

SE311 – Software Architecture

Smart City Application

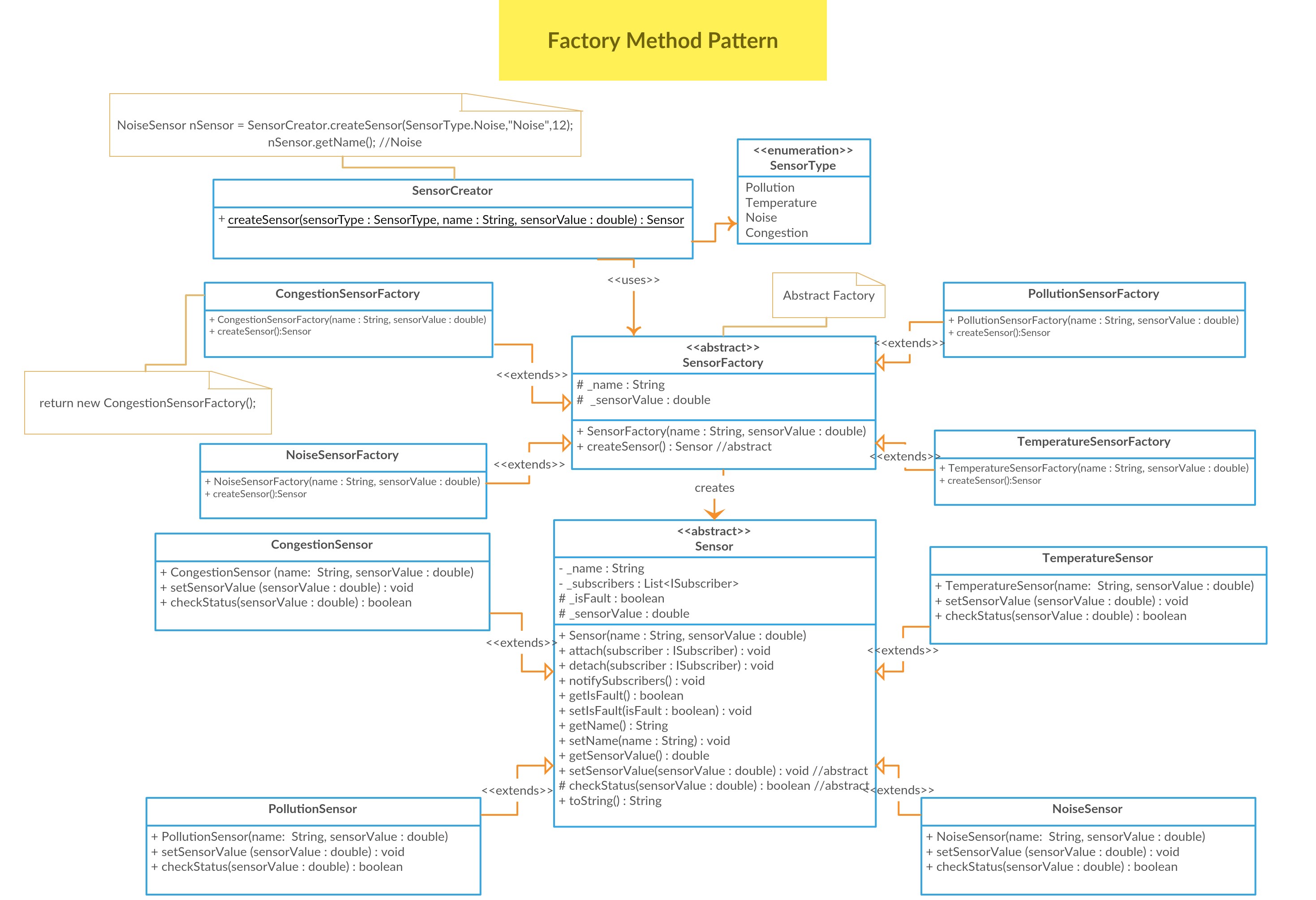
Mehmet Aydın KICIRTI -20160602105

Why We chosen below these patterns ?

