

## 71、编程用 JAVA 解析 XML 的方式.

答:用 SAX 方式解析 XML, XML 文件如下:

```
<?xml version="1.0" encoding="gb2312"?>
```

```
<person>
```

```
  <name>王小明</name>
```

```
  <college>信息学院</college>
```

```
  <telephone>6258113</telephone>
```

```
  <notes>男,1955 年生,博士, 95 年调入海南大学</notes>
```

```
</person>
```

事件回调类 SAXHandler.java

```
import java.io.*;
```

```
import java.util.Hashtable;
```

```
import org.xml.sax.*;
```

```
public class SAXHandler extends HandlerBase
```

```
{
```

```
  private Hashtable table = new Hashtable();
```

```
  private String currentElement = null;
```

```
  private String currentValue = null;
```

```
  public void setTable(Hashtable table)
```

```
  {
```

```
    this.table = table;
```

```
  }
```

```
  public Hashtable getTable()
```

```
  {
```

```
    return table;
```

```
  }
```

```
  public void startElement(String tag, AttributeList attrs)
```

```
  throws SAXException
```

```
  {
```

```
    currentElement = tag;
```

```
  }
```

```
  public void characters(char[] ch, int start, int length)
```

```
  throws SAXException
```

```
  {
```

```
    currentValue = new String(ch, start, length);
```

```
  }
```

```
  public void endElement(String name) throws SAXException
```

```
  {
```

```
    if (currentElement.equals(name))
```

```
      table.put(currentElement, currentValue);
```

```
  }
```

```
}
```

JSP 内容显示源码,SaxXml.jsp:

```
<HTML>
```

```
<HEAD>
```

```
<TITLE>剖析 XML 文件 people.xml</TITLE>
</HEAD>
<BODY>
<%@ page errorPage="ErrPage.jsp"
contentType="text/html;charset=GB2312" %>
<%@ page import="java.io.*" %>
<%@ page import="java.util.Hashtable" %>
<%@ page import="org.w3c.dom.*" %>
<%@ page import="org.xml.sax.*" %>
<%@ page import="javax.xml.parsers.SAXParserFactory" %>
<%@ page import="javax.xml.parsers.SAXParser" %>
<%@ page import="SAXHandler" %>
<%
File file = new File("c:\\people.xml");
FileReader reader = new FileReader(file);
Parser parser;
SAXParserFactory spf = SAXParserFactory.newInstance();
SAXParser sp = spf.newSAXParser();
SAXHandler handler = new SAXHandler();
sp.parse(new InputSource(reader), handler);
Hashtable hashTable = handler.getTable();
out.println("<TABLE BORDER=2><CAPTION>教师信息表</CAPTION>");
out.println("<TR><TD>姓名</TD>" + "<TD>" +
    (String)hashTable.get(new String("name")) + "</TD></TR>");
out.println("<TR><TD>学院</TD>" + "<TD>" +
    (String)hashTable.get(new String("college")) + "</TD></TR>");
out.println("<TR><TD>电话</TD>" + "<TD>" +
    (String)hashTable.get(new String("telephone")) + "</TD></TR>");
out.println("<TR><TD>备注</TD>" + "<TD>" +
    (String)hashTable.get(new String("notes")) + "</TD></TR>");
out.println("</TABLE>");
%>
</BODY>
</HTML>
```

## 72、EJB 的基本架构

答:一个 EJB 包括三个部分:

Remote Interface 接口的代码

```
package Beans;
import javax.ejb.EJBObject;
import java.rmi.RemoteException;
public interface Add extends EJBObject
{
    //some method declare
}
```

Home Interface 接口的代码

```
package Beans;
import java.rmi.RemoteException;
import javax.ejb.CreateException;
import javax.ejb.EJBHome;
public interface AddHome extends EJBHome
{
    //some method declare
}
EJB 类的代码
package Beans;
import java.rmi.RemoteException;
import javax.ejb.SessionBean;
import javax.ejb.SessionContext;
public class AddBean implements SessionBean
{
    //some method declare
}
```

### 73、如何校验数字型?

```
var re=/^\d{1,8}$|^\d{1,2}$/;
var str=document.form1.all(i).value;
var r=str.match(re);
if (r==null)
{
    sign=-4;
    break;
}
else{
    document.form1.all(i).value=parseFloat(str);
}
```

### 74、将一个键盘输入的数字转化成中文输出

(例如:输入:1234567 输出:一百二十拾三万四千五百六拾七)

