

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه اول

معرفی Swing

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



جاوا مجموعه ای از کتابخانه ها را برای ایجاد رابط کاربری گرافیکی فراهم کرده است.

این آموزش برای کسانی که قصد دارند برنامه های کاربردی ویندوزی و بازی با استفاده از زبان جاوا طراحی کنند بصورت ساده با گام های آسان تهیه شده است.

ما در این آموزش گرافیک را در محیط ایکلیپس کار می کنیم.

پیشنیازهای مبحث Swing در جاوا:

قبل از شروع مطالعه این مبحث شما باید با مباحث پایه ای جاوا ، کار با ویرایشگرهای (IDE) جاوا، نحوه اجرای برنامه و... آشنایی داشته باشید.

اگر اولین بار است که قصد یادگیری زبان جاوا را دارید لطفا از این مبحث صرف نظر کرده و از جلسات مقدماتی جاوا که در کانال تلگرام [javapro.ir](http://www.javapro.ir) و سایت www.javapro.ir از صفر آموزش داده شده است شروع به یادگیری کنید.

گرافیک در جاوا:

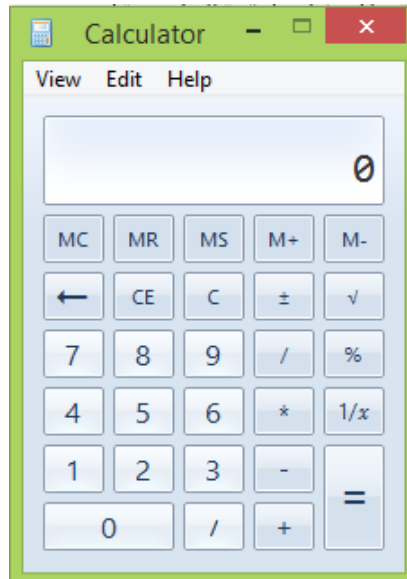
برای ایجاد برنامه های کاربردی گرافیکی مبتنی بر ویندوز ، جاوا دو پکیج در اختیار ما گذاشته است ابتدا پکیج AWT و بعدش پکیج Swing ، این دو پکیج شباهت ها و تفاوت هایی باهم دارند، از آن جهتی که پکیج Swing جدیدتر هستش ما قصد داریم گرافیک در جاوا با استفاده از پکیج Swing را کار کنیم هر چند که در هنگام نوشتن پروژه اگر نیاز به بعضی از مباحث AWT داشتیم بخش های مورد نیاز آن را آموزش خواهیم داد.

برای نوشتن برنامه های گرافیکی مبحث `javafx` هم هستش که فعلا ما باهاش کاری نداریم!

در پکیج `javax.swing` کلاس هایی نظیر `JButton`, `JTextField`, `JTextArea`, `JRadioButton`, `JCheckbox`

`JMenu`, `JColorChooser` و... وجود دارد که به ما در نوشتن یک برنامه گرافیکی ویندوزی کمک می کنند.

