

## TP5 : Patrons de conception 2

### Objectifs

- ✓ Manipuler les patterns Observable et Décorateur
- ✓ Comprendre le principe SOLID

### Exercice 1 : Le patron de conception observable

Pour cet exercice nous allons prendre un exemple très connu directement tiré de « Head First Design Patterns » à savoir la station météo (voir Figure 1).

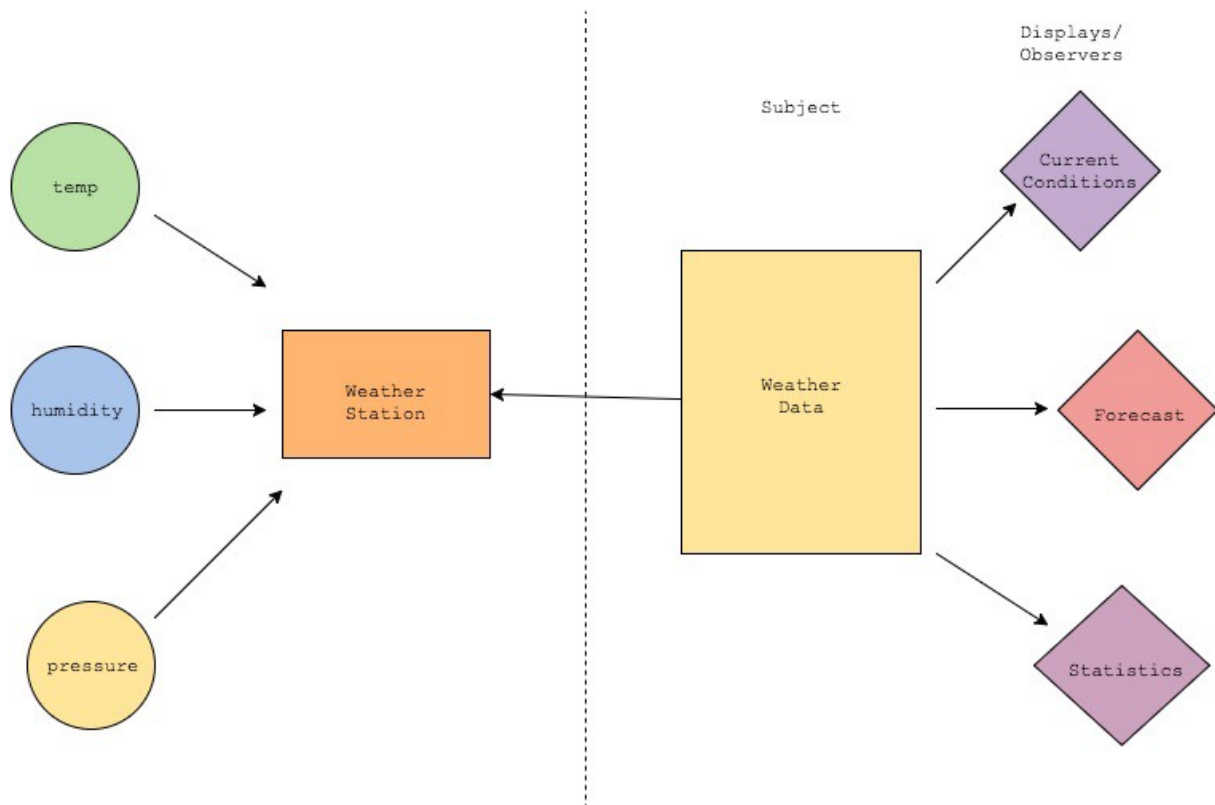


Figure 1 Exemple de la station météo

L'idée générale de cette application est qu'une station météo récupère des valeurs de température, d'humidité et de pression auprès de capteurs (les cercles) et les fournit à un service chargé de les afficher. Dans cet exemple, l'objet **WeatherData** est le sujet observé et les afficheurs (les losanges) sont les observateurs. Nous devons réaliser cette deuxième partie de l'application. Nous supposons donc dans un premier temps que les capteurs et la station météo existent.



Une spécification incomplète nous est fournie pour réaliser notre application (voir Figure 2)

WeatherData	
getTemperature() getHumidity() getPressure() measurementsChanged()	<pre>1 public class WeatherData { 2 3     // Déclaration des variables 4 5     public void measurementsChanged(){ 6         float temp = getTemperature(); 7         float humidity = getHumidity(); 8         float pressure = getPressure(); 9 10        currentConditionsDisplay.update(temp, humidity, pressure); 11        statisticsDisplay.update(temp, humidity, pressure); 12        forecastDisplay.update(temp, humidity, pressure); 13    } 14 15    // methodes de getter de temperature, humidite et pression 16 }</pre>

Figure 2:Spécifications

Cette solution n'est pas satisfaisante car elle ne respecte pas les principes SOLID. En effet si on doit changer le nombre d'affichage, il est nécessaire de modifier le code de notre application. De plus, bien que les trois observateurs aient choisi d'utiliser la même méthode d'affichage il n'y a pas pour l'instant d'interface définie et de plus on travaille directement avec les implémentations de cette interface ce qui couple fortement notre classe **WeatherData** avec les afficheurs.

### Consignes

- ✓ Proposez un diagramme classe basé sur le patron Observer qui permette de découpler la classe **WeatherData** des afficheurs.
- ✓ Implémentez votre solution proposée
- ✓ A l'aide d'objet **Mock**, proposez des tests permettant de tester le bon fonctionnement de votre solution

## Exercice 2 : Observable en Java 8

Depuis Java 8, le patron de conception **Observable** est disponible dans le paquetage **java.util**. En étudiant la JDK, il va falloir reprendre l'exercice précédent en utilisant cette fois ci directement **Observable**.

### Consignes

- ✓ Reprenez l'exercice précédent en utilisant **Observable**.
- ✓ Modifiez les tests pour permettre de prendre en compte ces modifications. En particulier, réfléchissez à comment utiliser Mockito pour tester la partie observable.



## Exercice 3 : Reprise de l'exercice du concours de galettes

L'objectif de cet exercice va être d'appliquer le patron Observable à notre concours de mangeurs de galettes. Pour cela nous allons ajouter une nouvelle classe qui représentera les **Supporteurs** du concours. Ces supporteurs sont là pour assister le concours et applaudisse à chaque fois qu'un concurrent est éliminé.

### Consignes

- ✓ Modifiez le diagramme de classe pour intégrer cette classe **Supporteur**. Utiliser un patron observable pour découpler les spectateurs du concours.
- ✓ Ecrivez les différents tests unitaires associés.

## Exercice 4 : Finalisation du concours

Le succès grandissant de notre concours entraine l'apparition de plus en plus de concurrents qui proposent d'ajouter toujours plus de galettes. La conception actuelle reposant sur des fabriques ne devient plus maintenable, vous décidez donc de reprendre la conception en utilisant cette fois ci un patron de conception Décorateur.

### Consignes

- ✓ Modifiez le diagramme de classe pour introduire ce nouveau patron dans la gestion des ingrédients des galettes
- ✓ Modifiez le code de votre application
- ✓ Modifiez les tests au fur et à mesure de votre conception