

Java 数据库编程中查询结果的 表格式输出

陕西财经学院 于波

摘要: 本文较为详细地介绍了在Java数据库编程中, 利用表格输出查询结果的方法; 并对所涉及的类作了简要说明。

关键词: 类 对象 接口

利用Java开发数据库应用系统时, 经常需要在用户界面上显示查询结果。由于SUN公司提供的JDK1.x开发工具包不是可视化的集成开发环境(IDE), 不能象Delphi、VB那样方便地把查询结果在DBGrid等表格中显示出来。因此, 只能靠自己编写代码来实现。在实际应用中, 可以利用Vector、JTable、AbstractTableModel等三个类较好地解决这一问题。以下详细介绍实现方法。

三种类简介

1. 类 Vector

类Vector是Java的历史集合类, 隶属于java.util包。它包装了异构链表和数组杂合体, 具有以下两个特点:

(1) 向量是异构的, 不要求每个元素的类型相同, 向量中可以混合多种对象类型;

(2) 向量是数组杂合体, 因为它们可以在增加元素时动态增大。

其异构性正好符合数据库记录中属性类型不一的特点, 而其动态性也正好符合数据库查询时, 结果集记录个数不定的特点。

类Vector定义如下:

```
public class Vector extends AbstractList  
implements List, Cloneable, Serializable { ... }
```

实现了向量成员的查找、新增、删除等方法。如: add(Object obj)可方便地加入一个对象; get(int index)可方便地得到向量中的一个对象; remove(Object obj)则可方便地删除向量中一个对象。

2. 类 JTable

JTable组件是Swing组件中比较复杂的小件, 隶属于javax.swing包, 它能以二维表的形式显示数据。类JTable定义如下:

```
public class JTable extends JComponent  
implements TableModelListener, Scrollable,  
TableColumnModelListener,  
ListSelectionListener, CellEditorListener, Accessible  
{ ... }
```

类JTable在显示数据时具有以下特点:

(1) 可定制性: 可以定制数据的显示方式和编辑状态;

(2) 异构性: 可以显示不同类型的数据对象, 甚至包括颜色、图标等复杂对象;

(3) 简便性: 可以以缺省方式轻松地建立起一个二维表。

其可定制性能满足不同用户和场合的要求, 异构性也正好符合数据库访问结果集中属性类型不一的特点。类JTable提供了极为丰富的二维表格操作方法, 如设置编辑状态、显示方式、选择行列等, 在此不一一赘述。使用类JTable显示数据之前, 必须根据情况生成定制模型、单元绘制器或单元编辑器。类AbstractListModel用来定制用户自己的数据模型, 这个类在后面要介绍。TableCellRenderer接口用来定制单元绘制器, TableCellEditor接口用来定制单元编辑器, 这两个接口主要用于颜色对象的处理上, 在示例中没有用到, 不做过多说明。

3. 类 AbstractTableModel

类 AbstractTableModel 是一个抽象类, 没有完全实现, 不能实例化, 使用时必须在程序中实现方法。它隶属于 javax.swing.table。类定义如下:

```
public abstract class AbstractTableModel extends Object
```

```
implements TableModel,Serializable {...
```

类 AbstractTableModel 提供了 TableModel 接口中绝大多数方法的缺省实现。TableModel 接口定义了 JTable 的基础数据结构。用户要生成自己的数据模型, 本来可以通过实现 TableModel 接口中所有方法来满足要求, 但管理听众表的功能对于所有数据模型是共同的, 所以在 javax.swing.table 中又定义了类 AbstractTableModel 来处理这个工作。它既管理听众表, 又为生成 TableModelEvents 事件并委托给听众提供了便利。

要想生成一个具体的 TableModel 作为 AbstractTableModel 的子类, 至少必须实现下面三个方法:

```
public int getRowCount();
public int getColumnCount();
public Object getValueAt(int row,int column);
```

至此, 我们可以建立一个简单二维表(5 × 5), 实现方法如下:

```
TableModel dataModel=new AbstractTableModel(){
    public int getColumnCount() {return 5;}
    public int getRowCount() {return 5;}
    public Object getValueAt(int row,int col) {return
new Integer(row*col);}
};
```

```
JTable table=new JTable(dataModel);
JScrollPane scrollpane=new JScrollPane(table);
```

数据库及其连接方法简介

示例采用 Sybase 数据库系统, 数据库存放在数据库服务器中。路径为: D: \ WORKER, 数据库名为: worker.dbf。具有以下字段:

字段名	类型
Wno(职工号)	VARCHAR
Wname(职工名)	VARCHAR
Sex(性别)	VARCHAR
Birthday(出生日期)	DATE
Wage(工资)	FLOAT

要连接此数据库, 需使用 java.sql 包中的类 DriverManager。此类是用于管理 JDBC 驱动程序的实用程序类。它提供了通过驱动程序取得连接、注册、撤消驱动程序, 设置登记和数据库访问登录超时等方法。具体连接方法如下:

第一步: 定位、装入和链接 SybDriver 类;

```
driver="com.sybase.jdbc.SybDriver";
```

```
SybDriver sybdriver=(SybDriver)Class.forName
(driver).newInstance();
```

