

**UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA – ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA INFORMÁTICA
71901072 – PROGRAMACIÓN ORIENTADA A OBJETOS (GRADO EN INGENIERÍA INFORMÁTICA /
TECNOLOGÍAS DE LA INFORMACIÓN)
SEPTIEMBRE 2016 – MODELO D – NO ESTÁ PERMITIDO EL USO DE MATERIAL ADICIONAL**

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: ¿Cuál es el resultado de ejecutar el siguiente código?

```
1. public abstract class ClaseA {
2.     public final void metodo1() {
3.         System.out.println("ClaseA");
4.     }
5.
6.     public static void main(String [] args) {
7.         ClaseA obj = new ClaseB();
8.         obj.metodo1();
9.     }
10. }
11.
12. class ClaseB extends ClaseA {
13.     public void metodo1() {
14.         System.out.println("ClaseB");
15.     }
16. }
```

- a. ClaseA
- b. ClaseB
- c. Error de compilación en la línea 7
- d. Error de compilación en la línea 13

Pregunta 2: De acuerdo a la bibliografía básica, el que un campo o miembro público de una clase sea estático implica que:

- a. Puedo acceder y modificar su valor sólo a través de un objeto.
- b. Puedo acceder y modificar su valor sin necesidad de haber instanciado objeto alguno.
- c. Todos los objetos tienen una copia de la variable.
- d. Es una variable global y se puede usar directamente en cualquier lugar sin hacer referencia a la clase correspondiente y sin instanciar objeto alguno.

Pregunta 3: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es incorrecta:

- a. Las clases pueden ser abstractas.
- b. Los diagramas de clases muestran las clases de una aplicación y la relación entre ellas.
- c. Las clases deben contener al menos un miembro de clase o campo.
- d. Las clases pueden contener métodos.

Pregunta 4: Dada la siguiente definición de clases:

```
1. public class ClaseA {
2.     protected void metodo1() {
3.         System.out.print("Entro en el método desde ClaseA");
4.     }
5. }
6.
7. class ClaseB extends ClaseA {
8.     public void metodo1() {
9.         System.out.print("Entro en el método desde ClaseB");
10.    }
11. }
```

¿Qué salida obtendremos al ejecutar el siguiente código dentro de un método *main* implementado en cualquiera de las clases?

```
15. ClaseA p = new ClaseB();
16. p.metodo1();
```

- a. Entro en el método desde Clase A
- b. Entro en el método desde Clase B
- c. Error de compilación en la línea 15
- d. Error de compilación en la línea 8

Pregunta 5: Dado el siguiente código indicar cuál de las afirmaciones es correcta. Suponga que las importaciones de librerías correspondientes se han realizado.

```
6. Set < Integer > conjunto = new HashSet < Integer > ();
7. conjunto.add(new Integer(86));
8. conjunto.add(75);
9. conjunto.add(new Integer(86));
10. conjunto.add(null);
11. conjunto.add(309);
12. Iterator i = conjunto.iterator();
13. while(i.hasNext()) {
14.     System.out.print(i.next());
15. }
```

- a. El código compila sin errores
- b. La salida que se muestra es 8675null309
- c. La línea 6 produce un error de compilación
- d. La línea 12 produce un error de compilación

Pregunta 6: ¿Cuál es el resultado de ejecutar el siguiente código?

```
6. List lista = new ArrayList();
7. lista.add("AA");
8. lista.add("BB");
9. lista.add(8);
10. for(int n=0;n<lista.size();n++) {
11.     System.out.print(lista.get(n));
12. }
```

- a. AABB
- b. AABBB8
- c. Error de compilación en la línea 6
- d. Error de compilación en la línea 9

Pregunta 7: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

- a. Una clase abstracta no puede implementar ninguna interface.
- b. Si una clase tiene sólo alguno de sus métodos abstractos, no es necesario declararla como abstracta.
- c. Una clase abstracta no puede extender otra clase que no sea abstracta.
- d. Una clase que hereda de una clase abstracta tiene que implementar todos los métodos abstractos para no ser abstracta.

Pregunta 8: Según el texto de la bibliografía básica de la asignatura, el alcance de una variable:

- a. Define el valor máximo que puede almacenar dicha variable.
- b. Define qué métodos de otras clases pueden acceder a la variable.
- c. Define qué clases externas pueden utilizar dicha variable.
- d. Ninguna de las anteriores.

