

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه دوازدهم

JMenuBar, JMenu and JMenuItem

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



کلاس JMenuBar

کلاس JMenuBar برای نمایش نوار منو در یک پنجره یا فریم برنامه استفاده می شود. شی ایجاد شده از کلاس JMenuBar خود می تواند شامل چندین منو شود.

کلاس JMenu

هدف از کلاس JMenu نمایش رو به پایین لیست آیتم های یک منو می باشد.

کلاس JMenuItem

بعد از این که JMenuBar یا نوار منو خود را در فریم ایجاد کردیم و بعد یک Menu به نوار منوی خود اضافه کردیم حالا می رسیم به مرحله اضافه کردن آیتم های خود به منو که با انتخاب منوی مربوطه لیست آیتم ها رو به پایین نمایش داده شوند.

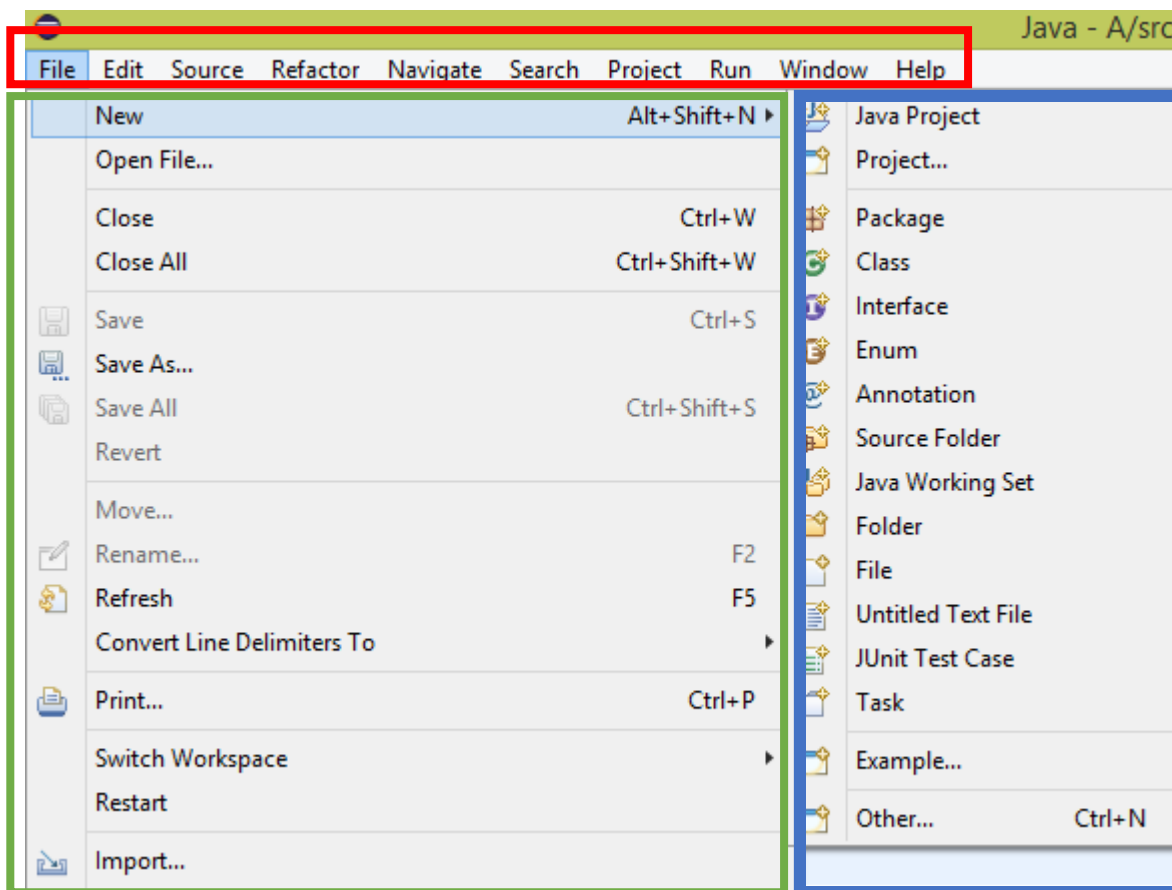
برای ایجاد آیتم های یک منو از کلاس JMenuItem استفاده می کنیم.

