**Observer Design Pattern:**

**Link:** [**https://refactoring.guru/design-patterns/observer**](https://refactoring.guru/design-patterns/observer)

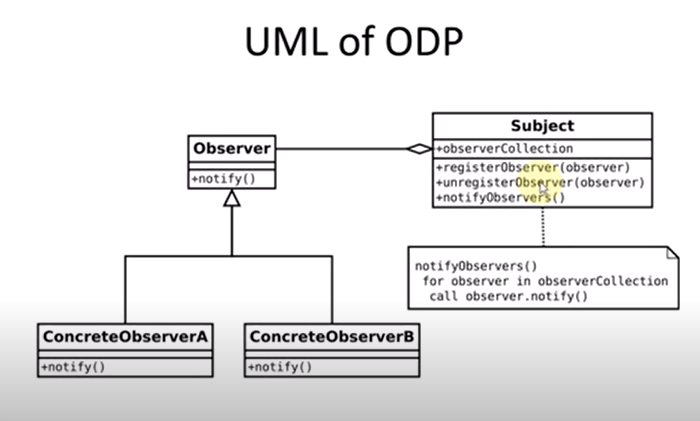
**YouTube Link:** <https://www.youtube.com/watch?v=KBkkEKNlE6I&list=PLk6CEY9XxSIDZhQURp6d8Sgp-A0yKKDKV&index=5>

Defines a one-to-many dependency between objects so that when one object changes state, all its

Dependents are notified and updated automatically.



**UML of ODP:**



**Application:**

Use the Observer pattern when changes to the state of one object may require changing other objects, and the actual set of objects is unknown beforehand or changes dynamically.