. public class ClaseA {
4.     static int var1 = 7;
5.     public static void main(String[] args) {
6.         new ClaseA().metodo(var1);
7.         System.out.print(" " + var1);
8.     }
9.     void metodo(int var1) {
10.        var1++;
11.        for(int var1 = 3; var1 < 6; var1++)
12.            ;
13.        System.out.print(" " + var1);
14.    }
15. }
```

- a. 8 8
- b. 8 7
- c. Error de compilación
- d. 5 7

**Pregunta 5:** Dado el siguiente código. ¿Cuál será el resultado?

```
4. public class ClaseA {
5.     public static void main(String[] args) {
6.         String s = "";
7.         Boolean b1 = true;
8.         Boolean b2 = false;
9.         if((b2 = false) | (21%5) > 2) s += "x";
10.        if(b1 || (b2 = true)) s += "y";
11.        if(b2 == true) s += "z";
12.        System.out.println(s);
13.    }
14. }
```

- a. Error de compilación
- b. El carácter 'x' estará incluido en la salida.
- c. El carácter 'y' estará incluido en la salida.
- d. El carácter 'z' estará incluido en la salida.

**Pregunta 6:** Para lograr que una clase entre en el depurador en BlueJ a hacer una instancia en BlueJ ¿Qué hay que hacer con el código fuente?

- a. Meter un punto de ruptura
- b. Lanzar directamente el depurador
- c. Se hace automáticamente al encontrar un error en el código
- d. Ninguna de las afirmaciones anteriores es cierta

**Pregunta 7:** Indica cual de las siguientes afirmaciones es correcta en relación a un método cohesionado:

- a. Será responsable de al menos una tarea bien definida, pero puede serlo de más.
- b. Es aquel método abstracto que se ha instanciado en una clase determinada.
- c. Es aquel que se crea en una clase interna para ser invocado desde la clase circundante.
- d. Ninguna de las afirmaciones anteriores es cierta.

**Pregunta 8:** Indica cual de las siguientes declaraciones de métodos de interfaz compilará:

- a. final void prueba();
- b. static void prueba();
- c. private void prueba();
- d. abstract public void prueba();

**Pregunta 9:** Indica 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. Un encapsulamiento apropiado en las clases reduce el acoplamiento.
- 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:** Indica cual de las siguientes afirmaciones es correcta:

- a. A extiende a B es correcto si y sólo si A es una clase y B una interfaz.
- b. A extiende a B es correcto si A y B son ambas o bien interfaces o clases.
- c. A extiende a B es correcto para todas las combinaciones de A y B siendo clases y/o interfaces.
- d. A extiende a B es correcto si y sólo si A es una interfaz y B es una clase.

**Pregunta 11:** Dados los siguientes dos ficheros, ¿Qué resultado se obtiene?:

```

1. package paqueteA;
2. public class Prueba {
3.     int a = 5;
4.     protected int b = 6;
5.     public int c = 7;
6. }

3. package paqueteB;
4. import paqueteA.*;
5. public class Principal {
6.     public static void main(String[] args) {
7.         Prueba p = new Prueba();
8.         System.out.print(" " + p.a);
9.         System.out.print(" " + p.b);
10.        System.out.print(" " + p.c);
11.    }
12. }
```

- a. 5 6 7
- b. Fallo de compilación en la línea 10
- c. Fallo de compilación en las líneas 8 y 9
- d. Fallo de compilación en la línea 7

**Pregunta 12:** Dado el siguiente código, ¿Qué resultado se obtiene?:

```

4. public class Persona extends SerVivo {
5.     public static void main(String[] args) {
6.         Short extremidades = 4;
7.         System.out.println(ContarExtremidades(extremidades, 2));
8.     }
9. }
10. class SerVivo {
11.     int ContarExtremidades(int x, int y) { return x + y; }
12. }

```

- a. Fallo de compilación en la línea 7.
- b. Fallo de compilación en la línea 11.
- c. 6
- d. Fallos de compilación debidos a múltiples errores.

**Pregunta 13:** Dada la siguiente declaración, ¿Cuál es la definición de clase correcta?:

```
public abstract interface A{ public void metodo(String s); }
```

- a. 

