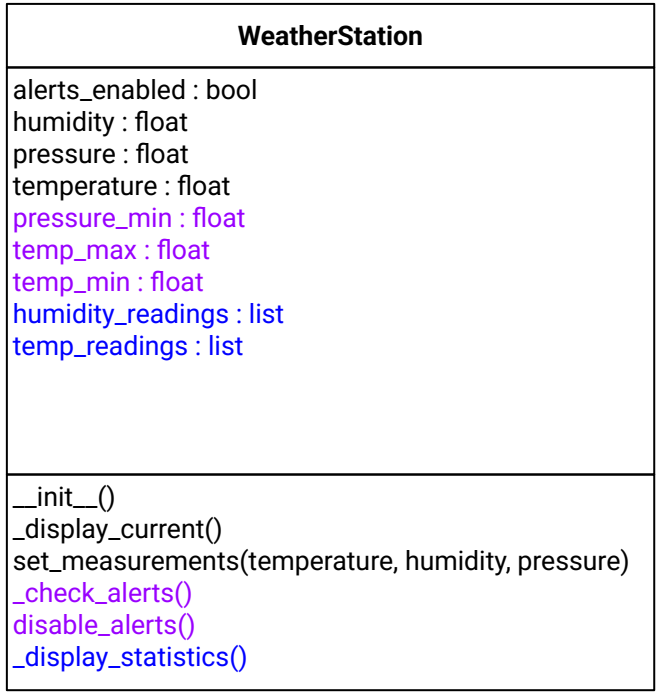
weather_mono.py
Class Diagram

start level 1
additions level 2 (alerts)
additions level 3 (statistics)
additions level 4 (forecast)



weather_mono.py
Sequence Diagram

start level 1
additions level 2 (alerts)
additions level 3 (statistics)
additions level 4 (forecast)

```
weather_station = WeatherStation()
```

```
weather_station.set_measurements(22.5, 60.0, 1013.2)
```

WeatherStation
alerts_enabled : True humidity : 0 pressure : 0 temperature : 0 pressure_min : 950 temp_max : 35 temp_min : -10 humidity_readings : [] temp_readings : []

WeatherStation
alerts_enabled : True humidity : 60.0 pressure : 1013.2 temperature : 22.5 pressure_min : 950 temp_max : 35 temp_min : -10 humidity_readings : [60.0] temp_readings : [22.5]
_display_current() _check_alerts() _display_statistics()