1. Factory Pattern
2. Command Pattern
3. Iterator Pattern
4. Composite Pattern
5. Singleton Pattern
6. Observer Pattern

2.Participant of Each Pattern

- 2.1) Factory Pattern =>



Participants =>

Abstract Factory => SensorFactory

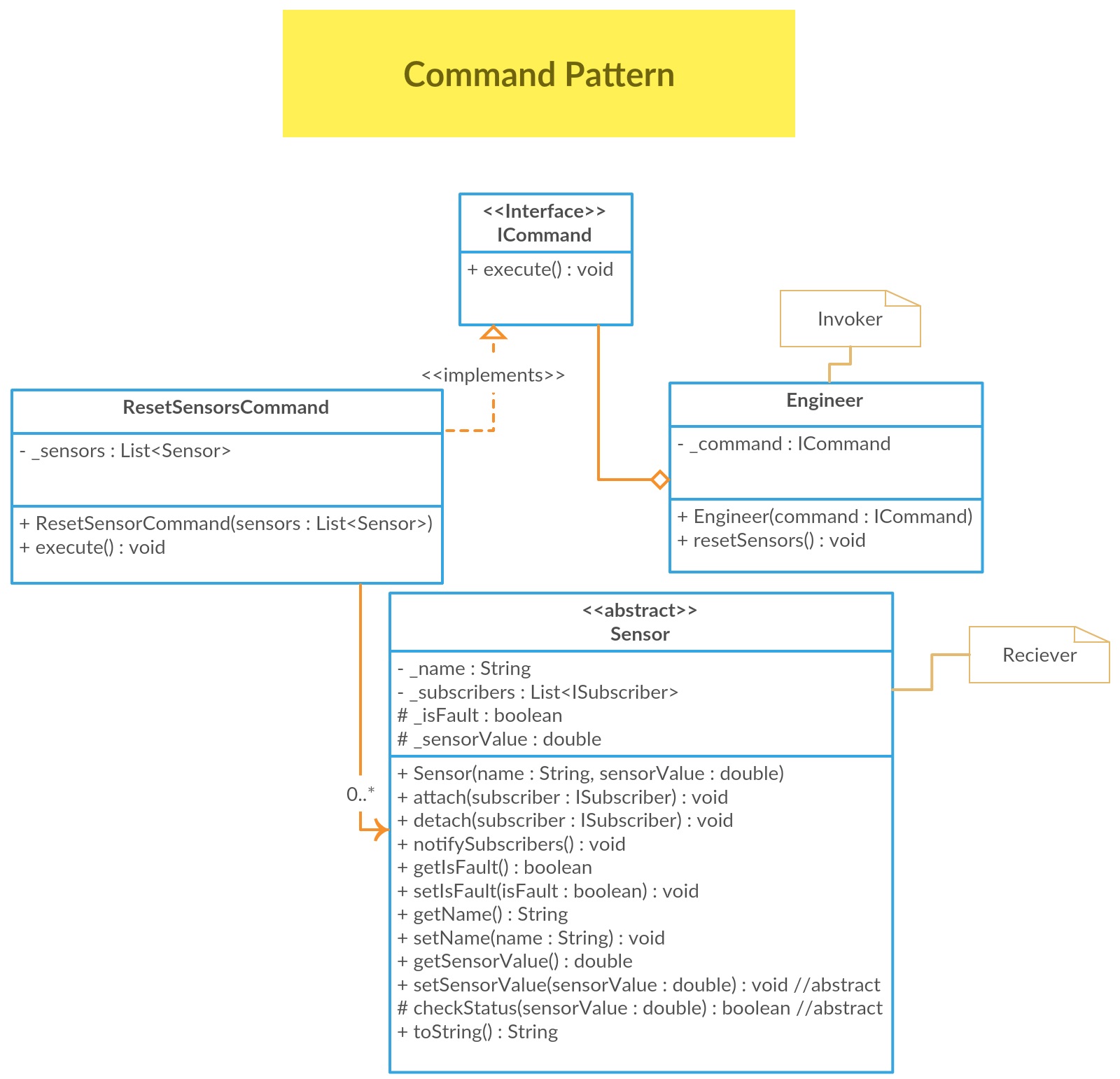
Concrete Factory => CongestionSensorFactory, PollutionSensorFactory,NoiseSensorFactory, TemperatureSensorFactory

AbstractProduct => Sensor

ConcreteProduct => NoiseSensor, CongestionSensor, TemperatureSensor, PollutionSensor

