

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه پنجم

کلاس JTextArea

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



سخن نویسنده:

در این جلسه قصد داریم یکی از اجزا (component) گرافیکی جاوا را بررسی کنیم. ما در مبحث گرافیک تک به تک اجزا مختلف گرافیک جاوا رو بصورت جلسات مجزا یاد خواهیم گرفت و در نهایت با آموزش پروژه محور از این اجزای گرافیکی برنامه های کاربردی و بازی خواهیم ساخت پس تا رسیدن به مرحله آموزش پروژه محور ما را دنبال کنید.

کلاس JTextArea:

همان طور که می دانید تمامی اجزای گرافیکی (component) در جاوا هر کدام یک کلاس هستند و ما با ایجاد شی به تعداد دلخواه از آنها، در برنامه خود از شون استفاده می کنیم.

وقتی از کلاس JTextArea شی ایجاد می کنیم یک ناحیه ای برای نمایش متن ایجاد می شود. این ناحیه متنی ایجاد شده بر خلاف TextField چندخطی یا چند سطری می باشد، یعنی می توان بعد از پایان سطر اول به سطر دوم و... برای نوشتن متن برویم. همچنین شی ایجاد شده از کلاس JTextArea به ما اجازه ویرایش تمام خطوط را می دهد.

پس کلاس JTextArea ویژگی های زیر را دارا می باشد:

۱. برای نمایش متن استفاده می شود.
۲. یک متن را می توان در چند سطر در آن قرار دهیم.
۳. متن های داخل آن قابل ویرایش می باشند.

سازنده های پر کاربرد کلاس JTextArea:

سازنده	کاربرد
JTextArea()	برای ایجاد یک ناحیه متنی خالی استفاده می شود.

سازنده	کاربرد
JTextArea(String s)	برای ایجاد یک ناحیه متنی با متن مورد نظر استفاده می شود.

سازنده	کاربرد
<code>JTextArea(int row, int column)</code>	برای ایجاد یک ناحیه متنی با تعداد سطر و ستون مشخص و نمایش بدون متن استفاده می شود.

سازنده	کاربرد
<code>JTextArea(String s, int row, int column)</code>	برای ایجاد یک ناحیه متنی با تعداد سطر و ستون مشخص و همچون نمایش متن مورد نظر استفاده می شود.

متدهای پر کاربرد کلاس JTextArea:

متد	کاربرد
<code>void setRows(int rows)</code>	برای مقداردهی کردن تعداد سطر مشخص استفاده می شود.

متد	کاربرد
<code>void setColumns(int cols)</code>	برای مقداردهی کردن تعداد ستون مشخص استفاده می شود.

متد	کاربرد
<code>void setFont(Font f)</code>	برای ست کردن فونت مورد نظر استفاده می شود

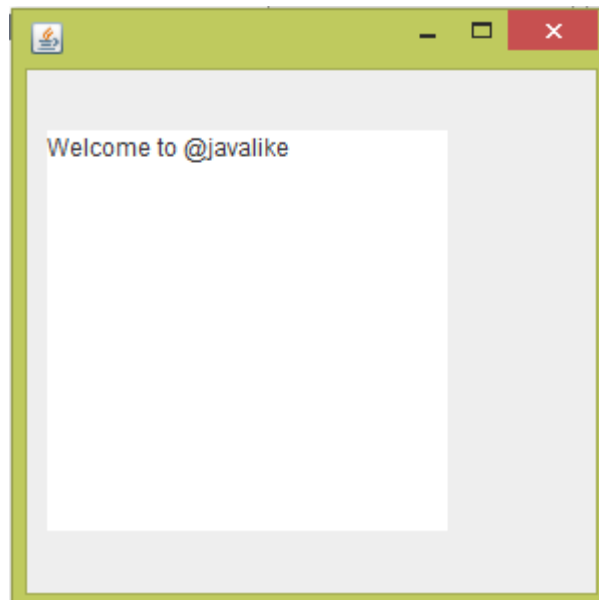
متد	کاربرد
<code>void insert(String s, int position)</code>	برای درج کردن یک متن مشخص در نقطه خاصی از ناحیه متنی استفاده می شود

مثال: بهترین راه یادگیری گرافیک در جاوا دیدن و حل کردن مثال های فراوان می باشد.

```
package swing_javalike;
import javax.swing.*;
public class TextAreaExample
{
    TextAreaExample(){
        JFrame f= new JFrame();
        JTextArea area=new JTextArea("Welcome to @javalike");
        area.setBounds(10,30, 200,200);
        f.add(area);
        f.setSize(300,300);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String args[])
    {
        new TextAreaExample();
    }
}
```

خروجی: این کد را اگر در یکی از IDE های جاوا اجرا کنید خروجی بصورت زیر خواهد بود: تصویر (۱)

نکته: منظور از IDE همان ویرایشگر یا محیطی است که کدهای جاوا را در آن اجرا می کنیم مثال ایدکیپس، نت بینز و...



