

Prácticas de Programación Orientada a Objetos.

Objetivo:

Desarrollar los conceptos teóricos del paradigma de programación orientada a objetos que han sido impartidos en clase. Reconocer y utilizar las buenas prácticas en la programación orientada a objetos. Conocer el lenguaje JAVA.

Enunciado:

La dirección de unos grandes almacenes, almacenes “*El Chollo*”, le ha solicitado la realización de una aplicación capaz de gestionar sus productos y sus recursos humanos.

Los almacenes El Chollo venden diferentes tipos de productos: de alimentación, tecnología y ropa. Se requiere que todos los productos tengan un precio, una descripción y un identificador único autoincrementado. Además, en el caso de los productos de alimentación será necesario conocer la fecha de caducidad, mientras que para los de tecnología se necesita registrar la duración del periodo de garantía, y para la ropa, la talla y color.

En los almacenes se considera también un tipo especial de producto llamado Cesta que está compuesto por distintos tipos de productos. Por ejemplo: CestaNavidad contiene un jamón y un delantal. El precio de la cesta será la suma del precio de todos los productos que lo componen.

Los clientes de los grandes almacenes están registrados para poder obtener beneficios, por tanto, de todos los clientes se conoce su nombre, DNI y número de tarjeta. Los clientes cuando se dan de alta tienen la opción de hacerse clientes VIP. Los clientes VIP deben facilitar su email y por ello reciben un descuento del 5% en todas sus compras.

De los empleados también se conocen sus datos personales (nombre y DNI) y otros datos de interés como su dirección, teléfono y sueldo.

Todo cliente VIP tiene asociado un empleado que será el que le atiende siempre de forma personalizada. Cuando un nuevo cliente VIP se da de alta en el sistema, se le asigna el empleado con menor número de clientes VIP asignados. Por cada cliente asignado el empleado recibe un incremento de 20€ en su sueldo. De esta forma, el sueldo de los empleados se calcula como: (número de horas mensuales * precio hora) + (número de clientes * 20€).

La aplicación para la gestión del almacén será programada en el lenguaje Java [1] utilizando el entorno de desarrollo Eclipse. La práctica será realizada en grupos de dos alumnos y será desarrollada en tres entregas. Tras cada entrega puede ser realizada una revisión y al finalizar todas las entregas se hará una entrevista para defender el trabajo de prácticas.

Entrega de prácticas:

Entrega febrero-junio:

La práctica se desarrollará en distintas fases, que repercutirán eventualmente en varios entregables. Los plazos de entrega de las prácticas son los siguientes (mirar las entregas para ver requisitos y forma de entrega):

- **Entrega 1: 23 de noviembre de 2014.**
- **Entrega 2: 22 de diciembre de 2014.**
- **Entrega 3 (final): 26 de enero de 2015.**

NOTA: Quien no realice alguna de las tres entregas propuestas, podrá entregar el proyecto final el **26 de enero de 2015**. Este proyecto final consiste en desarrollar **todas** las entregas planteadas. La evaluación del proyecto final será sobre el 100% de la nota sin contar la evaluación continua anterior.

Entrega 1. Análisis y Diseño:

El alumno deberá describir los requisitos básicos de la aplicación, describiendo las posibles clases que intervienen en la aplicación, y la relación entre dichas clases. Para el diseño, el alumno creará un diagrama de clases en lenguaje UML que modele las interacciones básicas entre las clases del dominio del problema. Para la elaboración de dichos diagramas se puede opcionalmente utilizar herramientas de modelado como Visual Paradigm [2] o Argouml [3].

Objetivos:

- Introducir al alumno en el ciclo de desarrollo del software.
- Conocer los principios básicos de la extracción de requisitos.
- Conocer y aplicar correctamente el concepto de clase.
- Establecer relaciones entre clases mediante relaciones de clientela.
- Utilizar el concepto de herencia como mecanismo de reutilización de código donde se ha de cumplir la relación “es-un” entre las clases.
- Aplicar los factores internos de calidad del software con el objetivo de conseguir los externos.

Fecha de entrega: 23 de noviembre de 2014.

