

بررسی الگوهای طراحی

# Adapter

توسط : محمد حسین زارع

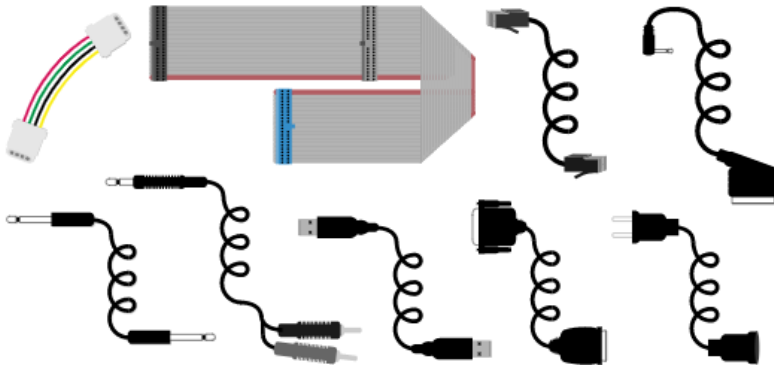
@zare88r: twitter

telegram

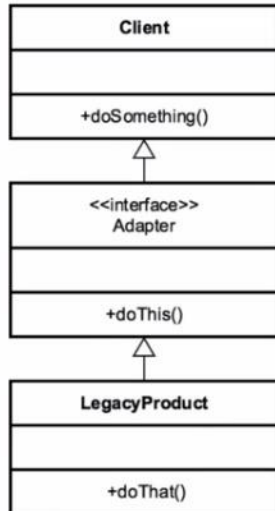
Join Us : @JavaLandCH

# مفاهیم

- استفاده از یک سازگار دهنده بین دو کلاس
- تبدیل یک Interface به یک interface دیگر
- ترجمه کردن درخواست ها
- سه موجودیت : client , adapter , adaptee
- نمونه ها
- Arrays -> Lists



# طراحی



- **Client** محور است
- نمونه کلاس های قدیمی را با اشیاء جدید تطبیق میدهد
- میتوان از **interface** استفاده کرد ولی اجباری نیست

# مثال 1

```
Integer[] array = new Integer[]{15, 537, 32, 3, 464, 1};  
List<Integer> integerList = Arrays.asList(array);
```

# یک نمونه

```
public class EmployeeAdapter extends Employee {  
  
    private EmployeeDto employeeDto ;  
  
    public EmployeeAdapter(EmployeeDto employeeDto) {  
        this.employeeDto = employeeDto;  
    }  
  
    @Override  
    public String getFullName() {  
        return employeeDto.getFirstName()+" "+employeeDto.getLastName();  
    }  
  
    @Override  
    public String getCode() {  
        return employeeDto.getCode();  
    }  
  
    @Override  
    public long getId() {  
        return employeeDto.getId();  
    }  
}
```

# یک نمونه

```
public class User {  
  
    public static void main(String[] args) {  
        EmployeeDto employeeDto = new EmployeeDto();  
        employeeDto.setCode("12");  
        employeeDto.setFirstName("Hosein");  
        employeeDto.setLastName("Zare");  
        employeeDto.setId(12L);  
  
        EmployeeDao employeeDao = new EmployeeDao();  
        employeeDao.saveEmployee(new EmployeeAdapter(employeeDto));  
    }  
}
```

# جمع بندی

- راه حل ساده
- پیاده سازی ساده
- از الگوی طراحی decorator استفاده میکند
- امکان ساخت چند adapter را میدهد