

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: Indique cuál de las siguientes declaraciones es válida para el método main:

- a. `public static void main(String args[]);`
- b. `static public void main(String);`
- c. `public static void main(String);`
- d. `public static int main(String args[]);`

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));
```

Indique cuál será la salida por pantalla (cada valor en una línea diferente):

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

Pregunta 4: Supongamos que queremos implementar una Agenda, ¿cuál sería la salida del siguiente código?

```
public class Agenda {

    public static void main(String argv[]){
        Agenda agenda = new Agenda();
    }

    protected Agenda(){
        for(int i=0; i<10; i++){
            System.out.println(i);
        }
    }
}
```

- a. Error de Compilación ya que los constructores no pueden ser declarados como "protected".
- b. Error en tiempo de ejecución ya que los constructores no pueden ser declarados como "protected".
- c. Compilación correcta y salida de los dígitos de 0 a 10.
- d. Compilación correcta y salida de los dígitos de 0 a 9.

Pregunta 5: Indique cuál de las siguientes afirmaciones es correcta:

- a. El lenguaje Java tiene tres tipos de ciclo: while, while-do y for-each.
- b. En caso de un bucle no relacionado con colecciones el bucle for-each no tiene utilidad.
- c. El tipo de la variable de ciclo no tiene porqué ser el mismo que el tipo del elemento declarado para la colección que estamos recorriendo con un ciclo.
- d. Un índice es un objeto que proporciona funcionalidad para recorrer todos los elementos de una colección.

Pregunta 6: Indique cuál de las siguientes afirmaciones es correcta:

- a. La interfaz de una clase describe lo que una clase hace y cómo se puede utilizar mostrando su implementación.
- b. El código fuente completo que define una clase es la interfaz de dicha clase.
- c. Se dice que un objeto es inmutable si su contenido o estado no puede cambiarse después de crearlo excepto si la interfaz es mutable.
- d. La documentación de la librería de clases Java muestra detalles acerca de todas las clases de la librería.

Pregunta 7: Indique cuáles de las siguientes expresiones resultan verdaderas:

- 1. `! (4 < 5)`
- 2. `(2 > 2) || ((4 == 4) && (1 < 0))`
- 3. `(2 > 2) || (4 == 4) && (1 < 0)`
- 4. `(2 > 2) || !((4 == 4) && (1 < 0))`
- 5. `(34 != 33) && ! false`

- a. Las expresiones 4 y 5.
- b. Las expresiones 3 y 4.
- c. Las expresiones 2 y 4.
- d. Las expresiones 3 y 5.

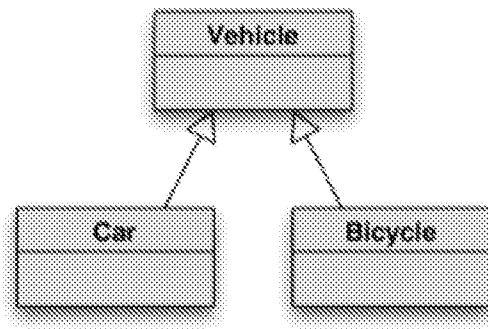
Pregunta 8: Indique cuál de las siguientes afirmaciones es correcta:

- a. El término cohesión describe la interconexión de las clases.
- b. El término acoplamiento describe lo bien que una unidad de código se corresponde con una tarea lógica o con una entidad.
- c. La duplicación de código produce errores de ejecución.
- d. Una adecuada encapsulación de las clases reduce el acoplamiento y conduce, por tanto, a un mejor diseño.

Pregunta 9: Indique cuál de las siguientes afirmaciones es correcta:

- a. La prueba es la actividad de descubrir si una pieza de código produce el comportamiento pretendido.
- b. La depuración es el intento de localizar y corregir el origen de un error.
- c. La prueba de unidad se refiere a las pruebas de las partes individuales de una aplicación, como los métodos y las clases.
- d. Todas las respuestas anteriores son correctas.

Pregunta 10: Dada la siguiente jerarquía de herencia:



Indique cuál de las siguientes asignaciones es correcta:

- a. `Vehicle v1 = new Vehicle();`
- b. `Vehicle v2 = new Bicycle();`
- c. `Vehicle v3 = new Car();`
- d. Todas las asignaciones anteriores son correctas.

Pregunta 11: Indique cuál de las siguientes afirmaciones es correcta:

- a. El tipo dinámico de una variable `v` es el tipo tal como está declarado en el código fuente.
- b. El tipo estático de una variable `v` es el tipo del objeto que está almacenado actualmente en `v`.
- c. Declarar un campo o un método protegido impide acceder directamente a él desde las subclases.
- d. Todas las afirmaciones anteriores son falsas.

Pregunta 12: Indique cuál de las siguientes opciones declarará un método en una clase que fuerza a una subclase a implementarlo:

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

Pregunta 13: Indique cuál de las siguientes afirmaciones es FALSA:

- a. Un método abstracto utiliza como prefijo la palabra clave `abstract`
- b. Un método abstracto tiene un cuerpo del método en su definición.
- c. Un método abstracto tiene una cabecera que termina con un punto y coma.
- d. Un método abstracto nunca se puede ejecutar.

Pregunta 14: En el siguiente fragmento de código hemos definido la ejecución de cinco bloques. Estos bloques se ejecutarán dependiendo de las excepciones que se produzcan en cada caso.

```
// Bloque1
try{
    // Bloque2
} catch (ArithmeticException e) {
    // Bloque3
} finally{
    // Bloque4
}
// Bloque5
```

Indique cuál de las siguientes afirmaciones es correcta:

- a. El Bloque4 se ejecutará aunque haya una excepción producida por un acceso a un objeto nulo (null) en el Bloque2
- b. El Bloque4 no se ejecutará si se produce una excepción de tipo aritmético en el Bloque2
- c. El Bloque4 no se ejecutará si se produce un acceso a un objeto nulo (null) en el Bloque2
- d. El Bloque4 se ejecutará antes que el Bloque3 si se produce una excepción de tipo aritmético en el Bloque2

Pregunta R1: Indique cuál de las siguientes afirmaciones es correcta en relación a BlueJ:

- a. Un punto de interrupción es un indicador asociado a un conjunto de líneas de código.
- b. Los puntos de interrupción se definen mediante la ventana del editor.
- c. Los puntos de interrupción solo se pueden definir dentro de las clases abstractas.
- d. Todas las afirmaciones anteriores son falsas.

Pregunta R2: Dado el siguiente fragmento de código,

```
int electrodomestico = 1;
boolean[] ventas = new boolean[3];
boolean financiado = ventas [electrodomestico];
```

Indique cuál de las siguientes afirmaciones es correcta en relación al valor de la variable "financiado".

- a. financiado tiene el valor false
- b. financiado tiene el valor 0
- c. financiado tiene el valor null
- d. Se produce una excepción y financiado no posee ningún valor

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 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 dar de alta un nuevo proyecto arquitectónico con los datos de los diferentes usuarios que participan en un proyecto de este tipo.
- c) **[2,0 puntos]** Implementar un método (o métodos) que permita producir diferentes listados del funcionamiento del estudio: un listado de clientes cuya vivienda tenga más de 45 años y otro listado con los certificados de habitabilidad que han cumplido 15 años.
- d) **[1,5 puntos]** Indique los cambios necesarios en el diseño de clases y en los métodos implementados anteriormente que permita añadir un nuevo usuario Delineante, que se encargue de: 1) los informes necesarios durante la ejecución de obras; 2) de los certificados y 3) toma de medidas y diseño de planos en caso de un proyecto arquitectónico.