

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: ¿Qué significa el siguiente fragmento de código Java?:

```
int uno() { return 1; }
```

- a. Hay un método “int uno” que no recibe ningún parámetro de entrada y devuelve el valor 1.
- b. Hay una variable “int” cuyo valor es “uno() { return 1; }”
- c. Hay un método “uno” que no recibe ningún parámetro de entrada y devuelve un entero cuyo valor es 1.
- d. El fragmento no representa un fragmento de código legal en Java.

Pregunta 2: Para lograr que una clase entre en el depurador en BlueJ a hacer una instancia en BlueJ, ¿qué hay que hacer con el código fuente?:

- a. Compilarlo de nuevo con la opción Debug activado.
- b. Meter un punto de ruptura.
- c. Lanzar directamente el depurador.
- d. Se hace automáticamente al encontrar un error en el código.

Pregunta 3: Dado el siguiente fragmento de código en Java:

```
int m, n;  
public void f() {  
    m = (m + 2) % n;  
    System.out.print(m + " ");  
}  
public void g() {  
    int i = 0; m = 0; n = 8;  
    while(i++<n) {  
        f();  
    }  
}
```

¿Cuál es el resultado de ejecutar g()?

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

Pregunta 4: Según el libro de la asignatura, ¿cómo se llama el código fuente de una clase?:

- a. La implementación de la clase.
- b. La interfaz de la clase.
- c. Los métodos de la clase.
- d. La visibilidad de la clase.

Pregunta 5: Para captar el evento de dar en un botón en Java, ¿qué interfaz hay que implementar?:

- a. `public class DemoBoton extends JPanel implements WindowListener`
- b. `public class DemoBoton extends JPanel implements EventListener`
- c. `public class DemoBoton extends JPanel implements ButtonListener`
- d. `public class DemoBoton extends JPanel implements ActionListener`

Pregunta 6: Según el libro de la asignatura, la duplicación de código es un síntoma de:

- a. Buena cohesión.
- b. Mala cohesión.
- c. La solución inevitable de un problema complejo.
- d. Mal encapsulamiento.

Pregunta 7: Según el libro de la asignatura, ¿qué significa “prueba de unidades”?:

- a. Una prueba completa de la aplicación.
- b. Una prueba sistemática de un método en concreto.
- c. Una prueba de las partes individuales de la aplicación.
- d. Una depuración completa de la aplicación.

Pregunta 8: Dado el siguiente fragmento de código de Java:

```
class I {}
class J extends I {}
class K extends J {}
public void hh () {
    J j = new J();
    boolean b1 = j instanceof K;
    boolean b2 = j instanceof J;
    boolean b3 = j instanceof I;
    boolean b4 = j instanceof Object;
    System.out.println(b1 + " " + b2 + " " + b3 + " " + b4);
}
```

¿Cuál sería la salida del método hh?

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

Pregunta 9: Un método abstracto se declara de la siguiente forma:

- a. `abstract String ss(){};`
- b. `abstract String ss();`
- c. `abstract String ss(){return void};`
- d. `abstract ss();`

Pregunta 10: Si existe un `ArrayList<String> as`, ¿cómo se consigue un iterador sobre `as`?

- a. `Iterator it = as.getIterator();`
- b. `Iterator it = as.iterator();`
- c. `Iterator it = new Iterator(as);`
- d. `Iterator it = as.nextIterator();`

Pregunta 11: Si las clases J y K heredan de la clase I y la clase L hereda de la clase J, entonces (indica la respuesta que daría un error de compilación):

- a. I i = new K();
- b. J j = new K();
- c. J j = new L();
- d. K k = new I();

Pregunta 12: En la práctica hay que leer información desde un archivo. ¿Cómo se puede generar una excepción si no se encuentra el archivo?:

- a. throw new FileNotFoundException("Archivo no encontrado");
- b. throw new NoFileException("Archivo no encontrado");
- c. throw new FileException("Archivo no encontrado");
- d. throw new NullPointerException("Archivo no encontrado");

Pregunta 13: Dado el siguiente fragmento de código en Java:

```
String s1 = 0 + "5"; // (1)
String s2 = "0" + 5; // (2)
String s3 = 0 + 5 + ""; // (3)
String s4 = java.lang.Integer.toString(0) + 5; // (4)
```

¿Cuáles son formas **válidas** de construir una cadena?:

- a. 1, 2 y 4
- b. 1, 2 y 3
- c. 4
- d. Todas las formas son válidas.

Pregunta 14: Según el libro de la asignatura, las instrucciones condicionales llevan a cabo (con la excepción del switch):

- a. una de dos acciones posibles.
- b. una de varias acciones posibles.
- c. las dos acciones posibles.
- d. dos o más de las acciones posibles.

Pregunta 15: ¿Cuál es la diferencia entre declarar una variable de clase **private** y **protected**?:

- a. private: Acceso solo dentro de la clase, protected: Acceso desde la clase y sus hijos.
- b. private: Acceso desde la clase y sus hijos, protected: Acceso solo dentro de la clase.
- c. private: Acceso solo dentro del paquete, protected: Acceso desde la clase y sus hijos.
- d. Se pueden aplicar private y protected solamente a la declaración de clases y no a variables.

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 tal y como se ha definido en el Nivel 2. 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) **[2,0 puntos]** Implementa la funcionalidad de 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).
- c) **[1,0 punto]** Implementa un método que dé una notificación cuando se venda el último ejemplar de un producto en el almacén para que se pueda pedir más.
- d) **[2,5 puntos]** Para la siguiente versión del software se quiere incluir un catálogo de los productos para que un cliente pueda mirarlo en pantallas dentro de la tienda y comprar los productos directamente desde el catálogo. Se trataría de una nueva interfaz sobre el TPV que proporcionaría algunas de las funciones (aquellas relevantes para el cliente) directamente al cliente. ¿Qué cambios serían necesarios en el diseño para incluir esta nueva funcionalidad? Implementa los cambios necesarios que permitan esta nueva funcionalidad.