تصویر (۱)

- همان طور که در تصویر (۱) مشاهده می کنید متن ما در ناحیه ای سفید رنگ نمایش داده شده است که این متن قابل ویرایش است.

```
TextAreaExample(){
    JFrame f= new JFrame();
    JTextArea area=new JTextArea("Welcome to @javalike");
}
```

- در سازنده کلاس یک شی از کلاس JFrame با نام f ایجاد کرده ایم. همان طور که می دانید frame نقش اسکلت یک برنامه را بازی می کند و ما سایر اجزای گرافیکی را روی فریم می چسبانیم.

```
JTextArea area=new JTextArea("Welcome to @javalike");
```

- برای ایجاد یک ناحیه متنی در فریم خود از کلاس JTextArea شی ایجاد کرده و متنی را به عنوان پارامتر به سازنده آن داده ایم که در ناحیه متنی نمایش داده می شود.

```
area.setBounds(10,30, 200,200);
f.add(area);
f.setSize(300,300);
```

- کارهای ثابتی که برای هر اجزای گرافیکی باید انجام دهیم برای TextArea نیز انجام می دهیم. نظیر تنظیم مختصات و عرض و ارتفاع آن، اضافه کردن اجزای گرافیکی به فریم، تنظیم اندازه فریم و.....

```
f.setLayout(null);
```

- چون قصد نداریم از طرح بندی خاصی استفاده کنیم مقدار پارامتر این متد را null قرار می دهیم.

```
f.setVisible(true);
```

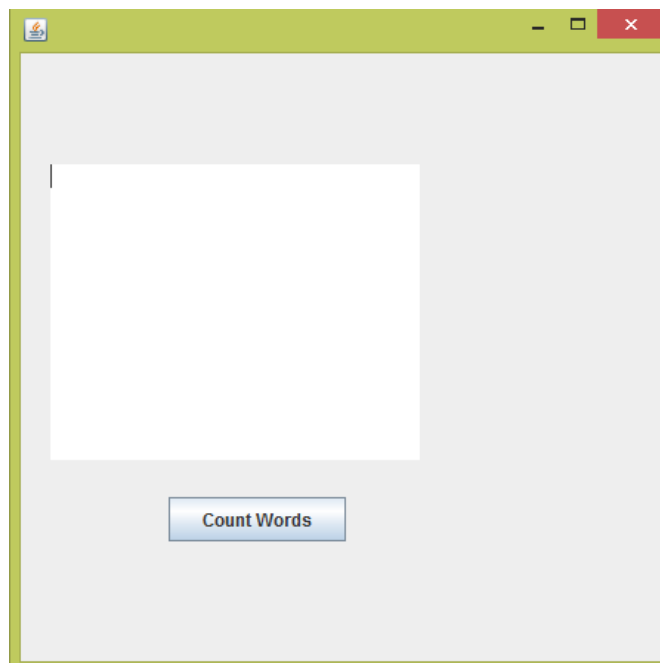
- برای نمایش فریم و تمامی اجزای گرافیکی باید مقدار پارامتر این متد true باشد.

مثال: برنامه زیر تعداد کلمات و تعداد کاراکترها (حروف) یک متن را می شمارد.:

```
package javalike;
import javax.swing.*;
import java.awt.event.*;
public class TextAreaExample implements ActionListener{
    JLabel l1,l2;
    JTextArea area;
    JButton b;
    TextAreaExample() {
        JFrame f= new JFrame();
        l1=new JLabel();
        l1.setBounds(50,25,100,30);
        l2=new JLabel();
```

