

### **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.

Las preguntas de reserva sólo tendrán utilidad en el caso de que alguna de las 14 preguntas iniciales del test sea anulada por cualquier circunstancia. Caso de ocurrir este hecho, si se produjera la anulación de alguna de las 14 preguntas iniciales, la primera pregunta de reserva sustituiría a la pregunta anulada. Caso de que una segunda pregunta de las 14 iniciales fuese anulada, entonces la segunda pregunta de reserva sustituiría a esta segunda pregunta anulada. En aquellos hipotéticos casos en los que se produjese la anulación de una tercera o sucesivas preguntas de las 14 iniciales, entonces sólo en ese caso, las preguntas tercera y sucesivas anuladas se considerarían como correctas (al no existir más preguntas de reserva que las sustituyan).

**Pregunta 1:** Según el texto de la bibliografía básica de la asignatura, ¿Para qué utilizamos los métodos?

- a. Para declarar una clase.
- b. Para declarar un objeto.
- c. Para comunicarnos con los objetos.
- d. Ninguna de las anteriores.

**Pregunta 2:** Según el texto de la bibliografía básica de la asignatura, el texto de una clase puede dividirse en dos partes principales: un envoltorio exterior (A) y una parte interior (B), donde:

- a. A = un envoltorio exterior que simplemente da nombre a la clase. B = la parte que se encarga de realizar todo el trabajo.
- b. A = un envoltorio exterior que se encarga de realizar todo el trabajo. B = la parte que simplemente da forma a la clase.
- c. A = B = la parte que da funcionalidad a la clase.
- d. Ninguna de las anteriores.

**Pregunta 3:** Dado el siguiente método:

```
public void setCredits(int creditValue)
{
    credits = creditValue;
}
```

¿Cuál es el nombre de este método y el nombre y el tipo de su parámetro?

- a. creditValue y setCredits
- b. void y creditValue
- c. setCredits y int
- d. setCredits y void

**Pregunta 4:** Según el texto de la bibliografía básica de la asignatura, una clase puede contener más de un

constructor siempre que cada uno tenga un conjunto diferente de tipos de parámetros. Dicha característica se llama:

- a. Encapsulación
- b. Sobrecarga
- c. Abstracción
- d. Métodos internos

**Pregunta 5:** Según el texto de la bibliografía básica de la asignatura, ¿cuál de las siguientes características **NO TIENE** la clase ArrayList?:

- a. Es capaz de incrementar su capacidad interna según sea necesario.
- b. No mantiene el orden de los elementos que se inserten en la lista.
- c. Mantiene su propio contador privado, el número de elementos que almacena en cada instante.
- d. Mantiene el orden de los elementos que se inserten en la lista.

**Pregunta 6:** Según el texto de la bibliografía básica de la asignatura, ¿qué técnica se realiza automáticamente cuando se utiliza un valor de tipo primitivo en un contexto que necesita una clase envolvente?

- a. Abstracción
- b. Ocultación
- c. Encapsulamiento
- d. Autoboxing

**Pregunta 7:** Según el texto de la bibliografía básica de la asignatura, una adecuada encapsulación de las clases reduce:

- a. Los errores
- b. El acoplamiento
- c. La cohesión
- d. Ninguna de las anteriores

**Pregunta 8:** Según el texto de la bibliografía básica de la asignatura, sobre la herencia, se puede decir que:

- a. El constructor de una superclase debe siempre invocar al constructor de su subclase.
- b. El constructor de una subclase no tiene porque siempre invocar al constructor de su superclase.
- c. El constructor de una subclase debe siempre invocar al constructor de su superclase.
- d. Ninguna de las anteriores

**Pregunta 9:** Dado el siguiente fragmento de código en Java:

```
public class Foo {  
    int num;  
    Foo (int num) {  
        CONTENIDO_CONSTRUCTOR  
    }  
}
```

¿Cuál será el CONTENIDO\_CONSTRUCTOR?

