

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

El test consta de 14 preguntas y 2 preguntas adicionales de reserva. 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. La signatura está formada por los parámetros de un método y proporciona la información necesaria para invocarlo.
- b. La signatura es el encabezado de un método y puede tener parámetros para proporcionar información adicional para realizar una tarea.
- c. La signatura es el encabezado de un método y proporciona la información necesaria para invocarlo.
- d. La signatura es el nombre de un método y puede tener parámetros para proporcionar información adicional para realizar una tarea.

**Pregunta 2:** Indique el orden seguido en los ejemplos del texto de la bibliografía básica de la asignatura en cuanto a la parte interna de una clase:

```
public class NombreClase
{
    PARTE INTERNA DE UNA CLASE
}
```

- a. Constructores, Métodos y Campos
- b. Métodos, Constructores y Campos
- c. Campos, Constructores y Métodos
- d. Campos, Métodos y Constructores

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

```
int A = 9;
float B = 3.3F;
char C = 'w';

System.out.println(A + B > 12);
System.out.println(A >= 8 && C != 'w');
System.out.println((C == 'c') || ((A + B) == 12));
```

Indica cual será la salida por pantalla (cada valor en una línea diferente):

- a. True true false
- b. True false false
- c. True false true
- d. False false false

**Pregunta 4:** Indica cual de las siguientes afirmaciones es correcta en relación a que tipo de bucle se debe utilizar:

- a. Si tenemos un bucle que no está relacionado con colecciones habrá que elegir el bucle for-each
- b. El bucle for es preferible si, al principio del bucle, no sabemos cuantas veces tenemos que ejecutarlo.
- c. Si necesitamos iterar a través de todos los elementos de una colección, el bucle for-each es casi siempre la opción más elegante.
- d. El bucle for-each es adecuado cuando nos hace falta utilizar de manera explícita el contador del bucle.

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

- a. Un objeto de tipo String puede ser modificado una vez que está creado, por tanto no es un ejemplo de objeto inmutable
- b. Un objeto es inmutable si su contenido o su estado no puede ser cambiado una vez que se ha creado
- c. La clase String tiene un método de nombre trim que permite modificar caracteres en cualquier posición de una cadena
- d. Como regla general, las cadenas de texto de tipo String se suelen comparar mediante el operador ==

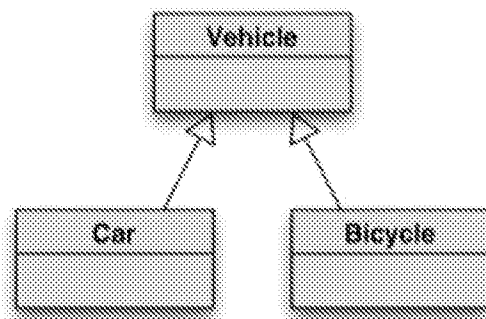
**Pregunta 6:** Indica cual de las siguientes afirmaciones es correcta:

- a. Una clase cohesionada representa una entidad bien acoplada.
- b. Un método cohesionado será responsable de varias tareas bien definidas.
- c. La segunda mayor ventaja de la cohesión es que ofrece un mayor potencial de utilización.
- d. Las dos más importantes maneras en que una alta cohesión beneficia a un diseño son el acoplamiento y la encapsulación.

**Pregunta 7:** Indica cual de las siguientes afirmaciones es correcta:

- a. Un seguimiento es la actividad de trabajar a través de un segmento de código línea por línea, mientras se observan cambios de estado y otros comportamientos de la aplicación.
- b. Un banco de pruebas es un conjunto de objetos en un estado indefinido que sirven como base para realizar pruebas de unidades.
- c. La aserción es la actividad de descubrir si una pieza de código produce el comportamiento pretendido.
- d. Una prueba es una expresión que establece una condición que esperamos que resulte verdadera.

**Pregunta 8 :** Dada la siguiente jerarquía de herencia:



Indica cual de las siguientes asignaciones es correcta:

- a. Car c1 = new Vehicle();
- b. Car c2 = new Vehicle();
- c. Vehicle v1 = new Car();
- d. Todas las asignaciones anteriores son correctas.

