

PRÁCTICA 3: PATRONES DE DISEÑO

División de Tareas: Monkaps

[1] Iván.- Crear toda la implementación respecto a Decorator: Configuración de clase abstracta Tarifa y sus clases concretas

[2] Andrés.- Actualizar Clientes (Particulares y Empresas) y crear la primera parte de la implementación de Factory:

- Crear correctamente la clase abstracta cliente y sus concretas Particulares y Empresas
- Crear los objetos concretos para cada uno de los tipos de clientes que puedes gestionar.

[3] Andrés y solo al estar acabado el punto 1 .-Crear la segunda parte de Factory:

- Crear los distintos tipos de tarifas que se pueden aplicar a un cliente.

[4] Andrés e Iván Quedada para ver coherencia (17 de Abril del 2018)

1.- Objetivo

El objetivo de esta práctica es que puedas poner en práctica lo que has aprendido en clases de teoría sobre los patrones de diseño. En concreto, te pedimos que apliques los patrones *Factoría* y *Decorador*.

Recuerda que el patrón *Factoría* te permite desacoplar clases clientes de la creación de otras clases con las que trabajan.

Por su lado, el patrón *Decorador* te permite añadir nueva funcionalidad a una clase sin hacer uso de la herencia para ello.

2.-Enunciado

Sobre el código en Java que escribiste para realizar la *Práctica 2* debes introducir ciertas modificaciones/mejoras.

Las **tarifas** (Clase implementada con **Patrón Decorador**) van a experimentar cambios. Un cliente tendrá una tarifa básica consistente en un precio en céntimos por minuto. Pero una tarifa podrá ser complementada con distintas tarifas en función del día y hora en que se realice la llamada:

- Incorporaremos tarifas según **franja horaria** (por ejemplo de tardes, de 16:00 a 20:00) que tendrán un determinado coste.
- También existirán tarifas para un determinado **día** (por ejemplo los domingos) donde todas las llamadas de ese día tendrán un coste determinado por minuto.

A la hora de calcular el coste total de una factura se deberán calcular los costes individuales de cada una de las llamadas de esa factura, y a cada llamada se le aplicará la tarifa del cliente que le resulte más beneficiosa. Por ejemplo:

Un cliente tiene contratada:

- una tarifa básica de 15 céntimos por minuto,
- y una tarifa de tardes entre las 16:00 y las 20:00 a 5 céntimos por minuto
- y una tarifa de domingos gratis.

Si realiza una llamada un domingo a las 17:00, ésta le resultará gratis, pero si la efectúa cualquier otro día, le costará a 5 céntimos por minuto. Una llamada un día que no sea domingo y que no sea entre las 16:00 y las 20:00 costará a 15 céntimos cada minuto.

3.- Metodología

Esta práctica la vas a desarrollar en tres sesiones. Como en la práctica anterior, en cada una de las sesiones te vamos a proponer que te concentres en un aspecto particular, pero el ritmo de desarrollo lo marcas tú, lo que aquí te presentamos es una sugerencia de cómo avanzar.

Sesión 1

Actualiza la estructura de clases que obtuviste en la *Práctica 2* para adaptarla al enunciado de esta práctica.

Emplea el patrón *Decorador* para implementar los distintos tipos de Tarifa. La idea es que dada una tarifa básica, ésta pueda ser «decorada» con diferentes nuevas tarifas para situaciones especiales como son un precio especial para una franja horaria o un determinado día de la semana.

En las sesiones de teoría hemos visto cómo utilizar este patrón de diseño para calcular el precio total de un vehículo; en ese caso, el precio final era la suma del precio del accesorio más el precio del vehículo que se decoraba (que a su vez podía ser un nuevo accesorio aplicado al vehículo). En este caso el algoritmo es diferente, para devolver la mejor tarifa a aplicar se ha de comparar la tarifa que decora (accesorio en el caso del ejemplo de los vehículos), con la tarifa que se decora (que de nuevo puede ser una tarifa que está decorando a otra), y finalmente nos quedaremos con la de importe más bajo. Esto lo debemos hacer así porque no sabemos el orden en el que se aplicaron las tarifas y puede que la más conveniente para la franja en la que se realizó la llamada no sea la última tarifa que se contrató, sino alguna otra que se contrató antes.

Sesión 2

Utiliza el patrón de diseño Factoría con dos propósitos:

- Crear los objetos concretos para cada uno de los tipos de clientes que puedes gestionar.
- Crear los distintos tipos de tarifas que se pueden aplicar a un cliente.

Sesión 3

Escribe las pruebas necesarias para comprobar la correcta implementación de los dos patrones de diseño que has implementado.

Fuentes de información

- [Head first design Patterns](#) Capítulos 4 y 6.
- [Patrones de diseño](#) Capítulos 3 y 5.
- [Desarrollo de proyectos informáticos con tecnología Java](#) Capítulo 16.