**Observer Pattern**

publish ⇔ subscribe

Architecture, Modeling, DesignPattern, Behavioral

```
public final class Observer<T> {
    public interface Subscriber<T> {
        void onSubscribe(T data);
    }

    private final HashMap<String, Subscriber<T>> manager = new HashMap<>();

    public final void subscribe(Subscriber<T> subscriber) {
        if(subscriber == null)
            return;

        manager.put(subscriber.getClass().getName(), subscriber);
    }

    public final void publish(T data) {
        if(data == null)
            return;

        for(Subscriber<T> listener : manager.values()) {
            listener.onSubscribe(data);
        }
    }
}
```

From:
https://ledyx.xyz/ - **Ledyx WIKI**

Permanent link:
**https://ledyx.xyz/doku.php?id=design_pattern:observer_pattern&rev=1612667712**

Last update: **2026/10/24 07:03**