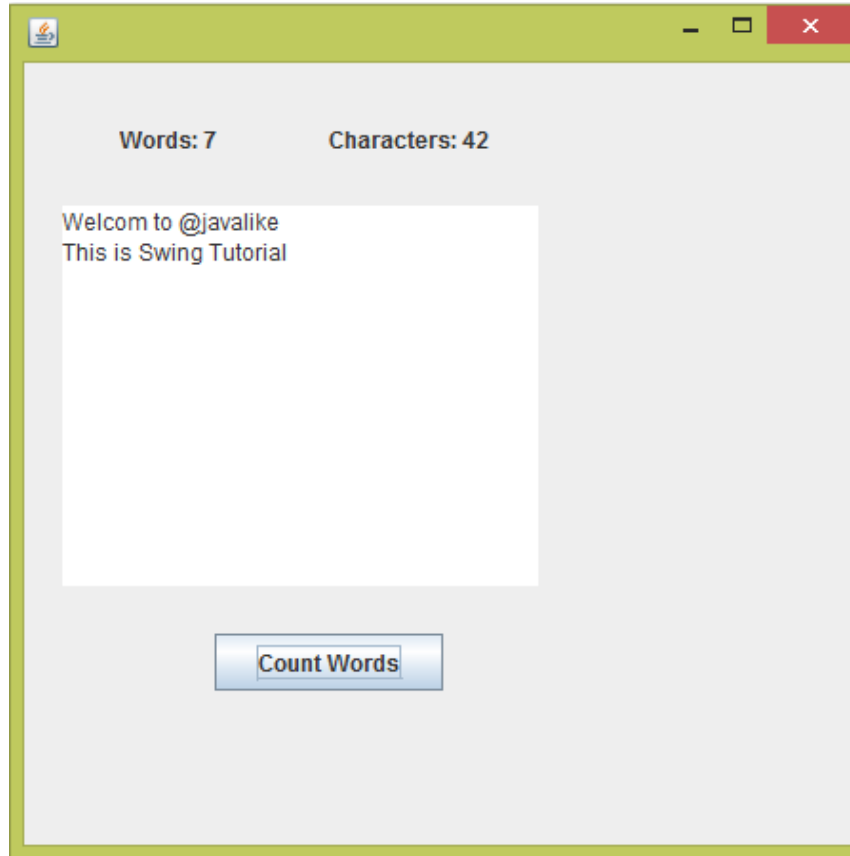
```
l2.setBounds(160,25,100,30);
area=new JTextArea();
area.setBounds(20,75,250,200);
b=new JButton("Count Words");
b.setBounds(100,300,120,30);
b.addActionListener(this);
f.add(l1);f.add(l2);f.add(area);f.add(b);
f.setSize(450,450);
f.setLayout(null);
f.setVisible(true);
}
public void actionPerformed(ActionEvent e){
String text=area.getText();
String words[]=text.split("\\s");
l1.setText("Words: "+words.length);
l2.setText("Characters: "+text.length());
}
public static void main(String[] args) {
new TextAreaExample();
}
}
```

خروجی: وقتی برنامه را اجرا میکنید با تصویر (۲) روبرو می شوید:



تصویر (۲)

حال در ناحیه سفید رنگ متن خود را تایپ می کنیم و برای شمردن تعداد کلمات و کاراکترهای متن تایپ شده و نمایش آنها روی دکمه Count Words کلیک می کنیم: تصویر (۳)



تصویر (۳)

```
public class TextAreaExample implements ActionListener{
```

- در برنامه خود قصد داریم از دکمه button استفاده کنیم و از طرفی برای دکمه خود قصد داریم عمل و رویدادی در نظر بگیریم بنابراین کلاس مون رو implements به اینترفیس ActionListener می کنیم.

```
JLabel l1,l2;  
JTextArea area;  
JButton b;
```

- برنامه ما از دو label برای نمایش تعداد کلمات و کاراکترها، یک TextArea برای نمایش و ویرایش متن مورد نظر و یک button به عنوان دکمه ای که عمل خاصی را برای ما انجام می دهد.

```

TextAreaExample() {
    JFrame f= new JFrame();
    l1=new JLabel();
    l1.setBounds(50,25,100,30);
    l2=new JLabel();
    l2.setBounds(160,25,100,30);
    area=new JTextArea();
    area.setBounds(20,75,250,200);
    b=new JButton("Count Words");
    b.setBounds(100,300,120,30);
    b.addActionListener(this);
    f.add(l1);f.add(l2);f.add(area);f.add(b);
    f.setSize(450,450);
    f.setLayout(null);
    f.setVisible(true);
}

```

- اجزای گرافیکی خود را در سازنده کلاس مون ایجاد کرده ایم.
- در این بخش سازنده کلاس تمام اجزای گرافیکی خود نظیر label ، TextArea ، و... را صدا زده ایم.
- همچنین مختصات و ابعاد اجزای گرافیکی خود را با متد setBounds مشخص کرده ایم.

```
b.addActionListener(this);
```

