



دانشکده فنی و مهندسی
کارشناسی ارشد مهندسی کامپیوتر نرم افزار
گروه مهندسی کامپیوتر و فناوری اطلاعات

معماری نرم افزار

موضوع :

پیاده سازی الگوی طراحی observer

در جاوا اسکریپت

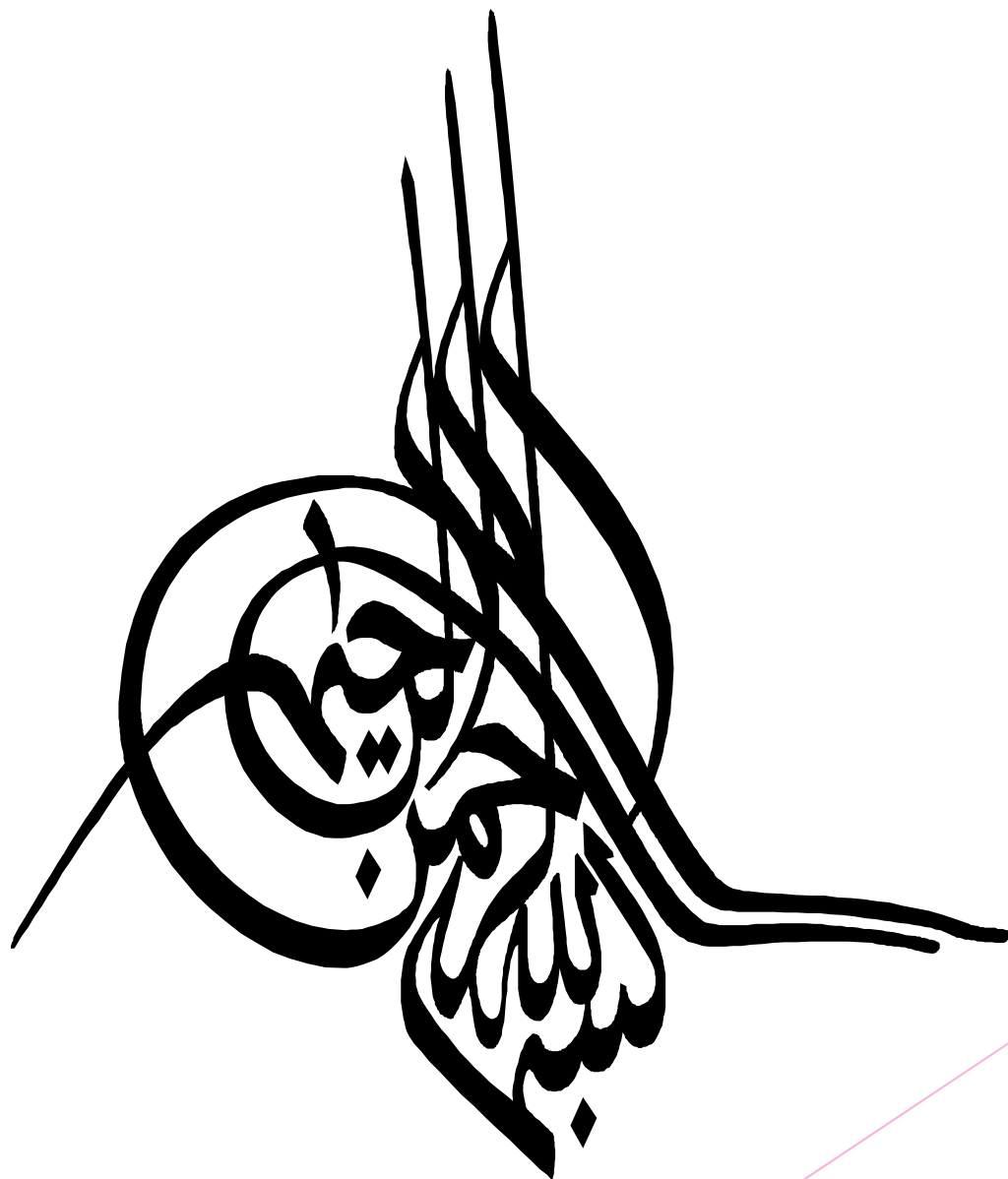
دانشجو::

