

TPV

Práctica de la Asignatura de Programación Orientada a Objetos
Escenario para el Curso 2014/2015 – Febrero de 2014 – Versión 1.1

Departamento de Lenguajes y Sistemas Informáticos
Escuela Técnica Superior de Ingeniería Informática - UNED



1.- Introducción

Los objetivos que se plantean en la realización de esta práctica son los siguientes:

- Familiarización con la Programación Orientada a Objetos (POO): definición de clases e instancias, uso de la herencia, definición/uso de métodos estáticos y abstractos,
- Realización del diseño orientado a objetos de un problema.
- Implementación de un programa sencillo donde se manejen conceptos relacionados con POO.

La práctica se va a implementar en Java 2 Estándar Edition (J2SE). El compilador de Java que se usará será BlueJ, tal y como se define en el programa de la asignatura. Información y material adicional referente a la práctica y su desarrollo puede encontrarse en el Campus Virtual de la asignatura.

2.- Programación Orientada a Objetos en Java

El paradigma de programación orientada a objetos define un programa como una colección de entidades que se relacionan para resolver un problema. Estas entidades, que se conocen genéricamente como objetos, están tienen por un conjunto de propiedades y métodos, y están organizadas en torno a una jerarquía de clases.

En Java cada objeto puede tener variables y métodos privados y públicos. Se puede modificar dicha visibilidad de una clase usando los modificadores de acceso a miembros. Las dos maneras más habituales de especificar la accesibilidad son:

private – la variable o método está disponible solamente para esta clase,

public – la variable o método está disponible para todas las clases,

Una clase puede heredar los variables y métodos públicos de otra clase a través del mecanismo de herencia y la palabra clave extends. Por ejemplo:

```
//clase base que va a contener información sobre vehículos de nuestra empresa:
public vehiculo {
    private int noPuertas;
    private int noRuedas;
    private String modelo;
    public vehiculo(){}
    public void setNoPuertas(int np) {
        noPuertas = np;
    }
    //etc.
}
//una clase para tratar a los coches en general...
public coche extends vehiculo {
    private
    private boolean airbags;
```

```

    public coche(){}
    public void setAirbags(Boolean a) {
        airbags = a;
    }
    //etc.
}
//y, por fin, una clase para tratar a los coches deportivos
public final cocheDeportivo extends vehículo {
    private String capacidadMotor;
    private int maxVelocidad;
    public cocheDeportivo(){}
    public void setCapacidadMotor(String cm) {
        capacidadMotor = cm;
    }
    //etc.
    //se puede llamar a cualquier método en las superclases como
    //si estuvieran dentro
    //de esta misma clase, p.ej.:
    setNoPuertas(2);
}

```

Notas: Las clases que extienden otras clases tienen el nombre de subclases y las clases que son extendidas por otras clases tienen el nombre de superclases.

Hay que tener cuidado a la hora de planificar las relaciones de herencia entre clases en Java porque una clase solamente puede heredar variables y métodos de otra (y sus superclases). Es decir, que no hay herencia múltiple en Java como hay en lenguajes como C++ (aunque se puede reproducir la técnica de herencia múltiple usando interfaces...). De todas formas, la manera más habitual para tratar está tema es simplemente usar una clase dentro de otra, por ejemplo, si hay una clase para el aparcamiento de una empresa que ya es una extensión de una clase base aparcamiento, dicha clase no puede heredar ninguna otra clase, por lo tanto, se incluirán las clases de coches, camiones, motos, etc., así:

```

public aparcamientoEmpresa extends aparcamiento {
    private String nombreEmpresa;
    private cocheDirector = new cocheDeportivo(...);
    public aparcamientoEmpresa(){}
    //etc.
    //para llamar a algún método en una clase hay que especificar
    //la variable de la instancia...
    cocheDirector.setCapacidadMotor("4.5l");
}

```

3.- Descripción de la Práctica

La práctica del presente curso va a ser diseñar e implementar una terminal punto de venta (por sus siglas, TPV). Esto servirá para estudiar y practicar los mecanismos de la Programación Orientada a Objetos. Se plantearán diferentes niveles de acabado, permitiéndose así que el alumno pueda optar de antemano a diferentes calificaciones conociendo previamente los requisitos a cumplir para cada una de esas calificaciones.

