

Funcionamento

O padrão **Observer** funciona como assinaturas de jornais e revistas, ou seja, temos uma editora que publica as edições e pessoas que assinam os jornais ou revistas dessa editora e sempre recebem as novas edições assim que elas são publicadas. Enquanto a pessoa é assinante ela continua recebendo as edições na sua casa. Se a pessoa cancelar a assinatura do jornal ou da revista ela para de receber as edições.

O padrão Observer funciona da mesma forma, no entanto, tem-se que a editora (que publica) é o chamado **SUBJECT** no Padrão Observer e os assinantes (que recebem as novas publicações) são os chamados **OBSERVER**.

Os **OBSERVERs** registram-se no **SUBJECT** para receber atualizações quando os dados do **SUBJECT** são alterados. Os **OBSERVERs** também podem cancelar o seu registro e dessa forma não receber mais nenhuma atualização do **SUBJECT**.

O Diagrama de classe abaixo mostra mais detalhes sobre o funcionamento do padrão Observer.

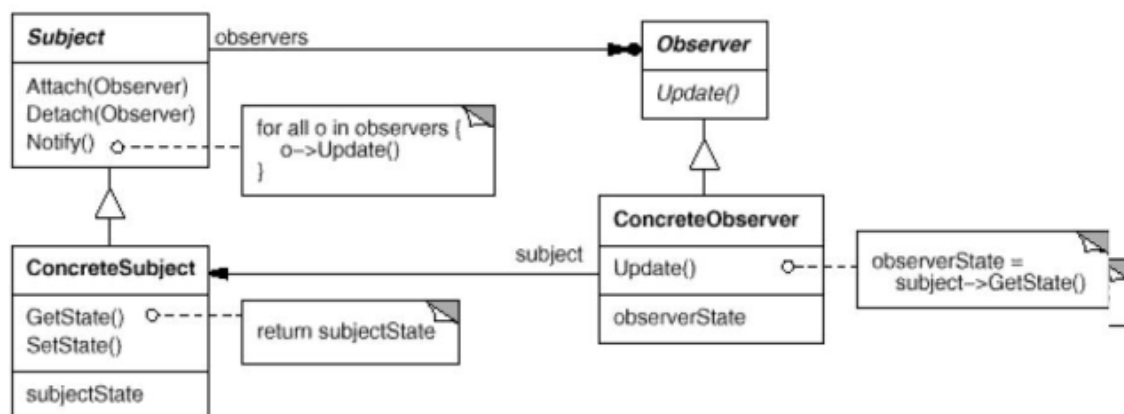


Figura 1: Diagrama de classe do Padrão Observer

No diagrama de classe acima nota-se a presença da interface **Subject** e da sua classe concreta **ConcreteSubject** que define o comportamento dos objetos para se registrarem (`Attach(anexar)`) e também para serem removidos (`Detach(separar)`). O **ConcreteSubject** é quem implementa

a interface Subject, além de definir os métodos da interface ele ainda define o seu próprio estado. O método notify() será utilizado para atualizar todos os observadores registrados sempre que o seu estado mudar. Do outro lado do diagrama tem-se o Observer e o ConcreteObserver que tem o método update() que é chamado quando o estado do Subject é alterado. A classe ConcreteObserver implementa a interface Observer, definindo assim o método update().

Podemos notar que aqui há uma relação UM-PARA-MUITOS, ou seja, temos UM Subject para MUITOS Observers.

A definição formal do Padrão Observer é: “O Padrão Observer define uma dependência um-para-muitos entre os objetos de modo que quando um objeto muda de estado, todos os seus dependentes são notificados e atualizados automaticamente”.

Princípio da Ligação Leve

O padrão Observer adere ao princípio de projeto em que se busca Designs levemente ligados entre objetos que interagem, isto é, eles interagem normalmente, no entanto sabem muito pouco um do outro.

O Subject sabe apenas que um Observer implementa uma interface comum a todos os Observers (interface Observer), apenas isso. O Subject não sabe a classe concreta que a implementa e nem sabe o que ela faz ou o qualquer outra coisa a respeito. Isto fica mais claro na implementação do padrão em que, como é de praxe dos padrões, programa-se para interface e não para classes concretas.

Essa ligação leve também proporciona mais flexibilidade ao padrão, pois como pode-se notar podemos adicionar novos Observers a qualquer momento e a única coisa que o Subject dependerá continua sendo uma lista de objetos que implementam Observer. Também podemos substituir Observers em tempo de execução ou remove-los e o Subject continuará se comportando da mesma forma e sem nenhuma alteração na sua estrutura.

Além disso, se alterarmos o Subject ou o Observer nota-se que não há nenhum impacto um no outro, eles apenas precisam continuar implementando as suas interfaces.