Ejercicios sobre patrones. GoF 2

Ejercicio 1.

Se desea implementar un módulo que calcula las nóminas de los empleados de una empresa (en un planeta donde no existen los impuestos). Los empleados de una empresa tienen una escala que define su salario base (escala C=900€, escala B=1100€, escala A=1300€). A mayores, los empleados tienen diversos complementos. Algunos de los complementos dependen de la antigüedad, como son los sexenios (cada 6 años, +50€ en la nómina), los quinquenios (cada 5 años, +20€ en la nómina) y los trienios (cada 3 años, +10€ en la nómina). Por otra parte, existen otros complementos, como haber participado en cargos de gestión de la empresa, lo que añade 15€ más de forma vitalicia.

La base de datos de empleados es un fichero de la siguiente forma:

NOMBRE<TABULADOR>ESCALA<TABULADOR>AÑOS\_TRABAJADOS<TABULADOR>CARGO\_GESTION(SI/NO)

Un fichero de ejemplo se puede ser el siguiente:

JUAN    A       10      SI  
PEPE    B       25      NO

Se pide:

1. Diseñar el sistema empleando el/los patron/es GoF que consideres adecuados teniendo en cuenta que 1) es posible que se añadan distintos complementos en el futuro, 2) se desea evitar una explosión de clases para cada una de las posibles combinaciones a la hora de calcular una nómina y 3) el código que genera las órdenes de pago al banco no debe verse afectado porque una nómina tenga más o menos complementos (simplemente quiere poder calcular el total de cada empleado).
2. Implementar el sistema en Java que tome como entrada un fichero como el del ejemplo y saque órdenes de pago, por ejemplo:

JUAN 1435

PEPE 1480

Entregar

1. Diagrama de clases y breve justificación del patrón (PDF).
2. Código fuente (carpeta del proyecto eclipse).

Ejercicio 2.

Amplia el Ejercicio 3 de la práctica anterior (GoF-1) para que permita monitorizar la inicialización y finalización de programas. Un mensaje pondrá por pantalla cada vez que se cree o destruya un programa **cuántos están actualmente en curso**. Por ejemplo: "Creado un programa. Actualmente hay 2 en curso". Emplea un patrón GoF para que 1) más de un código pueda monitorizar los trabajos y 2) los trabajos sean independientes de las clases concretas que implementan el comportamiento ante la creación/finalización de programas.

Entregar

1. Código fuente.