

JTable in Java Swing Example

BY DINESH THAKUR Category: [Swing](#)

The JTable class is another Swing component that has no equivalent in AWT. JTable provides a very flexible possibility to create and display tables. Lets build tables from arrays or vectors of objects or from objects that implement the TableModel interface.

The JTableModel interface defines methods for objects that specify the contents of a table. The class provides an implementation AbstractTableModel the JTableModel predetermined interface. This class is typically extended to provide a custom implementation of model table. The JTable class provides the ability to edit tables. the method setCellEditor () allows an object of the interface is TableCellEditor identified as the editor table cells.

Constructors and methods of the class JTable.

Constructors and Methods	Description
JTable()	Constructs a default JTable Initialized to default data model, default column and selection model.
JTable(int nrows, int ncols)	Constructs a new JTable with <i>nrows</i> and <i>ncols</i> of empty cells using DelaultTableModel.
JTable(Object[][] rowdata, Object[] colnames)	Constructs a JTable to display the data contained in the two-dimensional array, rowdata, having the column names specified in the array, colnames.
JTable(Vector rowvector, Vector colvector)	Constructs a JTable to display the data available in vector of vectors, rowvector, with column names available in colvector.
JTable(TableModel dm)	Constructs a new JTable initialized to specify data model, dm, with the default column model and the default selection model.
JTable(TableModel dm, TableColumnModel cm)	Constructs a JTable initialized to specified data model, dm, also initialized to specified column model, cm, with default selection model.
JTable(TableModel dm, TableColumnModel cm, ListSelectionModel lsm)	Constructs a JTable initialized to specified data model, dm. specified column model, cm, and specified list selection model, lsm.
void addColumn(TableColumn col)	Appends a column to the array of columns.
int getColumnCount()	Returns the number of columns in the column model. (Note: number of columns here may differ from the number of columns In the table model).
int getRowCount()	Returns the number of rows in the table.
int getSelectedColumn()	Returns the index of first selected column or returns -1 if no column is selected.

int getSelectedColumnCount()	Returns the number of columns selected.
int[] getSelectedColumns()	Returns the indices of all columns selected.
int getSelectedRow()	Returns the index of first selected row or returns -1 if no row is selected.
int getSelectedRowCount()	Returns the number of rows selected.
int[] getSelectedRows()	Returns the index of all the rows selected.
Object getValueAt(int r, int c)	Returns the value at rth row and cth column of JTable.
void removeColumn(Table Column tc)	Deletes or removes a column from JTable's array of columns.

```

import java.awt.*;
import javax.swing.*;
import javax.swing.table.*;
public class JavaExampleTableInJApplet extends JApplet
{
    Object[] Dt = new Object[5];
    DefaultTableModel DfltTblModl = new DefaultTableModel();
    JTable Tbl = new JTable(DfltTblModl);
    public void init()
    {
        for(int clmn = 0; clmn < 5; clmn++)
        {
            DfltTblModl.addColumn("Column"+clmn);
        }
        for(int row = 0; row < 10; row++)
        {
            for(int clmn = 0; clmn < 5; clmn++)
            {
                Dt[clmn] = "Cell " + row + "," + clmn;
            }
            DfltTblModl.addRow(Dt);
        }
        getContentPane().add(new JScrollPane(Tbl));
    }
}

```

