

PARTE TEÓRICA - TEST [2,5 PUNTOS]:

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, ¿qué es el *estado* de un objeto?

- a. La instancia de una clase.
- b. El conjunto de valores de todos los métodos que definen a un objeto.
- c. El conjunto de valores de todos los atributos que definen a un objeto.
- d. Ninguna de las anteriores.

Pregunta 2: ¿A qué tipo pertenecen los siguientes campos?

```
private boolean listo;  
private Alumno alumno;  
private String nombre;
```

- a. boolean, Alumno, String.
- b. private, private, private.
- c. Boolean, alumno, String.
- d. listo, alumno, nombre.

Pregunta 3: Según el texto de la bibliografía básica de la asignatura, ¿qué almacenan las variables declaradas a partir de una clase?

- a. Objetos.
- b. Referencias a objetos.
- c. Copias de objetos.
- d. Ninguna de las anteriores.

Pregunta 4: Si llamamos al método toUpperCase() de una instancia (nombre) de la clase String de la siguiente forma:

```
nombre.toUpperCase();
```

¿Cuál sería el resultado de compilar y/o ejecutar el código?

- a. El código generaría un error de compilación.
- b. El código generaría un error de ejecución.
- c. El código convertiría la cadena nombre en mayúsculas.
- d. Ninguna de las anteriores.

Pregunta 5: En el siguiente fragmento de código, ¿cómo se crea la instancia del ArrayList archivos en la línea 8?

```
1      import java.util.ArrayList;
2
3      public class MusicOrganizer {
4          private ArrayList<String> archivos;
5
6          public MusicOrganizer()
7          {
8
9          }
10 }
```

- a. archivos = new ArrayList;
- b. archivos = new ArrayList(String)();
- c. archivos = new ArrayList<String>();
- d. archivos = new ArrayList<String>;

Pregunta 6: Según el texto de la bibliografía básica de la asignatura, ¿Cómo se llama la idea de que cada clase debe ser responsable de gestionar sus propios datos?:

- a. El acoplamiento.
- b. La cohesión.
- c. El diseño dirigido por responsabilidad.
- d. Ninguna de las anteriores.

Pregunta 7: JUnit nos permite comprobar que un método devuelve un valor concreto. ¿Cómo se haría esa prueba para comprobar que el método comprobarPrecio() de una instancia (ventas) de una clase devuelva el valor de 500?

- a. assertEquals(500, ventas.comprobarPrecio());
- b. assertEquals(500, ventas.comprobarPrecio());
- c. assertEquals(500, ventas.comprobarPrecio());
- d. Ninguna de las anteriores.

Pregunta 8: Según el texto de la bibliografía básica de la asignatura, las ventajas de la herencia incluyen (indica la respuesta **incorrecta**):

- a. Facilita la duplicación de código.
- b. Facilita la reutilización de código.
- c. Facilita el mantenimiento de código.
- d. Facilita la ampliabilidad de código.

Pregunta 9: Según el texto de la bibliografía básica de la asignatura, para averiguar la clase de un objeto, se usa el operador:

- a. super
- b. instanceof
- c. instanceof
- d. Ninguna de las anteriores.

Pregunta 10: Según el texto de la bibliografía básica de la asignatura, un mapa es una colección que almacena (indique la afirmación correcta):

- a. pares llave/valor como entradas.
- b. tríos llave/índice/valor como entradas.
- c. pares índice/valor como entradas.
- d. tríos índice/posición/valor como entradas.

Pregunta 11: ¿Cómo se podría añadir un menú nuevo a una barra de menús (barraMenus) en Java?

- a. JMenu archivoMenu = new JMenu("Archivo"); barraMenu.add(archivosMenu);
- b. JMenu archivoMenu = new JMenu("Archivo"); barraMenu.addMenu(archivosMenu);
- c. JMenuItem archivoMenu = new JMenuItem("Archivo"); barraMenu.add(archivosMenu);
- d. Ninguna de las anteriores

Pregunta 12: Respecto a File y Path, podemos afirmar lo siguiente:

- a. La interfaz File permite a un programa consultar los detalles relativos a un archivo externo, de una forma independiente del sistema de archivos concreto sobre el que se esté ejecutando el programa.
- b. La interfaz Path dispone por sí misma de sendos métodos exists y canRead.
- c. La interfaz Files proporciona un gran número de métodos estáticos para consultar los atributos de un objeto Path.
- d. Ninguna de las anteriores.

Pregunta 13: Dado el siguiente código:

```
1 public class PruebaExcepciones {
2     public PruebaExcepciones(){}
3     public void lanzarExcepcion(int prueba)
4     {
5         if(prueba == 1) {
6             throw new IllegalArgumentException("Excepción lanzada");
7         }
8         System.out.println("Prueba superada");
9     }
10 }
```

¿Cuál es el resultado de compilar / llamar el método lanzarExcepcion(1)?

- a. El problema se compila y se ejecuta generando una excepción: java.lang.IllegalArgumentException.
- b. El problema se compila y se ejecuta generando la salida Prueba superada.
- c. Se produce un error de compilación en la línea 1.
- d. Se produce un error de compilación en la línea 2.

