

بررسی الگوهای طراحی - Iterator

توسط : محمد حسین زارع

@zare88r: twitter

telegram

Join Us : @JavaLandCH

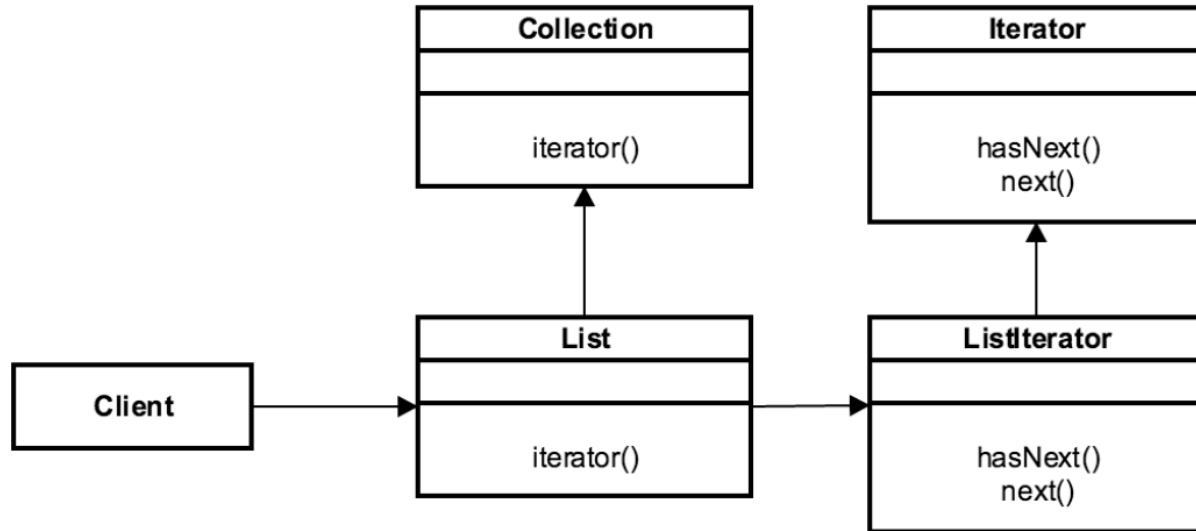
مفاهیم

- پیمایش یک **container**
- پنهان سازی ساختار پشت صحنه
- جداسازی الگوریتم ها
- **Sequential** (دارای توالی)
- نمونه ها :
- **java.util.Iterator**
- **java.util Enumeration**

طراحی

- یک interface برای iterate کردن
- یک متود factory
- **Iterator**
- **Concrete Iterator**

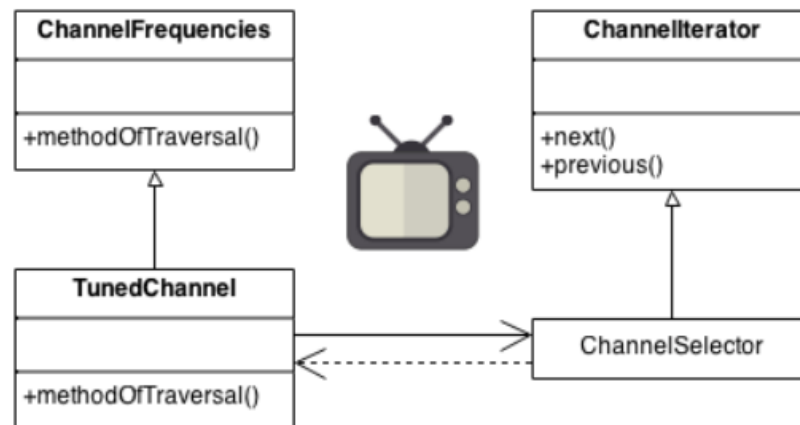
طراحی



مثال 1

```
List<String> strings = Arrays.asList(  
    "Str1", "Str2", "Str3", "Str4");  
  
Iterator<String> iterator = strings.iterator();  
  
while (iterator.hasNext()) {  
    System.out.println(iterator.next());  
}
```

مثال 1



یک نمونه

```
public class FileContentReader implements Iterable<String>
{
    private File file;

    private FileReader fileReader;
    private BufferedReader bufferedReader;
```

یک نمونه

```
public class FileContentReader implements Iterable<String>
{
    private File file;

    private FileReader fileReader;
    private BufferedReader bufferedReader;
```


یک نمونه

```
public class FileContentReader implements Iterable<String>
{
    private File file;

    private FileReader fileReader;
    private BufferedReader bufferedReader;
```

یک نمونه

```
@Override
public Iterator<String> iterator() {

    return new Iterator<String>() {
        String nextLine = null;
        @Override
        public boolean hasNext() {
            //WRITE HAS NEXT METHOD
        }
        @Override
        public String next() {
            //WRITE NEXT LINE
        }
    };
}
```

یک نمونه

```
FileContentReader reader =  
    new FileContentReader("/home/user/checkstyle-rules.xml");  
Iterator<String> iterator = reader.iterator();  
while (iterator.hasNext()) {  
    System.out.println(iterator.next());  
}
```

جمع بندی

- برای پیمایش روی یک مجموعه از اطلاعات کاربرد دارد
- پیچیدگی را از کلاینت پنهان میکند
- دسترسی به index ها ندارد