

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: ¿Qué entendemos por *signatura*?

- a. La cabecera de un método.
- b. El tipo de datos que devuelve un método.
- c. El tipo de datos que devuelve un constructor.
- d. Ninguna de las anteriores.

Pregunta 2: Queremos compilar el siguiente código que se puede encontrar en el texto base de la asignatura y que ha sido modificado convenientemente. ¿Cuál es el resultado que obtenemos al compilar?

```
1      public class Prueba
2      {
3          public static void main (String args[])
4          {
5              String cadena1 = new String("ejemPLO");
6              String cadena2 = new String("EJEMPL0");
7              cadena1.toUpperCase();
8              if (cadena1.toString().equals(cadena2.toString()))
9              {
10                 System.out.println("Son iguales");
11             }
12             else
13             {
14                 System.out.println("Son diferentes");
15             }
16         }
17     }
```

- a. Se produce una excepción y la ejecución falla.
- b. Se imprime por pantalla el mensaje: Son iguales.
- c. Se imprime por pantalla el mensaje: Son diferentes.
- d. Ninguna de las anteriores.

Pregunta 3: Según el texto de la bibliografía básica de la asignatura, ¿cuál de las siguientes afirmaciones es correcta respecto al texto de una clase?

- a. El envoltorio exterior contiene la cabecera de la clase.
- b. El propósito principal del envoltorio exterior es proporcionar un constructor a la clase.
- c. El envoltorio exterior permite sobrecargar cualquier método de la clase.
- d. Ninguna de las anteriores.

Pregunta 4: Supongamos que reescribimos el ejemplo `BouncingBall` del libro de la forma en que se muestra a continuación:

```
1      public class BouncingBall {
2          static int n = 10;
3          static Integer num = new Integer(n);
4
5          public static void main (String args [])
6          {
7              if (n!=0)
8              {
9                  incrementar (num);
10                 System.out.println("Valor: " + num.intValue());
11             }
12         }
13
14         public static void incrementar (Integer numero)
15         {
16             numero++;
17         }
18     }
```

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

- a. El programa no compila. Se produce un error en la línea 16
- b. El programa no compila. Se produce un error en la línea 10
- c. El programa compila e imprime el resultado "Valor: 11"
- d. Ninguna de las anteriores.

Pregunta 5: Según el texto de la bibliografía básica de la asignatura, ¿qué se entiende por abstracción?

- a. Proceso de dividir un todo en partes bien definidas que puedan construirse y examinarse por separado y que interactúen de formas bien definidas.
- b. Capacidad de ignorar los detalles de las distintas partes, para centrar la atención en un nivel superior de un problema.
- c. Capacidad de dividir un todo en partes bien definidas ignorando los detalles de las distintas partes.
- d. Ninguna de las anteriores.

Pregunta 6: Según el texto de la bibliografía básica de la asignatura, con respecto al constructor de la subclase podemos afirmar que ...

- a. Debe siempre invocar al constructor de su superclase como primera instrucción. Si el código fuente no incluye esa llamada, Java intentará insertar una llamada automáticamente.
- b. No debe invocar nunca al constructor de su superclase como primera instrucción. Si la incluye esa llamada, Java ignorará esta llamada automáticamente.
- c. Debe siempre invocar al constructor de su superclase como última instrucción. Si no incluye esa llamada, Java intentará insertar una llamada automáticamente.
- d. Debe siempre invocar al constructor de su superclase como última instrucción. Si no incluye esa llamada, Java generará un error de compilación.

Pregunta 7: ¿Cómo se llama el entorno de pruebas que soporta la prueba estructurada de unidades y las pruebas de regresión en Java?

- a. JDK.
- b. JBoss.
- c. Javadoc.
- d. JUnit.

Pregunta 8: Sea el siguiente código modificado de la clase `MusicOrganizer` mostrada en el libro base:

```
1    import java.util.*;
2
3    public class MusicOrganizer {
4
5        public static void main (String args []) {
6
7            ArrayList <Double> a = new ArrayList <Double> (5);
8            Integer numero = new Integer(0);
9
10           for (int i=0; i<=5; i++)
11           {
12               numero = Integer(i);
13               a.add(numero);
14           }
15           System.out.println(a.toString());
16       }
17   }
```

¿Qué ocurre cuando se intenta compilar y ejecutar el código?

- a. No compila. Hay que sustituir la línea 12 por la siguiente: `numero = i;` Haciendo esto, el programa compila y proporciona el resultado [0.0, 1.0, 2.0, 3.0, 4.0, 5.0]
- b. No compila. Hay que sustituir la línea 13 por la siguiente: `a.add(new Double(numero.intValue()));` Haciendo esto, el programa compila y proporciona el resultado [0.0, 1.0, 2.0, 3.0, 4.0, 5.0]
- c. No compila. Hay que sustituir la línea 12 por la siguiente: `numero = i;` y la línea 13 por la siguiente: `a.add(new Double(numero.intValue()));` Haciendo esto, el programa compila y proporciona el resultado [0.0, 1.0, 2.0, 3.0, 4.0, 5.0]
- d. Compila y proporciona el resultado [0.0, 1.0, 2.0, 3.0, 4.0, 5.0].

