# 观察者模式

1. **基本概念**

### **1、定义：**

观察者（observer）模式的定义：观察者模式定义了一种一对多的依赖关系，让多个观察者对象同时监听某一个主题对象。这个主题对象在状态发生变化的时候，会通知所有观察者对象，使他们能够自动更新自己。

### 2、组成环节：

主题类（Subject）：把所有对观察者对象的引用保存在一个聚集里。每个主题都可以有任何数量的观察者。提供一个接口，可以增加或者删除观察者对象。

抽象观察类（Observer）：是为了所有具体观察者定义一个接口，在得到主题更新时通知自己。

具体主题类（ConcreteSubject）：将有关状态存入具体观察者对象，在具体主题内部改变的时候，给具体观察者发出通知。

具体观察类（ConcreteObserver）：实现抽象观察者角色所要求的更新接口，以便于本身的状态和主题状态相协调。

## 二、go语言中实现

##### 1、首先定义一个observer的抽象观察类

type observer interface {  
 Update()  
}

##### 2、再定义一个subject的主题类

type subject interface {  
 Attach(o observer)  
 Detach(o observer)  
 Notify()  
}

##### 3、创建两个具体主题类，并进行方法的实现

type concretesubjectA struct {  
 os []observer  
}  
  
func (ca \*concretesubjectA) Attach(o observer) {  
 ca.os = append(ca.os, o)  
}  
  
func (ca \*concretesubjectA) Detach(o observer) {  
 for i, oo := range ca.os {  
 if oo == o {  
 ca.os = append(ca.os[:i], ca.os[i+1:]...)  
 }  
 }  
}  
  
func (ca \*concretesubjectA) Notify() {  
 for \_, oo := range ca.os {  
 oo.Update()  
 }  
}  
  
type concretesubjectB struct {  
 os []observer  
}  
  
func (cb \*concretesubjectB) Attach(o observer) {  
 cb.os = append(cb.os, o)  
}  
  
func (cb \*concretesubjectB) Detach(o observer) {  
 for i, oo := range cb.os {  
 if oo == o {  
 cb.os = append(cb.os[:i], cb.os[i+1:]...)  
 }  
 }  
}  
  
func (cb \*concretesubjectB) Notify() {  
 for \_, oo := range cb.os {  
 oo.Update()  
 }  
}

##### 4、创建具体观察类，并进行方法实现

type concreteobserverA struct {  
 name string  
}  
  
func (ca \*concreteobserverA) Update() {  
 fmt.Println(ca.name + " concreteobserverA is update.")  
}  
  
type concreteobserverB struct {  
 name string  
}  
  
func (cb \*concreteobserverB) Update() {  
 fmt.Println(cb.name + " concreteobserverB is update.")  
}

##### 5、通过main函数实现

func main() {  
 csa := &concretesubjectA{}  
 csb := &concretesubjectB{}  
 coa := &concreteobserverA{name: "xiaoming"}  
 cob := &concreteobserverB{name: "xiaohong"}  
  
 csa.Attach(coa)  
 csa.Attach(cob)  
 csa.Notify()  
  
 csb.Attach(coa)  
 csb.Notify()  
}

##### 6、输出结果

**xiaoming concreteobserverA is update.**

**xiaohong concreteobserverB is update.**

**xiaoming concreteobserverA is update.**