Pregunta 9: ¿Cuál es el resultado de ejecutar el siguiente código?

```
10. String [] array= {"AA", "BB", "CC"};
11. for(int n = 0; n < array.length; n++) {
12.     System.out.print(array[n]);
13. }
14. System.out.print(n);
```

- a. AABBBCC
- b. AAB
- c. AABB
- d. El código no compila

Pregunta 10: ¿Cuál es el resultado de ejecutar el siguiente código?

```
1. public class ClaseA{
2.     private String miembro1;
3.     private boolean miembro2;
4.
5.     public static void main(String [] args ) {
6.         ClaseA obj = new ClaseA();
7.         if(!obj.miembro2) {
8.             System.out.println("Miembro1 = " + obj.miembro1);
9.         }
10.    }
11.}
```

- a. Error de compilación en la línea 6
- b. Error de compilación en la línea 7
- c. Error de compilación en la línea 8
- d. Miembro1 = null

Pregunta 11: ¿Cuál es el resultado de ejecutar el siguiente código?

```
6. String cadena1 = "Examen";
7. String cadena2 = new String(cadena1);
8. if(cadena1 == cadena2) {
9.     System.out.println("cadena1 == cadena2");
10. }
11. if(cadena1.equals(cadena2)) {
12.     System.out.println("cadena1.equals(cadena2)");
13. }
```

- a. No se muestra nada
- b. cadena1 == cadena2
- c. cadena1.equals(cadena2)
- d. B y C son ciertas

Pregunta 12: Según el texto de la bibliografía básica de la asignatura, indique cuál de las siguientes afirmaciones es correcta:

- a. El acoplamiento describe el encapsulamiento de las clases.
- b. El encapsulamiento apropiado en las clases reduce su cohesión.
- c. El encapsulamiento apropiado en las clases reduce su acoplamiento.
- d. La cohesión de una unidad de código refleja su acoplamiento.

Pregunta 13: ¿Cuál es el resultado de ejecutar el siguiente código?

```
1. class ClaseA {
2.     public float metodo1(double d) {
3.         System.out.println("ClaseA");
4.         return 1.0F;
5.     }
6. }
7.
8. public class ClaseB extends ClaseA {
9.     public double metodo1(double d) {
10.        System.out.println("ClaseB");
11.        return 1.0;
12.    }
13.
14.    public static void main(String [] args) {
15.        new ClaseB().metodo1(0.0);
16.    }
17. }
```

- a. ClaseA
- b. ClaseB
- c. 1.0
- d. El código no compila

Pregunta 14: De acuerdo a la bibliografía básica, ¿qué significa instanciar una clase?

- a. Duplicar una clase.
- b. Heredar de una clase.
- c. Crear un objeto a partir de una clase.
- d. Conectar dos clases entre sí.

RESERVA 1: Respecto a las excepciones en Java, ¿Cuál de las siguientes afirmaciones es correcta?

- a. Todas las subclases de la clase estándar de Java RuntimeException son excepciones comprobadas.
- b. Todas las subclases de la clase estándar de Java Exception son excepciones comprobadas.
- c. Error es una subclase directa de Throwable, mientras que Exception es una subclase directa de Error.
- d. Tanto Error como Exception son subclases directas de Throwable.

RESERVA 2: ¿Cuál de las siguientes afirmaciones es correcta?

- a. En Java no se permite la herencia múltiple de clases, ni tampoco la implementación múltiple de interfaces.
- b. En Java se permite la herencia múltiple de clases, pero no la implementación múltiple de interfaces.
- c. En Java no se permite la herencia múltiple de clases, pero sí la implementación múltiple de interfaces.
- d. En Java se permite la herencia múltiple de clases, y también la implementación múltiple de interfaces.

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 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]** Se desea incluir en la biblioteca un nuevo tipo de recurso, los videojuegos para diferentes consolas (PS3, PS4, Xbox 360, Xbox o Nintendo). Indica los cambios que habría que realizar en el sistema de gestión de préstamos, así como la estructura de clases resultante.
 - c) **[2 puntos]** Implementar los métodos de préstamo y devolución de videojuegos teniendo en cuenta, en el caso de las devoluciones, las implicaciones que dichos préstamos pueden acarrear: sanciones (si se entrega fuera de plazo), actualización de ficheros que contienen los préstamos (indicar la solución que se plantea en este caso), actualización de las reservas que haya sobre ese videojuego, etc. Justifíquese las opciones y decisiones que se tomen.
 - d) **[2 puntos]** Proporcione un método (o métodos) que permita mostrar por pantalla un formulario básico en **modo gráfico** que permita generar las estadísticas de los préstamos de videojuegos que se encuentran almacenados en un fichero. La pantalla permitirá elegir entre dos listados: videojuegos más prestados, usuarios más “jugones” (con mayor número de préstamos). Pedirá un rango de fechas y aplicará ese criterio a la hora de buscar los contenidos en los ficheros. Los mostrará por pantalla de mayor a menor. Justifíquese las opciones y decisiones que se tomen.