برای درک این این سه کلاس تصویر (۱) را مشاهده کنید:



تصویر (۱)

- یک JMenuBar می تواند از چندین JMenu تشکیل شود.
- یک JMenu می تواند از چندین JMenuItem تشکیل شود.
- همچنین می توانیم به یک JMenu ، JMenu دیگری به عنوان یکی از آیتم های آن اضافه کنیم. تصویر (۲)



تصویر (۲)

در تصویر (۲):

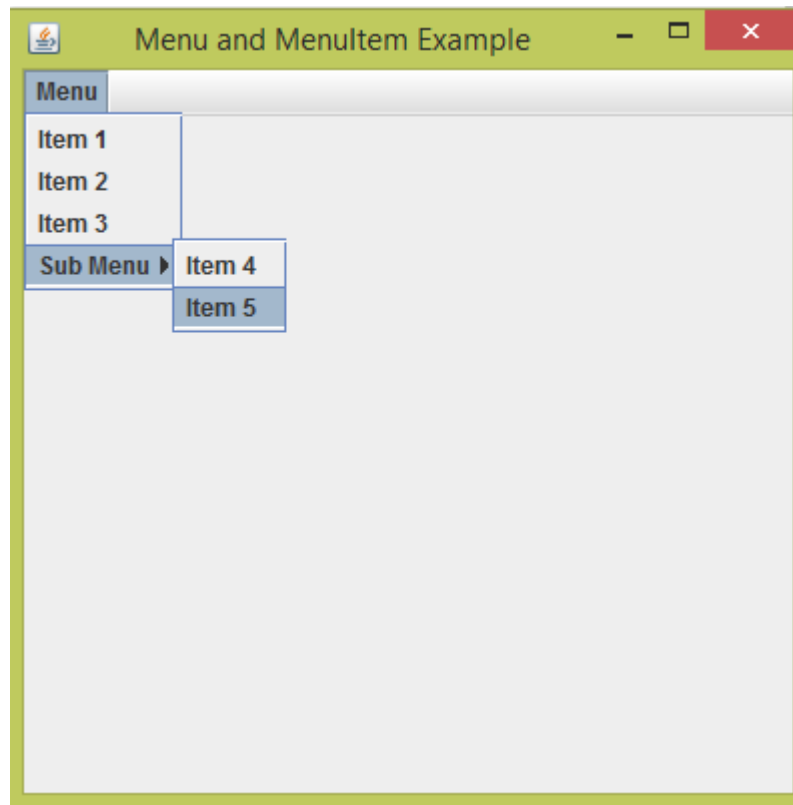
۱. بخش قرمز رنگ: JMenuBar ما که شامل چندین JMenu است.
۲. بخش سبز رنگ: آیتم های منو File که بصورت لیست رو به پایین نمایش داده شده اند.
۳. بخش آبی رنگ: آیتم های منو New را شامل می شود. منو New خود یک آیتم از منو File می باشد. پس می توانیم یک JMenu را به عنوان یک آیتم از JMenu دیگر در نظر بگیریم.

خب بریم سراغ بررسی این مفاهیم با حل مثال:

مثال:

```
package javalike;
import javax.swing.*;
class MenuExample
{
    JMenu menu, submenu;
    JMenuItem i1, i2, i3, i4, i5;
    MenuExample(){
        JFrame f= new JFrame("Menu and MenuItem Example");
        JMenuBar mb=new JMenuBar();
        menu=new JMenu("Menu");
        submenu=new JMenu("Sub Menu");
        i1=new JMenuItem("Item 1");
        i2=new JMenuItem("Item 2");
        i3=new JMenuItem("Item 3");
        i4=new JMenuItem("Item 4");
        i5=new JMenuItem("Item 5");
        menu.add(i1); menu.add(i2); menu.add(i3);
        submenu.add(i4); submenu.add(i5);
        menu.add(submenu);
        mb.add(menu);
        f.setJMenuBar(mb);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String args[])
    {
        new MenuExample();
    }
}
```

خروجی: تصویر (۳)



تصویر (۳)

توضیحات:

```
JMenu menu, submenu;
JMenuItem i1, i2, i3, i4, i5;
```

- دو JMenu و پنج JMenuItem که آیتم های منوهای ما را تشکیل می دهند را تعریف کرده ایم.

```
JMenuBar mb=new JMenuBar();
```

- یک منو بار برای چسباندن منوهای خود به آن تعریف کرده و از آن شی ایجاد کرده ایم. پس فونداسیون و پایه منوهای ما را منو بار تشکیل می دهد.

```
menu=new JMenu("Menu");
submenu=new JMenu("Sub Menu");
```

با صدا زدن سازنده های کلاس JMenu دو شی به نام های menu و submenu ایجاد می کنیم.

```
i1=new JMenuItem("Item 1");
i2=new JMenuItem("Item 2");
i3=new JMenuItem("Item 3");
i4=new JMenuItem("Item 4");
```

```
i5=new JMenuItem("Item 5");
```

