

Programación con Objetos 2

TP Final: Concesionaria Automotriz

Alejandro Slabkov [alejandroslabkov@gmail.com](mailto:alejandroslabkov@gmail.com)

Martín Serna [martin.eduardo.serna@gmail.com](mailto:martin.eduardo.serna@gmail.com)

**Resolución**

El trabajo práctico fue realizado siguiendo los conceptos vistos en clase y la bibliografía recomendada.

Después de algunas discusiones respecto al enunciado llegamos al diseño que puede apreciarse en el Diagrama UML adjunto.

Donde los participantes del plan de ahorro constituyen una subscripción que contiene todo lo pertinente a su participación en el plan, como por ejemplo los pagos y cupón de adjudicación.

El concesionario por su parte conoce todos sus planes y través de ellos sus subscriptores y adjudicados.

**Patrones de Diseño utilizados**

Wrapper

La subscripción envuelve un cliente que la compone

Strategy

En adjudicación y financiamiento para manejar los diferentes comportamientos del plan de acuerdo a su naturaleza.

Observer

Para cumplir con la funcionalidad del stock del concesionario. Planta y Fábrica implementan ICambioStock que es una interfaz de tipo observable customizada para el negocio, que maneja las notificaciones de cambio de stock a sus observadores que implementan IStockObserver.

De esta forma las plantas (observable) al producir notifican a la Fábrica (observador y observable) que notifica los cambios de stock al concesionario (observador).

Factory

Implementamos un GeneradorDeDocumentos que es el encargado de construir tanto los cupones de adjudicación como los comprobantes de pago. Esto libera al Plan de Ahorro de la responsabilidad de saber cómo generarlos.

**Interfaces Complementarias**

Utilizamos tres interfaces complementarias para facilitar el desarrollo del sistema.

IGoogleMaps: calcula la distancia entre dos direcciones. A fines prácticos no está implementada y es reemplazada por un double para testear.

IAseguradoraDePlanes: Es la Interfaz que deben implementar todas las compañías de seguro que trabajen con el concesionario. Habilitaría la posibilidad de tener más de una compañía de seguro en el futuro con una estrategia diferente para calcular los costos.

IFleteCotizador: La implementarían las compañías de flete para establecer el valor de acuerdo a la distancia a recorrer.

Herramientas utilizadas:

* UML: violet UML Editor
* Código: Java 8 (Eclipse IDE)
* Testing: junit / mockito
* Code Coverage: EclEmma
* Control de versiones : git
* Repositorio Remoto: github