



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

معماری نرم افزار

موضوع :

پیاده سازی الگوی طراحی strategy

در جاوا اسکریپت

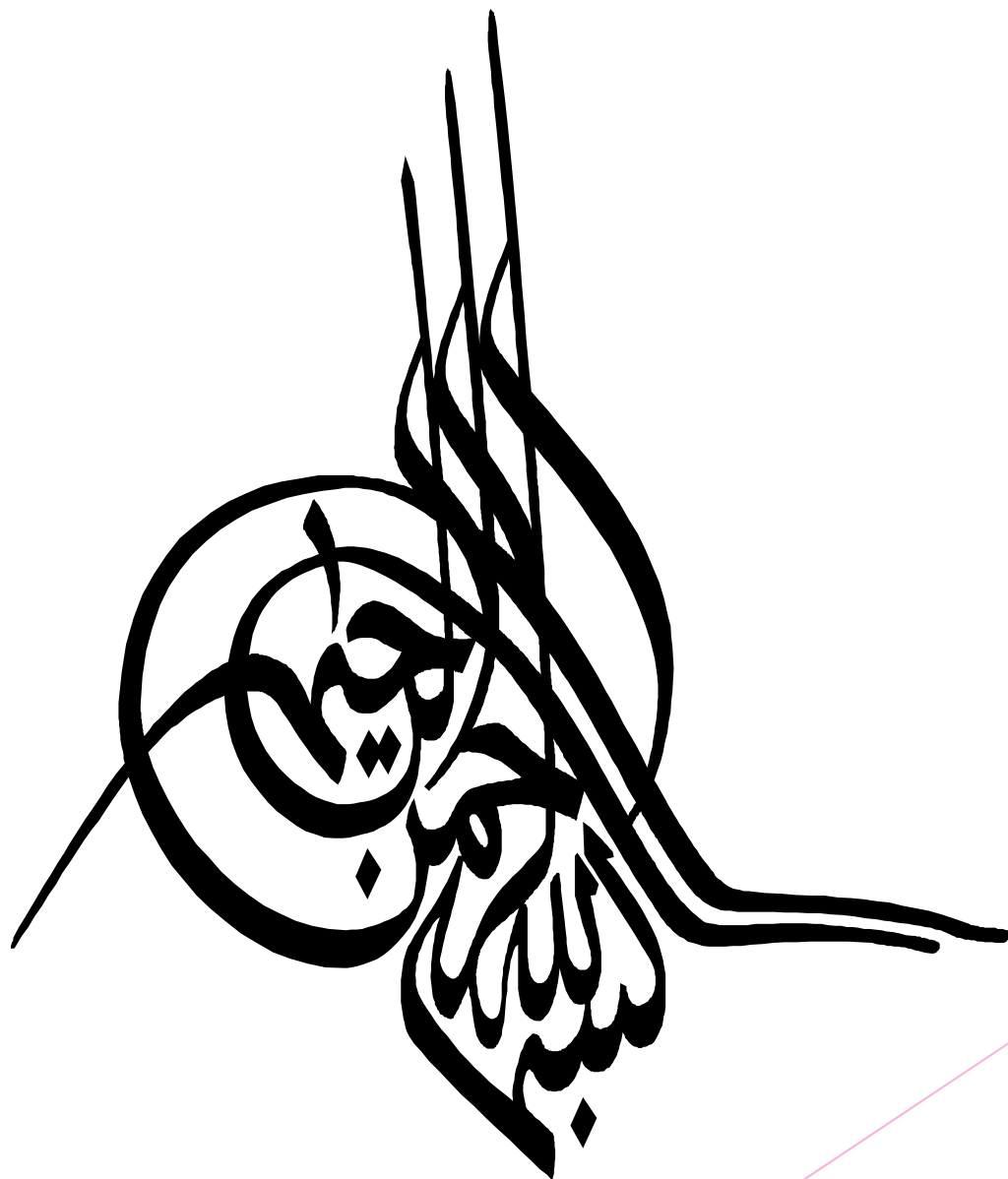
دانشجو::

