

## **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:** Dado el siguiente código: ¿Cuál de las siguientes declaraciones **NO** es un ejemplo correcto de paso de parámetros?

- a. `public boolean valida(Fecha i) {return false;}`
- b. `public void imprimir(String i) {}`
- c. `public int calcArea (altura i, anchura *j) {return area;}`
- d. `public double d(double [] i) {return i[0];}`

**Pregunta 2:** Si tenemos un objeto **persona** con un método **edad()**, ¿cómo se invoca el método en BlueJ?

- a. Hacer clic con el botón izquierdo del ratón sobre el objeto persona y seleccionar el método.
- b. Hacer doble clic con el botón izquierdo del ratón sobre el objeto y seleccionar el método.
- c. Hacer clic con el botón derecho del ratón sobre el objeto y seleccionar el método.
- d. Todas las respuestas anteriores son correctas.

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

```
class Persona {
    void resumir(){System.out.println("resumiendo...");}
}

class Medico extends Persona {
    void resumir(){System.out.println("resumiendo médico");}
}

class Ingeniero extends Persona {
    void resumir(){System.out.println("resumiendo ingeniero");}
}

public class Empleados {
    public static void main(String args[]){
        Persona p;
        p = new Medico();p.resumir();
        p = new Ingeniero();p.resumir();
    }
}
```

¿Cuál es el resultado que se produce?:

- a. resumiendo médico  
resumiendo ingeniero
- b. resumiendo médico  
resumiendo médico
- c. Error de compilación
- d. Error de ejecución

**Pregunta 4:** Según el texto de la bibliografía básica de la asignatura, la **abstracción**:

- a. es la capacidad de ignorar los detalles de partes de un programa.
- b. no permite ignorar la complejidad de un programa
- c. es la capacidad de destacar los detalles de partes de un programa.
- d. Ninguna de las anteriores.

**Pregunta 5:** Según el texto de la bibliografía básica de la asignatura, los métodos principales de un ArrayList son (indica la respuesta FALSA):

- a. add()
- b. isFull()
- c. size()
- d. get()

**Pregunta 6:** Dado el siguiente código:

```
public class Persona {
    String nombre;
    Persona(String n) {
        nombre = n;
    }
    Persona() {
        this("Arturo");
    }
    public static void main(String args[]) {
        Persona p1 = new Persona("Miguel");
        Persona p2 = new Persona();
    }
}
```

Después de ejecutar el método main, ¿qué valor tendría la variable n para los objetos p1 y p2?

- a. Arturo, null
- b. Arturo, Arturo
- c. Miguel, Arturo
- d. Miguel, null

**Pregunta 7:** Según el texto de la bibliografía básica de la asignatura, la duplicación de código es un síntoma de:

- a. Buena cohesión.
- b. La evidencia de un problema complejo.
- c. Mal encapsulamiento.
- d. Mal diseño.

**Pregunta 8:** Según el texto de la bibliografía básica de la asignatura, **un conjunto** es:

- a. una colección que almacena cada elemento individual varias veces.
- b. una colección que almacena cada elemento individual como máximo una vez.
- c. un arreglo que almacena cada elemento individual varias veces.
- d. un arreglo que almacena cada elemento individual como máximo una vez.

**Pregunta 9:** Según el texto de la bibliografía básica de la asignatura, el **diseño dirigido por responsabilidad** es el proceso de:

- a. diseñar clases asignando unas responsabilidades bien definidas a cada clase.
- b. diseñar todas las clases por iguales.
- c. diseñar clases asignando unas responsabilidades bien definidas a algunas clases.
- d. Ninguna de las anteriores.

**Pregunta 10:** Dado el siguiente código:

Nº de línea	Código
1	public class Saludar {
2	public static void main(String[] args) {
3	String nombre = "Pepe";
4	saludar(nombre);
5	System.out.println(nombre);
6	}
7	public static void saludar(String nombre) {
8	private nombre = "Hola " + nombre;
9	}
10	}

¿Cuál sería el resultado de compilar/ejecutar el código?:

- a. Pepe
- b. Hola Pepe
- c. Error en la línea 7
- d. Error en la línea 8

**Pregunta 11:** Según el texto de la bibliografía básica de la asignatura, un **depurador** es:

- a. una herramienta *software* que proporciona soporte para realizar un recorrido de un segmento de código.
- b. una herramienta *software* que proporciona soporte para comprobar el estilo de un segmento de código.
- c. una técnica que proporciona soporte para realizar un recorrido de un segmento de código.
- d. una técnica que proporciona soporte para comprobar el estilo de un segmento de código.

