

### **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:** Respecto a los conceptos de clases y objetos, según el texto de la bibliografía básica de la asignatura, podemos afirmar ...

- a. Los objetos se crean a partir de clases
- b. Un objeto describe el tipo de clase
- c. Las clases representan las instanciaciones individuales de un objeto
- d. Todas las respuestas anteriores son falsas

**Pregunta 2:** Respecto al concepto de palabras clave o palabras reservadas, según el texto de la bibliografía básica de la asignatura, podemos afirmar ...

- a. En Java existen unas 100 palabras de este tipo.
- b. Las palabras clave en Java pueden contener o no letras mayúsculas.
- c. En determinadas circunstancias, `endOfFile` puede ser una palabra reservada.
- d. La palabra `public` es una palabra reservada de Java.

**Pregunta 3:** Sea el siguiente fragmento de código:

```
1    public class Ejemplo {
2        static int num1 = 10;
3        public static void main (String args []) {
4            int num2 = 5;
5            new Ejemplo ();
6        }
7        public Ejemplo () {
8            int aux = this.num2;
9            if (aux > 1) {
10                System.out.println(aux);
11            }
12        }
13    }
```

¿Cuál es el resultado que produce?

- a. Se produce un error de compilación.
- b. Se produce un error de ejecución.
- c. No se produce ningún error pero no muestra nada por pantalla.
- d. No se produce ningún error y muestra por pantalla el valor 5.

**Pregunta 4:** Sea el siguiente código:

```

1    import java.util.*;
2    public class Ejemplo {
3        public static void main (String args []) {
4            ArrayList <String> a = new ArrayList (5);
5            for (int i=0; i<=5; i++)
6            {
7                a.add("Hola");
8            }
9            System.out.println("Funciona");
10        }
11    }

```

¿Cuál es el resultado de compilar y ejecutar este código?

- a. Se produce un error de ejecución al definir un ArrayList de 5 elementos y querer insertar 6 elementos.
- b. La línea 5 provoca un error en tiempo de ejecución.
- c. La línea 7 provoca un warning pero se ejecuta sin problemas proporcionando el resultado por pantalla "Funciona".
- d. El resultado que se obtiene no está recogido en ninguna de las anteriores opciones de respuesta.

**Pregunta 5:** Según el texto de la bibliografía básica de la asignatura, ¿En qué condiciones puede volverse a invocar un constructor de una clase para un objeto después de que ese objeto haya sido creado?

- a. Cuando queremos resetear todos los campos del objeto a sus valores.
- b. Cuando se ha creado un objeto abstracto y se le quiere dar valores iniciales a sus atributos.
- c. Cuando se implementa una interfaz para el objeto en cuestión.
- d. Nunca.

**Pregunta 6:** Sea el siguiente fragmento de código de código (se omiten las sentencias `import` para importar las clases que correspondan):

```

1    Random randomGenerator;
2    randomGenerator = new Random(0);
3    char index = (char) randomGenerator.nextInt(0);
4    System.out.println(index);

```

¿En qué línea del código anterior se produce un error de compilación?

- a. En la línea 1.
- b. En la línea 2.
- c. En la línea 3.
- d. No se produce error de compilación.

**Pregunta 7:** ¿De qué manera podemos crear un objeto matriz en Java?

- a. `int hoursCounts = new int[24];`
- b. `int hoursCounts[] = new int[24];`
- c. `int hoursCounts[] = new int[];`
- d. `int hoursCounts = new int[];`

**Pregunta 8:** Según el texto de la bibliografía básica de la asignatura, un método cohesionado ...

- a. será responsable de al menos una tarea bien definida, pero puede serlo de más.
- b. será responsable de una y sólo una tarea bien definida.
- c. es aquel método abstracto que se ha instanciado en una clase determinada.
- d. es aquel que se crea en una clase interna para ser invocado desde la clase circundante.

**Pregunta 9:** Queremos compilar el siguiente código que se puede encontrar en el texto base de la asignatura y que hemos modificado. ¿Qué ocurre al compilarlo con BlueJ?

```
1  import java.util.List;
2
3  public abstract class Animal
4  {
5      public void act (List<Animal> newAnimals, char animals) {}
6      public static void act (List<Animal> newAnimals, int animals) {};
7      abstract public static void act (List<Animal> newAnimals, String animals);
8  }
```

- a. La línea 5 provoca error de compilación.
- b. La línea 6 provoca error de compilación.
- c. La línea 7 provoca error de compilación.
- d. No se produce ningún tipo de error de compilación.

**Pregunta 10:** Según el texto de la bibliografía básica de la asignatura, respecto al apartado de pruebas, cuál de las siguientes afirmaciones ES FALSA:

- a. Una prueba positiva es la prueba de un caso que se espera que funcione correctamente.
- b. La automatización de pruebas simplifica el proceso de pruebas de integración.
- c. BlueJ incluye soporte para las pruebas de regresión.
- d. JUnit tiene sistemas similares disponibles en otros lenguajes de programación.