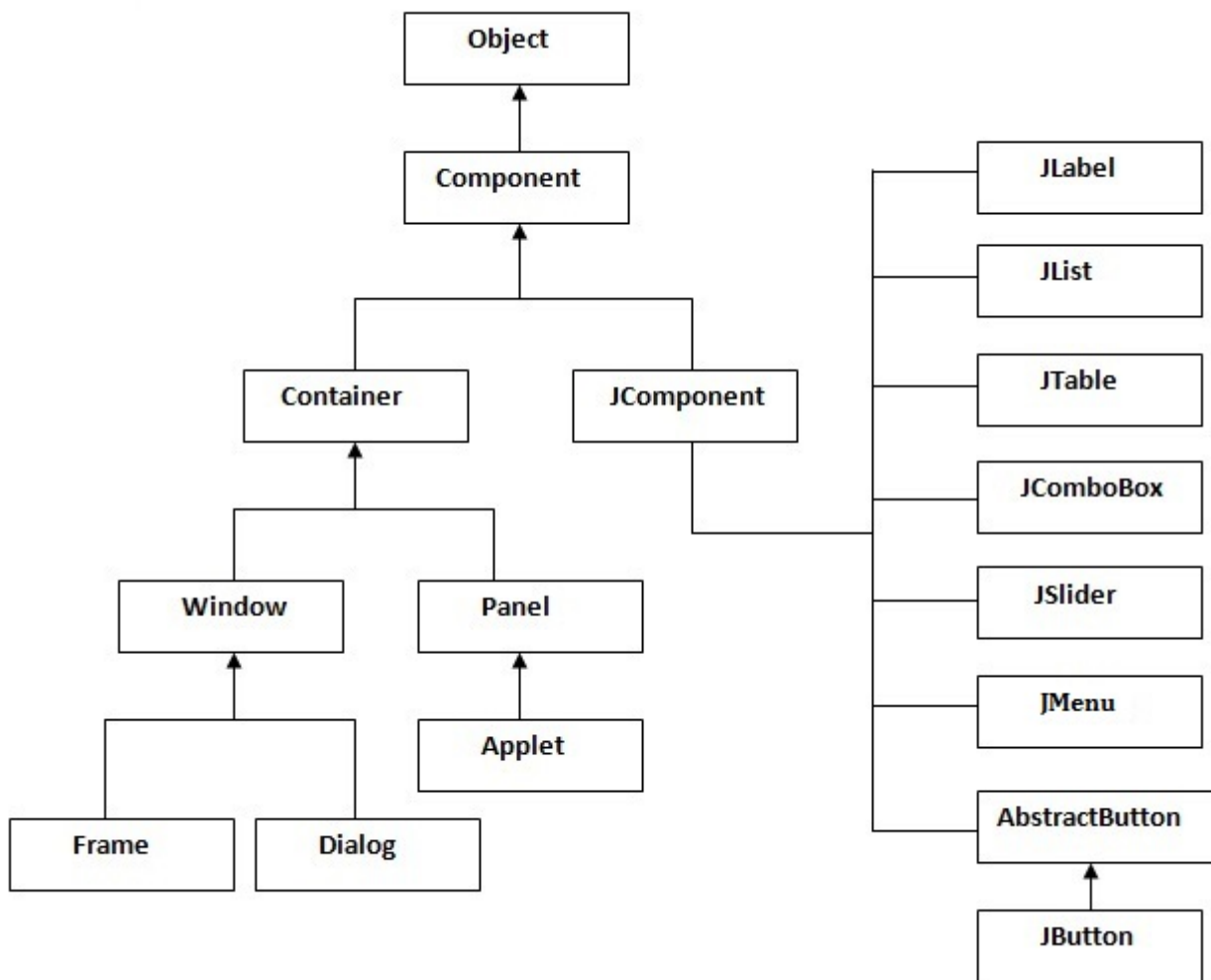
برای درک بهتر این کلاس ها در پکیج `javax.swing` برنامه ماشین حساب در کامپیوتر را باز می کنیم: تصویر(۱)



تصویر(۱)

همان طور که در تصویر(۱) مشاهده می کنید این برنامه ماشین حساب از چندین دکمه بر انتخاب عدد یا عملیات مورد نظر ، یک صفحه برای نمایش اعداد و نتیجه عملیات، یک منو که از بخش های متنوعی تشکیل شده و هر یک کار خاصی را انجام می دهند تشکیل شده است. همه اجزای این ماشین حساب از دکمه تا اون صفحه و منوها خود یک کلاس مجزا هستند که در برنامه ماشین حساب خود استفاده کرده ایم ، مثلا کلاس `JButton` در جاوا کار دکمه های ماشین حساب را می تواند انجام دهند و... بگذریم تنها هدف از این مثال پیدا کردن یک ذهنیت درمورد اجزای گرافیکی در جاوا بود در ادامه بیشتر به این اجزای گرافیکی می پردازیم.

