

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.

Pregunta 1: Indica cual de las siguientes afirmaciones es correcta:

- a. Los métodos pueden devolver información de algún objeto mediante un valor de retorno.
- b. Los métodos siempre tienen parámetros con los que obtener la información necesaria.
- c. A partir de una clase tan solo se puede crear un solo objeto.
- d. El estado de los objetos se representa mediante los métodos implementados.

Pregunta 2: Dado el siguiente fragmento de código,

```
int estudiante = 1;
boolean[] matriculas = new boolean[3];
boolean aprobado = matriculas [estudiante];
```

Indica cual de las siguientes afirmaciones es correcta en relación al valor de la variable aprobado.

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

Pregunta 3: Indica cual de las siguientes afirmaciones es correcta:

- a. Los campos también son conocidos como variables de estado.
- b. El alcance de una variable define la sección de código desde donde la variable puede ser declarada.
- c. Los constructores permiten que cada objeto sea preparado adecuadamente cuando es creado.
- d. El tiempo de vida de una variable describe el número de veces que es utilizada en un método.

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: Indica cual de las siguientes afirmaciones es correcta:

- a. Los campos se definen dentro de los constructores y de los métodos.
- b. Los campos se usan para almacenar datos que nunca persisten durante la vida del objeto.
- c. Los campos tienen un tiempo de vida que perdura después de terminar el objeto.
- d. La accesibilidad de los campos se extiende a toda clase y por este motivo pueden usarse dentro de cualquier constructor o método de clase en la que estén definidos.

Pregunta 6: Indica cual de las siguientes afirmaciones es correcta:

- a. El lenguaje Java tiene tres variantes del ciclo for : for-each, for y for-do.
- b. Un ciclo while es similar en su estructura y propósito que el ciclo for-each.
- 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 7: Indica cual de las siguientes afirmaciones es correcta:

- a. La prueba es la actividad de descubrir si una pieza de código produce el comportamiento pretendido.
- b. Una aserción es una expresión que establece una condición que esperamos que resulte verdadera.
- c. Un seguimiento es la actividad de trabajar a través de un segmento de código línea por línea, mientras se observan cambios de estado y otros comportamientos de la aplicación.
- d. Todas las respuestas anteriores son correctas.

Pregunta 8: Indica cual de las siguientes afirmaciones es correcta:

- a. Un objeto de tipo String puede ser modificado una vez que está creado, por tanto no es un ejemplo de objeto inmutable
- b. La clase String tiene un método de nombre trim que permite modificar caracteres en cualquier posición de una cadena
- c. Las cadenas de texto de tipo String solamente se pueden comparar mediante el operador “==”
- d. Un objeto es inmutable si su contenido o su estado no puede ser cambiado una vez que se ha creado

Pregunta 9 : Basado en el ejemplo de la Base de Datos de CDs y DVDs visto en la asignatura en el capítulo 8, ¿cuál sería la salida del siguiente código?

```
public class BaseDeDatos {  
  
    public final void metodoAgregarElemento(){  
        System.out.println("Agregar Elemento");  
    }  
}  
  
public class BaseDeDatosDeMusica {  
  
    public static void main(String argv[]){  
        BaseDeDatos db = new BaseDeDatos();  
        db.metodoAgregarElemento();  
    }  
}
```

- a. Error en tiempo de compilación indicando que una clase con métodos finales deben ser declarada también como final.
- b. Error en tiempo de compilación indicando que no se puede heredar de una clase con métodos finales.
- c. Error en tiempo de ejecución indicando que BaseDeDatos no ha sido definida como final.
- d. Éxito en la compilación y salida "Agregar Elemento".