第二步: 注册 SybDriver 类:

```
DriverManager.registerDriver(sybdriver);
```

第三步: 取得连接(SybConnection)对象引用。

```
user="sa";
```

```
password="";
```

```
url="jdbc:sybase:Tds:202.117.203.114:5000/
WORKER";
```

```
SybConnection connection=(SybConnection)
DriverManager.getConnection
```

```
(Url,user,password);
```

建立完连接后, 即可通过 Statement 接口进行数据库的查询与更改。

实现方法

限于篇幅, 在此只给出核心代码, 包引入、界面处理、变量定义等部分不再介绍。

第一步: 对象声明

```
AbstractTableModel tm;//声明一个类 Abstract-
TableModel 对象
```

```
JTable jg_table;//声明一个类 JTable 对象
```

```
Vector vect;//声明一个向量对象
```

```
JScrollPane jsp;//声明一个滚动杠对象
```

```
String title [] = {"职工号", "职工名", "性别",
"出生日期", "工资"}; //二维表列号
```

第二步: 定制表格

1. 实现抽象类 AbstractTableModel 对象 tm 中的方法

```
vect=new Vector();//实例化向量
```

```
tm=new AbstractTableModel(){
```

```
public int getColumnCount(){
```

```
return title.length;} //取得表格列数
```

```
public int getRowCount(){
```

```
return vect.size();} //取得表格行数
```

```

public Object getValueAt(int row,int column){
    if(!vect.isEmpty())
        rerutn
        ((Vector)vect.elementAt(row)).elementAt(column);
    return null;} // 取得单元格中的属性值
public String getColumnName (int column){
    return title [column];} // 设置表格列名
public void setValueAt(Object value,int row,int
column) {} // 数据模型不可编辑, 该方法设置为空
public Class getColumnClass(int c){
    return getValueAt(0,c).getClass();
} // 取得列所属对象类
public boolean isCellEditable(int row,int
column){
    return false;} // 设置单元格不可编辑, 为缺省
实现};

```

2. 定制表格

```

jg_table=new JTable(tm); // 生成自己的数据模型
jg_table.setToolTipText("显示全部查询结果");//设置
帮助提示

```

```

jg_table.setAutoResizeMode(JTable.AUTO-
RESIZE_OFF); // 设置表格调整尺寸模式

```

```

jg_table.setCellSelectionEnabled(false); // 设置单元
格选择方式

```

```

jg_table.setShowVerticalLines(true); // 设置是否显
示单元格间的分割线

```

```

jg_table.setShowHorizontalLines(true);

```

```

jsp=new JScrollPane(jg_table); // 给表格加上滚动杠

```

第三步: 显示查询结果

(1) 连接数据库: 第二部分已给出。

(2) 数据库查询:

```

Statement stmt=connection.createStatement();
ResultSet rs=stmt.executeQuery("select*from worker");

```

(3) 显示查询结果:

```

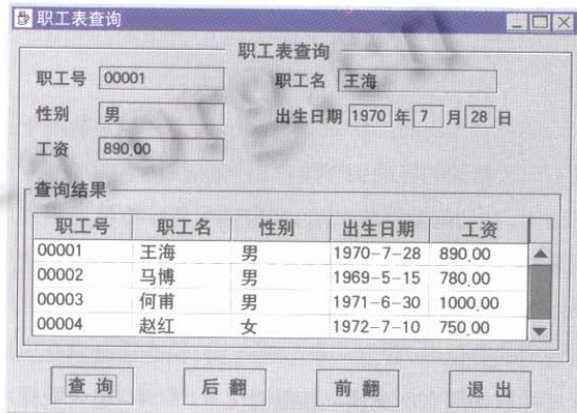
vect.removeAllElements(); // 初始化向量对象
tm.fireTableStructureChanged(); // 更新表格内容
while(rs.next()){
    Vector rec_vector=new Vector();
    // 从结果集中取数据放入向量 rec_vector 中
    rec_vector.addElement(rs.getString(1));
    rec_vector.addElement(rs.getString(2));

```

```

rec_vector.addElement(rs.getString(3));
rec_vector.addElement(rs.getString(4));
rec_vector.addElement(new Float(rs.getFloat(5));
rec.addElement(rec_vector); // 向量 rec_vector 加入
向量 vect 中
}
tm.fireTableStructureChanged(); // 更新表格, 显示
向量 vect 的内容例图如下:

```



若要实现示图中记录前翻、后翻的效果有两种方法:

1. 如果软件环境支持 JDBC2.0, 可直接利用 rs.prevoius() 和 rs.next() 获得记录, 然后通过类 JTextField 中的 setText() 方法, 显示出各个字段值。

2. 如果不支持 JDBC2.0, 则可利用向量 Vector 按行取出 JTable 中数据。自定义一个指针, 用来记录位置。当指针加 1 时, 取出上一行数据放入 Vector 中显示; 指针减 1 时, 取出下一行数据显示。显示方法同上。

需要说明的是: 代码中没有给出捕捉例外的部分, 如 SQLException, 实际应用中必须给出。此外有些系统中, 文本域中的汉字不一定能正确显示, 需借助其他方法实现。■

参考文献

- 1 《Java2 从入门到精通》, [美] John Zukowski 著, 电子工业出版社;
- 2 《Java2 高级开发指南》, 电子工业出版社;