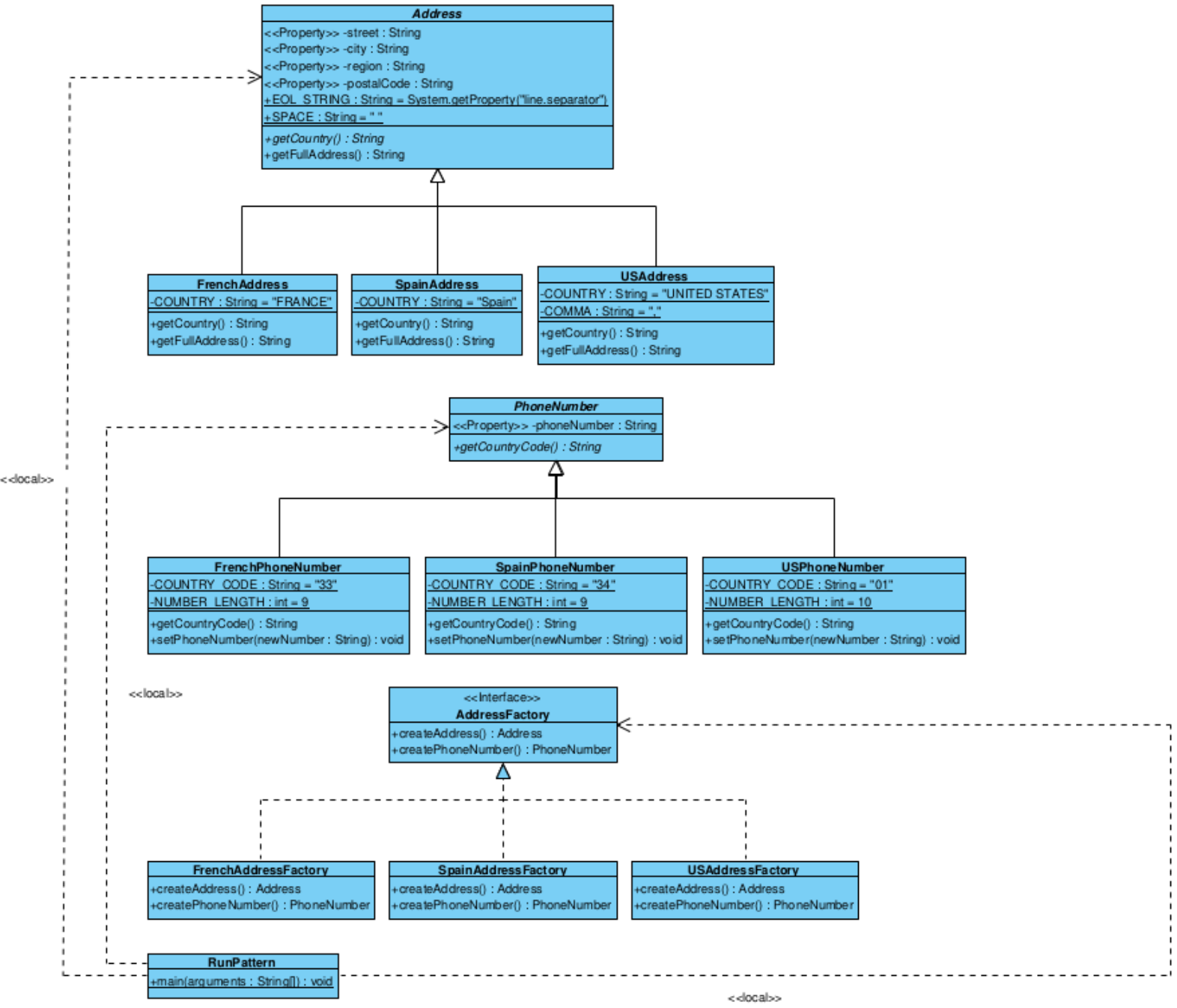
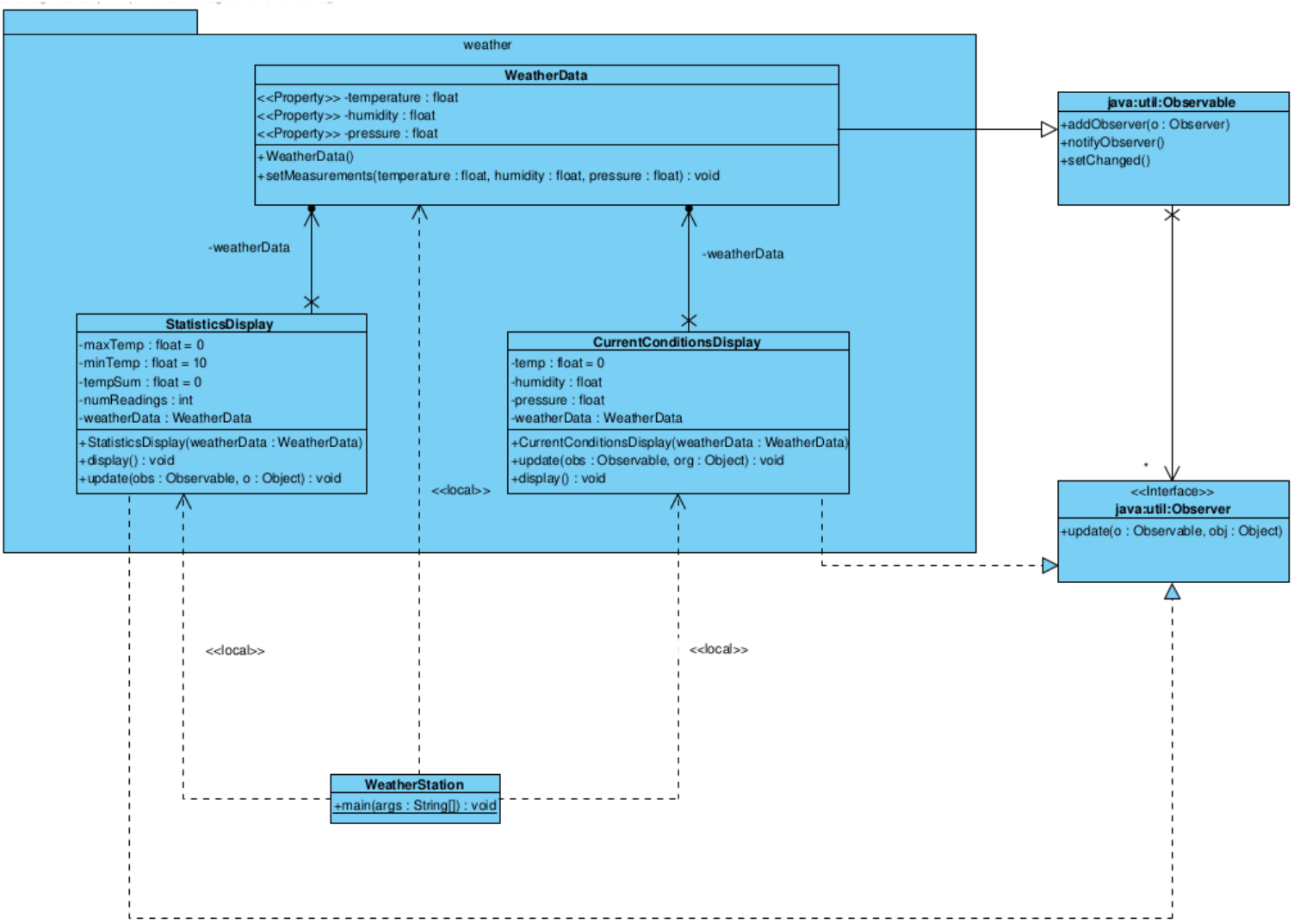
Conception Java : TP6
Antoine Gicquel