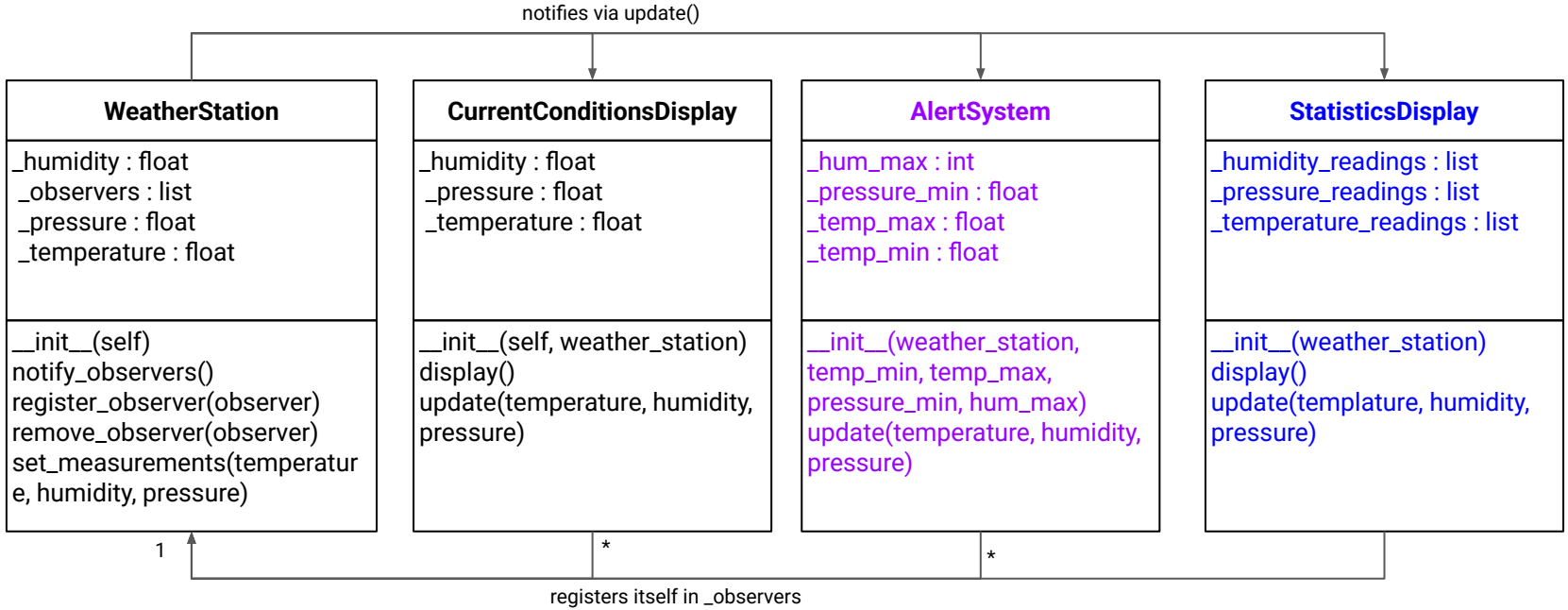
```python
def _display_current(self):
    """Display current conditions."""
    print(f"Current: {self.temperature:.1f}°C, "
          f"{self.humidity:.1f}% humidity, {self.pressure:.1f} hPa")
```

```python
def _check_alerts(self):
    """Check for weather alerts."""
    # Temperature alerts
    if self.temperature < self.temp_min:
        print(f"🚨 COLD ALERT: Temperature {self.temperature:.1f}°C is below minimum!")
    elif self.temperature > self.temp_max:
        print(f"🚨 HEAT ALERT: Temperature {self.temperature:.1f}°C is above maximum!")

    # Pressure alerts
    if self.pressure < self.pressure_min:
        print(f"🚨 STORM ALERT: Low pressure {self.pressure:.1f} hPa detected!")

    # Humidity alerts
    if self.humidity > 90:
        print(f"🚨 HUMIDITY ALERT: Very high humidity {self.humidity:.1f}%!")
```

```python
def _display_statistics(self):
    """Display weather statistics."""
    if len(self.temp_readings) < 2:
        print("Statistics: Insufficient data")
        return

    min_temp = min(self.temp_readings)
    max_temp = max(self.temp_readings)
    avg_temp = sum(self.temp_readings) / len(self.temp_readings)

    min_hum = min(self.humidity_readings)
    max_hum = max(self.humidity_readings)
    avg_hum = sum(self.humidity_readings) / len(self.humidity_readings)

    print(f"Statistics: Temp {min_temp:.1f}-{max_temp:.1f}°C (avg {avg_temp:.1f}°C), "
          f"Humidity {min_hum:.1f}-{max_hum:.1f}% (avg {avg_hum:.1f}%)")
```

level1_single_observer_modular.py
Class Diagram

start level 1
additions level 2 (alerts)
additions level 3 (statistics)
additions level 4 (forecast)