امیر حسین روشن مهر

استاد راهنما:

دکترسید علی ابراهیمی رضوی

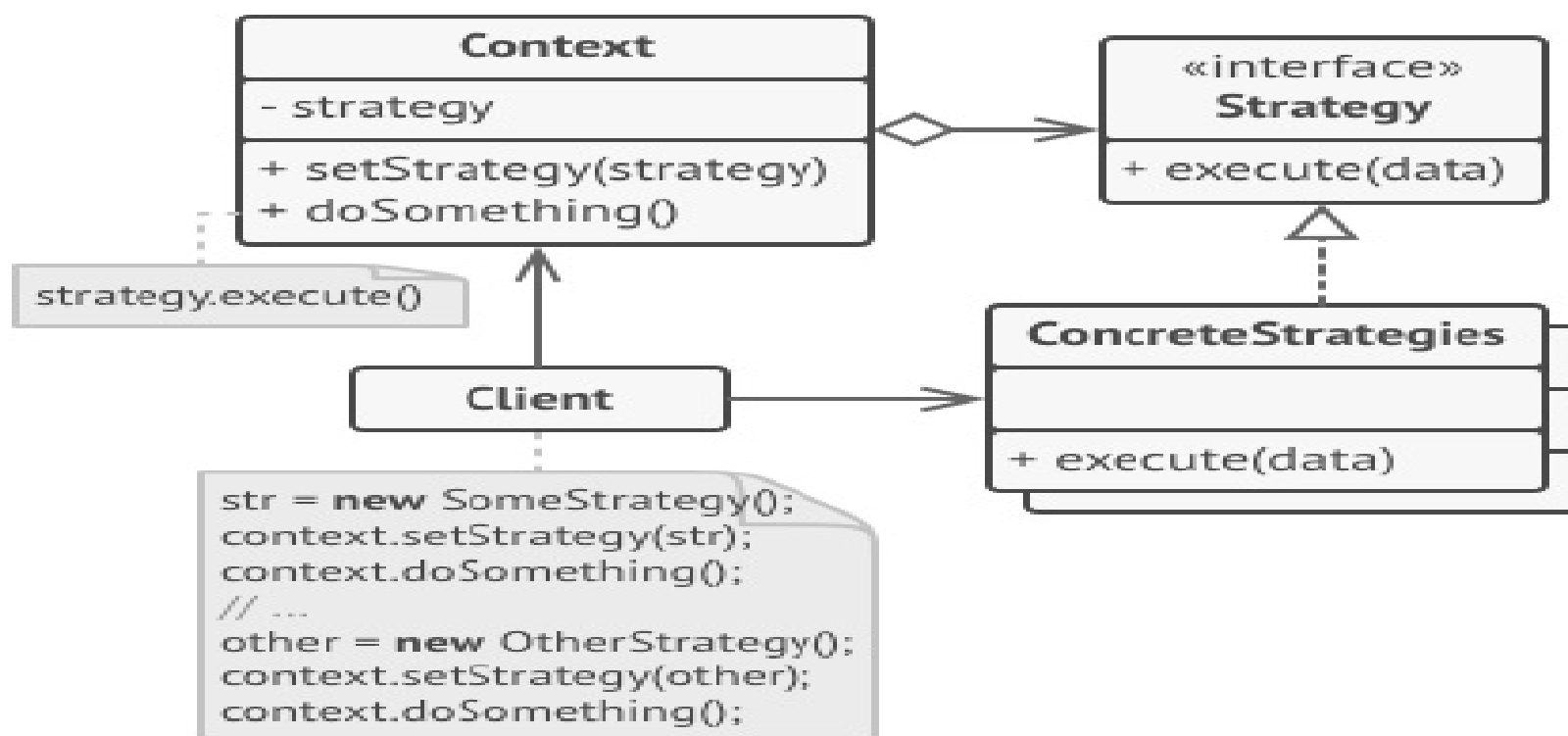
دی ۱۳۹۹



الگوی طراحی strategy در جاوا اسکریپت

► وقتی به زندگی روزمره می اندیشیم به این نتیجه میرسیم که گاهی در زندگی در شرایط مختلف باید برنامه های متفاوتی داشته باشیم و به شکل های مختلفی عمل کنیم اگر شما هم در چنین شرایطی بودید باید بدانید که ندانسته از الگوی طراحی strategy استفاده کرده اید در ادامه به طور کامل به این الگو خواهیم پرداخت.