Pregunta 14: Como parte del proceso de diseño de un programa orientado a objetos se pueden usar las tarjetas CRC. ¿Qué significa CR?

- a. Collaborators/Responsibilities/Collaboration.
- b. Class/Responsibilities/Class.
- c. Class/Responsibilities/Collaborators.
- d. Ninguna de las anteriores.

RESERVA 1: En BlueJ, con un proyecto abierto, ¿cómo se puede acceder al código fuente de una clase? (Indica la respuesta incorrecta)

- a. Haciendo clic con el botón izquierdo dos veces sobre una clase.
- b. Haciendo clic con el botón derecho y seleccionando "Abrir Editor".
- c. Seleccionando la clase e ir al menú Herramientas (Tools) y seleccionando la entrada "Abrir Editor".
- d. Todas las anteriores.

RESERVA 2: Sobre los tipos estático y dinámico de una variable se puede decir, según el texto de la bibliografía básica de la asignatura:

- a. El tipo estático de una variable es el tipo tal como está declarado en el código fuente. El tipo dinámico de una variable es el tipo del objeto que está almacenado actualmente.
- b. El tipo dinámico de una variable es el tipo tal como está declarado en el código fuente. El tipo estático de una variable es el tipo del objeto que está almacenado actualmente.
- c. Tanto el tipo estático como el tipo dinámico de una variable son el tipo tal como está declarado en el código fuente.
- d. Tanto el tipo estático como el tipo dinámico de una variable son el tipo del objeto que está almacenado actualmente.

PARTE PRÁCTICA [6,5 PUNTOS]:

La práctica del presente curso consiste en diseñar e implementar un sistema integrado de gestión de un taller de vehículos. Todos los dueños de algún tipo de vehículo (moto, coche, furgoneta, etc.) tienen experiencia en llevar su vehículo al taller para algún tipo de revisión y/o reparación. Puede que sea una puesta a punto antes de las vacaciones o algún viaje largo, o debido a un problema concreto o ruido que hace, o alguna fuga de líquido, etc. Hace años, el responsable de un taller no haría más que apuntar en un libro una breve descripción del vehículo, el motivo por el cual ha acudido al taller su dueño y algún número de contacto. Hoy en día, debido en parte a la competencia entre talleres y al deseo de aportar servicios de calidad para poder fidelizar al cliente, los talleres usan sistemas de gestión para todo el proceso de recepción, procesado y facturación de cada vehículo que pasa por el taller.

En general, las funciones que tienen un sistema de gestión de un taller de coches son varias:

- **Recepción del vehículo:** al entrar un vehículo en el taller hay que generar una ficha con los datos (si ya no forma parte del registro histórico del taller) más importantes (marca, modelo, matrícula, cliente, etc.), dejar constancia del motivo de la visita (problema mecánico, revisión, etc.) e imprimir el formulario de autorización que el dueño tiene que firmar para autorizar el trabajo.
- **Asignación de los trabajos a los mecánicos:** a medida que vayan terminando trabajos, el jefe del taller les va asignando nuevos vehículos.
- **Procesado de los vehículos:** un mecánico, al terminar la reparación de un vehículo, acude al sistema para ver los siguientes trabajos que le corresponden. Según la información del sistema, tiene que averiguar el problema, llevar a cabo el trabajo correspondiente y dejar constancia del proceso realizado en el sistema para que el comercial pueda informar al cliente de que su vehículo está listo para recoger. Una vez que el mecánico termine con un coche, en el sistema le aparece reflejado el siguiente vehículo asignado. Si por el motivo que sea (por ejemplo, falta de piezas o que un cliente no haya dado su autorización a realizar una reparación debido al coste), un mecánico deja un trabajo en un estado sin completar, anota en el sistema el motivo y pasa al siguiente trabajo.
- **Gestión de usuarios:** altas, bajas, modificaciones de las personas que figuran en el sistema (miembros del taller [jefe, mecánico, comercial] y clientes). La primera vez que acude un cliente al taller hay que darle de alta en el sistema.
- **Gestión de clientes por parte del comercial:** comunicar a los clientes el precio de una reparación, informarles de que su vehículo está listo para recoger, presentar ofertas especiales (por ejemplo, revisión antes de la inspección técnica del vehículo [ITV] o puesta a punto para las vacaciones, revisión de los neumáticos, frenos).

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 permitan gestionar las diferentes ofertas y promociones que puede ofrecer el taller: puesta a punto del vehículo antes de los periodos vacacionales principales (Semana Santa, verano, Navidad), cambio de neumáticos, cambio de filtro y aceite, etc. Justifiquen las opciones y decisiones que se tomen.
- c) **[2 puntos]** Implementar un método (o métodos) que permita(n) que cada mecánico vea las fichas que le corresponde gestionar y pueda editar los datos dejando constancia del trabajo realizado y el estado de reparación (por ejemplo, pendiente, en proceso, parado [hace falta piezas, pendiente de confirmación del cliente], fase de prueba, terminado).
- d) **[2,0 puntos]** Se quiere utilizar el sistema de gestión del taller para gestionar una empresa de alquiler de vehículos. Indique los cambios que serían necesarios en el diseño y la implementación para permitir esa nueva funcionalidad.