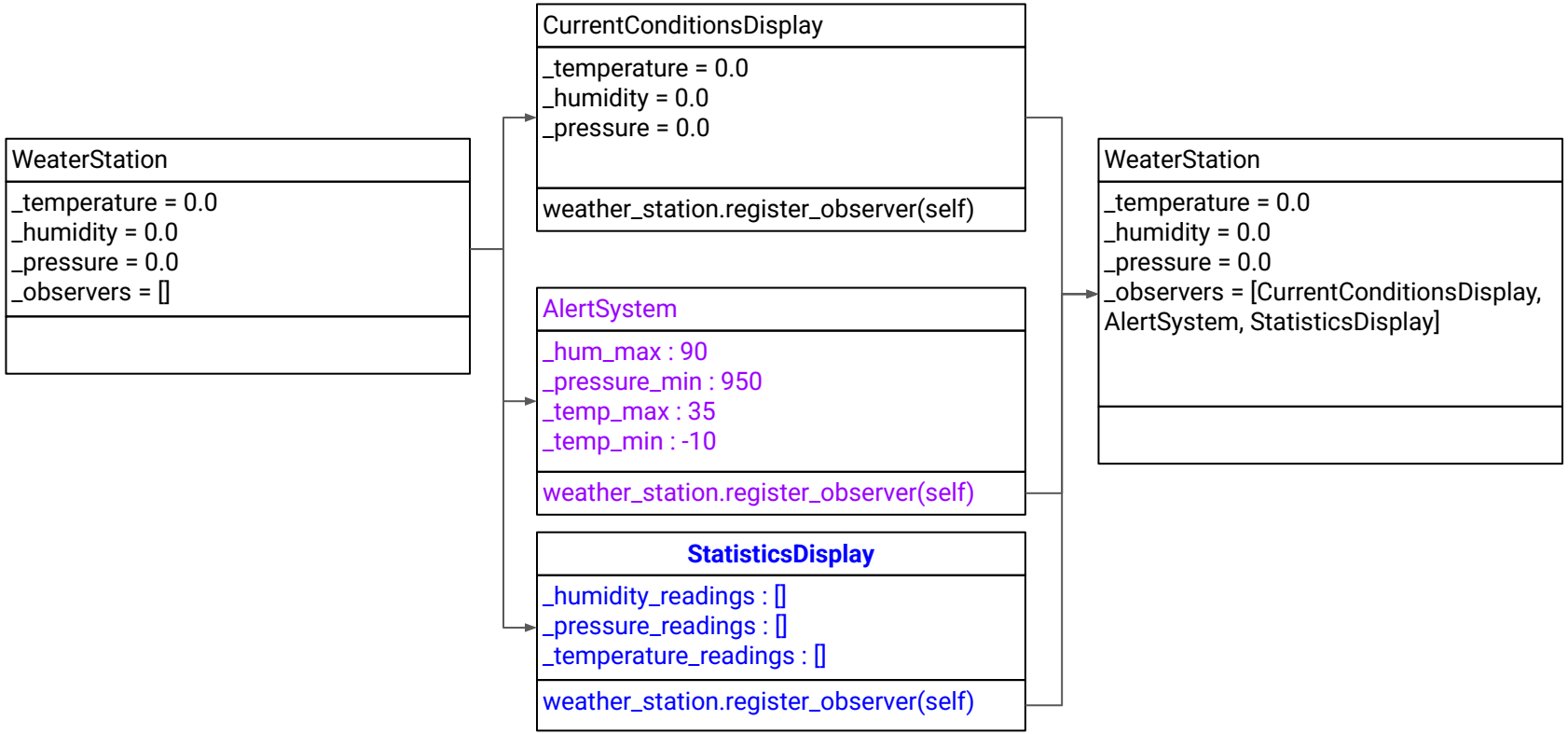


Initialise WeatherStation
Sequence Diagram

start level 1
additions level 2 (alerts)
additions level 3 (statistics)
additions level 4 (forecast)

