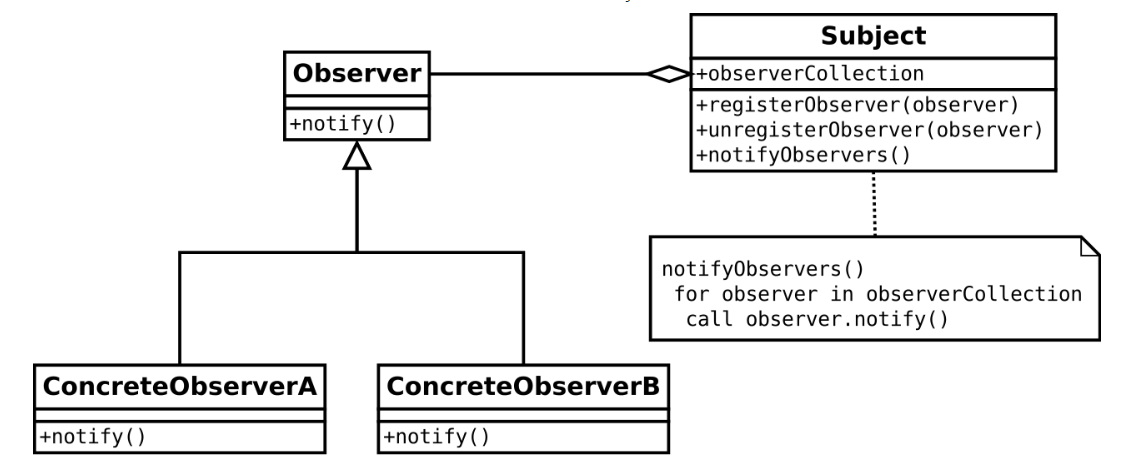
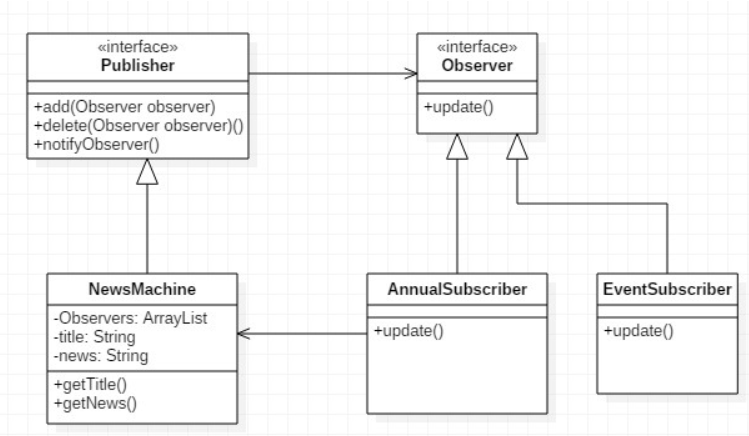
|  |  |
| --- | --- |
| Name | Observer |
| Description | -객체의 상태 변화를 관찰하는 관찰자들, 즉 observer들의 목록을 객체에 등록하여 상태 변화가 있을 때마다 method 등을 통해 객체가 직접 목록의 각 observer에게 통지하도록 하는 디자인 패턴  -주로 분산 이벤트 핸들링 시스템을 구현하는데 사용되며 발행/구독 모델로 알려져 있기도 함  -객체의 상태를 display하는 것과 객체 자체를 분리하여 다양한 display가 제공되도록 지원  -The Observer defines a one-to-many relationship so that when one object changes state, the others are notified and updated automatically. |
| Problem description | -상태정보에 대해 다양한 방법으로 display하고자 할 때  -적용되는 경우  (1) 상태 변화를 수신해야 하는 객체가 여러 개거나 전달 받아야 할 정보가 많을 때  (2) 상태 정보가 한가지 이상의 방법으로 display될 때  (3) 상태 정보를 유지하고 있는 객체는 어떤 방식으로 이 정보가 display 되는지 알 필요 없게 할 때 |
| Solution description | -Two abstract objects: Subject, Observer   -Two concrete objects: ConcreteSubject, ConcreteObserver  -Subject가 Observer에 대해서 아는 것은 Observer가 특정 인터페이스(Observer 인터페이스)를 구현 한다는 것  -Observer는 언제든지 새로 추가할 수 있음 (Subject는 Observer 인터페이스 구현하는 객체 목록에만 의존하기 때문)  -새로운 형식의 Observer를 추가하려 해도 Subject를 전혀 변경할 필요가 없음. (새로운 클래스에서 Observer 인터페이스만 구현해주면됨)  -Subject나 Observer가 바뀌더라도 서로에게 전혀 영향을 주지 않음 => Subject와 Observer는 서로 독립적으로 재사용할 수 있음  -추상객체는 모든 경우에 적용될 수 있는 일반적인 operations만을 포함   -ConcreteSubject는 Subject 클래스를 상속 받아 정의되며, 화면에 보일 내용을 유지함   -Observer는 상태가 변화되었음을 알려주는 역할을 담당함=> Subject는Observer를 추가하거나 제거할 수 있음  -ConcreteObserver는 추상객체인 Observer의 Update() method를 구현하여 ConcreteSubject의 상태를 복사하여 유지   -ConcreteObserver는 상태가 변화 될 때마다 자동적으로 상태변화를 화면에 보여줌 |
| Consequences | Subject는 abstract observer에 대해서만 알고 있으며 concrete class에 대해서는 아는 바 없음=> 해당 객체간의 coupling이 최소화 |



[그림1. Observer pattern UML]



[그림1. Observer pattern 예시 UML]