- شی b دکمه ما را تشکیل می دهد.حالا قصد داریم عمل خاصی را به دکمه خود نسبت دهیم برای این کار متد addActionListener را صدا زده و کلمه کلیدی this که در اینجا معادل شی از کلاسی است که در آن قرار داریم را به عنوان پارامتر به این متد داده ایم.
- دلیل این که از کلمه this استفاده کردیم بصورت زیر است:

از انجایی که کلاس خود را implements به اینترفیس ActionListener کردیم دیگر نیازی نیست در پارامتر متد addActionListener از اینترفیس ActionListener مستقیم شی ایجاد کنیم.تنها کافیسست شی ای از کلاسی که implements به اینترفیس ActionListener شده است را جایگزین پارامتر متد addActionListener کنیم که در اینجا کلمه کلیدی this به شی کلاسی که در آن قرار داریم اشاره دارد.

میدونم شاید باز گنگ باشه توضیح! به طور کلی هر وقت خواستیم در کلاس مون یک دکمه یا اجزای گرافیکی که عملی خاص را با کلیک کردن روی آن برامون انجام بده را ایجاد کنیم موارد زیر را به صورت ثابت در برنامه مون انجام می دهیم.

1. کلاسی که قصد داریم برای اجرای یک عمل خاص، یک button یا هر اجزای گرافیکی دیگری در آن تعریف کنیم ، implements به اینترفیس ActionListener می کنیم.


```
public class TextAreaExample implements ActionListener{
```

۲. با شی ایجاد شده از `button` یا هر اجزای گرافیکی که قراره با کلیک کردن روی آن عمل خاصی را اجرا کند متد `addActionListener` را صدا می زنیم.

```
b.addActionListener();
```

۳. شی ای از کلاسی که در آن قرار داریم را جایگزین پارامتر متد `addActionListener` می کنیم. همچنین می توانیم به جای ایجاد شی از کلاس از کلمه کلیدی `this` که در اینجا اشاره به شی کلاس دارد استفاده کنیم.

```
b.addActionListener(this);
```

۴. دستورات و عملیاتی را که قصد داریم هنگام کلیک کردن روی `button` یا هر اجزای گرافیکی اجرا شود را درون متد زیر قرار می دهیم:

```
public void actionPerformed(ActionEvent e){
    }
}
```

```
f.add(l1);f.add(l2);f.add(area);f.add(b);
```

- تمامی اجزای گرافیکی را به فریم خود اضافه میکنیم.

```
f.setSize(450,450);
f.setLayout(null);
f.setVisible(true);
```

- خوب اینم کارای همیشگی که برای فریم خود باید انجام دهیم 😊

```
public void actionPerformed(ActionEvent e){
    String text=area.getText();
    String words[]=text.split("\\s");
    l1.setText("Words: "+words.length);
    l2.setText("Characters: "+text.length());
}
}
```

- این متد وقتی رویدادی رخ دهید دستورات درون آن اجرا می شود.
- منظور از رویداد کلیک کردن روی اجزای گرافیکی خاص و... می باشد.

```
String text=area.getText();
```

- این دستور متن موجود در ناحیه متنی یا همون `TextArea` را برمیگرداند و درون یک متغیر از نوع رشته می ریزد.

```
String words[]=text.split("\\s");
```

- این دستور برای استخراج کلمات متن استفاده می شود.
- منطق این دستور این است که می‌گه تمام کلمات متن را بر اساس فضای خالی (space) تفکیک کن و بریز درون یک آرایه.

```
11.setText("Words: "+words.length);
```

- این برنامه طول آرایه words که برابر با تعداد کلمات درون متن TextArea هستش را درون لیبل l1 قرار می دهد.

```
12.setText("Characters: "+text.length());
```

- این دستور طول متن دریافت شده از TextArea را به عنوان تعداد کاراکترها درون لیبل l2 قرار می دهد.

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

- در پایان هم با شی سازی از کلاس درون متد main می توانیم برنامه را اجرا کنیم.

امیدوارم از این جلسه آموزشی رضایت داشته باشید. ما در تلاش هستیم محتوای فارسی در زمینه برنامه نویسی جاوا را از لحاظ کیفیت و کمیت بهبود بخشیم. برای حمایت از ما اگر سایت، وبلاگ، شبکه اجتماعی، مجموعه ایمیل، گروه و... دارید لینک سایت و کانال آموزش آسان جاوا را به اشتراک بگذارید. سپاسگزاریم

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.