Valoración de la práctica: 20% de la nota final.

Instrucciones de la Entrega 1

¿Qué debo entregar? → Un documento pdf nombrado con el DNI que incluya el análisis, diseño e implementación de una estructura de clases correcta identificando roles del sistema y usando diagrama de clases UML

¿Cómo debo entregarlo? → En el campus virtual, siguiendo las especificaciones propuestas en el documento “Formato documentación practicas POO” que se encuentra en el campus virtual.

¿Dónde debo entregarlo? → Campus virtual, tarea “**Práctica. Entrega1**”.

Evaluación de la Entrega 1:

- ✓ La estructura de clases es correcta y cumple con los principios de la programación orientada a objetos (80% de la nota de la Entrega 1).
- ✓ Existe una explicación fundamentada para la estructura de clases diseñada (10 % de la nota de la Entrega 1).
- ✓ La memoria está bien redactada y estructurada (10 % de la nota de la Entrega 1).

Entrega 2. Fase de implementación del modelo:

El alumno deberá implementar las clases diseñadas en el análisis y diseño siguiendo los objetivos de esta entrega, e incluyendo la siguiente funcionalidad mínima. Además el alumno podrá incluir sus aportaciones a la aplicación.

Funcionalidad mínima exigida en Junio:

Para superar la práctica con un 5 deberá estar implementada correctamente la siguiente funcionalidad:

1. Añadir, modificar, eliminar y consultar empleados.
2. Añadir, modificar, eliminar y consultar clientes y clientes VIP.
3. Añadir, modificar, eliminar y consultar productos.
4. Buscar clientes y empleados por DNI.
5. Mostrar para cada cliente la lista de productos que ha comprado.
6. Guardar/Recuperar los datos de la aplicación en fichero (usando la interfaz Serializable)
7. Datos creados para probar la práctica.

Objetivos:

- Introducir al alumno en el ciclo de desarrollo del software.
- Aplicar el principio de ocultación de la información en la definición de una clase.
- Definir correctamente la visibilidad de las declaraciones.
- Definir adecuadamente métodos de acceso y modificación para los atributos de una clase.
- Entender el concepto de propiedad calculada. Valorar cuando una propiedad conviene representarla con un atributo o realizar un cálculo para su obtención.

- Comprender la utilidad de los constructores como mecanismo de inicialización de los objetos. Asimismo, aplicar reutilización en la definición de constructores.
- Aplicar la sobrecarga en la definición de métodos como facilidad del lenguaje para definir firmas coherentes de métodos y reutilizar la implementación entre métodos sobrecargados.
- Definición y uso de constantes.
- Conocer la diferencia entre atributos o propiedades de instancia y de clase, y aplicarlos correctamente.
- Comprender la semántica de las referencias en Java.
- Establecer relaciones entre clases mediante relaciones de clientela.
- Utilizar el concepto de herencia como mecanismo de reutilización de código donde se ha de cumplir la relación “es-un” entre las clases.
- Entender la necesidad de la redefinición de métodos en la aplicación de la herencia.
- Comprender el concepto de polimorfismo, ligadura dinámica y visibilidad para la herencia.
- Comprender el concepto de igualdad de objetos frente a identidad.
- Definir correctamente la copia de objetos en Java.
- Valorar la utilidad de la representación textual de un objeto y realizar su implementación.
- Documentar correctamente las clases en Javadoc.
- Utilizar el concepto de paquete como mecanismo de organización del código.
- Comprender la importancia de la aplicación de una convención de nombrado en la escritura de código. Aplicar la convención de nombres de Java.

Fecha de entrega: 22 de diciembre de 2014.

Valoración de la práctica: 40% de la nota final.

Instrucciones de Entrega 2

¿Qué debo entregar? → Un zip nombrado con el DNI que incluya: el proyecto Eclipse (cuyo nombre también será el dni), el documento pdf nombrado con el DNI que incluya el análisis, diseño e implementación de la estructura de clases correcta identificando roles del sistema y usando diagrama de clases UML, y la documentación Javadoc del código.