Pregunta 9: Según el texto de la bibliografía básica de la asignatura, el acceso protegido ...

- a. No puede aplicarse a métodos y constructores de una clase, sólo a los campos de una clase.
- b. No puede aplicarse a los campos de una clase, sólo a métodos y constructores de una clase.
- c. Puede aplicarse a cualquier miembro de una clase, aunque suele reservarse a campos y métodos, no constructores.
- d. Puede aplicarse a cualquier miembro de una clase, aunque suele reservarse a constructores y métodos, no campos.

Pregunta 10: Respecto a `File` y `Path`, podemos afirmar ...

- 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 11: Queremos compilar el siguiente código que se puede encontrar en el texto base de la asignatura y que ha sido convenientemente modificado. ¿Qué ocurre cuando lo compilamos?

```
1    import java.awt.*;
2    public class ImageViewer extends Frame{
3        public static void main(String argv[]){
4            ImageViewer MiImageViewer=new ImageViewer();
5        }
6
7        ImageViewer(){
8            Button BotonHola=new Button("Hola");
9            Button BotonAdios=new Button("Adios");
10           add(BotonHola);
11           add(BotonAdios);
12           setSize(300,300);
13           setVisible(true);
14       }
15   }
```

- a. Compila, y muestra dos botones juntos ocupando todo el frame. “Hola” en la izquierda y “Adios” en la derecha.
- b. Compila, y muestra un botón ocupando todo el frame diciendo “Hola”.
- c. Compila, y muestra un botón ocupando todo el frame diciendo “Adios”.
- d. No compila.

Pregunta 12: Según el texto de la bibliografía básica de la asignatura, ¿qué permite el patrón Método Factoría?

- a. Garantiza que solo se cree una instancia de una clase y proporciona un acceso unificado a la misma.
- b. Proporciona una interfaz para crear objetos, pero deja que las subclases decidan qué clase específica se crea.
- c. Trata con el problema de añadir funcionalidad a un objeto existente.
- d. Define una relación uno-a-muchos, de modo que cuando un objeto cambie su estado, muchos otros pueden ser notificados.

Pregunta 13: Se quiero compilar y ejecutar el siguiente código obtenido del manual de referencia y que ha sido oportunamente modificado. ¿Cuál es el resultado que obtenemos?

```
1    public abstract class Simulator {
2        public static void main(String argv[]){
3            System.out.println(mensajePantalla());
4        }
5
6        public static String mensajePantalla(){
7            return ("Hola");
8        }
9    }
```

- a. El problema compila y se ejecuta correctamente, mostrando por pantalla el mensaje “Hola”.
- b. Se produce un error de compilación en la línea 1.
- c. Se produce un error de compilación en la línea 2.
- d. Se produce un error de compilación en la línea 3.

Pregunta 14: ¿Qué suceso se genera cuando se hace clic en un botón o se mueve el ratón?

- a. Se genera un suceso `ActionEvent` en ambos casos.
- b. Se genera un suceso `ActionPerform` en ambos casos.
- c. Se genera un suceso `ActionListener` en ambos casos.
- d. Ninguna de las anteriores.

RESERVA 1: ¿Cuál de las siguientes es una característica de la clase `java.lang.Error`?

- a. `extends Exception`.
- b. `implements Throwable`.
- c. `implements Exception`.
- d. Ninguna de las anteriores.

RESERVA 2: La siguiente operación: `System.out.println (4 | 3)`; ¿qué imprime por pantalla?

- a. 0
- b. 1
- c. 6
- d. 7

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 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) **[2,5 puntos]** Implementar un método (o métodos) que permitan generar los listados de vehículos que se han reparado en un día determinado, indicando, para cada uno de los vehículos, los mecánicos que han intervenido, las piezas que se han utilizado y una descripción de cada uno de los trabajos realizados. El listado aparecerá ordenado ascendentemente por la matrícula del vehículo (en el listado aparecen antes aquellos que tengan una matrícula más antigua). Justifique las opciones y decisiones que se tomen.
- c) **[2,0 puntos]** Implementar un método (o métodos) que implementen la gestión del servicio integral de revisión técnica (ITV) de los vehículos: el cliente puede dejar su coche, el taller lleva a cabo una revisión, lleva el coche para la ITV y realiza las reparaciones adicionales necesarias. Justifique las opciones y decisiones que se tomen.
- d) **[1,0 puntos]** Proporcione un método (o métodos) que muestre por pantalla un formulario básico en modo gráfico que recoja los parámetros necesarios para dar de alta un nuevo vehículo en el sistema. Se pide expresamente la parte gráfica (asumiendo que la lógica de funcionamiento está desarrollada). El objetivo es ver el conocimiento y destreza en el uso de las librerías Swing y/o AWT. No desarrolle código asociado a la funcionalidad del alta. Justifique las opciones y decisiones que se tomen.