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 en función de la calificación a la que se aspire. De este modo, una mayor complejidad a desarrollar implicará una calificación mayor en la evaluación de la práctica.

Es importante considerar que para optar a la calificación de un nivel superior han de cumplirse todas y cada una de las funcionalidades especificadas en el nivel inmediatamente anterior. Caso de no ser así (no cumplir con todas los requerimientos de un nivel) no se podrá obtener una calificación superior a la marcada por el nivel cuyas restricciones no se cumplen en su totalidad. Del mismo modo, los niveles han de implementarse en el orden que se indican, no siendo posible implementar niveles no consecutivos para obtener calificaciones superiores..

Para cada uno de los niveles se van a indicar unos requisitos mínimos de cumplimiento. Esto quiere decir que para cualquier otro detalle de diseño que no se encuentre descrito expresamente en lo indicado en este enunciado, el alumno tiene libertad para tomar cuantas decisiones considere oportunas.

Nivel 1 - Puntuación Total Máxima a Obtener: 1 punto.

Lo que se pretende que el alumno desarrolle en este nivel son las relaciones de clase, herencia y demás que van asociadas al desarrollo de la práctica. Así, se pide realizar las siguientes tareas:

- Planteamiento del Problema: actores participantes, relaciones entre actores, funcionalidad a cumplir por la práctica a desarrollar.
- Establecimiento de diferentes clases a intervenir en la práctica, relaciones de dependencia entre clases, identificar diferentes jerarquías de clases, etc.
- Elaboración de un documento escrito (memoria de la práctica) que contenga el primer punto y los correspondientes ficheros para BlueJ que implementen lo segundo.

Nivel 2 - Puntuación Total Máxima a Obtener: 3 puntos.

Los alumnos que implementen este nivel de finalización de la práctica podrán sumar a la calificación que obtuvieron por realizar el Nivel 1 (que fue de 1 punto) hasta 2 puntos adicionales, con lo que la implementación del Nivel 2 acarreará la consecución de una

puntuación máxima de 3 puntos. Sólo se podrá optar a este Nivel si se ha implementado satisfactoriamente y en su totalidad los requerimientos especificados en el Nivel 1.

Lo que se pretende que el alumno desarrolle en este nivel es la parte de **gestión de inventario** del establecimiento comercial. De este modo, el sistema deberá permitir lo siguiente:

- 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).
- Modificación del documento escrito desarrollado en el Nivel 1 (memoria de la práctica) y los correspondientes ficheros para Blue] de modo que se implemente esta funcionalidad y quede reflejada dicha modificación en la memoria correspondiente. Del mismo modo, la memoria contendrá un Manual de Usuario simple que muestre la forma de manejar la aplicación con las funcionalidades añadidas en este nivel. La implementación puede ser realizada en modo textual (sin usar librerías gráficas).

Nivel 3 - Puntuación Total Máxima a Obtener: 5 puntos.

Los alumnos que implementen este nivel de finalización de la práctica podrán sumar a la calificación que obtuvieron por realizar el Nivel 2 (que fue de 3 puntos) hasta 2 puntos adicionales, con lo que la implementación del Nivel 3 acarreará la consecución de una puntuación máxima de 5 puntos. Sólo se podrá optar a este Nivel si se ha implementado satisfactoriamente y en su totalidad los requerimientos especificados en el Nivel 2.