用 java 语言实现,,请编一段程序实现!

```
public class Reader {
    private String strNum;
    private String strNumChFormat;
    private String strNumTemp;
    private int intNumLen;
    private String strBegin;
    public Reader(String strNum) {
        this.strNum = strNum;
    }
    public boolean check(String strNum) {
        boolean valid = false;
```

```
if (strNum.substring(0,1).equals("0")){
    this.strNum = strNum.substring(1);
}
try {
    new Double(strNum);
    valid = true;
}
catch (NumberFormatException ex) {
    System.out.println("Bad number format!");
}
return valid;
}

public void init() {
    strNumChFormat = "";
    intNumLen = strNum.length();
    strNumTemp = strNum;
    strNumTemp = strNumTemp.replace('1', '一');
    strNumTemp = strNumTemp.replace('2', '二');
    strNumTemp = strNumTemp.replace('3', '三');
    strNumTemp = strNumTemp.replace('4', '四');
    strNumTemp = strNumTemp.replace('5', '五');
    strNumTemp = strNumTemp.replace('6', '六');
    strNumTemp = strNumTemp.replace('7', '七');
    strNumTemp = strNumTemp.replace('8', '八');
    strNumTemp = strNumTemp.replace('9', '九');
    strNumTemp = strNumTemp.replace('0', '零');
    strNumTemp = strNumTemp.replace('.', '点');
    strBegin = strNumTemp.substring(0, 1);
}

public String readNum() {
    if (check(strNum)) {
        init();
        try {
            for (int i = 1, j = 1, k = 1; i < intNumLen; i++) {
                if (strNumTemp.charAt(intNumLen - 1) == '零' && i == 1) {
                    strNumChFormat = "位";
                }
                else if (strNumTemp.charAt(intNumLen - i) == '零' && j == 1) {
                    strNumChFormat = "位" + strNumChFormat;
                }
                else if (strNumTemp.charAt(intNumLen - i) == '点') {
                    j = 1;
                    k = 1;
                    strNumChFormat = strNumTemp.charAt(intNumLen - i) + strNumChFormat;
                    continue;
                }
            }
        }
    }
}
```

```
else {
    strNumChFormat = strNumTemp.charAt(intNumLen - i) + strNumChFormat;
}
if (strNumTemp.charAt(intNumLen - i - 1) != '位' &&
    strNumTemp.charAt(intNumLen - i - 1) != '零') {
    if (j == 1 && i < intNumLen) {
        strNumChFormat = '拾' + strNumChFormat;
    }
    else if (j == 2 && i < intNumLen) {
        strNumChFormat = '百' + strNumChFormat;
    }
    else if (j == 3 && i < intNumLen) {
        strNumChFormat = '千' + strNumChFormat;
    }
}
if (j == 4 && i < intNumLen) {
    j = 0;
}
if (k == 4 && i < intNumLen) {
    strNumChFormat = '万' + strNumChFormat;
}
else if (k == 8 && i < intNumLen) {
    k = 0;
    strNumChFormat = '亿' + strNumChFormat;
}
j++;
k++;
}
while (strNumChFormat.indexOf("位") != -1) {
    strNumChFormat = strNumChFormat.replaceAll("位", " ");
}
if (strNumChFormat.substring(0, 2) == "一拾") {
    strNumChFormat = strNumChFormat.substring(1, strNumChFormat.length());
}
if (strNumChFormat.indexOf("点") >= 0) {
    String rebegin = strNumChFormat.substring(0,
        strNumChFormat.indexOf("点"));
    String relast = strNumChFormat.substring(strNumChFormat.indexOf("点"),
        strNumChFormat.length());
    for (int i = 1; i <= relast.length(); i++) {
        relast = relast.replaceAll("拾", "");
        relast = relast.replaceAll("百", "");
        relast = relast.replaceAll("千", "");
        relast = relast.replaceAll("万", "");
        relast = relast.replaceAll("亿", "");
    }
}
```

```
        strNumChFormat = rebegin + relast;
    }
}
catch (ArrayIndexOutOfBoundsException ex) {
    ex.printStackTrace();
}
catch (Exception ex) {
    ex.printStackTrace();
}
int off = strNumChFormat.indexOf("点");
strNumChFormat = strBegin + strNumChFormat.substring(0);
}
else {
    strNumChFormat = "";
}
return strNumChFormat;
}
public static void main(String args[]) {
    try {
        String number = args[0].toString();
        System.out.println("The number is: " + number);
        Reader reader = new Reader(number);
        System.out.println("Output String: " + reader.readNum());
    }
    catch (Exception ex) {
        System.out.println("Please input like that: javac Reader <number>");
    }
}
}
```

**75、设计 4 个线程，其中两个线程每次对 j 增加 1，另外两个线程对 j 每次减少 1。写出程序。**

答:以下程序使用内部类实现线程，对 j 增减的时候没有考虑顺序问题。

```
public class ThreadTest1{
    private int j;
    public static void main(String args[]){
        ThreadTest1 tt=new ThreadTest1();
        Inc inc=tt.new Inc();
        Dec dec=tt.new Dec();
        for(int i=0;i<2;i++){
            Thread t=new Thread(inc);
            t.start();
            t=new Thread(dec);
            t.start();
        }
    }
    private synchronized void inc(){
```

```
j++;
System.out.println(Thread.currentThread().getName()+"-inc:"+j);
}
private synchronized void dec(){
j--;
System.out.println(Thread.currentThread().getName()+"-dec:"+j);
}
class Inc implements Runnable{
public void run(){
for(int i=0;i<100;i++){
inc();
}
}
class Dec implements Runnable{
public void run(){
for(int i=0;i<100;i++){
dec();
}
}
}
```