**Pregunta 9:** Dado el siguiente código:

```
public class testJunio {  
    public void setVar (int a, int b, float c) {  
    }  
}
```

Y los siguientes métodos:

1. private void setVar (int a, float c, int b) { }
2. protected void setVar (int a, int b, float c) { }
3. public int setVar (int a, float c, int b) {return a;}
4. public int setVar (int a, int b, float c) {return a;}
5. protected float setVar (int a, int b, float c) {return c;}

Indique qué métodos permiten una sobrecarga del método setVar de manera correcta:

- a. 3 y 5
- b. 3 y 4
- c. 1 y 2
- d. 1 y 3

**Pregunta 10:** Indique cual de las siguientes opciones declarará un método en una clase que fuerza a una subclase a implementarlo:

- a. static void methoda (double d1) { }
- b. public native double methoda();
- c. abstract public void methoda();
- d. protected void methoda (double d1){}

**Pregunta 11:** Indica cual de las siguientes afirmaciones es correcta en relación a un marco de Swing (JFrame):

- a. Está compuesto de tres partes: la barra de título, una barra de menú opcional y los cuadros de diálogo.
- b. Está compuesto de tres partes: la barra de título, el panel de contenido y el gestor de los bordes.
- c. Está compuesto de tres partes: la barra de título, una barra de menú opcional y el panel de contenido.
- d. Está compuesto de cuatro partes: la barra de título, una barra de menú opcional, el panel de contenido y el gestor de los bordes.

**Pregunta 12:** Indique cual de las siguientes afirmaciones es correcta:

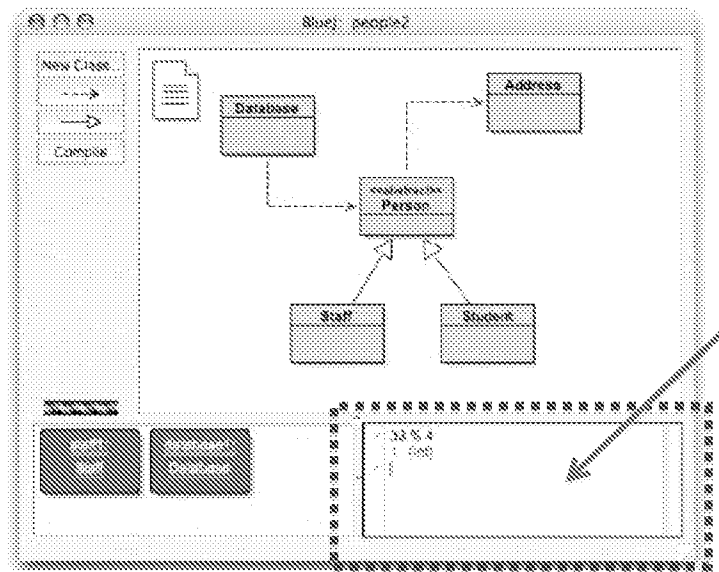
Todas las excepciones no comprobadas son subclases de

- a. Exception
- b. RuntimeException
- c. Throwable
- d. Error

**Pregunta 13:** Teniendo en cuenta el modelo en cascada presente en la construcción del software, indica cual de las siguientes fases NO pertenece al desarrollo de software:

- a. Análisis del problema.
- b. Prueba Unitaria.
- c. Prueba Secuencial.
- d. Entrega del sistema al cliente.

**Pregunta 14:** La siguiente imagen corresponde a un pantallazo de la aplicación BlueJ. En la parte inferior derecha hay un componente rectangular de BlueJ que en su interior contiene la expresión `33 % 4` que a su vez está rodeada por un rectángulo intermitente. ¿Qué componente o herramienta es esta?:



- a. Code Add
- b. Code Exp
- c. Code Pad
- d. Code Area

**Pregunta R1:** Indica cual de las siguientes afirmaciones es correcta:

- a. Un mapa es una colección que almacena pares llave/valor como entradas.
- b. Un mapa es una colección que almacena tríos llave/índice/valor como entradas.
- c. Un mapa es una colección que almacena pares índice/valor como entradas.
- d. Un mapa es una colección que almacena tríos índice/posición/valor como entradas.

**Pregunta R2:** Indique el resultado de ejecutar el siguiente código que se muestra a continuación:

```
public class test {
    public static void add3 (Integer i) {
        int val = i.intValue();
        val += 3;
        i = new Integer (val);
    }

    public static void main (String args[]) {
        Integer i = new Integer (0);
        add3 (i);
        System.out.println (i.intValue ( ) );
    }
}
```

- a. El programa indicará un fallo en tiempo de compilación
- b. El programa imprime por pantalla el valor 0
- c. El programa imprime por pantalla el valor 3
- d. El programa lanzará una excepción en la línea 3 (`int val = i.intValue();`)

## **PARTE PRÁCTICA [6,5 PUNTOS]:**

La Práctica del presente curso es diseño e implementación de un sistema integrado de gestión de una biblioteca (a partir de ahora, SIGB). En general, las funciones que tienen un SIGB son varias según el perfil de su usuario (que va desde el usuario de la biblioteca hasta su director) e incluyen las siguientes:

- Adquisiciones: la compra de materiales (libros en diferentes formatos, audiolibros, CDs de música, películas en DVD, etc.), gestión de compras, facturación, etc.
- Catalogar: la clasificación e indexación de los materiales de la biblioteca.
- Préstamos: prestar los materiales a los usuarios (tanto en papel como en otros formatos), reservas de materiales ya en préstamos, control de préstamos (emisión de avisos de materiales fuera de plazo), gestión de multas.
- Suscripciones: gestión de las suscripciones a revistas y periódicos.
- Catálogo en línea u OPAC (del inglés Online Public Access Catalog): interfaz pública a los servicios de la biblioteca (búsquedas, gestión de préstamos, etc.).
- Gestión de usuarios: altas, bajas, generación de tarjetas, historiales.

### **Funcionalidades**

Los SIGB permiten la implementación desde labores simples de gestión de una alta de usuario, hasta operaciones más complejas como es la gestión de préstamos o inventario. En esta práctica, se propondrán diferentes funcionalidades para el sistema de gestión bibliotecaria:

- Añadir nuevos materiales a la colección de la biblioteca (rellenando los datos de un formulario). Cada tipo de material debería tener su propia colección (libros, revistas, periódicos, audio, video, etc.).
- Borrar materiales de la colección.
- Realizar búsquedas sencillas sobre los materiales.
- Gestionar suscripciones a revistas y periódicos.
- Gestión de usuarios: altas, bajas, generación de tarjetas, historiales de préstamo, control de acceso (diferenciar entre dos perfiles: usuarios y bibliotecarios).
- Realización básica de Préstamos: prestar un material si está disponible en la biblioteca, asignar fechas de devolución.
- Producir listados de préstamos según el tipo de material.
- Realizar búsquedas flexibles sobre los materiales en la biblioteca combinando varios campos de búsqueda.
- Control de préstamos: número máximo de ítems de préstamo (6 por usuario, independiente de tipo de material), emisión de avisos de materiales fuera de plazo, gestión de multas, etc.
- Producir listados de los materiales prestados.
- Realizar búsquedas flexibles sobre los materiales en varias bibliotecas a la vez combinando varios campos de búsqueda.
- Préstamos entre bibliotecas: poder solicitar materiales a otras bibliotecas y procesar las solicitudes de otras bibliotecas. El procesamiento de dichas solicitudes se lleva a cabo usando archivos de solicitud de la siguiente manera:
  - Preparar y exportar una lista de solicitudes de materiales que se quiere hacer a una biblioteca. Se prepara la lista usando un formulario para identificar el nombre de la biblioteca, el nombre del libro, el autor y el nombre de esta biblioteca. Una vez terminado, se guarda la lista en un archivo de texto. No es necesario en esta práctica preocuparse de cómo se enviaría el archivo a otras bibliotecas.
  - Importar y procesar un archivo de solicitudes para materiales proveniente de otra biblioteca. Se debe actualizar el estatus de cada material para marcarse como prestado, pero en vez del identificador del usuario debería aparecer el identificador de la biblioteca.
- Control de reservas: poder reservar un material si está ya prestado, gestión de avisos (al usuario con el material que convendría devolverlo porque hay alguien esperando y al usuario con la reserva cuando el material ya está devuelto).

- 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]** Implementa un método (o métodos) que permitan añadir nuevos materiales a la colección de la biblioteca (rellenando los datos de un formulario). Cada tipo de material debería tener su propia colección (libros, revistas, periódicos, audio, video, etc.).
- c) **[2,0 puntos]** Implementa un método (o métodos) que permitan la importación y procesamiento de un archivo de solicitudes para materiales proveniente de otra biblioteca (préstamos entre bibliotecas). Se debe actualizar el estatus de cada material para marcarse como prestado, pero en vez del identificador del usuario debería aparecer el identificador de la biblioteca.
- d) **[1,5 puntos]** Para la siguiente versión del software se desea añadir la figura de la Editorial. De cada editorial se debe tener un listado de los libros y audiolibros que proporciona, así como su precio, que podría actualizarse manualmente después de la última venta. El sistema debería consultar el inventario a final de año, y de los libros más prestados (top 10%), realizar un pedido a la Editorial correspondiente de un 20% más de libros. El porcentaje de libros más prestados y el porcentaje de libros pedidos a las Editoriales, podrán ser configurables dependiendo del presupuesto de la biblioteca. ¿Qué cambios serían necesarios en el diseño para adaptar esta nueva funcionalidad? Implemente el método (o métodos) que permita esta nueva funcionalidad.