**Pregunta 12:** Dado el siguiente código:

```
public class Numeros {
    int n1, n2;
    public void num() {
        n1 = (n1 + 2) % n2;
        System.out.print(n1 + " ");
    }
    public void calc() {
        int i = 0; n1 = 0; n2 = 8;
        while(i++<n2) {
            num();
        }
    }
    public static void main(String args[]) {
        Numeros nos = new Numeros();
        nos.calc();
    }
}
```

¿Cuál sería el resultado de ejecutar el código?:

- a. 0 2 4 6 0 2 4 6
- b. 2 4 6 0 2 4 6 0
- c. 2 4 6 0 2 4 6
- d. 2 4 6 0 2 4 6 0 2

**Pregunta 13:** Según el texto de la bibliografía básica de la asignatura sobre los campos y constructores, se puede decir:

- a. Los campos almacenan datos de manera persistente dentro de un objeto.
- b. Los campos almacenan datos de manera no persistente dentro de un objeto.
- c. Los constructores son responsables de garantizar que un objeto se configure apropiadamente en cualquier momento.
- d. Los constructores implementan el comportamiento de un objeto.

**Pregunta 14:** Para manejar los errores de tiempo de ejecución en Java existe la construcción **try ... catch** que se utiliza de la siguiente forma:

```
try {
    // aquí se protege una o más sentencias
}
XXXXXXXXXX {
    // aquí se informa y se recupera de la excepción
}
```

En vez de XXXXXXXXXX habrá que poner:

- a. catch (Exception ex)
- b. catch (exception Ex)
- c. catch (Event ev)
- d. catch (event Ev)

**Pregunta R1:** Según el texto de la bibliografía básica de la asignatura sobre **los tipos estáticos y dinámicos**, se puede decir que para una variable v:

- a. El tipo dinámico es el tipo tal como está declarado en el código fuente y el tipo estático es el tipo del objeto que está almacenado actualmente en v.
- b. El tipo estático es el tipo tal como está declarado en el código fuente y el tipo dinámico es el tipo del objeto que está almacenado actualmente en v.
- c. El tipo estático es el tipo tal como está declarado en el código fuente y el tipo dinámico también.
- d. El tipo estático es el tipo tal como está declarado en el código fuente y el tipo dinámico es el tipo del objeto que ha estado almacenado en v.

**Pregunta R2:** Según el texto de la bibliografía básica de la asignatura, la definición de un método abstracto está compuesta de:

- a. una cabecera de método, sin que exista un cuerpo del mismo.
- b. un cuerpo sin cabecera de método.
- c. una cabecera de método con un cuerpo correspondiente.
- d. No se puede definir métodos abstractos en Java.

## **PARTE PRÁCTICA [6,5 PUNTOS]:**

La Práctica del presente curso consiste en implementar un sistema integrado de gestión de un estudio de arquitectura. Los estudios de arquitectura, descrito de una forma genérica, se dedican a dos tipos de tareas. Por un lado, desarrollan proyectos arquitectónicos para la construcción o rehabilitación de edificios. Por otro lado, llevan a cabo una tarea de certificación.

Por tanto, el estudio ha de guardar información acerca de los proyectos y/o certificados que ha solicitado cada cliente, en un historial que es único para cada uno de los clientes.

En la aplicación que se desea desarrollar, es necesario almacenar dos tipos distintos de persona, para diferenciar si se trata de un cliente del estudio, o de un empleado del mismo. Dentro de los empleados, encontramos cuatro subtipos: el administrador del sistema, los arquitectos que realizan los proyectos y certificados, los aparejadores que llevan el día a día de una construcción, y los contables que llevan la economía del estudio.

Dentro de los proyectos de arquitectura que se pueden realizar encontramos tres tipos: los proyectos residenciales que engloban edificios y viviendas unifamiliares; los proyectos no residenciales como naves industriales, museos, o cualquier otra construcción no residencial; y los proyectos de rehabilitación que consiste en reformar una construcción ya finalizada previamente.

Por último, los certificados que desarrolla un estudio de arquitectura pueden ser de cuatro tipos: los certificados de habitabilidad, que además caducan cada 15 años y hay que renovarlos; la inspección técnica de edificios que se aplica a edificios comunitarios y es obligatoria a partir de los 45 años de su construcción; los certificados de eficiencia energética que se emiten con una categoría que puede ser desde la "A" (la más eficiente) a la "G" (la menos eficiente); y finalmente los informes periciales.

