

Trabajo Práctico Integrador: Programación Avanzada **Curso 2023**

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 lenguaje de programación Java. El compilador de Java que se usará será IntelliJ IDEA. 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 vehiculo {  
    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

Se solicita diseñar e implementar un modelo de negocio a una terminal punto de venta, permitiendo poder cambiar el modelo por el grupo previo autorización del profesor. Esto servirá para estudiar y practicar los mecanismos de la Programación Orientada a Objetos. Se plantearán diferentes niveles de entrega, 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.

Primera Entrega

Lo que se pretende que el alumno desarrolle en esta primera entrega 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 que implementen lo segundo.
- Fecha: 08 de mayo.

Segunda Entrega

A definir.

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.

5.- Material a Entregar

Memoria: La memoria constará de los siguientes apartados:

- Carátula con título y los datos de los alumnos: Nombres, Apellidos.
- 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.

6.- Normas de Realización de la Práctica

1. La realización de la práctica es obligatoria.
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 IntelliJ IDEA.
3. El trabajo a realizar con la práctica tiene dos partes: el diseño y la implementación.
4. La práctica es grupal, pero tendrán que demostrar el trabajo individual aportado en el mismo. Las prácticas cuyo código coincida total o parcialmente con el de otro grupo serán motivo de supervisión.
5. Evidentemente se puede usar los foros para realizar consultas a los compañeros o profesor.