FICHA

Autor: Sergio Martín Santana

Nombre: Observer Pattern. Mejora Clima.

Entrega: 04/12/2015

INTRODUCCION

PATRON OBSERVER

El patrón observador, en inglés Observer, es un patrón de diseño que define una dependencia del tipo ***uno-a-muchos*** entre objetos, de manera que cuando uno de los objetos cambia su estado, notifica este cambio a todos los dependientes. Se trata de un patrón de comportamiento por lo que está relacionado con algoritmos de funcionamiento y asignación de responsabilidades a clases y objetos. Los patrones de comportamiento describen no solamente estructuras de relación entre objetos o clases sino también esquemas de comunicación entre ellos y se pueden clasificar en función de que trabajen con clases (Método Plantilla) u objetos (Cadena de Responsabilidad, Comando, Iterador, Recuerdo, Observador, Estado, Estrategia, Visitante).

Este patrón también se conoce como el patrón de publicación-inscripción o modelo-patrón. Estos nombres sugieren las ideas básicas del patrón, que son: El objeto de datos, que se le puede llamar Sujeto a partir de ahora, contiene atributos mediante los cuales cualquier objeto Observador o vista se puede suscribir a él pasándole una referencia a sí mismo. El Sujeto mantiene así una lista de las referencias a sus observadores. Los observadores a su vez están obligados a implementar unos métodos determinados mediante los cuales el Sujeto es capaz de notificar a sus observadores suscritos los cambios que sufre para que todos ellos tengan la oportunidad de refrescar el contenido representado. De manera que cuando se produce un cambio en el Sujeto, ejecutado, por ejemplo, por alguno de los observadores, el objeto de datos puede recorrer la lista de observadores avisando a cada uno. Este patrón suele observarse en los frameworks de interfaces gráficas orientados a objetos, en los que la forma de capturar los eventos es suscribir listeners a los objetos que pueden disparar eventos.

El patrón Observer es la clave del patrón de arquitectura Modelo Vista Controlador (MVC). De hecho el patrón fue implementado por primera vez en Smalltalk's MVC basado en un framework de interfaz. Este patrón está implementado en numerosos librerías y sistemas, incluyendo todos los toolkits de GUI.

OBJETIVO

El objetivo de este trabajo es modificar el código de la práctica 2, añadiendo a él un patrón observador para facilitar el uso de la misma, automatizando la entrada de variables: año, lugar y tipo de gráfico. Para ello tendremos tres métodos de hacerlo:

- Implementando nuestro propio "kit" Observer.
- Utilizando las clases Observable y la interfaz Observer, proporcionada por java.
- Utilizando la Interfaz ActionListener, también proporcionada por java.

CÓDIGO IMPLEMENTADO

Para la consecucion de esta práctica, nos hemos decantado por la tercera vía utilizando la interfaz ***ActionListener***, dada la estructura con la que contábamos era la vía por la cual modificabamos la estructura en lo mínimo. Para ello en la clase ***Graficos***, añadimos una Referencia a la Ventana que lanza el prgrama (***Dash***), para poder contar con todas las variables necesarias. También enlazaremos en esta la interfaz ***ActionListener***, por lo que deberemos de implementar todos los métodos con los que cuenta esta, en este caso es simplemente una, ***ActionPerformed(Actionevent e){}***.