



我将带大家一起去看一看在面向对象编程中出现的23大设计模式和9大设计原则。设计原则是我们开发研究设计模式的目的，而设计模式则是达到设计原则的手段和方法。在记录笔记过程中，会在不同的设计模式中穿插讲解设计原则。

本笔记知识点主要来自于《Head First 设计模式》

行为型模式之五：迭代器模式

定义：

提供一种方法访问顺序访问聚合元素，而又不暴露其内部的表示

设计原则：

- 单一责任原则：一个类应该只有一个引起变化的原因

组成：

- 抽象迭代器(Iterator)：定义了遍历所有元素的接口；
- 具体迭代器：实现了抽象迭代器，纪录遍历的位置；
- 抽象聚集角色／抽象集合角色：此接口定义了产生迭代器对象方法；
- 具体聚集角色：实现了抽象聚集角色接口，并且能根据自身的数据结构特性返回迭代器对象；

看代码：

```
1. 假设客户需要遍历被给出的不同的数据结构的元素，例如ArrayList和Array。单纯的遍历将会调用两次for循环，造成代码的重复。因此可以用迭代器封装具体数据类型，暴露给客户的只是遍历用的迭代器而已。
2. //自定义的迭代器接口
3. public interface Iterator {
4.     boolean hasNext();
5.     Object next();
6. }
7. //具体的迭代器接口
8. public class ListIterator implements Iterator {
9.     List<Object> list;
10.    int position = 0;
11.    public ListIterator(List list) {
12.        this.list = list;
13.    }
14.
15.    public boolean hasNext() {
16.        if(poistion >= list.size() || list.get(position) == null)
17.        {
18.            return false;
19.        }else {
20.            return true;
21.        }
22.
23.    public Object next() {
24.        Object res = list.get(position);
25.        position++;
26.        return res;
27.    }
28. }
29. //抽象的聚集类
30. public interface Menu {
31.     public Iterator createIterator();
32. }
33. //具体的聚集类
34. public class DineMenu implements Menu {
35.     List<MenuItem> list;
36.
37.     public DineMenu(List<MenuItem> list) {
38.         this.list = list;
39.     }
40.
41.     public Iterator createIterator() {
42.         return new ListIterator(list);
43.     }
44.     //其它方法，比如添加menuItem，删除等等
45. }
46.
47. //测试类,MenuItem类就不给出了。
48. public class Test {
49.     public static void main(String[] args) {
50.         List<MenuItem> list = new ArrayList<MenuItem>();
51.         list.add(new MenuItem("Bread", 0.99, "Sweet"));
52.         list.add(new MenuItem("Milk", 1.99, "Diet"));
53.         list.add(new MenuItem("Juice", 1.99, "Organic"));
54.         list.add(new MenuItem("Noddle", 2.99, "Spicy flavor"));
55.         Menu menu = new DineMenu(list);
56.         Iterator iterator = menu.createIterator();
57.         while (iterator.hasNext()) {
58.             System.out.println(iterator.next());
59.         }
60.     }
61. }
62. 事实上对于Java自带的集合类都已经实现了迭代器接口的，因此根本不需要像上面那样自己定义接口，然后实现它。不过对于那些没有实现迭代器接口的，这里可以作为一个例子教你如何实现迭代器遍历元素。
```