**Pregunta 11:** Queremos compilar el siguiente código que se puede encontrar en el texto base de la asignatura y que ha sido modificado convenientemente. Se produce un error de compilación. ¿Qué línea es la que contiene un error, tal que si la modificamos convenientemente, el código compila y no provoca ningún error de compilación en BlueJ?

```
1  import java.io.*;
2
3  public class Ejemplo
4  {
5      public static void main () throws IOException
6      {
7          String filename = new String ("EJEMPLO");
8          try {
9              throw IOException ();
10         }
11         catch (Exception e) {
12             System.out.println("Unable to save to "+ filename);
13         }
14     }
15 }
```

- a. El error está en la línea 7

- b. El error está en la línea 9
- c. El error está en la línea 11
- d. El código, tal y como se proporciona en el enunciado sin ningún cambio, compila y no produce error alguno.

**Pregunta 12:** Según el texto de la bibliografía básica de la asignatura, con respecto a la igualdad de referencias podemos afirmar ...

- a. La igualdad de referencia tiene en cuenta el contenido de los objetos.
- b. No es posible usar la igualdad de referencias para realizar comparaciones de cadenas de caracteres.
- c. El método equals heredado de la clase Object no permite comprobar que existe una igualdad de referencias.
- d. Ninguna de las anteriores.

**Pregunta 13:** Según podemos deducir de lo indicado en el texto de la bibliografía básica de la asignatura, ¿qué debilitaría la encapsulación?

- a. Emplear el acceso protegido a los métodos de una clase.
- b. Emplear el acceso protegido a los campos de una clase.
- c. Emplear el acceso protegido a los constructores de una clase.
- d. Ninguna de las anteriores.

**Pregunta 14:** Según el texto de la bibliografía básica de la asignatura, ¿qué podemos afirmar sobre el concepto de sustitución?

- a. Pueden utilizarse objetos de un supertipo en cualquier lugar en el que se espera objetos de un subtipo.
- b. Permite crear objetos de una clase que es abstracta.
- c. Permite que una variable almacena objetos de diferentes tipos (en concreto, del tipo declarado o de cualquier supertipo del tipo declarado).
- d. Ninguna de las anteriores

**RESERVA 1:** Queremos compilar el siguiente código que se puede encontrar en el texto base de la asignatura, convenientemente modificado. ¿Qué ocurre al compilarlo con BlueJ?

```
1    import java.util.List;
2    public abstract class Animal
3    {
4        private boolean alive;
5        private String field;
6        private String location;
7
8        public abstract Animal(String field, String location)
9        {
10            alive = true;
11            this.field = field;
12            this.location = location;
13        }
14
15        abstract public void act(List<Animal> newAnimals);
16    }
```

- a. Compila, no proporcionando ningún error en tiempo de compilación.

- b. No compila. Se soluciona sustituyendo la línea 15 por la siguiente: `public void act(List<Animal> newAnimals);`
- c. No compila. Se soluciona sustituyendo la línea 15 por la siguiente: `public abstract void act(List<Animal> newAnimals);`
- d. Ninguna de las anteriores.

**RESERVA 2:** Según el texto de la bibliografía básica de la asignatura, ¿qué se entiende por abstracción?

- a. Proceso de dividir un todo en partes bien definidas que puedan construirse y examinarse por separado y que interactúen de formas bien definidas.
- b. Capacidad de ignorar los detalles de las distintas partes, para centrar la atención en un nivel superior de un problema.
- c. Capacidad de dividir un todo en partes bien definidas ignorando los detalles de las distintas partes.
- d. Ninguna de las anteriores.

## **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 atracciones que han estado en funcionamiento en un día determinado, proporcionando además un listado de los responsables de las mismas cada día, así como de los diferentes ayudantes de atracción que trabajaban en ella, ordenados estos últimos alfabéticamente.
- c) **[2,0 puntos]** Implementar un método (o métodos) que desarrolle la siguiente funcionalidad: anualmente, por cada trabajador, se paga una cantidad fija a la Seguridad Social (que será un valor  $n$ , de tipo double), y un porcentaje que depende del sueldo bruto total del cada trabajador. Así, los empleados de atención al cliente tienen fijado este porcentaje en el 3%, los relaciones públicas el 4%, los responsables de atracción el 4% y los ayudantes de atracción el 2%. Proporcionar un listado de cuál es la cantidad que se tiene que pagar por cada uno de los grupos de trabajadores, así como el total que hay que desembolsar.
- d) **[1,5 puntos]** Implementar un método (o métodos) gestionar las estadísticas de afluencia de visitantes por día. Para ello se proporcionará la siguiente información:
  - Los 10 días del año en los que ha habido una mayor afluencia de niños (ordenados descendentemente, desde el de más afluencia)
  - Los 10 días del año en los que ha habido una menor afluencia de seniors (ordenados ascendentemente, desde el de menor afluencia).
  - Los 10 días del año en los que ha habido una mayor afluencia total de visitantes (ordenados descendentemente, desde el de mayor afluencia).