

**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)  
JUNIO 2016 – MODELO B – 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:** Según el texto de la bibliografía básica de la asignatura, un ArrayList tiene las siguientes características (indica la respuesta que **NO** es correcta):

- a. Puede incrementar su capacidad interna si es necesario.
- b. Mantiene su propio contador del número de elementos almacenados.
- c. Mantiene el orden de almacenamiento de los elementos.
- d. No permite la duplicación de objetos.

**Pregunta 2:** Según el texto de la bibliografía básica de la asignatura, ¿cuál de las siguientes clases **NO** es una colección?

- a. ArrayList
- b. HashSet
- c. HashMap
- d. ArraySet

**Pregunta 3:** La presencia de dos o más constructores en una clase se llama:

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

**Pregunta 4:** Dado el siguiente código:

```
1  import java.util.ArrayList;
2  class Persona {
3      private String nombre; private String telefono;
4      public Persona(String n, String t) {
5          nombre = n; telefono = t;
6      }
7      public String getNombre() {
8          return(nombre);
9      }
10 }
11 public class Prueba {
12     private ArrayList<Persona> amigos = new ArrayList<Persona>();
13
14     public Prueba() {
15         amigos.add(new Persona("Pepe","1234"));
16         amigos.add(new Persona("Sara","1236"));
17     }
```

```

18     public void borrarAmigo(String nombre) {
19         for (Persona p: amigos) {
20             if (p.getNombre().equals(nombre)) {
21                 amigos.remove(p);
22                 System.out.println("Amigo borrado");
23             }
24         }
25     }
26 }

```

¿Cuál es el resultado de compilar / llamar el método `borrarAmigo("Sara")`?

- Se produce un error de compilación.
- Se produce una excepción durante la ejecución
- Se produce un aviso(warning) durante la ejecución
- Se imprime por pantalla el mensaje: Amigo borrado y no se genera error o excepción alguno

**Pregunta 5:** Según el texto de la bibliografía básica de la asignatura, el estado de un objeto se denomina:

- El conjunto de parámetros que recibe y devuelve un objeto.
- El conjunto de valores de todos los atributos que definen al objeto.
- El conjunto de campos y métodos que componen el objeto.
- Ninguna de las anteriores.

**Pregunta 6:** ¿Cómo se activan por primera vez las herramientas de prueba de JUnit en BlueJ?

- No es necesario, ya vienen activadas.
- Con el botón derecho, seleccionando Activar (*Activate*) JUnit
- A través de la pestaña Miscelánea (*Miscellaneous*) del cuadro de diálogo Preferencias (*Preferences*).
- A través del menú.

**Pregunta 7:** Según el texto de la bibliografía básica de la asignatura, ¿qué es una aserción?:

- Una expresión que establece una condición que esperamos que sea cierta.
- Una expresión que resume la función de un método.
- Una instrucción que prueba la validez de una función.
- Ninguna de las respuestas anteriores.

**Pregunta 8:** Si simplificamos el ejemplo PhotoPost del libro de la asignatura de la siguiente manera:

```

1  public class Post {
2      private String usuario;
3
4      public Post(){}
5      public Post(String autor) {
6          usuario = autor;
7      }
8  }
9  class PhotoPost extends Post {
10     private String file;
11
12     public PhotoPost(String autor, String file) {
13         XXX
14         this.file = file;
15     }
16 }

```

¿Qué tendremos que añadir a la línea 13, en vez de XXX, para asignar el valor de autor a variable usuario en la clase Post?:

- `super();`
- `usuario = autor;`
- `super(autor);`
- `super.usuario = autor;`

**Pregunta 9:** Según el texto de la bibliografía básica de la asignatura, caracterizan los campos, constructores y métodos de la siguiente forma:

1. ... implementan el comportamiento de un objeto.
2. ... almacenan datos de manera persistente dentro de un objeto.
3. ... son responsables de garantizar que un objeto se configure apropiadamente a crearlo por primera vez.

¿Qué definición corresponde con que término?

- a. Campos = 1, Constructores = 2, Métodos = 3.
- b. Campos = 2, Constructores = 1, Métodos = 3.
- c. Campos = 2, Constructores = 3, Métodos = 1.
- d. Campos = 3, Constructores = 2, Métodos = 1.

**Pregunta 10:** Dado el siguiente fragmento de código del ejemplo del libro *ImageViewer*:

```

1 private void makeMenuBar(JFrame frame) {
2     final int SHORTCUT_MASK =
3         Toolkit.getDefaultToolkit().getMenuShortcutKeyMask();
4
5     JMenuBar menubar = new JMenuBar();
6     frame.setJMenuBar(menubar);
7
8     JMenu menu;
9     JMenuItem item;
10
11     // generar menú de preferencias
12     XXX
...

