

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: Indica cual de las siguientes afirmaciones es correcta:

- a. Los objetos Java no modelan los objetos pertenecientes a un dominio de problema.
- b. La clase describe el tipo de objeto.
- c. Los objetos representan las diferentes clases declaradas en un método.
- d. Podemos comunicarnos con los objetos mediante acciones sobre una clase de comunicación.

Pregunta 2: Dado el siguiente fragmento de código:

```
int indice = 1;  
boolean[] examen = new boolean[8];  
boolean poo = examen [indice];
```

Indica cual de las siguientes afirmaciones es correcta en relación al valor de la variable poo.

- a. poo tiene el valor 0
- b. poo tiene el valor null
- c. poo tiene el valor false
- d. Se produce una excepción y poo no posee ningún valor

Pregunta 3: Dadas las siguientes expresiones:

```
1. (8 == 8) | (10 > 8) == false | true == true  
2. (8 > 8) && (8 > 8) == (8 > 8) == false
```

Indica cual de las siguientes opciones es la correcta:

- a. La expresión 1 es evaluada como false y la expresión 2 como false.
- b. La expresión 1 es evaluada como false y la expresión 2 como true.
- c. La expresión 1 es evaluada como true y la expresión 2 como true.
- d. La expresión 1 es evaluada como true y la expresión 2 como false.

Pregunta 4: Indique cuál de las siguientes afirmaciones es correcta:

- a. Un objeto colección no puede almacenar un número arbitrario de otros objetos.
- b. Un bucle while consta de una cabecera, un cuerpo y un pie.
- c. Un bucle se puede utilizar para ejecutar un bloque de instrucciones repetidamente, teniendo que escribirlas múltiples veces.
- d. Un iterador es un objeto que proporciona funcionalidad para iterar a través de todos los elementos de una colección.

Pregunta 5: Indica cual de las siguientes afirmaciones es correcta:

- a. La documentación de la librería de clases Java muestra detalles acerca de todas las clases de la librería.
- b. La interfaz de una clase describe lo que una clase hace y cómo se puede utilizar mostrando su implementación.
- c. Se dice que un objeto es inmutable si su contenido o estado no puede cambiarse después de crearlo excepto si la interfaz es mutable.
- d. La palabra interfaz solo tiene un significado en el campo de la programación independientemente del contexto.

Pregunta 6: Dado el siguiente fragmento de código, indique cuál de las siguientes afirmaciones es el resultado de su ejecución:

```
class Test
{
    public static void main (String args [])
    {
        int n, c = 1, serie = 5;
        System.out.print ("Cantidad de terminos: ");
        n = 7;
        while (c < n)
        {
            System.out.print ("," + serie);
            serie += 5;
            c++;
        }
    }
}
```

- a. Cantidad de terminos: 5,10,15,20,25,30,
- b. Cantidad de terminos: ,5,10,15,20,25,30,35
- c. Cantidad de terminos: ,5,10,15,20,25,30,35,40
- d. Cantidad de terminos: ,5,10,15,20,25,30

Pregunta 7: Indique cuales de las siguientes expresiones resultan "true":

1. ! (4 < 5)
2. (2 > 2) || ((4 == 4) && (1 < 0))
3. (2 > 2) || (4 == 4) && (1 < 0)
4. (2 > 2) || !((4 == 4) && (1 < 0))
5. (34 != 33) && ! false

- a. Las expresiones 3 y 4.

- b. Las expresiones 2 y 4.
- c. Las expresiones 4 y 5.
- d. Las expresiones 3 y 5.

Pregunta 8: Indica cual de las siguientes afirmaciones es correcta:

- a. Una prueba negativa es una implementación que no ha funcionado correctamente.
- b. La automatización de pruebas no simplifica el proceso de pruebas de regresión pero se debe realizar.
- c. Una aserción es una expresión que establece una condición que esperamos que sea falsa.
- d. Un banco de pruebas es un conjunto de objetos en un estado definido que sirven como base para realizar pruebas de unidades.

Pregunta 9: ¿Cuál sería la salida del siguiente código? (Las clases estarían definidas en ficheros diferentes)

```
public class BaseDeDatos {

    public final void metodoAgregarElemento() {
        System.out.println("Agregar Elemento");
    }
}

public class BaseDeDatosDeMusica {

    public static void main(String argv[]) {
        BaseDeDatos db = new BaseDeDatos();
        db.metodoAgregarElemento();
    }
}
```

- a. Error en tiempo de compilación indicando que una clase con métodos finales deben ser declarada también como final.
- b. Error en tiempo de compilación indicando que no se puede heredar de una clase con métodos finales.
- c. Error en tiempo de ejecución indicando que BaseDeDatos no ha sido definida como final.
- d. Éxito en la compilación y salida " Agregar Elemento".

Pregunta 10: Indica cual de las siguientes afirmaciones es correcta:

- a. El tipo dinámico de una variable v es el tipo tal como está declarado en el código fuente.
- b. El tipo estático de una variable v es el tipo del objeto que está almacenado actualmente en v.
- c. Declarar un campo o un método protegido impide acceder directamente a él desde las subclases.
- d. Todas las afirmaciones anteriores son falsas.