Lo que se pretende que el alumno desarrolle en este nivel es la parte de **gestión de la venta** del establecimiento comercial. De este modo, el sistema deberá permitir lo siguiente:

- 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).
- Modificación del documento escrito desarrollado en el Nivel 2 (memoria de la práctica) y los correspondientes ficheros para Blue] de modo que se implemente esta funcionalidad y quede reflejada dicha modificación en la memoria correspondiente. Del mismo modo, la memoria contendrá un Manual de Usuario simple que muestre la forma de manejar la aplicación con las funcionalidades añadidas en este nivel. La implementación puede ser realizada en modo textual (sin usar librerías gráficas).

Nivel 4 - Puntuación Total Máxima a Obtener: 7 puntos.

Los alumnos que implementen este nivel de finalización de la práctica podrán sumar a la calificación que obtuvieron por realizar el Nivel 3 (que fue de 5 puntos) hasta 2 puntos adicionales, con lo que la implementación del Nivel 4 acarreará la consecución de una puntuación máxima de 7 puntos. Sólo se podrá optar a este Nivel si se ha implementado satisfactoriamente y en su totalidad los requerimientos especificados en el Nivel 3.

Lo que se pretende que el alumno desarrolle en este nivel es la parte de **facturación y clientes** del establecimiento comercial. De este modo, el sistema deberá permitir lo siguiente:

- 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.
- Modificación del documento escrito desarrollado en el Nivel 3 (memoria de la práctica) y los correspondientes ficheros para BlueJ de modo que se implemente esta funcionalidad y quede reflejada dicha modificación en la memoria correspondiente. Del mismo modo, la memoria contendrá un Manual de Usuario simple que muestre la forma de manejar la aplicación con las funcionalidades añadidas en este nivel. La implementación puede ser realizada en modo textual (sin usar librerías gráficas).

Nivel 5 - Puntuación Total Máxima a Obtener: 10 puntos.

Los alumnos que implementen este nivel de finalización de la práctica podrán sumar a la calificación que obtuvieron por realizar el Nivel 4 (que fue de 7 puntos) hasta 3 puntos adicionales, con lo que la implementación del Nivel 5 acarreará la consecución de una puntuación máxima de 10 puntos. Sólo se podrá optar a este Nivel si se ha implementado satisfactoriamente y en su totalidad los requerimientos especificados en el Nivel 4.

Lo que se pretende que el alumno desarrolle en este nivel es llevar a cabo la implementación en un entorno gráfico de todos los niveles planteados en la práctica (Niveles 1 a 4). De este modo, habrá de implementarse de manera gráfica tanto la gestión de inventario/stocks, como la gestión de venta, gestión de usuarios y listados. Los requerimientos de funcionamiento en el modo gráfico serán exactamente los mismos que los que se expusieron en cada uno de los niveles. Se deja al alumno libertad completa para decidir el estilo y diseño gráfico que puede desarrollar en este punto.

Del mismo modo, para la superación de este nivel se exigirá la modificación del documento escrito desarrollado en el Nivel 4 (memoria de la práctica) y los correspondientes ficheros para BlueJ de modo que se implemente esta funcionalidad y quede reflejada dicha modificación en la memoria correspondiente. Del mismo modo, la memoria contendrá un Manual de Usuario simple que muestre la forma de manejar la aplicación con las funcionalidades añadidas en este nivel.

4.- Plan de Trabajo

Para realizar la práctica se seguirá el siguiente método de trabajo:

- En primer lugar se leerá detenidamente el enunciado de esta práctica
- A continuación hay que diseñar, utilizando un paradigma orientado a objetos, los elementos necesarios para la aplicación explicada en el apartado anterior. 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. Del mismo modo, se hace constar que este diseño vendrá marcado, en parte, por el nivel de desarrollo de la práctica que el alumno quiera llevar a cabo.
- El código estará debidamente comentado.
- La clase principal que abre la aplicación deberá llamarse "tpv.class".
- A modo de indicación, se muestra una posible planificación temporal para desarrollar la práctica:

- Semana 1 / 2 / 3: Tareas iniciales de desarrollo de la práctica. Planteamiento inicial de la práctica. Identificación de las clases.
- Semana 4 / 5 / 6: Establecer el conjunto de atributos y métodos asociados a cada una de las clases identificadas y definidas anteriormente. Establecimiento de los diferentes constructores. Desarrollo de los principales métodos. Inicio de labores de codificación.
- Semana 7 / 8: Refinamiento de clases. Establecimiento de estructuras de almacenamiento auxiliares necesarias. Labores de codificación de métodos.
- Semana 9 / 10: Documentación del código generado. Establecimiento del plan de pruebas de la práctica.
- Semana 11 / 12: Verificación y validación de la práctica. Establecimiento de estrategias de control de errores y excepciones. Mantenimiento de código y eliminación de errores detectados en la fase de verificación y validación.
- Semana 13: Últimas pruebas. Generación de documentos finales asociados a la práctica.

5.- Material a Entregar

Memoria: La memoria constará de los siguientes apartados:

- Portada con título “Práctica de Programación Orientada a Objetos – Junio 2015” y los datos del alumno: Nombre, Apellidos, dirección de correo electrónico y teléfono.
- Análisis de la aplicación realizada, mostrando el funcionamiento del programa, estrategias implementadas, decisiones de diseño establecidas y, en general, toda aquella información que haga referencia a las diferentes decisiones tomadas a lo largo del desarrollo de la práctica, junto a una justificación de dichas decisiones.
- Diagrama de clases, detallando claramente el tipo de relación entre ellas (uso, agregación, herencia, ...).
- Un texto en el que se describa (de manera breve) cada clase/objeto, justificación de su existencia, métodos públicos que contiene y funcionalidad que realizan.
- Anexo con el código fuente de las clases implementadas que se consideren relevantes para el entendimiento del programa.

Archivos: incluyendo todos los ficheros *.java y *.class, así como la memoria en formato electrónico (preferiblemente html o pdf). El soporte estará libre de virus. No se corregirá ni se tendrá en cuenta ninguna práctica que esté infectada por un virus.

Entrega de la Práctica: se entregará necesariamente a través de la plataforma aLF, en el apartado habilitado para dicha función, y en el plazo que el tutor haya establecido necesariamente para sus alumnos. Indicar que una cosa es la fecha máxima en la que los alumnos pueden ser calificados por el tutor de la práctica (que es la fecha que aparece como válida en aLF) y otra diferente es aquella en la que los alumnos han de realizar la entrega de la misma, que forzosamente ha de ser anterior a la indicada en aLF. **Cada alumno debe ponerse en contacto con su tutor para confirmar con él la fecha de entrega.** Si un alumno envía al tutor la práctica fuera del plazo establecido por éste, el tutor puede disponer la no corrección de la misma para la convocatoria correspondiente.

6.- Normas de Realización de la Práctica

1. La realización de la práctica es obligatoria. Sólo se evaluará el examen si la práctica ha sido previamente aprobada.
2. Aunque si bien el desarrollo de aplicaciones Orientadas a Objetos usando el lenguaje de programación Java no requiere el uso concreto de ningún entorno de desarrollo, esta práctica ha de desarrollarse íntegramente empleando el entorno de desarrollo BlueJ, que es el que se muestra en el libro de texto básico de la asignatura.
3. Después de un trabajo inicial de diseño, las prácticas se hacen en los centros asociados supervisadas por el tutor de la asignatura. El trabajo a realizar con la práctica tiene dos partes: el diseño y la implementación. Cada alumno deberá llevar su diseño el día en que se realicen las prácticas para que el tutor lo supervise y el alumno puede realizar la implementación.
4. La práctica es individual. Las prácticas cuyo código coincida total o parcialmente con el de otro alumno serán motivo de suspenso para todos los implicados (copiadores y copiados), no pudiéndose examinar ninguno de ellos en el presente curso académico.
5. Cada tutor establecerá unas fechas para la realización de la práctica. El tutor remitirá, antes del 1 de Junio, a la asignatura de Programación Orientada a Objetos un informe de calificación de cada alumno.
6. No habrá sesión extraordinaria de prácticas ya que la asignatura ya debe estar implantada en todos los centros asociados. En caso de que algún alumno no tuviera tutor, deberá dirigirse a cualquier otro centro asociado donde se imparta la asignatura.
7. El equipo docente tendrá en cuenta prácticas con notas altas para aquellos alumnos cuyo examen esté cercano al aprobado.
8. El alumno debería dirigirse a su tutor para cualquier duda que tenga sobre su práctica y solamente al equipo docente (por correo electrónico) en el caso de que su tutor no pueda resolver su problema. En este caso pedimos al alumno que, además de sus datos personales, nos envíe el nombre del centro asociado en el que está matriculado y el de su tutor.
9. Evidentemente se puede usar los foros para realizar consultas a los compañeros pero nunca para intercambiar código.

