



الگوهای طراحی Design Patterns

دیزاین پترن COMPOSITE

STRUCTURAL DESIGN PATTERNS

یکی از زیرشاخه های الگوهای طراحی، Structural Design Patterns است. این الگوها در مورد چگونگی ترکیب اشیاء و کلاس ها بایکدیگر صحبت میکند.

COMPOSITE DESIGN PATTERN

این پترن یکی از زیرشاخه های الگوهای طراحی از نوع Structural است.

از کلمپوزیت پترن جهت ساخت اشیاء بصورت درخت سلسله مراتبی زمانی که قرار است با گروهی از اشیاء رفتار مشابهی انجام داد استفاده میشود.

میزان استفاده:  متوسط رو به بالا

همچنین در کدهایی مثل لیست پیوندی و گراف میتوانیم از دیزاین پترن کامپوزیت استفاده کنیم.

فرض کنید کلاسی با نام Employee به عنوان کلاس کارمند به صورت زیر داریم.

```
1 internal class Employee
2 {
3     #region Variables
4     public int PersonalId { get; set; }
5     public string FirstName { get; set; }
```

```

5     public string FirstName { get; set; }
6     public string LastName { get; set; }
7     public List EmployeeSubSetList { get; set; }
8     #endregion
9
10    #region Constructors
11    public Employee()
12    {
13        PersonalId = 0;
14        FirstName = "";
15        LastName = "";
16        EmployeeSubSetList = new List();
17    }
18    #endregion
19
20    #region Methods
21    public void AddSubEmployee(Employee employee)
22    {
23        if (EmployeeSubSetList == null)
24            EmployeeSubSetList = new List();
25        EmployeeSubSetList.Add(employee);
26    }
27    public void RemoveSubEmployee(Employee employee)
28    {
29        if (EmployeeSubSetList == null)
30            EmployeeSubSetList = new List();
31
32        if (EmployeeSubSetList.Any(x => x == employee))
33            EmployeeSubSetList.Remove(employee);
34    }
35    #endregion
36
37    #region Helper Methods
38    public override string ToString()
39    {
40        string result = "";
41        result += $"{FirstName}.{LastName} ({PersonalId})";
42        foreach (var employee in EmployeeSubSetList)
43            result += "\n\t" + employee.ToString();
44        return result;
45    }
46    #endregion
47 }

```

در این کلاس یک property لیستی از همین کلاس Employee داریم.

```

1 public List EmployeeSubSetList { get; set; }

```

همچنین دو متد جهت اضافه نمودن کارمند زبردست و حذف کارمند زبردست از این لیست نوشته شده است.

```

1 public void AddSubEmployee(Employee employee)
2 {
3     if (EmployeeSubSetList == null)
4         EmployeeSubSetList = new List();
5     EmployeeSubSetList.Add(employee);
6 }
7 public void RemoveSubEmployee(Employee employee)
8 {
9     if (EmployeeSubSetList == null)
10        EmployeeSubSetList = new List();
11
12    if (EmployeeSubSetList.Any(x => x == employee))
13        EmployeeSubSetList.Remove(employee);
14 }

```

حال نوبت به نوشتن متد main میرسد.

```
1 static void Main(string[] args)
2 {
3     try
4     {
5         var CEO = new Employee()
6         {
7             PersonalId = 1,
8             FirstName = "Mr",
9             LastName = "CEO"
10        };
11        var CTO = new Employee()
12        {
13            PersonalId = 10,
14            FirstName = "Mr",
15            LastName = "CTO"
16        };
17        var clerk = new Employee()
18        {
19            PersonalId = 100,
20            FirstName = "Mr",
21            LastName = "Clerk"
22        };
23        CTO.AddSubEmployee(clerk);
24        CEO.AddSubEmployee(CTO);
25
26        ShowMessage(CEO.ToString());
27        ShowMessage("=====");
28        ShowMessage(CTO.ToString());
29        ShowMessage("=====");
30        ShowMessage(clerk.ToString());
31    }
32    catch (Exception ex)
33    {
34        ShowError(ex.Message);
35    }
36    Console.ReadLine();
37 }
```

در متد main برنامه، یک مدیرعامل با عنوان CEO تعریف کرده ایم. یک مدیر فنی با عنوان CTO تعریف کرده ایم و یک کارمند تحت عنوان Clerk تعریف کرده ایم.

اگر بخواهیم ترتیب این اعضا را مشخص کنیم، به این صورت است که کارمند زیرمجموعه مدیرفنی و مدیرفنی زیرمجموعه مدیرعامل است.

جهت نشان دادن این سلسله مراتب باید به ترتیب کارمند را زیرمجموعه مدیرفنی کنیم و سپس مدیرفنی را زیرمجموعه مدیرعامل.

```
1 CTO.AddSubEmployee(clerk);
2 CEO.AddSubEmployee(CTO);
```

خروجی کد main بصورت زیر است.

```
1 Mr.CEO (1)
2     Mr.CTO (10)
3         Mr.Clerk (100)
4 =====
5 Mr.CTO (10)
6     Mr.Clerk (100)
```



HOLOSEN

هولوسن

با من یاد بگیر

آموزش های بیشتر در وبسایت هولوسن : <https://holosen.net>

برچسبها دیزاین پترن