

Datos Meteorológicos

Documentación y aclaraciones

El patrón de diseño utilizado en la solución a este problema ha sido el de “Observable - Observador”.

Aclaraciones sobre las clases mas importantes:

- Observador: Es una interfaz que tan solo tiene la cabecera del método manejarEvento el cual utilizarán todas las clases Observadoras (PantallaTemperatura, PantallaHumedad, PantallaPresion) para poder actualizar sus datos con respecto a los Observables.
- Observable: Esta clase abstracta la cual dispone de un atributo que es una lista de observadores, y dos métodos, uno para añadir mas observadores y otro para notificar a los observadores de que ha habido algún cambio en los valores. Esta clase la extenderán o utilizarán las tres clases observables de manera que no tienen que sobrescribir ninguno de sus métodos (debido a que el array de observadores es de Observador en general).
- Simulador: Esta clase extiende de Thread para poder lanzar la acción de actualizar datos cada X tiempo y mantener a la hebra en reposo mientras no se necesite. Por otro lado es la encargada de calcular los valores aleatorios de temperatura (calculando también los máximos y mínimos teniendo en cuenta el mes del año en el que nos encontramos), humedad y presión.
- ObservableHumedad, ObservableTemperatura, ObservablePresion: Son clases singleton de manera que no se podrán instanciar mas de una sola vez.

Tanto las clases asociadas a la humedad y a la presión las añadí ya que se pedía algún datos aparte de la temperatura.

Diagrama de Clases