امیر حسین روشن مهر

استاد راهنما:

دکترسید علی ابراهیمی رضوی

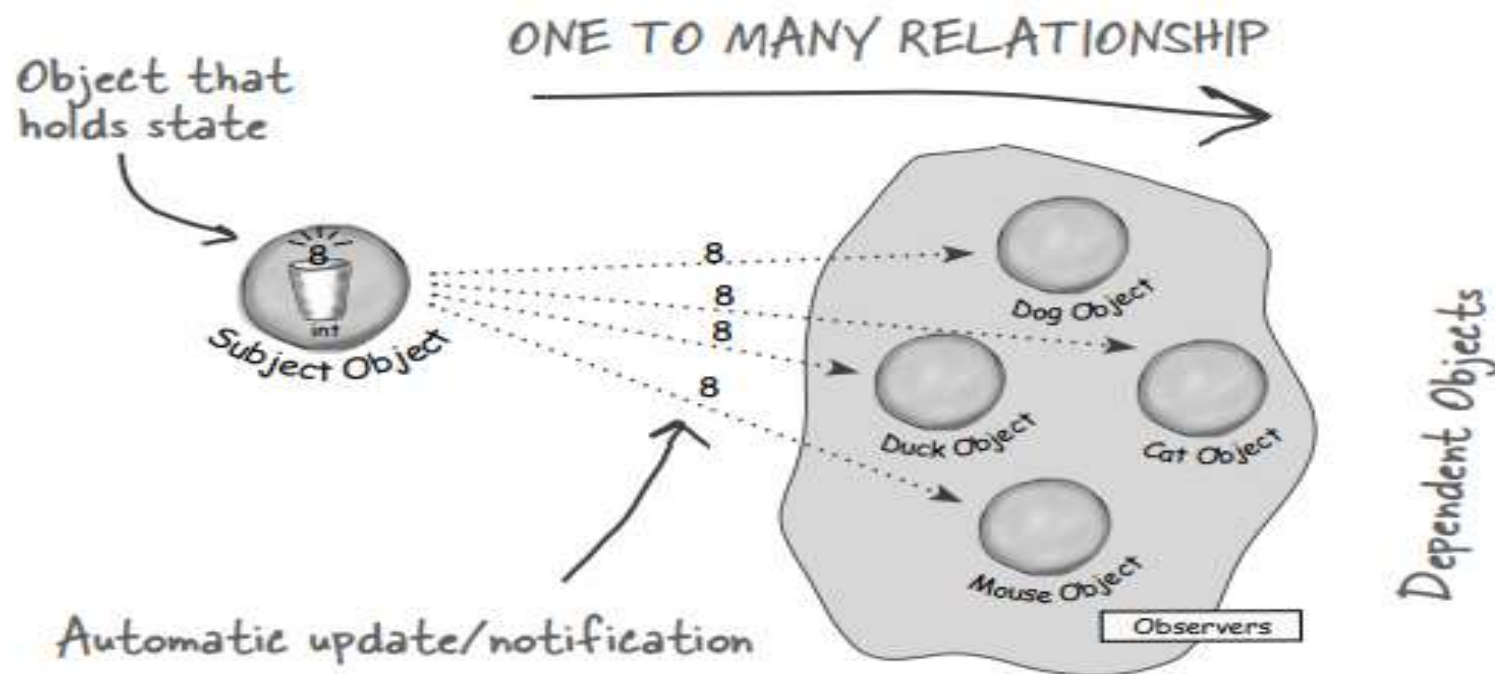
دی ۱۳۹۹



الگوی طراحی observer در جاوا اسکریپت

▶ در هنگام برنامه نویسی شرایطی ایجاد میشود که شما نیاز به الگویی پیدا میکنید که در آن یک تغییر را به تک تک عناصر یک مجموعه اطلاع بدهید بدون اینکه عناصر از هم اطلاع داشته باشند، در ادامه الگویی را به شما معرفی میکنم که این امکان را برای شما فراهم میکند

observer design pattern الگویی است که در آن یک رابطه یک به چند بین object ها برقرار است بطوریکه تغییر ایجاد شده در یک object بطور اتوماتیک به بقیه نیز اطلاع رسانی میشود