 # Create weather station
weather_station = WeatherStation()

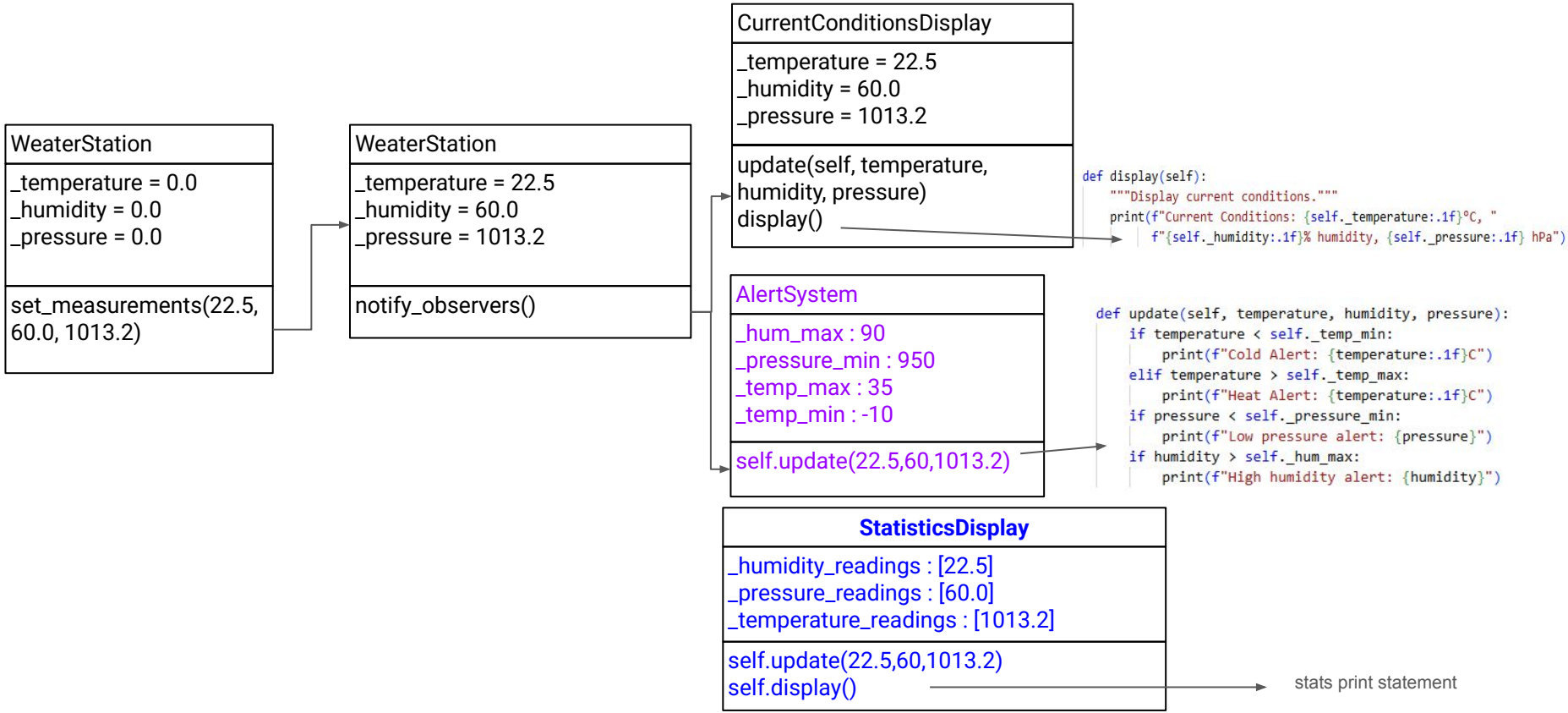
 # Create and register display
current_display = CurrentConditionsDisplay(weather_station)
alert_sytem = AlertSystem(weather_station)
stats_display = StatisticsDisplay(weather_station)



Update Measurements
Sequence Diagram

start level 1
additions level 2 (alerts)
additions level 3 (statistics)
additions level 4 (forecast)

```
# Simulate weather updates
weather_station.set_measurements(22.5, 60.0, 1013.2)
```



Update Measurements
Sequence Diagram

start level 1
additions level 2 (alerts)
additions level 3 (statistics)
additions level 4 (forecast)

```
# Simulate weather updates
weather_station.set_measurements(22.5, 60.0, 1013.2)
```

```
Participant: Main
Participant: WeatherStation
Participant: CurrentConditionsDisplay


Main -> WeatherStation: set_measurements(22.5, 60.0, 1013.2)
   WeatherStation: _temperature = 22.5
   WeatherStation: _humidity    = 60.0
   WeatherStation: _pressure    = 1013.2

   WeatherStation -> WeatherStation: notify_observers()
      loop for each observer in _observers
         WeatherStation -> CurrentConditionsDisplay: update(22.5, 60.0, 1013.2)
            CurrentConditionsDisplay: _temperature = 22.5
            CurrentConditionsDisplay: _humidity    = 60.0
            CurrentConditionsDisplay: _pressure    = 1013.2

            CurrentConditionsDisplay -> CurrentConditionsDisplay: display()
               (prints: "Current Conditions: 22.5°C, 60.0% humidity, 1013.2 hPa")
      end loop
   return from notify_observers
return from set_measurements
```