<http://www.cnblogs.com/Alandre/p/3908729.html>

摘要: 摘要: 原创出处: http://www.cnblogs.com/Alandre/ 泥沙砖瓦浆木匠 希望转载，保留摘要，谢谢！

##### 记起2年前，高中生活最多的是老师的批评。谢谢！ -泥沙砖瓦浆木匠

[一.初识Enumeration](http://www.cnblogs.com/Alandre/p/3908729.html#jeffli01)

[二.功能的使用](http://www.cnblogs.com/Alandre/p/3908729.html#jeffli02)

[三.演示Enumeration接口的使用](http://www.cnblogs.com/Alandre/p/3908729.html#jeffli03)

# 一.初识Enumeration

    当我写网络编程的时候，我发现了这个接口。很神奇的是以前都没用过，NetworkInterface.getNetworkInterfaces（）返回的是Enumeration载体这NetworkInterface对象。Enumeration是java.util中的一个接口类，在Enumeration中封装了有关枚举数据集合的方法。Enumeration接口本身不是一个数据结构。但是，对其他数据结构非常重要。 Enumeration接口定义了从一个数据结构得到连续数据的手段。

### 注意：此接口的功能与 Iterator 接口的功能是重复的。此外，Iterator 接口添加了一个可选的移除操作，并使用较短的方法名。新的实现应该优先考虑使用 Iterator 接口而不是 Enumeration 接口。

# 二.功能的使用

## java.util Enumeration<E>

通常用 Enumeration 中的以下两个方法打印向量中的所有元素：

(1) boolean [**hasMoreElements**](mk:@MSITStore:E:%5C?????????%5CvideoForStudy%5C??????????????????????????????12???JAVAEE+??????%5C????????????%5CJDK6API????????????.chm::/java/util/Enumeration.html#hasMoreElements())()  
          测试此枚举是否包含更多的元素。

(2) [E](mk:@MSITStore:E:%5C?????????%5CvideoForStudy%5C??????????????????????????????12???JAVAEE+??????%5C????????????%5CJDK6API????????????.chm::/java/util/Enumeration.html) [**nextElement**](mk:@MSITStore:E:%5C?????????%5CvideoForStudy%5C??????????????????????????????12???JAVAEE+??????%5C????????????%5CJDK6API????????????.chm::/java/util/Enumeration.html#nextElement())()  
          如果此枚举对象至少还有一个可提供的元素，则返回此枚举的下一个元素。

用法一：

vec 是接口实现类 Vector( 集合 ) 的对象， vec.elements() 获得 Vector( 集合 ) 的一个元素。

for(Enumeration e = vec.elements(); e.hasMoreElements();)

{

System.out.println(e.nextElement());

}

用法二：

e is an object that implements the Enumeration interface

[复制代码](javascript:void(0);)

while (e.hasMoreElements()) {

Object o= e.nextElement();

System.out.println(o);

}

[复制代码](javascript:void(0);)

下面举个例子：

[复制代码](javascript:void(0);)

public class TestEnumeration{

public static void main(String[] args){

Vector v = new Vector();

v.addElement("Li");

v.addElement("YYY");

v.addElement("Brown");

Enumeration e = v.elements();

while(e.hasMoreElements()){

String value = (String)e.nextElement();

System.out.println(value);

}

}

}

[复制代码](javascript:void(0);)

你可以看到以下输出

Li

YYY

Brown

# 三.演示Enumeration接口的使用

下面我们就实现这个接口来完成一个小功能：

MyDataStruct类用于实例化一个简单的、可以提供enumeration对象，给使用程序的数据结果对象

class MyDataStruct

{

String[] data;

// 构造器

MyDataStruct()

{

data = new String[4];

data[0] = "zero";

data[1] = "one";

data[2] = "two";

data[3] = "three";

}

// 返回一个enumeration对象给使用程序

Enumeration getEnum()

{

return new MyEnumeration(0, data.length, data);

}

}

MyEnumeration类实现Enumeration接口

[复制代码](javascript:void(0);)

class MyEnumeration implements Enumeration

{

int count; // 计数器

int length; // 存储的数组的长度

Object[] dataArray; // 存储数据数组的引用

// 构造器

MyEnumeration(int count, int length, Object[] dataArray)

{

this.count = count;

this.length = length;

this.dataArray = dataArray;

}

public boolean hasMoreElements()

{

return (count < length);

}

public Object nextElement() {

return dataArray[count++];

}

}

[复制代码](javascript:void(0);)

测试类 MainClass

[复制代码](javascript:void(0);)

class MainClass

{

public static void main(String[] args)

{

// 实例化MyDataStruct类型的对象

MyDataStruct mySataStruct = new MyDataStruct();

// 得到描述myDataStruct类型对象的enumeration对象

Enumeration myEnumeration = mySataStruct.getEnum();

// 使用对象循环显示myDataStruct类型的对象中的每一个元素

while (myEnumeration.hasMoreElements())

System.out.println(myEnumeration.nextElement());

}

}

[复制代码](javascript:void(0);)

你可以看到下面输出：

zero

one

two

three

# 四.感谢知识来源和小结

[一.初识Enumeration](http://www.cnblogs.com/Alandre/p/3908729.html#jeffli01)

[二.功能的使用](http://www.cnblogs.com/Alandre/p/3908729.html#jeffli02)

[三.演示Enumeration接口的使用](http://www.cnblogs.com/Alandre/p/3908729.html#jeffli03)

来自：java基础