سلسله مراتب کلاس های درون پکیج `Swing` را در زیر مشاهده می کنید: تصویر(۲)



تصویر (۲)

در تصویر (۲) کلاس **Object** را مشاهده می کنید که پدر همه کلاس ها در جاوا می باشد. همچنین اجزای گرافیکی در جاوا که هر کدام یک کلاس مجزا می باشند. ما با استفاده از همین کلاس هایی که در تصویر (۲) مشاهده می کنید برنامه گرافیکی خود را می توانیم طراحی کنیم.

نکته بسیار مهم : یادگیری مبحث گرافیک در جاوا به صورت تئوری زیاد جالب نیست!!! از جهتی خوب باید تئوری ها هم گفت!!! اما اصلا نگران نباشید! بهترین راه یادگیری مفهوم گرافیک در جاوا بررسی مثال ها و تمرینات کدنویسی در این زمینه می باشد، پس اگر از تئوری هایی که میگویم گیج کننده است اصلا نگران نشوید ما با مثال سعی می کنیم بصورت ساده این مبحث رو آموزش دهیم.

کلاس Component :

یکی از اجزای گرافیکی شکل دهنده برنامه ویندوزی ما می باشد.

نکته: ما به هر یک از اجزای گرافیکی در جاوا یک **component** می گوییم. مثلا کلاس **JButton** یک **component** می باشد.

متدهای متداول این کلاس که بصورت گسترده استفاده می شود بصورت زیر می باشد:

متد	توصیف
<code>public void add(Component c)</code>	اضافه کردن یک component به component دیگر
<code>public void setSize(int width,int height)</code>	ست کردن اندازه یک component
<code>public void setLayout(LayoutManager m)</code>	برای مدیریت طرح بندی component
<code>public void setVisible(boolean b)</code>	بعد از اضاف کردن یک component ، برای این که component ما نمایش داده شود از این متد استفاده می شود. مقدار پیشفرض این متد false می باشد، که برای دیده شدن و نمایش component باید این مقدار را true کنیم

میدونم مطمئنا گمراه کننده است ، خودم هم قبول دارم، تنها پیشنهادی که برای یادگیری این مبحث براتون دارم به مثال هایی که براتون خواهیم زد توجه کنید. 😊

یک توضیحی از جدول بالا براتون میگم ، همان طور که میدانید یک برنامه گرافیکی از اجزای گرافیکی تشکیل شده است! اجزای گرافیکی چی هستند؟! اجزای گرافیکی همان دکمه ها، لیبل ها، منوها ، دکمه های رادیویی و... که تاحالا در برنامه های موجود در کامپیوتر زیاد دیدیم!! به هر یک از این اجزای گرافیکی یک **component** می گوییم! خب ما برای این که یک اجزای گرافیکی یا **component** را در برنامه خود استفاده کنیم نیاز هست ابتدا این اجزای گرافیکی را اضافه کنیم! مثلا ما یک فریم داریم ، قصد داریم یک دکمه یا باتن را به فریم خود اضافه کنیم خب با استفاده از متد **add** آن را به فریم مون اضافه می کنیم که به این کار میگن اضافه کردن یک **component** به **component** دیگر! بعد نیاز هست اندازه این دکمه را مشخص کنیم بعدش طرح بندی این دکمه را مشخص کنیم یعنی بگیم کجای فریم قرار بگیر بعدش با **true** کردن مقدار پارامتر متد **setVisible** دکمه یا **component** ما در فریم نمایش داده می شود.

بازم نگران نباشید!!!! میریم سراغ مثال 😊

مثال های Swing :

نکته: دوستان گرافیک در جاوا مفاهیم خیلی زیاد است و همه این ها رو همیشه در قالب یک جلسه آورد! خب چطور یادشون بگیریم؟! ما پرکاربردترین و متداول ترین مفاهیم گرافیک در جاوا را آموزش خواهیم داد، سایر مفاهیم در قالب پروژه اگه بهشون نیاز پیدا کردید می تونید یاد بگیرید یعنی تا زمانی که نیاز پیدا نکردید نیازی نیست بر کل مفاهیم گرافیک در جاوا مسلط باشید. پس مهم ترین اساس یادگیری نیاز است!

