DESIGN PATTERN(S)

C Abstract Factory	**S** Facade	**S** Proxy	
S Adapter	**C** Factory Method	**B** Observer	
S Bridge	**S** Flyweight	**C** Singleton	
C Builder	**B** Interpreter	**B** State	
B Chain of Responsibility	**B** Iterator	**B** Strategy	
B Command	**B** Mediator	**B** Template Method	
S Composite	**B** Memento	**B** Visitor	
S Decorator	**C** Prototype		

Erfinder *Architekt Christopher Alexander ~1977*

Design patterns are proven solution templates for recurring design problems in architecture as well as in software development.
A design pattern usually contains several classes (templates), so a single design pattern is a collection of different templates. E.g. The shape of a single step of stairs **AND** the railing (steelwork or eg woodwork) for the complete stairs **AND** for example the shape for the carpet of each step

Anwendungsfälle

Architecture, Software development, psychology (Archetypes), Sociology (Archetype-Combinations)

Observer Pattern

Subject

+observerCollection

+registerObserver(observer)
+unregisterObserver(observer)
+notifyObservers()

```
notifyObservers()
  for observer in observerCollection
    call observer.notify()
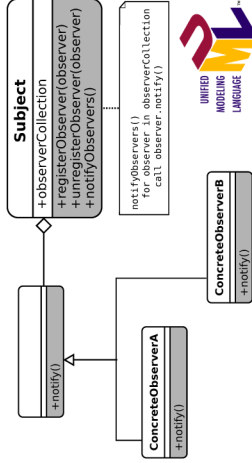```

**ConcreteObserverB**
+notify()

**ConcreteObserverA**
+notify()

## Inventor *Erich Gamma, Richard Helm ... ~1994*

The observer pattern (also called listener pattern) is a design pattern in software development. It belongs to the category of behavior patterns. In the programming language Java it would be **public interface Observer** and it would be used with **"implements Observer"** (needs exercise)

## Use Cases

To transfer changes to objects to a complete structures that dependent on this object, ev. an application with different views