En general, las funciones que tiene el sistema de gestión son las siguientes:

- Gestión de usuarios: altas, bajas, modificaciones de las personas que figuran en el sistema (empleados -- administradores, arquitectos, aparejadores y contables -- y clientes). La primera vez que acude un cliente al estudio hay que darle de alta en el sistema.
- Desarrollo de un proyecto arquitectónico: los clientes acuden al estudio para solicitar un proyecto arquitectónico. Este proyecto puede ser de los tipos indicados anteriormente, y será necesario almacenar la fecha de solicitud, de entrega al cliente, duración prevista de la obra y presupuesto de ejecución. Una vez contratada la construcción también será necesario almacenar las fechas de inicio de construcción y duración prevista, y al finalizar la obra la fecha de fin de la obra. Un cliente puede solicitar todos los proyectos que desee y cada solicitud tiene que registrar los datos del cliente y del arquitecto que los desarrollaron. Cada proyecto arquitectónico tiene que registrar cierta información asociada al edificio. En el caso de proyectos residenciales tendrá que constar la dirección, superficie del terreno, superficie del edificio, plantas, habitaciones y baños. En el caso de proyectos no residenciales, será necesario registrar la dirección, superficie del terreno, superficie

del edificio y finalidad de la obra. Para los proyectos de rehabilitación habrá que registrar la dirección y la superficie a reformar. Además, cada proyecto tiene un coste que será determinado por un contable, el cual también tendrá que estar asociado al proyecto y quedar constancia en el sistema. Finalmente, un aparejador tendrá que llevar a cabo el control de la obra y por tanto habrá que asignar un aparejador a cada proyecto.

- Desarrollo de un certificado: los clientes acuden al estudio para solicitar un certificado. Este certificado puede ser de los tipos indicados anteriormente, y será necesario almacenar la fecha de solicitud y de entrega al cliente. Un cliente puede solicitar todos los certificados que desee y cada solicitud tiene que registrar los datos del cliente y del arquitecto que los desarrollaron. Además, cada proyecto tiene un coste que será determinado por un contable, el cual también tendrá que estar asociado al certificado y quedar constancia en el sistema. El sistema tendrá que almacenar un histórico para cada vivienda o edificio sobre los que se ha solicitado un certificado y en el que se puedan consultar todos los certificados asociados a esa vivienda o edificio. En el caso de los certificados de habitabilidad tendrá que haber una visita de un aparejador del estudio al edificio y un arquitecto desarrollará el certificado. Las fechas de la visita del aparejador y de la emisión del certificado tendrán que quedar registradas en el sistema, ya que a los 15 años caducan y el estudio mediante el administrador tendrá que ponerse en contacto con el cliente para informar de este hecho y preguntar por su posible renovación. La inspección técnica de edificios se aplica a edificios comunitarios y es obligatoria a partir de los 45 años de su construcción, por tanto, en el momento en que una vivienda para la que se ha desarrollado un proyecto cumple 45 años desde el fin de obra, el administrador se pondrá en contacto con el cliente de dicho proyecto para informar de este hecho y preguntar por su posible inspección por parte del estudio. Los certificados de eficiencia energética se emiten con una categoría que puede ser desde la “A” (la más eficiente) a la “G” (la menos eficiente) y además de esta categoría tendrá que registrar la fecha de emisión del certificado y del aparejador que haya realizado la visita. Por último, los informes periciales son desarrollados por un arquitecto del estudio y tendrán una fecha de emisión.
- Comunicación con los clientes: la función del administrador del estudio será la de ponerse en contacto con los clientes que o bien han solicitado un proyecto o un certificado. Para ello podrá obtener en todo momento el listado de dichos clientes. Además, podrá generar un listado de clientes cuya vivienda tenga más de 45 años para informar de la necesidad de obtener un certificado de inspección técnica de edificios y otro listado con los certificados de habitabilidad que han cumplido 15 años con el objetivo de informar a dichos clientes de la necesidad de renovarlos.
- Ejecución de obras: Una vez el cliente ha obtenido el proyecto y decide que el propio estudio desarrolle la obra, el administrador gestionará el calendario teniendo en cuenta que el estudio no puede tener más de tres obras en marcha al mismo tiempo. Para ello el administrador consultará la duración prevista de la obra que consta en el proyecto y asignará la fecha de inicio de la obra. Cuando la obra haya finalizado, también actualizará la fecha de fin de obra.



Se pide realizar las siguientes tareas:

- a) **[1,0 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]** Implementar un método (o métodos) que permita(n) la gestión de usuarios (altas, bajas y modificaciones) en el sistema. Se debe justificar las opciones y decisiones que se tomen.
- c) **[2 puntos]** Implementar un método (o métodos) que permita(n) almacenar un histórico para cada vivienda o edificio sobre los que se ha solicitado un certificado y en el que se puedan consultar todos los certificados asociados a esa vivienda o edificio. Se debe justificar las opciones y decisiones que se tomen.
- d) **[2,0 puntos]** Implementar un método (o métodos) que permita(n) la visualización de todos los datos asociados a un proyecto. Se debe justificar las opciones y decisiones que se tomen.