```
public abstract class B implements A{
    public abstract void metodo(String s) {}
}
```
- b. 

```
public class B implements A{
    public void metodo(String i) {}
    public void metodo(Integer s) {}
}
```
- c. 

```
public class B extends A{
    public void metodo(Integer i) {}
}
```
- d. 

```
public class B implements A{
    public void metodo(Integer i) {}
}
```

**Pregunta 14:** Dado el siguiente código. ¿Cuál será el resultado?

```

class ClaseA {
    private final void metodo() { System.out.println("ClaseA"); }
}

public class ClaseB extends ClaseA {
    public final void metodo() { System.out.println("ClaseB"); }
    public static void main(String [] args) {
        new ClaseB().metodo();
    }
}

```

- a. ClaseB
- b. ClaseA
- c. ClaseA  
ClaseB
- d. Error de compilación

**RESERVA 1:** Dado el siguiente código. ¿Qué habrá que insertar en la línea 9 para que compile?

```

1. class A { void metodo1() { } }
2. class B extends A { void metodo2() { } }
3.
4. class Prueba {
5.     public static void main(String [] args) {
6.         A a1 = new A();
7.         A a2 = new B();
8.         B b1 = new B();
9.         // CODIGO A INSERTAR
10.    }
11. }

```

- a. a2.metodo2();
- b. (B)a2.metodo2();
- c. ((B)a2).metodo2();
- d. Ninguna de las anteriores respuestas compila.

**RESERVA 2:** Dada el siguiente código. ¿Cuál será el resultado?