- پنج شی از JMenuItem ایجاد می کنیم. این پنج شی آیتم های منوهای مربوطه را تشکیل می دهند.

```
menu.add(i1); menu.add(i2); menu.add(i3);
```

- آیتم های یک تا سه را به شی menu اضافه می کنیم.

```
submenu.add(i4); submenu.add(i5);
```

- آیتم های چهار و پنج را به شی submenu اضافه می کنیم.

```
menu.add(submenu);
```

- انگار زور شی menu از نوع کلاس JMenuItem بیشتر از شی submenu از نوع کلاس JMenuItem می باشد. به عبارتی منو submenu جز زیر منو یا آیتمی از منو menu شده است. تصویر (۳)

```
mb.add(menu);
```

- در پایان شی menu از نوع کلاس JMenuItem را به منو بار خود اضافه می کنیم.

```
f.setJMenuBar(mb);
```

- با این متد منو بار خود را به فریم اضافه می کنیم.

مثال:

```
package javalike;

import javax.swing.*;
import java.awt.event.*;

public class MenuExample implements ActionListener {
    JFrame f;
    JMenuBar mb;
    JMenuItem file, edit, help;
    JMenuItem cut, copy, paste, selectAll;
    JTextArea ta;

    MenuExample() {
        f = new JFrame();
        cut = new JMenuItem("cut");
        copy = new JMenuItem("copy");
        paste = new JMenuItem("paste");
        selectAll = new JMenuItem("selectAll");
        cut.addActionListener(this);
        copy.addActionListener(this);
    }
}
```

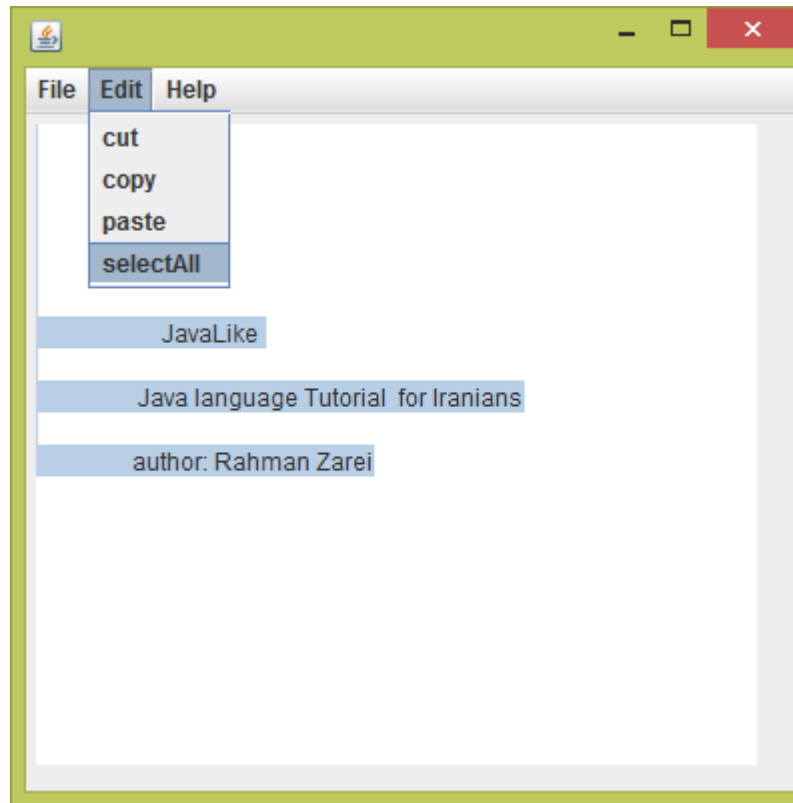
```
paste.addActionListener(this);
selectAll.addActionListener(this);
mb = new JMenuBar();
file = new JMenu("File");
edit = new JMenu("Edit");
help = new JMenu("Help");
edit.add(cut);
edit.add(copy);
edit.add(paste);
edit.add(selectAll);
mb.add(file);
mb.add(edit);
mb.add(help);
ta = new JTextArea();
ta.setBounds(5, 5, 360, 320);

f.add(ta);
f.setJMenuBar(mb);
f.setLayout(null);
f.setSize(400, 400);
f.setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    if (e.getSource() == cut)
        ta.cut();
    if (e.getSource() == paste)
        ta.paste();
    if (e.getSource() == copy)
        ta.copy();
    if (e.getSource() == selectAll)
        ta.selectAll();
}

public static void main(String[] args) {
    new MenuExample();
}
}
```

خروجی: تصویر (۴)



تصویر(۴)

- یک برنامه ویرایشگر ساده را مشاهده می کنید که در بخش منو `Edit` آیتم `selectAll` را انتخاب کرده ایم که موجب انتخاب همه متن های درون ویرایشگر شده است.

