

Desarrollo de Software - Práctica 1.

Ejercicio 2 - Patrón Observador.

Adrián Portillo Sánchez

Para el ejercicio 2, el cual pedía implementar una aplicación meteorológica he utilizado el patrón observador.

En primer lugar este programa tiene dos partes diferenciadas, los Observadores y los Observables; en primer lugar hablaremos de los primeros.

Un Observador es una implementación de una interfaz con un método manejarEvento() el cual se llamará cuando sea notificado por los Observables; en mi aplicación tengo cinco observadores, todos ellos clases Singleton: dos de ellos simplemente guardan el valor de la temperatura y humedad del sistema, un tercero guarda otro valor que en mi programa, por cuestiones de sencillez es dependiente del valor de la humedad (será temperatura + 1): el tiempo satelital; las otras dos clases son las que varían un poco: la primera, BotonCambio tiene una llamada pulsarBoton() que obliga a una actualización de los valores en los Observables; y la segunda, GraficaTemperatura crea una gráfica con todos los valores de temperatura observados a lo largo del tiempo, gracias a una librería externa.

Nota Importante: Para crear la gráfica se usan dos librerías externas que estará incluidas en el .zip del ejercicio, para poder compilar el ejercicio se tendrán que añadir las librerías en el IDE.

Un Observable es una implementación de una clase abstracta la cual guarda un array de Observadores, sus métodos genéricos son añadir y notificar a todos los Observadores de este array, llamando a su método manejarEvento(). De esta clase heredan dos subclases en mi aplicación: la primera será un observable que guarda la temperatura en el sistema, que simplemente será un dato entero con sus getter y sus setter, y la segunda será lo mismo pero guardando la humedad.

Una vez implementado todo esto, para los cambios de temperatura se crea un simulador, el cual actualizará los valores de temperatura y humedad con valores aleatorios cada 60 segundos, o cuando la clase BotonCambio llama a pulsarBoton(); esta clase también instanciará los Observadores y Observables por primera vez.

Por último en una interfaz se tomarán las instancias de las clases de Observadores, y se mostrarán de forma adecuada. Dicha interfaz posee dos botones, un botón que obliga al cambio de los valores del sistema, es decir, llama al método pulsarBoton() de BotonCambio, y un botón que actualiza los valores mostrados por pantalla, ya que cambiar la temperatura del sistema guarda nuevos valores en el sistema, pero no es hasta que el usuario lo desea que estos se muestran.

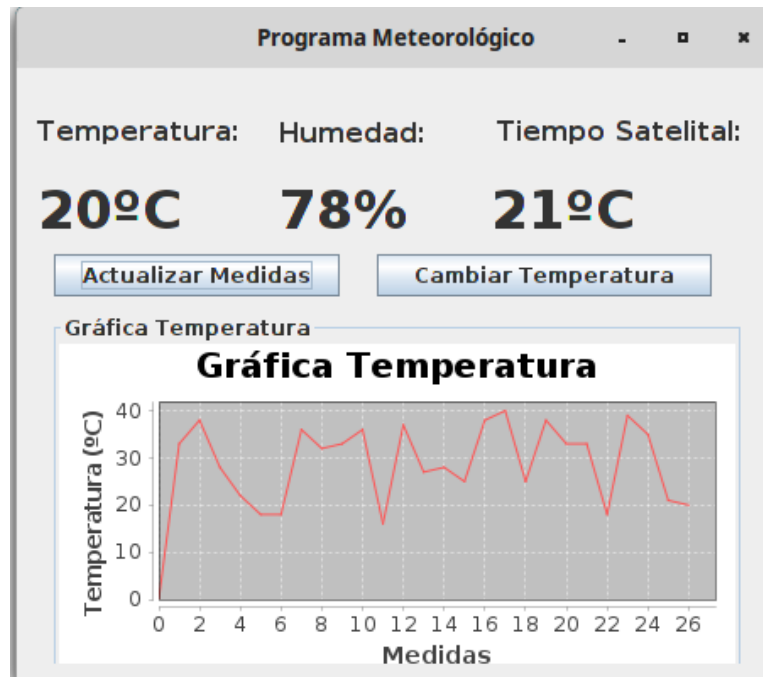


Figura 1: Muestra de la interfaz del programa.

A continuación se muestra el diagrama de clases del programa donde podemos ver de una forma más gráfica esto que he descrito, por cuestiones de visualización también dejo una copia en formato .png junto a este documento.