Client => SensorCreator

* 2.2) Command Pattern =>



Participants

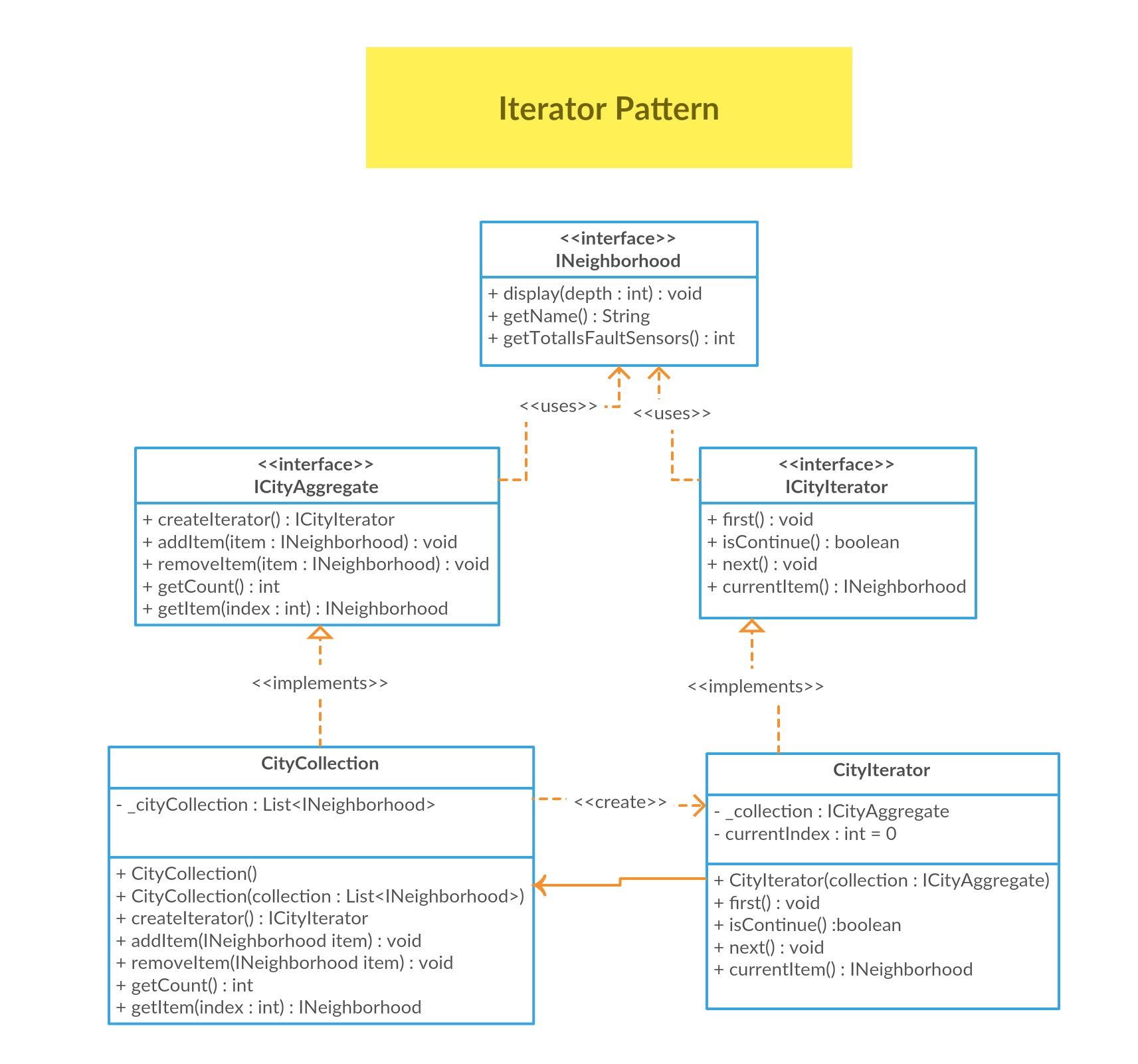
Command => ICommand

ConcreteCommand => ResetSensorsCommand

Invoker => Engineer

Reciever => Sensor

- 2.3) Iterator Pattern



Participants

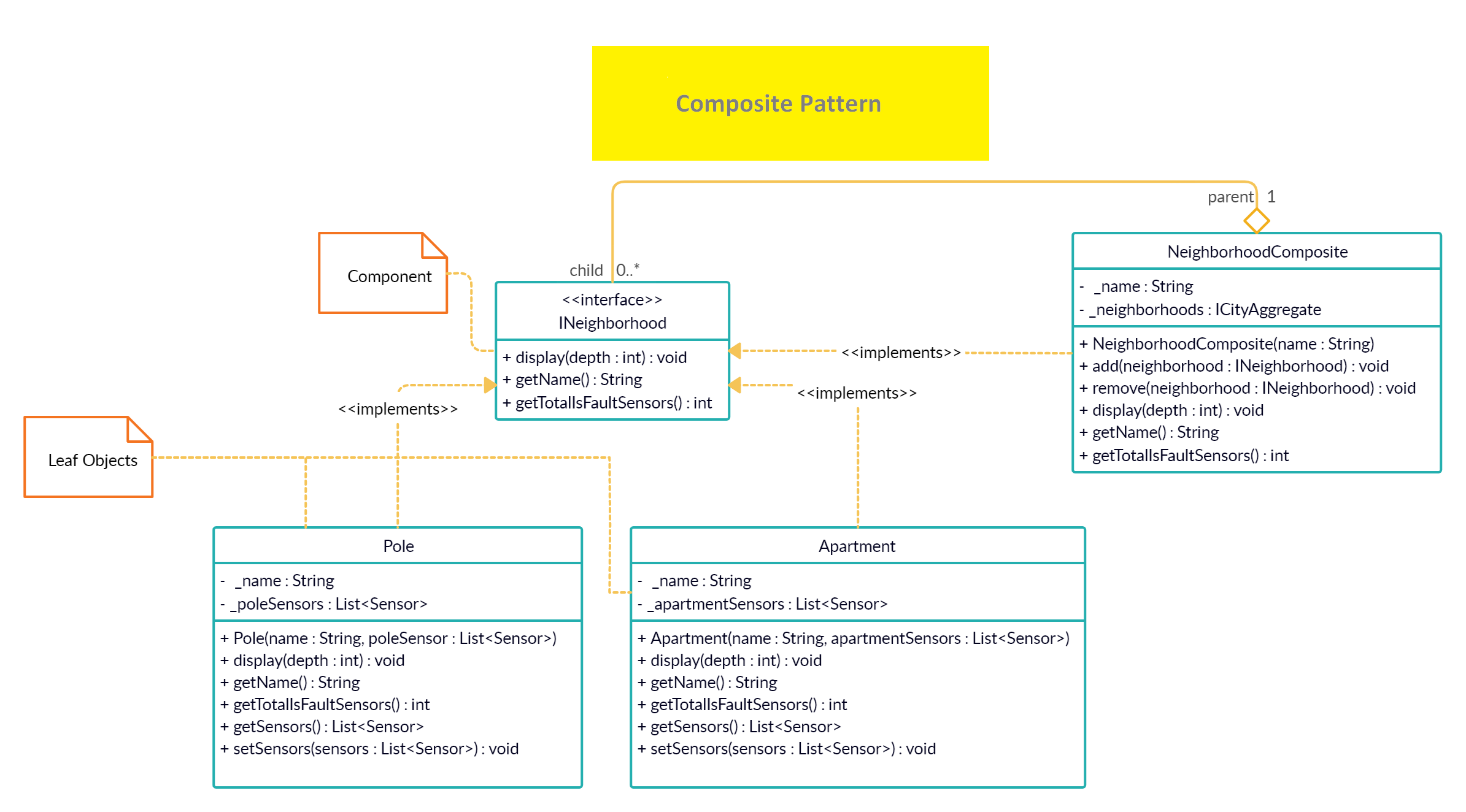
Aggregate => ICityAggregate

Iterator => ICityIterator