Le diagramme de classe final de l'exercice 2 :



Le diagramme de classes de l'exercice 3 intégrant le pattern Observer :



Le code complet de l'exercice 3 :

```
package weather;

public class WeatherStation {
    public static void main(String[] args) {
        WeatherData weatherData = new WeatherData();
        CurrentConditionsDisplay currentDisplay = new
CurrentConditionsDisplay(weatherData);
        StatisticsDisplay sd = new StatisticsDisplay(weatherData);
        weatherData.setMeasurements(12, 65, 1040);
        currentDisplay.update();
    }
}


package weather;

public class CurrentConditionsDisplay implements java.util.Observer {
    private float temp = 0;
    private float humidity;
    private float pressure;
    private WeatherData weatherData;

    public CurrentConditionsDisplay(WeatherData weatherData) {
        this.weatherData = weatherData;
        weatherData.addObserver(this);
    }

    public void update(java.util.Observable obs, Object org) {
        temp = weatherData.getTemperature();
        humidity = weatherData.getHumidity();
        pressure = weatherData.getPressure();
        this.display();
    }

    public void display() {
        System.out.println("Current conditions : ");
        System.out.println("* temperature : " + temp);
        System.out.println("humidity : "+ humidity);
        System.out.println("pressure : "+ pressure);
    }
}
```

```
package weather;

import java.util.*;

public class WeatherData extends java.util.Observable {
    private float temperature;
    private float humidity;
    private float pressure;

    public void setMeasurements(float temperature, float humidity, float
pressure) {
        this.temperature = temperature;
        this.humidity = humidity;
        this.pressure = pressure;
        this.setChanged();
        this.notifyObservers();
    }

    public float getTemperature() {
        return temperature;
    }

    public float getHumidity() {
        return humidity;
    }

    public float getPressure() {
        return pressure;
    }
}
```

```
package weather;

public class StatisticsDisplay implements java.util.Observer {
    private float maxTemp = 0;
    private float minTemp = 10;
    private float tempSum= 0;
    private int numReadings;
    private WeatherData weatherData;

    public StatisticsDisplay(WeatherData weatherData) {
        this.maxTemp = 0;
        this.minTemp = 10;
        this.tempSum= 0;
        this.weatherData = weatherData;
        this.weatherData.addObserver(this);
    }

    public void update(java.util.Observable obs, Object o) {
        float temp = weatherData.getTemperature() ;
        tempSum += temp;
        numReadings++;
        if (temp > maxTemp)
            maxTemp = temp;
        else if (temp < minTemp)
            minTemp = temp;
        this.display();
    }

    public void display() {
        System.out.println("Avg/Max/Min temperature = "+ (tempSum / numReadings)
+ "/" + maxTemp + "/" + minTemp);
    }
}
```

Le diagramme de classes détaillé de l'exercice 4 :