

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه دهم

کلاس JTable

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



یکی دیگر از کلاس های موجود در پکیج Swing کلاس JTable می باشد. کلاس JTable برای نمایش داده ها در یک فرم جدولی استفاده می شود. همان طور که هر جدولی دارای سطر و ستون می باشد ، شی ایجاد شده از کلاس JTable ترکیبی از سطر و ستون می باشد. برای درکی از شکل ظاهری اجزای گرافیکی JTable تصویر (۱) را مشاهده کنید:

Symbol	Name	Price
AMZN	Amazon	67 9/16
AOL	America Online	68 3/4
BOUT	About.com	56 3/8
YHOO	Yahoo!	151 1/8

تصویر (۱)

- در تصویر (۱) یک JTable داریم که از تعدادی سطر و ستون تشکیل شده است.

سازنده های پر کاربرد کلاس JTable:


سازنده	کاربرد
JTable()	ایجاد یک جدول با سلول های خالی
JTable(Object[][] rows, Object[] columns)	ایجاد یک جدول با داده های مشخص شده

- دیگه فعلا تئوری رو رها می کنیم میریم سراغ حل مثال! در خلال بررسی مثال ها سایر مفاهیم ،ویژگی ها و متدهای کلاس JTable را بررسی می کنیم.

مثال:

```
package swing_javalike;
import javax.swing.*.*;
public class TableExample {
    JFrame f;
    TableExample(){
        f=new JFrame();
        String data[][]={ {"101", "hasan", "670000"},
                           {"102", "jafar", "780000"},
                           {"103", "maryam", "700000"}};
        String column[]={"ID", "mina", "SALARY"};
        JTable jt=new JTable(data,column);
        jt.setBounds(30,40,200,300);
        JScrollPane sp=new JScrollPane(jt);
        f.add(sp);
        f.setSize(300,200);
        f.setVisible(true);
    }
    public static void main(String[] args) {
        new TableExample();
    }
}
```

خروجی: تصویر (۲)



ID	mina	SALARY
101	hasan	670000
102	jafar	780000
103	maryam	700000

تصویر (۲)

```
String data[][]={ {"101","hasan","670000"},  
{"102","jafar","780000"},  
{"103","maryam","700000"}};
```

- یک آرایه دو بعدی تعریف کرده و داده های خود را درون آن میریزیم. این آرایه دوبعدی قراره داده های جدول `JTable` ما را تشکیل خواهد داد.

```
String column[]={ "ID","mina","SALARY"};
```

- یک آرایه یک بعدی تعریف کرده و داده هایی را درون آن می ریزیم، داده های این آرایه عنوان های ستون جدول `JTable` ما را تشکیل خواهد داد.
- پس تا اینجا داده های جدول را درون یک آرایه دو بعدی و داده های عناوین ستون جدول را درون یک آرایه یک بعدی ریختیم.

```
JTable jt=new JTable(data,column);
```

- برای ایجاد جدول در برنامه از کلاس `JTable` شی ایجاد کرده ایم. درون سازنده این کلاس آرایه دوبعدی `data` که حاوی داده های جدول و آرایه یک بعدی `column` که حاوی عنوان های ستون جدول می باشد را قرار داده ایم.

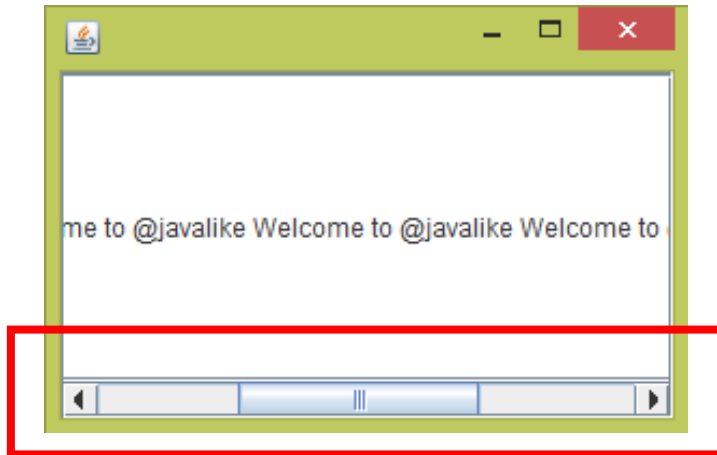
```
jt.setBounds(30,40,200,300
```

- تعیین مختصات و ابعاد جدول خود در فریم

```
JScrollPane sp=new JScrollPane(jt);
```

کلاس JScrollPane :