توضیحات:

```
public class MenuExample implements ActionListener {
```

چون قصد داریم رویدادهای مربوط به کلیک کردن روی آیتم های منو را دریافت کنیم اینترفیس `ActionListener` را در کلاس خود `implements` می کنیم.

```
1. JFrame f;
2. JMenuBar mb;
3. JMenu file, edit, help;
4. JMenuItem cut, copy, paste, selectAll;
5. JTextArea ta;
```

۱. تعریف یک شی از نوع `JFrame`

۲. تعریف یک شی از نوع کلاس `JMenuBar` که منو بار ما را تشکیل می دهد.

۳. تعریف چند شی به عنوان منوهای برنامه

۴. تعریف چند شی به عنوان آیتم های منوهای برنامه

۵. تعریف یک شی از نوع کلاس JTextArea که محیط ویرایشگر ما را تشکیل می دهد.

```
cut = new JMenuItem("cut");
copy = new JMenuItem("copy");
paste = new JMenuItem("paste");
selectAll = new JMenuItem("selectAll");
```

- آیتم های خود را ایجاد می کنیم.

```
cut.addActionListener(this);
copy.addActionListener(this);
paste.addActionListener(this);
selectAll.addActionListener(this);
```

- رویدادهای ActionListener را به آیتم های خود اضافه می کنیم. در صورت کلیک کردن روی هر ایتام واکنش نسبت داده شده به هر ایتام اجرا می شود.

```
mb = new JMenuBar();
```

- منو بار خود را ایجاد می کنیم.

```
file = new JMenu("File");
edit = new JMenu("Edit");
help = new JMenu("Help");
```

- منوهای خود را می سازیم.

```
edit.add(cut);
edit.add(copy);
edit.add(paste);
edit.add(selectAll);
```

- آیتم های خود را به منو Edit اضافه می کنیم.

```
mb.add(file);
mb.add(edit);
mb.add(help);
```

- حال منوهای خود را به منو بار اضافه می کنیم.

```
ta = new JTextArea();
ta.setBounds(5, 5, 360, 320);
```

- ناحیه متنی خود را ایجاد کرده و مختصات و ابعاد آن را تعیین می کنیم.

```
f.add(ta);
```

- **TextArea** خود را به فریم اضافه می کنیم.

```
f.setJMenuBar(mb);
```

- منو بار خود را به فریم اضافه می کنیم.

```
public void actionPerformed(ActionEvent e) {
    if (e.getSource() == cut)
        ta.cut();
    if (e.getSource() == paste)
        ta.paste();
    if (e.getSource() == copy)
        ta.copy();
    if (e.getSource() == selectAll)
        ta.selectAll();
}
```

- متد **actionPerformed** رویدادهای رخ داده مثل کلیک کردن روی ایتمی از منو یا دکمه باتن و... را دریافت می کند.

```
if (e.getSource() == cut)
    ta.cut();
```

- در این دستور گفته شده اگر رویداد رخ داده برابر با آیتم **cut** (cut یکی از آیتم های منو **Edit** می باشد) باشد، متد **cut** را صدا بزن. در صورت صدا زدن متد **cut** اگر متنی درون **TextArea** بود **cut** می شود. متد **cut** یکی از متدهای درون کلاس **JTextArea** می باشد.

```
if (e.getSource() == paste)
    ta.paste();
```

- در این دستور نیز اگر آیتم **paste** موجود در منو **Edit** را انتخاب کنیم، متد **paste** صدا زده می شود و عملاً پیست کردن انجام می شود.

```
if (e.getSource() == copy)
    ta.copy();
```

- در این دستور نیز اگر آیتم **copy** موجود در منو **Edit** را انتخاب کنیم، متد **copy** که عمل کپی کردن را انجام میدهد صدا زده می شود.

```
if (e.getSource() == selectAll)
    ta.selectAll();
```

- اگر آیتم `selectAll` موجود در منو `Edit` را انتخاب کنیم ، متد `selectAll` صدا زده شده و تمام متن های درون `TextArea` انتخاب می شوند.

نکته : متدهای `paste.copy.cut` و `selectAll` متدهای آماده موجود در کلاس `JTextArea` می باشند. که ما می توانیم از آنها به راحتی استفاده کنیم.

دوستان این جلسات آموزشی برای آشنایی و استارت کار برای شروع هر مبحث از جاوا می باشد و بعد از مطالعه آن تئوری خواندن را رها کنید و تمرین و مثال های بیشتری حل کنید تا به چالش کشیده شوید و جزئیات بیشتری از هر مبحث را یاد بگیرید.

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 جاوالایک به دور از اخلاق حرفه ای ست و حرام می باشد.