```

Si se quisiera añadir una nueva entrada en la barra de menús con el nombre Preferencias, ¿que habrá que añadir en la línea 13 en vez de XXX?:

- a. menu = new JMenuItem("Preferencias"); menubar.add(menu);
- b. menu = new JMenu("Preferencias"); frame.add(menu);
- c. menu = (Jmenu)new JMenuItem("Preferencias"); menubar.add(menu);
- d. menu = new JMenu("Preferencias"); menubar.add(menu);

**Pregunta 11:** Según el texto de la bibliografía básica de la asignatura, para crear una representación de un objeto en forma de String tenemos que implementar el método:

- a. getString()
- b. toString()
- c. printString()
- d. parseString()

**Pregunta 12:** Se quiere proporcionar dos constructores en la clase del reloj ClockDisplay del libro para iniciar el reloj de dos formas diferentes (fijando las horas y los minutos y fijando los minutos y los segundos):

```

1 public ClockDisplay(int hour, int minute)
2 {
3     hours = new NumberDisplay(24);
4     minutes = new NumberDisplay(60);
5     seconds = new NumberDisplay(60);
6     setTime(hour, minute, second);
7     this.hour = hour;
8     this.minute = minute;
9     this.second = 0;
10 }
11
12 public ClockDisplay(int minute, int second)
13 {
14     hours = new NumberDisplay(24);
15     minutes = new NumberDisplay(60);
16     seconds = new NumberDisplay(60);
17     setTime(hour, minute, second);
18     this.hour = 0;
19     this.minute = minute;

```

```

20         this.second = second;
21     }

```

¿Cuál es el resultado de compilar / llamar al constructor con los valores 2 y 15?

- Generamos un objeto ClockDisplay con la hora 02:15:00
- Generamos un objeto ClockDisplay con la hora 00:02:15
- Tanto la respuesta a como la b son posibles.
- Se produce un error de compilación en la línea 12.

**Pregunta 13:** Para definir una nueva clase de excepción, como se ha hecho en el ejemplo *AddressBook* del libro, se hace de la siguiente forma:

```

1     public class NoMatchingDetailsException extends XXX {
2     private String key;
3
4     public NoMatchingDetailsException(String key) {
5         this.key = key;
6     }
7
8     ...

```

¿Que habrá que añadir en la línea 13 en vez de XXX para declarar correctamente la nueva clase?:

- RuntimeException
- Exception
- IOException
- java.util.Exception

**Pregunta 14:** Si se quiere insertar el siguiente método en la clase de la máquina expendedora de billetes, ¿cuál es el resultado de compilar / ejecutar el método con un valor de coste de 50?

```

1     public void probarDinero(int dinero)
2     {
3         if (dinero = 50) {
4             System.out.println("No se admiten billetes de 50€.");
5         }
6         else {
7             System.out.println("Impresión de billete en curso.");
8         }
9     }

```

- Se produce un error de compilación en la línea 3.
- Se produce un error de ejecución en la línea 4.
- Se imprime por pantalla el mensaje: No se admiten billetes de 50€.
- Se imprime por pantalla el mensaje: Impresión de billete en curso.

**Reserva 1:** Supongamos que reescribimos una parte de la simulación de los zorros y los conejos del libro de la forma que se muestra a continuación:

```

1     import java.util.List;
2     public abstract class Animal {
3
4         public Animal() {}
5
6         abstract public void act(List<Animal> newAnimals);
7         protected void isAlive() {}
8         protected void setDead() {}
9     }
10
11     class Rabbit extends Animal {
12
13         XXX
14         private void incrementAge() {}
15         private void giveBirth(List<Animal> newRabbits) {}
16     }

```

¿Qué tendremos que añadir a la línea 13, en vez de XXX, para que el código no genere un error de

compilación?:

- a. `public Rabbit();`
- b. `public void act(List<Animal> newRabbits) {}`
- c. `public void act(List<Animal> newRabbits);`
- d. El código se compilará sin ningún error de todas formas.

**Reserva 2:** Según el texto de la bibliografía básica de la asignatura, ¿cómo se usa *instanceof* para averiguar si el tipo dinámico de un objeto (miobj) es de una cierta clase (MiClase)?

- a. `MiClase instanceof(miobj)`
- b. `java.lang.Object instanceof(miobj, MiClase)`
- c. `miobj instanceof MiClase`
- d. `Object instanceof(miobj, MiClase)`

## **PARTE PRÁCTICA [6,5 PUNTOS]:**

La Práctica del presente curso va a consistir en diseñar e implementar un sistema integrado de gestión de una biblioteca (a partir de ahora, SIGB). Han existido versiones sencillas de estos sistemas incluso antes de la existencia de los computadores, donde se almacenaba información sobre los materiales de la biblioteca (por aquél entonces, libros, revistas, periódicos) en fichas en formato papel, guardadas en cajones clasificados. El primer paso hacia la informatización de estos sistemas, hacia lo que hoy en día es un SIGB, tuvo lugar en 1936 en la Universidad de Texas, donde la información sobre los libros estaba representada en tarjetas perforadas y cargada en su computador central. Con el avance de la informática a lo largo de los años, el manejo de los recursos de una biblioteca, a través de sistemas integrados de gestión, se ha podido llevar a cabo con una amplia gama de dispositivos, desde terminales tontas hasta teléfonos móviles.

### **Funcionalidades**

Un SIGB proporciona las siguientes funcionalidades:

- 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 guardar 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 esta devuelto).
- 
- a) **[1 punto]** Diseñar utilizando el 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 punto]** Implementar el método que permite producir listados de préstamos según el tipo de material. Justifíquese las opciones y decisiones que se tomen.
  - c) **[2 puntos]** Implementar la funcionalidad de la preparación y exportación de una lista de solicitudes que se quiere hacer a una biblioteca como parte del préstamo entre bibliotecas. Justifíquese las opciones y decisiones que se tomen.
  - d) **[2,5 puntos]** Se quiere añadir una cafetería a la biblioteca para que los clientes puedan tomar algo mientras miran los libros, revistas o periódicos. Se quiere utilizar el sistema de gestión de la biblioteca para gestionar la compra y venta de los productos de la cafetería. Indique los cambios que serían necesarios en el diseño y la implementación para permitir esa nueva funcionalidad.