7.- Centros Asociados vs. Prácticas en Asignaturas Obligatorias

Las prácticas son esenciales en las titulaciones de Informática porque, entre otras cosas, permiten a los alumnos adquirir conocimientos importantes sobre los aspectos más aplicados de ciertas asignaturas, lo cual resulta de gran relevancia e interés a la hora de acceder a un puesto laboral relacionado con la Informática. Para orientar y ayudar a los alumnos, así como para comprobar que realmente un alumno ha realizado su práctica de forma satisfactoria, ésta se debe realizar en un Centro Asociado bajo la supervisión de un tutor, quien decide, en última

instancia, la forma en la cual se organiza el desarrollo de la misma en su Centro Asociado (existencia o no de sesiones presenciales obligatorias, forma de entrega, etc.)

De vez en cuando sucede que un alumno se pone en contacto con un Equipo Docente del Departamento de Lenguajes y Sistemas Informáticos (L.S.I.) porque se ha matriculado en una asignatura obligatoria en un Centro Asociado que no le proporciona un tutor para supervisar la práctica, aún cuando se le ha permitido matricularse. El alumno busca en el Equipo Docente que se le proporcione una solución a este problema, como por ejemplo, la posibilidad de asistir a unas sesiones extraordinarias de prácticas en la Sede Central de la U.N.E.D. en Madrid o la posibilidad de realizar la práctica por su cuenta en casa, enviándola a continuación al Equipo Docente para su corrección. Sin embargo, los Equipos Docentes de L.S.I. no disponen de recursos para poder llevar a cabo ninguna de estas dos alternativas.

Un Centro Asociado que ha permitido a un alumno matricularse en una asignatura obligatoria de una carrera de Informática debería ayudarle a encontrar una solución al problema de la realización de las prácticas. Si se trata de una asignatura donde no se han matriculado muchos alumnos, quizás el centro no cuente con recursos para proporcionar un tutor específicamente para la asignatura. Si hay otro Centro Asociado cerca que dispone de tutor, quizás el alumno pueda realizar la práctica allí. Pero si no es así, el Centro Asociado debería proporcionar un tutor para supervisar y corregir las prácticas de sus alumnos. Lo más razonable sería que fuera un tutor de otra asignatura de Informática en el mismo Centro el que hiciera la sesión de prácticas para los alumnos de la asignatura en cuestión, y al final de la sesión evaluara los trabajos de los alumnos, según las pautas marcadas por el Equipo Docente, haciendo llegar a éste las calificaciones otorgadas.

Por lo tanto, un alumno que tras haberse matriculado en una asignatura obligatoria en un Centro Asociado, se encuentre con que el centro no tiene tutor para dicha asignatura, debería dirigirse al Director del Centro Asociado, para solicitar de él una solución, tal como se ha presentado aquí, es decir, alguien que pueda supervisar y corregir su práctica con plenas garantías. En el caso de que el Director no le proporcione una solución, el alumno debería comunicárselo, por escrito, lo antes posible, al Director del Departamento de L.S.I., Dr. D. Julio Gonzalo.