- a. num = num;
- b. this.num = num;
- c. int num = num;
- d. Ninguna de las anteriores

**Pregunta 10:** ¿Cuál de los siguientes fragmentos de código en Java muestra como se puede implementar la

interfaz MilInterfaz en la clase Maquina en Java?

- a. `public class Maquina = new Class.MilInterfaz { CÓDIGO NO INCLUIDO }`
- b. `public class Maquina implements MilInterfaz { CÓDIGO NO INCLUIDO }`
- c. `public class Maquina interface MilInterfaz { CÓDIGO NO INCLUIDO }`
- d. Ninguna de las anteriores.

**Pregunta 11:** Dado el siguiente fragmento de código:

```
class Vehiculo {
    String nombre = null;
    public String toString() {return(nombre);}
}
class Moto extends Vehiculo {String nombre ="Mi moto";}
class Coche extends Vehiculo {String nombre ="Mi coche";}
public class Garaje {
    public static void main (String [] args) { Vehiculo v = new Moto();
        Moto h = (Moto) v;
        System.out.println(v.toString() + " " + h.toString());
    }
}
```

¿Cuál es el resultado de compilar / llamar el método main()?

- a. Se produce un error de compilación.
- b. Se produce un error de ejecución.
- c. El código compilará y generará el siguiente: Mi moto Mi coche
- d. El código compilará y generará el siguiente: null null

**Pregunta 12:** ¿Cuál es la diferencia entre **final** y **finalize** en Java?

- a. **final** = palabra reservada y **finalize** = un método
- b. **final** = un método y **finalize** = palabra reservada
- c. **final** = **finalize** = un método
- d. Ninguna de las anteriores.

**Pregunta 13:** ¿De qué tipo son los siguientes objetos en Java?

- "Hoy es un día importante"
  - 55.5
  - true
- a. `char[]`, `int`, `String`
  - b. `String`, `long`, `boolean`
  - c. `String`, `double`, `boolean`
  - d. Ninguna de las anteriores.

**Pregunta 14:** Si metemos el siguiente código en Bluej:

```

class Saludo {
    public Saludo (String nombre) {
        System.out.print ("Hola " + nombre);
    }
}
public class Saludar extends Saludo {
    public Saludar (String apellido) {
        System.out.print (" " + apellido);
    }
    public static void main (String [] args) {
        Saludar miSaludo = new Saludar("Juan Latorre");
        System.out.println (" ");
    }
}

```

¿Cuál es el resultado de compilar / llamar el main()?

- a. Se produce un error de compilación en la línea public Saludar ...
- b. Se produce un error de compilación en la línea public class Saludar ...
- c. El código compilará y generará el siguiente: Hola Juan Latorre
- d. El código compilará y generará el siguiente: Hola

**RESERVA 1:** ¿A qué paquete de Java pertenece Throwable?

- a. java.error
- b. java.lang
- c. java. Útil
- d. java.io

**RESERVA 2:** Dado el siguiente fragmento de código:

```

int a = 10;
int b = 37;
int z = 0;
int w = 0;
if (a == b) { z = 3; }
else if (a > b) { z = 6; }

w = 10 * z;

```

¿Cuál es el valor de w?

- a. 37
- b. 3
- c. 0
- d. 10

## **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 especialista en 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 punto]** 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) **[1,5 puntos]** Implementar un método (o métodos) que permitan la gestión de la venta de los diferentes tipos de entradas al parque. Justifica las opciones y decisiones que se tomen.
- c) **[2 puntos]** Implementar un método (o métodos) que permita(n) que la gestión del equipo humano de cada atracción. Dichos métodos deberían permitir montar un equipo de responsables, asignar a cada uno un papel, unas responsabilidades, un horario de trabajo y disponer de un historial de problemas que ha habido con la atracción. Justifica las opciones y decisiones que se tomen.
- d) **[2,0 puntos]** Se quiere colocar dentro del parque unas pantallas táctiles para dar a los visitantes acceso al servicio de atención al cliente. Implementar el sistema de gestión del parque para gestionar la atención al cliente que se puede usar desde estas pantallas. Justifica las opciones y decisiones que se tomen.