▶ در این سناریو به object هایی که گوش میدهند observer یا subscriber میگویند

▶ و به object ای که بقیه به او گوش میدهند observable یا subject میگویند

▶ برای اینکه مفهوم الگوی observer رو کاملاً درک کنید من قصد دارم در یک مثال عملی تعدادی input را به عنوان observer و یک input دیگر را به عنوان observable در نظر بگیرم بطوریکه وقتی observable تغییر کند تمام observer ها بروزرسانی شوند اما قبل از آن باید نحوه ساخت الگوی observer را به شما توضیح بدهم

▶ طبق تعریف سه چیز برای پیاده سازی این الگو نیاز داریم:

▶ یک لیست که مجموعه observerها رو در خود نگه دارد

▶ یک متد که observer را داخل لیست قرار دهد

▶ یک متد که تغییرات را به بقیه observerها اطلاع رسانی کند

▶ پس یک آرایه به نام observers داریم که لیستی از observerها داخل آن قرار میگیرند و یک متد به نام subscribe که هر observer را داخل لیست قرار میدهد

```
class ObserverPattern {  
  constructor(){  
    this.observers = []  
  }  
  subscribe(observer){  
    this.observers.push(observer)  
  }  
}
```

و از طرفی یک متد به نام `notify` لازم داریم که تغییر را به بقیه اطلاع رسانی میکند

```
class ObserverPattern {  
  constructor(){  
    this.observers = []  
  }  
  subscribe(observer){  
    this.observers.push(observer)  
  }  
  notify (change){  
    this.observers.forEach(observer => observer(change))  
  }  
}
```


حال الگوی observer ما به ساده ترین شکل تشکیل شد و من میتوانم از آن استفاده کنم

► **خب به مثال عملی زیر که با الگوی observer پیاده سازی شده است توجه کنید**
وقتی روی دکمه افزودن کلیک میکنید یک input به لیست observer ها
افزوده شده و در صفحه نیز نمایش داده میشود و از طرفی وقتی داخل input
آبی رنگ متنی تایپ میکنید تغییرات آن به تمام input هایی که از طریق دکمه
افزودن به صفحه اضافه شده اند اطلاع رسانی میشود

HTML

CSS

JS

Result

EDIT ON
CODEPEN

```
//Observer Design Pattern
class ObserverPattern {
  constructor(){
    this.observers = []
  }
  subscribe(observer){
    this.observers.push(observer)
  }
  notify (change){
```

افزودن

Resources

1x

0.5x

0.25x

Rerun

		55555	افزودن
55555	55555	55555	
55555	55555	55555	

HTML

```
1 <button type="button" id="add">افزودن</button>
2
3 <input type="text" id="observable">
4
5 <div id="observers"></div>
```

CSS

JS

```
14 //1- Create a instance of ObserverPattern class
15 var observerInstance = new ObserverPattern ()
16
17 //2- Subscribe to observers after clicking on the button
18 $('#add').click(() => {
19     let input = document.createElement('input');
20     input.type = 'text';
21     $('#observers').append(input);
22
23     let observer = value => ( input.value = value )
24     observerInstance.subscribe(observer)
25 })
26
27 //3- Notify input changes to observers
28 $('#observable').keyup( e => observerInstance.notify(e.target.value) )
29
```

HTML

```
1 <button type="button" id="add">الزائد</button>
```

```
2
```

```
3 <input type="text" id="observable"> |
```

```
4
```

```
5 <div id="observers"></div>
```

پایان

This document was created with Win2PDF available at <http://www.win2pdf.com>.
The unregistered version of Win2PDF is for evaluation or non-commercial use only.
This page will not be added after purchasing Win2PDF.