Pregunta 10: Indica cual de las siguientes afirmaciones es correcta:

- a. El término acoplamiento describe cuánto se ajusta una unidad de código a una tarea lógica o a una entidad
- b. El acoplamiento describe la conectividad de los propios objetos de una clase
- c. Un encapsulamiento apropiado en las clases reduce el acoplamiento
- d. Un sistema debilmente acoplado se caracteriza por la imposibilidad de modificar una de sus clases sin tener que realizar cambios en ninguna otra

Pregunta 11: Dado el siguiente fragmento de código que pretende mostrar un ejemplo de sobrescritura:

```
class Examen {  
    private float pregunta = 1.0f ;  
    protected float getNota () {return pregunta;}  
}  
  
class Test extends Examen {  
    private float nota = 2.0f;  
    //Insertar código aquí  
}
```

Indique cual de las siguientes opciones completaría el código anterior para dar lugar a un ejemplo correcto de sobrescritura:

- a. public float getNota (float valor) { return valor;}
- b. public float getNota () { return nota;}
- c. float getNota () { return nota;}
- d. float double getNota () { return nota;}

Pregunta 12: Indica cual de las siguientes afirmaciones es correcta:

- a. Una IGU se construye mediante visores que se ubican en la pantalla.
- b. La distribución de los componentes en la pantalla se lleva a cabo mediante gestores de disposición.
- c. Los componentes se ubican en una ventana agregándolos a la barra de estado o al panel agregador.
- d. Un objeto puede escuchar los eventos de los componentes implementando una interfaz interpretadora de eventos.

Pregunta 13: 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 cual de las siguientes afirmaciones es correcta:

- a. El Bloque4 no se ejecutará si se produce una excepción de tipo aritmético en el Bloque2
- b. El Bloque4 se ejecutará antes de que la excepción producida por un acceso a un objeto nulo (null) en el Bloque2 se propague hacia arriba
- 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 14: Indica cual de las siguientes afirmaciones es correcta:

- a. Una superclase es una clase que es implementada por otra.
- b. Una subclase es una clase que implementa a otro objeto.
- c. La herencia nos permite heredar pero no reutilizar en un nuevo contexto clases que fueron escritas previamente.
- d. Las clases que están vinculadas mediante una relación de herencia forman una jerarquía de herencia.

Pregunta 15: Indica cual de las siguientes afirmaciones es correcta:

- a. La interfaz de una clase describe lo que hace la clase y cómo puede usarse pudiendo mostrar parte de su implementación.
- b. Un mapa es una colección que almacena entradas de ternas de valores llave/valor/posición.
- c. La documentación de una clase debe ser suficientemente detallada como para que otros programadores puedan usar la clase sin necesidad de leer su implementación.
- d. Los modificadores de acceso definen las restricciones de uso de un objeto para determinados métodos, constructores o campos.

PARTE PRÁCTICA [6,5 PUNTOS]:

La práctica del presente curso ha sido una terminal punto de venta (por sus siglas, TPV) que ha servido para estudiar y practicar los mecanismos de la Programación Orientada a Objetos.

Definición de TPV y Características

Según la Wikipedia (www.wikipedia.org), un terminal punto de venta (cuyo acrónimo es TPV hace referencia al dispositivo y tecnologías que ayudan en la tarea de gestión de un establecimiento comercial de venta al público que puede contar con sistemas informáticos especializados mediante una interfaz accesible para los vendedores.

Los TPV permiten la creación e impresión del tique de venta mediante las referencias de productos, realizan diversas operaciones durante todo el proceso de venta, así como cambios en el inventario. También generan diversos reportes que ayudan en la gestión del negocio. Los TPV se componen de una parte hardware (dispositivos físicos) y otra software (sistema operativo y programa de gestión).

En nuestro caso concreto, el hardware será un ordenador tipo PC o similar y nuestro software será una aplicación desarrollada en Java que se ejecutará sobre dicho equipo.

Funcionalidades

Los TPV permiten la implementación desde labores simples de gestión de una venta, hasta operaciones más complejas como es la gestión de almacén o inventario, gestión de facturación o gestión de clientes. En esta práctica, se propondrá diferentes funcionalidades para el sistema de gestión del TPV:

- Llevar un control de diferentes elementos que existen en nuestro establecimiento. Así, los productos habrán de estar identificados en el sistema por, al menos, los siguientes datos: código descriptivo (por ejemplo, el código de barras), descripción, precio unitario sin IVA, IVA aplicable, precio unitario con IVA, cantidad disponible en stock.
- El sistema debe permitir dar de alta nuevos productos, dar de baja productos existentes así como modificar los datos del mismo.
- Realizar la importación y/o exportación de los productos a/desde ficheros (u otro método similar que el alumno considere en su lugar).
- Llevar un control de las diferentes ventas que se producen. Así, el sistema deberá llevar un control de tickets generados, de modo que cada ticket se considerará una venta. Cada ticket tiene que tener un código de identificador único. Una forma de generar un código único podría ser de la forma AAAAMMDDHHMM, donde AAAA es el año en curso, MM el mes en que se genera la venta, DD el día de la venta, HHMM las horas y minutos en las que se inicia la venta. Asumiremos que sólo hay un TPV, por lo que no procede que haya dos ventas simultáneas.
- La venta consistirá en la inclusión de varios productos en una lista, generándose una línea por cada producto vendido. Cada línea mostrará, al menos, el código del producto, la descripción del producto, la cantidad de unidades vendidas, el precio unitario con IVA, el IVA que se le aplica y el importe total de la venta de ese producto según el número de unidades vendidas.
- El proceso de venta implicará automáticamente un proceso de actualización del inventario. De este modo, si se introduce un código que no pertenece a ningún producto, o si se introduce un producto que no existe en stock (o más unidades de las existentes), el programa deberá mostrar los errores correspondientes.
- El sistema deberá permitir también introducir un producto a vender en el ticket haciendo una búsqueda por la descripción, además de con el código que lo identifica.
- Realizar la importación y/o exportación de los diferentes tickets de ventas a/desde ficheros (u otro método similar que el alumno considere en su lugar).
- Llevar un control de los diferentes clientes que trabajan con el establecimiento comercial. Así, los

clientes habrán de estar identificados en el sistema por, al menos, los siguientes datos: código identificativo del cliente, NIF o CIF, nombre y apellidos / razón social, domicilio, fecha de alta en el sistema.

- El sistema debe permitir dar de alta nuevos clientes, dar de baja clientes existentes así como modificar los datos de los mismos.
 - Realizar la importación y/o exportación de los clientes a/desde ficheros (u otro método similar que el alumno considere en su lugar).
 - Permitir generar facturas a partir de un conjunto de tickets. Puede generar facturas agrupando diferentes tickets siempre y cuando pertenezcan al mismo cliente y se han realizado dentro del mismo periodo fiscal (es decir, dentro del mismo año). La información que irá en cada factura deberá ser, al menos, la siguiente: número de la factura (identificador único), CIF del vendedor, razón social del vendedor, fecha de emisión de la factura, datos del cliente (los indicados con anterioridad, excepto la fecha de alta en el sistema), listado de los diferentes productos vendidos (especificando para cada producto, el ticket en el que se encuentra, su cantidad vendida e importe total) así como suma del total de la venta (valor total de la factura).
 - Realizar la importación y/o exportación de las facturas a/desde ficheros (u otro método similar que el alumno considere en su lugar).
 - Generación de listados: se deberá implementar, al menos, la emisión de tres listados, a saber: ventas realizadas en un intervalo de tiempo determinado agrupadas estas ventas por clientes, ventas realizadas en un intervalo de tiempo determinado a un cliente y ranking de productos más vendidos en un intervalo de tiempo determinado.
- 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) **[1,5 puntos]** Implementa la funcionalidad que permita dar de alta nuevos clientes, dar de baja clientes existentes así como modificar los datos de los mismos. Justifique las opciones y decisiones que se tomen.
- c) **[1,5 puntos]** Implementa la funcionalidad que permita la generación de listados: se deberá implementar, al menos, la emisión de tres listados, a saber: ventas realizadas en un intervalo de tiempo determinado agrupadas estas ventas por clientes, ventas realizadas en un intervalo de tiempo determinado a un cliente y ranking de productos más vendidos en un intervalo de tiempo determinado. Justifique las opciones y decisiones que se tomen.
- d) **[2,5 puntos]** Para la siguiente versión del software se desea añadir la figura del proveedor. De cada proveedor se debe tener un listado de los productos que sirve, así como su precio, que podría actualizarse manualmente después de la última venta. El sistema debería consultar el inventario después de cada venta y por debajo de un mínimo de productos realizar un pedido al proveedor más barato, de manera automática. El número de productos pedido, podrá ser configurable para que el vendedor establezca un valor por cada producto. ¿Qué cambios serían necesarios en el diseño para adaptar esta nueva funcionalidad? Implemente el método (o métodos) que permita esta nueva funcionalidad.