برای افزودن قابلیت اسکرول به اجزای گرافیکی ، مدیریت نمایش، افقی یا عمودی بودن اسکرول بارها و نمایش عنوان های سطر و ستون استفاده می شود. در تصویر (۳) یک اسکرول بار افزوده شده به یک `TextField` را مشاهده می کنید:



تصویر (۳)

- برای افزودن یک اسکرول بار و تعیین عنوان سطر و ستون و.. کافیست از کلاس `JScrollPane` شی ایجاد کنیم و درون پارامتر سازنده کلاس شی از اجزای گرافیکی مربوطه قرار دهیم.
- برای افزودن اسکرول بار و تعیین ستون برای جدول خود از `JScrollPane` استفاده می کنیم. روش کار هم به این صورته که ابتدا از کلاس `JTable` شی ایجاد می کنیم و سی ایجاد شده را درون پارامتر سازنده کلاس `JScrollPane` قرار می دهیم:

```
JTable jt=new JTable(data,column);
JScrollPane sp=new JScrollPane(jt);
```

پس هنگام ایجاد جدول `JTable` در برنامه حتما باید از کلاس `JScrollPane` استفاده کنیم.

```
1. f.add(sp);
2. f.setSize(300,200);
3. f.setVisible(true);
```

۱. افزودن شی از `JScrollPane` به فریم خود

۲. تعیین ابعاد فریم

۳. برای نمایش فریم و تمام اجزای گرافیکی برنامه کاربرد دارد

مثال از `JTable` و استفاده از اینترفیس `ListSelectionListener`:

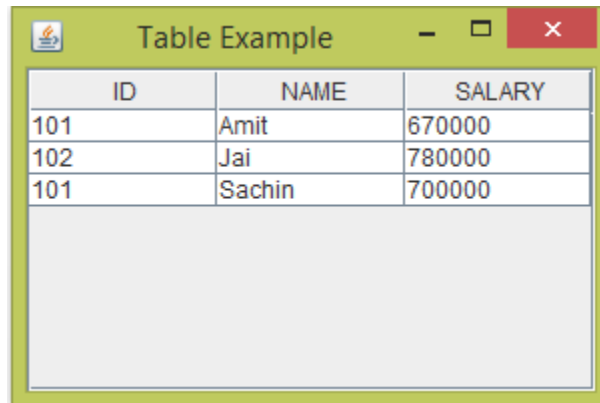
```
package javalike;

import javax.swing.*;
import javax.swing.event.*;

public class TableExample {
    public static void main(String[] a) {
        JFrame f = new JFrame("Table Example");
        String data[][] = { { "101", "Amit", "670000" },
                            { "102", "Jai", "780000" }, { "101", "Sachin", "700000" }
        };

        String column[] = { "ID", "NAME", "SALARY" };
        final JTable jt = new JTable(data, column);
        jt.setCellSelectionEnabled(true);
        ListSelectionModel select = jt.getSelectionModel();
        select.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
        select.addListSelectionListener(new ListSelectionListener() {
            public void valueChanged(ListSelectionEvent e) {
                String Data = null;
                int[] row = jt.getSelectedRows();
                int[] columns = jt.getSelectedColumns();
                for (int i = 0; i < row.length; i++) {
                    for (int j = 0; j < columns.length; j++) {
                        Data = (String) jt.getValueAt(row[i],
                columns[j]);
                    }
                }
                System.out.println("Table element selected is: " + Data);
            }
        });
        JScrollPane sp = new JScrollPane(jt);
        f.add(sp);
        f.setSize(300, 200);
        f.setVisible(true);
    }
}
```

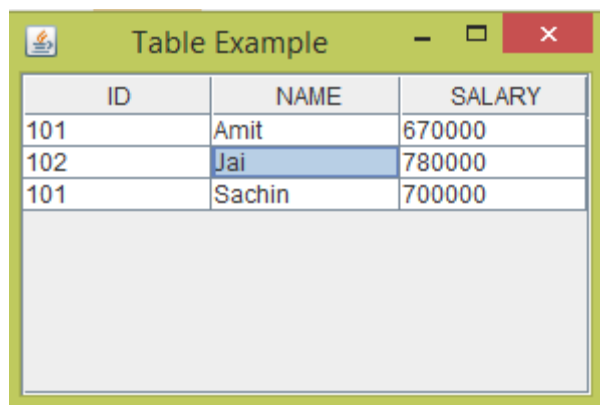
خروجی: تصویر (۴)



ID	NAME	SALARY
101	Amit	670000
102	Jai	780000
101	Sachin	700000

تصویر (۴)

اگر هر عنصر درون سلول های جدول را انتخاب کنید ، مقدار عنصر انتخاب شده در کنسول نمایش داده می شود، مثلا اگر ما عنصر **Jai** موجود در سلول جدول را انتخاب کنیم خروجی بصورت زیر خواهد بود: تصویر (۵)