```

3. class ClaseA {
4.     public void metodo1() { System.out.print("b "); }
5. }
6. class ClaseB extends ClaseA {
7.     public void metodo2() { System.out.print("c "); }
8.     public void metodo1() { System.out.print("a "); }
9. }
10. public class ClaseC {
11.     public static void main(String[] args) { new ClaseC().metodo(); }
12.     void metodo() {
13.         new ClaseB().metodo1();
14.         ((ClaseA) new ClaseB()).metodo1();
15.         ((ClaseA) new ClaseB()).metodo2();
16.     }
17. }

```

- a. a a c
- b. a b c
- c. Fallo de compilación en la línea 15
- d. Fallo de compilación en la línea 14

## **PARTE PRÁCTICA [6,5 PUNTOS]:**

La Práctica del presente curso consiste en el modelado de la realidad asociada a un Parque de Atracciones. Entre las especificaciones clave que se indicaban en el enunciado del presente curso, podemos reseñar, a modo de recordatorio, lo siguiente:

- Es habitual encontrar los siguientes grupos de turistas de los parques:
  - Adultos: es común considerar que un adulto es toda persona comprendida entre los 13 años y los 64 años (ambos inclusive). Pueden existir subgrupos restringidos: estudiantes, poseedores de carné joven, diversidad funcional o desempleados.
  - Niños: habitualmente se considera niño aquellos visitantes con una edad comprendida entre los 3 años y los 12 años, ambos inclusive, disfrutando de gratuidad en la entrada

aquellos que tengan una edad igual o inferior a 2 años. Van asociados a una entrada de adulto. Pueden existir subgrupos restringidos: diversidad funcional, por ejemplo.

- Senior: es frecuente establecer una categoría específica para aquellos adultos con edad igual o superior a los 65 años. Le suelen aplicar los mismos subgrupos que a adultos.
- Es habitual encontrar los siguientes periodos de uso y agrupaciones específicas:
  - Entrada general: sea adulto, niño o senior, el acceso al parque tiene un tipo de entrada que facilita el acceso a cualquier hora del día cualquier día de la semana. Suelen distinguirse temporadas (alta, media y baja), donde el valor de la entrada en temporada alta es un incremento del precio respecto a la entrada de temporada media, mientras que el valor de la entrada de temporada baja suele ser un descuento sobre el precio fijado para la entrada de temporada media. El beneficio de comprar una entrada general es que es válida para cualquier día que el cliente quiera dentro de la temporada asignada. No hay restricciones horarias de acceso al parque en ningún caso.
  - Entrada de día laborable: es común aplicar un descuento para acudir al parque de lunes a jueves no festivos (si es festivo, rige la entrada general).
  - Entrada de tarde: Se suele aplicar un porcentaje de descuento a la tarifa que correspondiese en ese caso (generalmente, a partir de las 16.00h).
  - Entradas familiares: La familia tipo habitual es de dos adultos y dos niños. En estos casos, además de aplicar un porcentaje a cada una de las entradas adquiridas individualmente (con sus casuísticas determinadas), se les suele aplicar otro tipo de ventaja: parking gratuito, acceso preferente a las atracciones, bono de comida gratuito.
  - Otras entradas bonificadas: ofrecidas a lo largo del año, ofrecen descuento sobre la tarifa general que corresponda (siendo estos descuentos inferiores a los ya vigentes, pero donde las restricciones para poder acceder a ellos son escasos o nulos).
- Suele ser común facilitar a los clientes la posibilidad de, previo pago, disfrutar de un mecanismo de “espera preferente”. Este suplemento va asociado a la entrada (hay que disponer de una entrada para poder abonar el suplemento), y el hecho de comprar varias entradas (familiares, individuales) requiere que se adquiera ese suplemento para todas y cada una de las entradas a las que se quiera aplicar el beneficio.
- Las atracciones de los parques suelen tener restricciones de acceso a las mismas. Esto es independiente del tipo de entrada que se haya adquirido.
- Dentro de la organización de un parque de atracciones, es habitual contar con un equipo de trabajo muy diverso:
  - Atención al cliente: es el perfil cuya responsabilidad principal es la de atender las necesidades que un cliente del parque pueda tener. No está relacionado directamente con las atracciones como tales (ya que cuando un cliente tiene un problema en una determinada atracción o en un restaurante concreto, a quien ha de dirigirse es al responsable de la atracción o del restaurante), pero sí que actúa en esos casos como segundo nivel de “queja”. Sí que representan la primera línea de atención al cliente en aspectos como problemas con las entradas, gestión de algún tipo de pago/devolución, recogida de sugerencias de los clientes y otras similares.
  - Relaciones públicas: de ellos depende la existencia de las diferentes ofertas dentro del parque. Una oferta que se realiza o se propone siempre va a ir de la mano de un relaciones públicas que la ha desarrollado previo análisis de mercado.

- Responsables de atracción: cada atracción, necesariamente, tiene que tener un responsable de su funcionamiento, gestión y, en definitiva, organización. Los responsables de atracción son ayudantes de atracción que un momento determinado, por motivos diferentes, son promocionados a dicha categoría. Tiene a su cargo a un número variable de ayudantes de atracción. Este número dependerá de la atracción en cuestión (una atracción puede tener un número de ayudantes significativamente diferente de otra, si bien ese número es constante).
- Ayudantes de atracción: son las personas que se encargan de posibilitar que una atracción pueda funcionar de manera correcta. Un ayudante puede estar asignado a una única atracción en un día determinado, si bien es posible que, al realizar la planificación semanal de las atracciones, un ayudante de atracción pueda estar asignado a más de una atracción.

Llegados a este punto, se quiere desarrollar una solución empleando Paradigma de Programación Orientado a Objetos y el lenguaje de programación Java que permita gestionar de manera informática toda esta casuística asociada a los parques de atracciones.

Se pide realizar las siguientes tareas:

- a) **[1,0 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 (mediante el uso de un diagrama de clases) 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.
- b) **[2,0 puntos]** Implementar un método o métodos que permitan calcular, para un día específico, las ganancias brutas del parque de atracciones, así como el dinero que el parque ha dejado de ganar debido a las bonificaciones y calculado sobre el coste bruto suponiendo que todas las personas que accedieron al parque ese día pagaron una entrada general correspondiente a la temporada a la que pertenece el día solicitado.
- c) **[2,0 puntos]** Implementar un método o métodos que permitan calcular las ganancias brutas por temporada.
- d) **[1,5 puntos]** Implementar un método o métodos que permitan calcular el promedio de entradas preferentes por temporada y que, además, calcule su porcentaje con respecto al número de entradas vendidas.