الگوی طراحی strategy در تعریف به ما میگوید که الگوریتم کارهای مختلفی که یک کلاس باید انجام دهد را میتوان در کلاس های جدا به نام strategy پیاده سازی کرد



▶ اگر به شکل بالا دقت کنید این الگو از سه بخش اصلی تشکیل شده:

▶ context کلاسی که قرار است چند کار مختلف انجام دهد

▶ client کسی که تعیین میکند کدام استراتژی روی context اعمال شود

▶ strategy الگوریتم کاری که قرار است context دهد

▶ نکته بسیار مهم این است که context هیچ اطلاعی از استراتژی ها ندارد ، بلکه این کلاینت است که تعیین میکند کدام استراتژی روی آن اعمال شود

► فرض کنید یک فروشگاه فروش کتاب داریم که با دو رویکرد کتاب های خود را به فروش میرساند، عرضه رایگان و فروش

► در سیستم این فروشگاه با استفاده از الگوی طراحی استراتژی تصمیم گیری میشود که کتاب های رایگان و پولی چگونه به فروش برسند

► برای پیاده سازی این الگو کافیت context و استراتژی های مربوط به آن را شناسایی کنید و بر اساس آنها پیاده سازی را انجام دهید

► مثلاً در مثال فروشگاه اینترنتی خود فروشگاه نقش context و عرضه رایگان و فروش هر کدام استراتژی های ما هستند، پس شروع به پیاده سازی میکنم در ابتدا یک کلاس به نام Shopping تشکیل میدهم که خود دارای دو متد `calculate` و `setStrategy` میباشد

```
class Shopping {  
  constructor () {  
    this.strategy = ''  
  }  
  setStrategy (strategy) {  
    this.strategy = strategy;  
  }  
  calculate () {  
    document.write(product.name + ' Price: ', this.strategy.calculate(product))  
  }  
}
```


و از طرفی دو استراتژی داریم پس دو کلاس با نام های Free و Premium تشکیل می‌دهیم که خود دارای الگوریتم محاسبه قیمت می‌باشند

```
class Free {  
    calculate (product) { return "0$" }  
}  
class Premium {  
    calculate (product) { return (product.kpi * 1.5) + '$' }  
}
```

► برای اجرای الگو یک محصول فرضی ساخته سپس با استفاده از متد `setStrategy` یک استراتژی تعیین میکنیم و در انتها با استفاده از متد `calculate` میتوانیم قیمت مورد نظر را محاسبه کنید

► نتیجه اجرای آن را نیز میتوانید در ادامه مشاهده کنید

JS

Result

EDIT ON
CODEPEN

//CONTEXT

```
class Shopping {  
  constructor () {  
    this.strategy = ''  
  }  
  setStrategy (strategy) {  
    this.strategy = strategy;  
  }  
  calculate () {
```

Javascript design Price: 0\$

Resources

1x

0.5x

0.25x

Rerun

```
var product = { name:'Javascript design', kpi:1.254 };
```

```
// the 2 strategies
```

```
var FREE = new Free();
```

```
var PREMIUM = new Premium();
```

```
var SHOPPING = new Shopping();
```

```
SHOPPING.setStrategy(FREE)
```

```
SHOPPING.calculate(product)
```

HTML

CSS

JS

```
1 var product = { name:'Javascript design', kpi:1.254 };
2
3
4 // the 2 strategies
5 var FREE = new Free();
6 var PREMIUM = new Premium();
7
8 var SHOPPING = new Shopping();
9
10
11 SHOPPING.setStrategy(PREMIUM)
12 SHOPPING.calculate(product)
```

Javascript design Price: 1.881\$

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.