ID	NAME	SALARY
101	Amit	670000
102	Jai	780000
101	Sachin	700000

تصویر (۵)

خروجی در کنسول:

```
Table element selected is: Jai
Table element selected is: Jai
```

توضیحات:

```
jt.setCellSelectionEnabled(true);
```

این متد قابلیت انتخاب سطر و ستون سلول های جدول را تعیین می کند.

```

ListSelectionModel select = jt.getSelectionModel();
select.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
select.addListSelectionListener(new ListSelectionListener() {
    public void valueChanged(ListSelectionEvent e) {
        String Data = null;
        int[] row = jt.getSelectedRows();
        int[] columns = jt.getSelectedColumns();
        for (int i = 0; i < row.length; i++) {
            for (int j = 0; j < columns.length; j++) {
                Data = (String) jt.getValueAt(row[i],
columns[j]);
            }
        }
        System.out.println("Table element selected is: " + Data);
    }
});

```

- این دستورات برای این است که وقتی یکی از سلول های جدول انتخاب می شود عمل خاصی رخ دهد. به این صورت که وقتی ما یکی از سلول های جدول را انتخاب می کنیم یک رویداد رخ میدهد که متد `valueChanged` آن را دریافت می کند و دستورات درون بدنه خود را اجرا می کند. که در این جا مقدار سلولی که انتخاب شده درون کنسول چاپ می شود.
- متد `getSelectionModel` جریانی از مدل انتخاب را برمیگرداند.
- جریانی از مدل های انتخاب شده از جدول را درون شی از کلاس `ListSelectionModel` می ریزیم.
- کلاس `ListSelectionModel` مجموعه ای از مدل های انتخاب را در خود جا می دهد.

```
select.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
```

- در این دستور در هر زمان تنها یک ایندکس لیست را می توانیم انتخاب کنیم.
- با متد `addListSelectionListener` شی از اینترفیس `ListSelectionListener` به `ListSelectionModel` اضافه می کنیم.
- متد `valueChanged` در برابر تغییر انتخاب واکنش نشان می دهد. این متد رویداد مربوط به انتخاب را دریافت می کند. خب این متد درون اینترفیس `ListSelectionListener` قرار دارد، پس برای استفاده از این متد باید از اینترفیس `ListSelectionListener` شی ایجاد کنیم.
- میدونم از توضیحات بالا الان گیج شدید!!! کلا توضیحات رو بی خیال! در کل اگر خواستیم با انتخاب سلول های جدول واکنش خاصی داده شود از دستورات زیر استفاده می کنیم:


```
final JTable jt = new JTable(data, column);  
  
    ایجاد جدول مربوطه  
  
jt.setCellSelectionEnabled(true);  
  
    تنظیم قابلیت انتخاب سلول های جدول  
  
ListSelectionModel select = jt.getSelectionModel();  
  
    دریافت جریانی از مدل های انتخاب از جدول مربوطه  
  
select.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);  
  
    تعیین مدل انتخاب که در اینجا مدل تنها یک بار انتخاب در هر لحظه  
  
select.addListSelectionListener(new ListSelectionListener()  
{  
  
    افزودن شی از اینترفیس ListSelectionListener به شی select  
    public void valueChanged(ListSelectionEvent e) {  
        واکنش در برابر تغییر انتخاب  
    }  
});
```

- من توضیحات رو زیاد برای یادگیری گرافیک در جاوا نمی پسندم!!! بهترین راه یادگیری جاوا مخصوصا گرافیک اینه که کدها رو با چشمتون خوب ببینید بعد اجرا کنید بعد برا خودتون تغییر بدید کد هارو اینجوری عملی بهتر یادمیگیرید تا توضیحات!!! این کد بالا هم اگر با توضیحات براتون روشن نشد نگران نباشید!! در ایکلیپس یا هر IDE دیگه اجرا کنید و تغییر بدید کاربردشون دستتون میاد.

پیروز و موفق باشید

سایت آموزش زبان جاوا به زبان ساده، آسان و شیرین!!!

www.JAVAPro.ir

آموزش جاوا SE را با تجربه شخصی و به زبان خودمونی یاد بگیرید!!!!

بازدید از کانال

بازدید از سایت

هر روز مفاهیم و مثال های جدید به سایت اضافه می شود برای اطلاع از مطالب جدید روی سایت عضو کانال شوید.

دخل و تصرف ، ویرایش و کپی زدن تمامی آموزش های جاوا لایک به دور از اخلاق حرفه ای ست و حرام می باشد.