## 迭代器（Iterator）

### Intent

提供一种顺序访问聚合对象元素的方法，并且不暴露聚合对象的内部表示。

### Class Diagram

* Aggregate 是聚合类，其中 createIterator() 方法可以产生一个 Iterator；
* Iterator 主要定义了 hasNext() 和 next() 方法；
* Client 组合了 Aggregate，为了迭代遍历 Aggregate，也需要组合 Iterator。

### Implementation

public interface Aggregate {
 Iterator createIterator();
}

public class ConcreteAggregate implements Aggregate {

 private Integer[] items;

 public ConcreteAggregate() {
 items = new Integer[10];
 for (int i = 0; i < items.length; i++) {
 items[i] = i;
 }
 }

 @Override
 public Iterator createIterator() {
 return new ConcreteIterator<Integer>(items);
 }
}

public interface Iterator<Item> {

 Item next();

 boolean hasNext();
}

public class ConcreteIterator<Item> implements Iterator {

 private Item[] items;
 private int position = 0;

 public ConcreteIterator(Item[] items) {
 this.items = items;
 }

 @Override
 public Object next() {
 return items[position++];
 }

 @Override
 public boolean hasNext() {
 return position < items.length;
 }
}

public class Client {

 public static void main(String[] args) {
 Aggregate aggregate = new ConcreteAggregate();
 Iterator<Integer> iterator = aggregate.createIterator();
 while (iterator.hasNext()) {
 System.out.println(iterator.next());
 }
 }
}

### JDK

* [java.util.Iterator](http://docs.oracle.com/javase/8/docs/api/java/util/Iterator.html)
* [java.util.Enumeration](http://docs.oracle.com/javase/8/docs/api/java/util/Enumeration.html)