¿Cómo debo entregarlo? → En el campus virtual, siguiendo las especificaciones propuestas en el documento “Formato documentación practicas POO” que se encuentra en el campus virtual.

¿Dónde debo entregarlo? → Campus virtual, tarea **“Práctica. Entrega2”**.

Evaluación de la Entrega 2:

A la hora de evaluar la Entrega 2 de la práctica se tendrán en cuenta varios criterios que determinan el grado en la calificación; estos son:

- ✓ La estructura de clases es correcta y cumple con los principios de la Programación Orientada a Objetos.
- ✓ El código está optimizado.
- ✓ El código está comentado.

Antes de calificar se comprueba que:

- ✓ La memoria esté bien redactada y estructurada.
- ✓ La funcionalidad mínima haya sido implementada.

Entrega 3. Implementación de la interfaz gráfica:

La tercera entrega consistirá en el diseño de la interfaz gráfica necesaria para la realización de la funcionalidad mínima implementada en la segunda entrega. Dicha interfaz gráfica constará de una ventana principal desde la cual se tenga acceso para realizar todos los procesos de gestión indicados.

El alumno tendrá libertad para elegir aquel diseño que considere más apropiado.

Fecha de entrega: 26 de enero de 2015.

Valoración de la práctica: 40% de la nota final.

Instrucciones de entrega

¿Qué debo entregar? → Un zip nombrado con el DNI que incluya: el proyecto Eclipse (cuyo nombre también será el dni), el documento pdf nombrado con el DNI que incluya el análisis, diseño e implementación de la estructura de clases correcta identificando roles del sistema y usando diagrama de clases UML, y la documentación Javadoc del código.

¿Cómo debo entregarlo? → En el campus virtual, siguiendo las especificaciones propuestas en el documento “Formato documentación practicas POO” que se encuentra en el campus virtual.

¿Dónde debo entregarlo? → Campus virtual, tarea **“Práctica. Entrega3”**.

Evaluación de la Entrega 3:

A la hora de evaluar la Entrega 3 de la práctica se tendrán en cuenta varios criterios que determinan el grado en la calificación; estos son:

- ✓ La estructura de clases es correcta y cumple con los principios de la Programación Orientada a Objetos.
- ✓ Existe una explicación fundamentada para la estructura de clases diseñada.
- ✓ Incluye estructuras polimórficas.
- ✓ El código está optimizado.
- ✓ El código está comentado.
- ✓ Existe un correcto control de errores tanto a nivel de excepción en la ejecución como de interfaz de usuario.
- ✓ Se valorará como puntuación extra la realización de la funcionalidad ampliada para septiembre.

Antes de calificar se comprueba que:

- ✓ La memoria esté bien redactada y estructurada.
- ✓ La funcionalidad mínima haya sido implementada

Entrega septiembre:

La entrega de septiembre consiste en desarrollar las tres entregas que se describen a continuación, junto con los requisitos adicionales para esta convocatoria.

¿Qué debo entregar? → Un zip nombrado con el DNI que incluya: el proyecto Eclipse (cuyo nombre también será el DNI) y un documento con la memoria.

¿Cómo debo entregar? → En el campus virtual, siguiendo las especificaciones propuestas en el documento “Formato documentación practicas POO” que se encuentra en el campus virtual.

¿Cuándo debo entregar? → 4 de septiembre de 2015.

¿Dónde? → Campus virtual, tarea “Práctica septiembre”.

Funcionalidad mínima exigida en Septiembre:

Para superar la práctica con un 5 deberá estar implementada correctamente la siguiente funcionalidad:

1. Toda la funcionalidad mínima de convocatorias anteriores.
2. Introducir diferentes criterios de búsqueda para los empleados y clientes (por nombre, DNI, etc.) haciendo uso de interfaces.

Referencias básicas de la práctica:

- [1] P. Deitel y H. Deitel. JAVA cómo programar. 9 edth. Ed. Pearson, 2012.
- [2] Visual Paradigm, <http://www.visual-paradigm.com/> (último acceso 23 de octubre de 2013)
- [3] ArgoUML, <http://argouml.tigris.org/> (último acceso 23 de octubre de 2013)