Pregunta 11: Indica cual de las siguientes afirmaciones es correcta:

- a. Un método abstracto se marca con la palabra clave abstractive.
- b. Una clase abstracta es una clase que no está pensada para crear instancias.
- c. La definición de un método abstracto está compuesta de cabecera y cuerpo.
- d. Solamente los métodos pueden declararse como abstractos, las clases no.

Pregunta 12: Dado el siguiente fragmento de código,

```
int electrodomestico = 1;
boolean[] ventas = new boolean[3];
boolean financiado = ventas [electrodomestico];
```

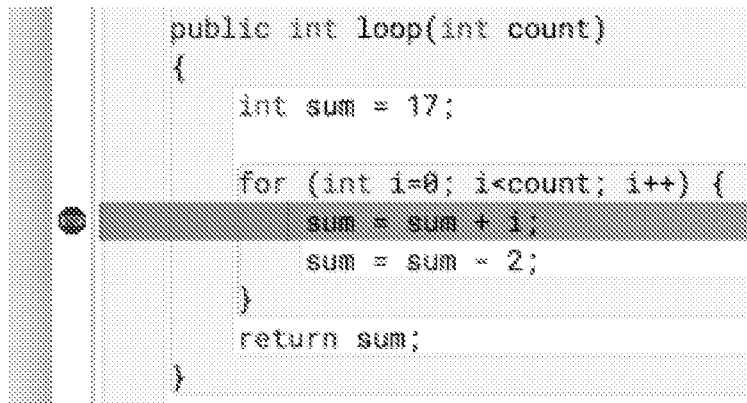
Indica cual de las siguientes afirmaciones es correcta en relación al valor de la variable `financiado`.

- a. `financiado` tiene el valor `false`
- b. `financiado` tiene el valor `0`
- c. `financiado` tiene el valor `null`
- d. Se produce una excepción y `financiado` no posee ningún valor

Pregunta 13: Indica cual de las siguientes afirmaciones es correcta:

- a. Una excepción es un método que representa los detalles de un fallo del programa.
- b. Una excepción no comprobada es un tipo de excepción cuyo uso requiere una comprobación por parte del compilador.
- c. Las excepciones comprobadas en Java requieren que se usen cláusulas *throws* e instrucciones *try*.
- d. Una excepción permite leer y escribir en una única operación objetos completos, así como jerarquías de objetos.

Pregunta 14: Dada la siguiente imagen de BlueJ, indica cual de las siguientes afirmaciones es correcta:



- a. En el editor, la parte correspondiente a la izquierda del texto se denomina área de edición.
- b. Un punto de interrupción es un indicador asociado a una línea de código fuente.
- c. El símbolo de "STOP" a la izquierda del código indica un error de compilación.
- d. El símbolo de "STOP" a la izquierda del código no implica que se detenga la ejecución del método `loop` en el momento en que se alcance ese punto, solo se comprobará el valor de la variable `sum`.

RESERVA 1: Indique cuál de las siguientes afirmaciones es correcta:

- a. Una superclase es una clase que es implementada por otra.
- b. Las clases que están vinculadas mediante relaciones de herencia forman una jerarquía de herencia.
- c. Una subclase es una clase que implementa a otra clase.
- d. La herencia nos permite heredar pero no reutilizar en un nuevo contexto clases que fueron escritas previamente.

RESERVA 2: Indique cuál de las siguientes afirmaciones es correcta:

- a. Un encapsulamiento apropiado en las clases reduce el acoplamiento.
- b. El término acoplamiento describe cuánto se ajusta una unidad de código a una tarea lógica o a una entidad.
- 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

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 permita generar un listado de cada una de las entradas vendidas en un día determinado, diferenciando entre los diversos tipos de entrada (general, laborable, tarde, etc.).
- c) **[2,0 puntos]** Implementar un método (o métodos) que desarrolle la siguiente funcionalidad: el libro de mantenimiento digital de una atracción. Este libro de mantenimiento consta de toda la información posible acerca de cada una de las reparaciones hechas en la atracción. La dirección del parque puede acceder a este servicio que requiere saber todas las incidencias relacionadas con una determinada atracción, así como su responsable de atracción y ayudantes de atracción que atendieron dicha incidencia. Justifique las opciones y decisiones que se tomen.
- d) **[1,5 puntos]** Implementar un método (o métodos) que permita gestionar las estadísticas de incidencias en un periodo determinado de tiempo. Para ello se proporcionará la siguiente información:
 - Los 10 días del año en los que ha habido un mayor número de incidencias (ordenados descendientemente, desde el de más incidencias).
 - Las 10 atracciones que han sufrido más incidencias en el último año (ordenados descendientemente, desde el de más incidencias).
 - Las 10 atracciones en las que han intervenido un mayor número de trabajadores (responsable y ayudantes de atracción) en el último año (ordenados descendientemente, desde el que más personas ha empleado).