ConcereteAggregate => CityCollection

ConcreteIterator=>CityIterator

- 2.4) Composite Pattern



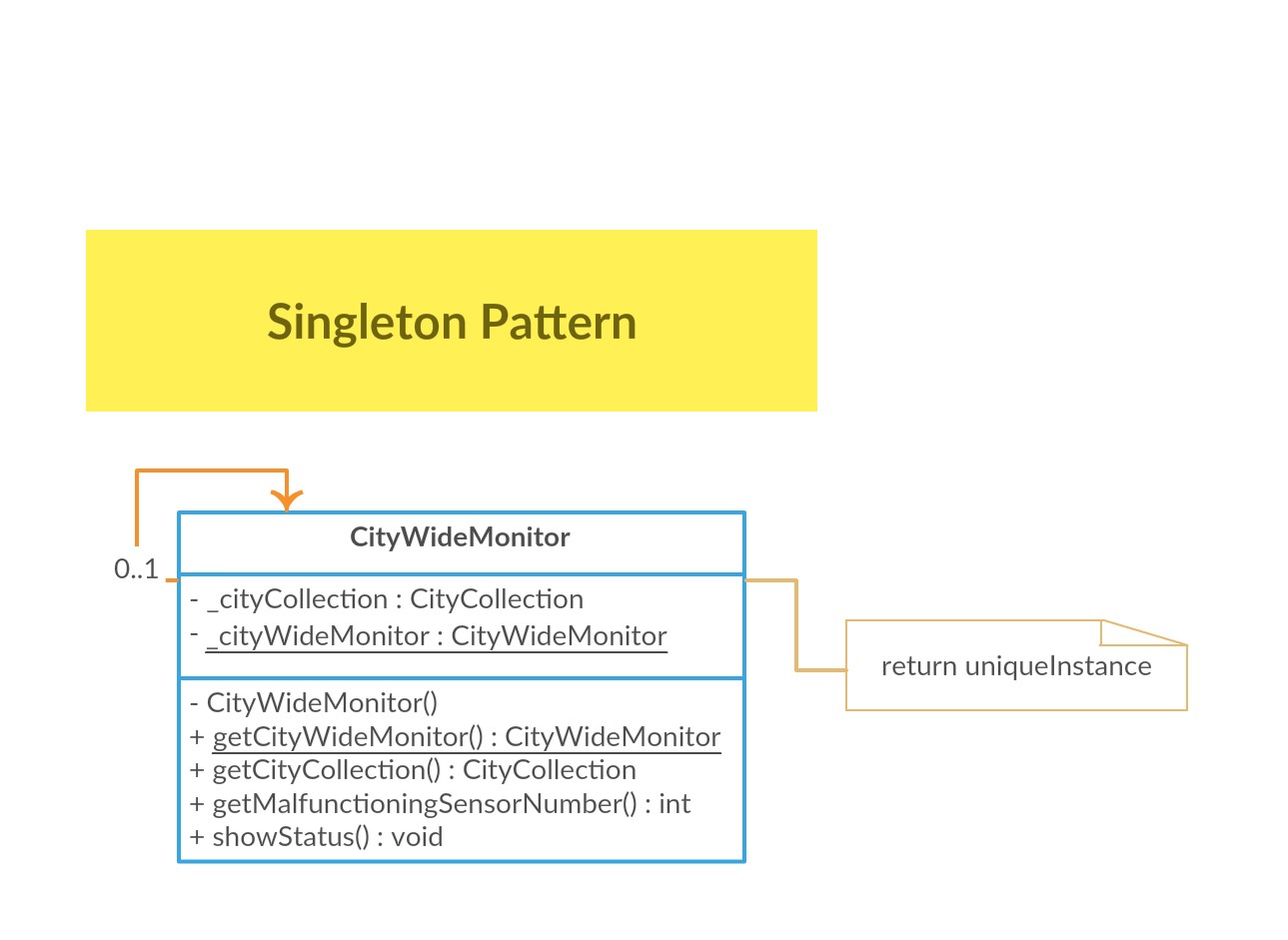
-- Participants

Component => INeighborhood

Composite => NeighborhoodComposite

Leaf => Pole, Apartment

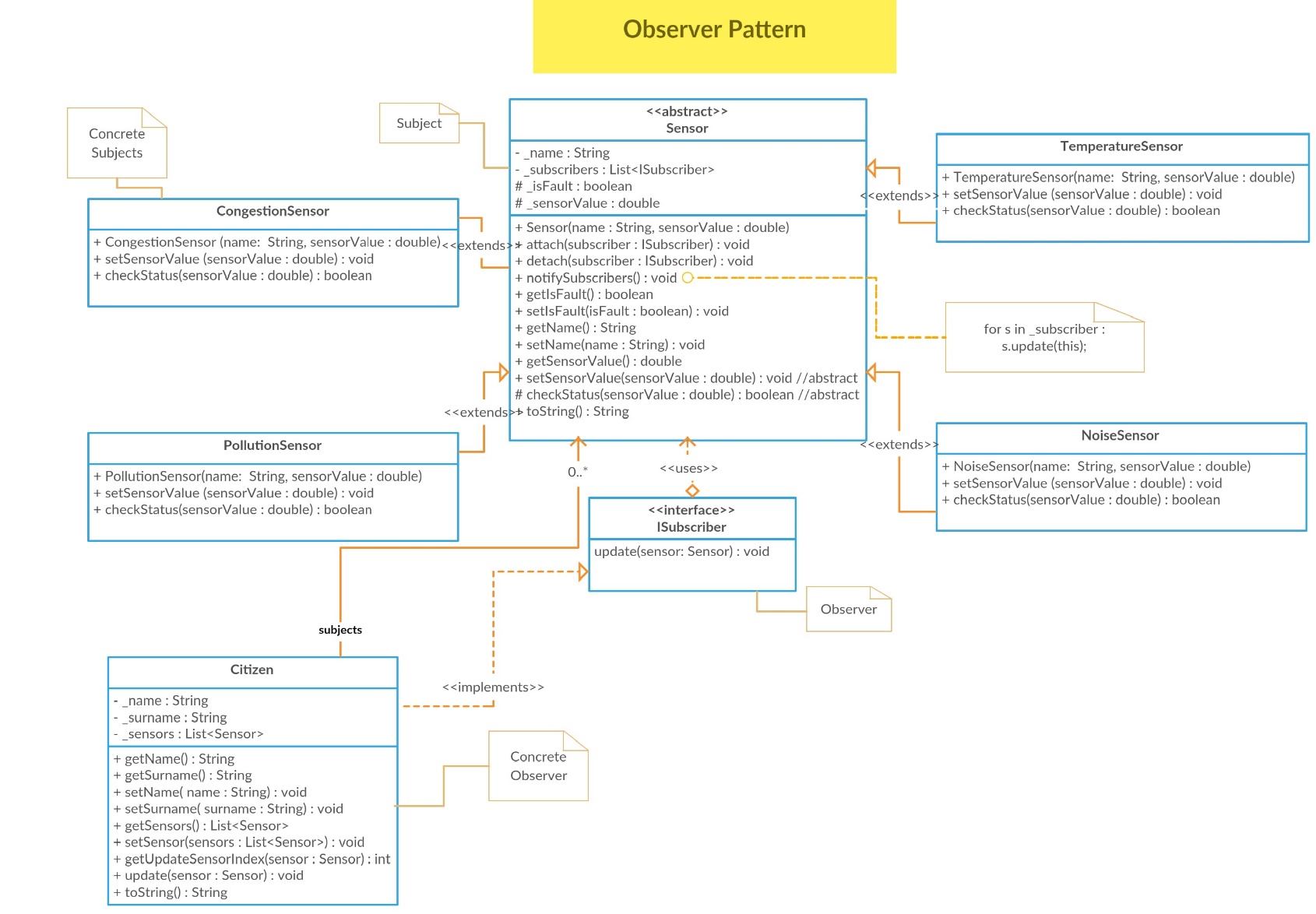
- 2.5) Singleton Pattern



* Participants

Singleton : CityWideMonitor

- 2.6) Observer Pattern



Participants =>

Subject : Sensor

Observer : ISubscriber

Concerete Subject : CongestionSensor, TemperatureSensor, PollutionSensor, NoiseSensor

Concerete Observer : Citizen

Conclusion

In the Smart City Application, we tried to implement 6 patterns. Before drawing the UML, We tried to think about the best choice patterns from the scenario. After that, we started to draw the UML, but this part was so hardest part for us. Especially, especially when we were trying to connect classes in UML. We used Creately to draw the class diagram, then we were adapting to own structures on application. As a result, we aimed to understand the best solution to the use of design patterns.