خب ما میخوایم یک برنامه گرافیکی را طراحی کنیم! اولین اجزای گرافیکی که نیاز داریم **frame** است، **frame** کارش چیه؟ ما تمام اجزای گرافیکی یا همون **component** ها رو روی **frame** سوار می کنیم! چیزی شبیه تنه موتورسیکلت! که تمامی اجزای موتورسیکلت بر روی تنه آن سوار و چسبیده می شود!!

- برای ایجاد **frame** در جاوا دو راه وجود دارد:

۱. ایجاد یک شی از کلاس **Frame**

۲. به ارث بردن کلاس **Frame**

- می توانیم اجزای گرافیکی خود را درون متد **main**، **constructor** یا هر متد دیگری در کلاس خود ایجاد کنیم.

- یک مثال ساده از یک برنامه گرافیکی که از پکیج **Swing** استفاده شده است.

مثال:

در برنامه زیر قصد داریم یک **Frame** ایجاد کنیم، برای این کار بصورت زیر عمل می کنیم:

```
package swing_javalike;

import javax.swing.*;

public class FirstSwingExample {
    public static void main(String[] args) {
        JFrame f = new JFrame();// creating instance of JFrame
    }
}
```

- برای نوشتن یک برنامه گرافیکی با استفاده از **Swing** نیاز است که ابتدا پکیج زیر را **import** کنیم:

```
import javax.swing.*;
```

- خب بعدش یک کلاس ایجاد می کنیم و در متد **main** یک شی از کلاس **JFrame** ایجاد می کنیم.

```
JFrame f = new JFrame();
```

در ادامه قصد داریم اندازه **frame** خود را مشخص کنیم برای این کار از متد **setSize** استفاده می کنیم:

```
package swing_javalike;

import javax.swing.*;

public class FirstSwingExample {
    public static void main(String[] args) {
        JFrame f = new JFrame();// creating instance of JFrame
        f.setSize(400, 500);
    }
}
```

• در کد بالا ۴۰۰ عرض و ۵۰۰ ارتفاع **frame** ما را مشخص می کند.

خب قصد مدیریت طرح بندی برای **frame** خود نداریم پس مقدار پارامتر متد **setLayout** را **null** می کنیم:

```
package swing_javalike;

import javax.swing.*;

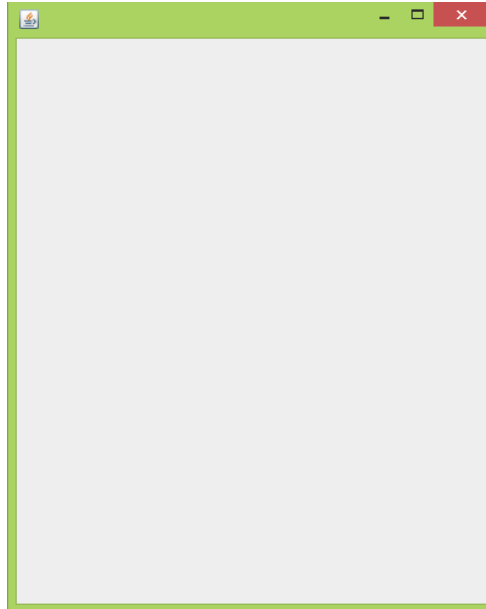
public class FirstSwingExample {
    public static void main(String[] args) {
        JFrame f = new JFrame();// creating instance of JFrame
        f.setSize(400, 500);
        f.setLayout(null);
    }
}
```

خب برای این که **frame** ما هنگام اجرای برنامه نمایش داده شود نیاز است که متد **setVisible** را صدا زده و مقدار پارامتر آن را **true** کنیم:

```
package swing_javalike;
import javax.swing.*;

public class FirstSwingExample {
    public static void main(String[] args) {
        JFrame f = new JFrame();// creating instance of JFrame
        f.setSize(400, 500);
        f.setLayout(null);
        f.setVisible(true);
    }
}
```

- همان طور که مشاهده می کنید در کد بالا شی ای که از کلاس **JFrame** ایجاد کرده ایم **f** نام دارد. حال با شی **f** متدهای مربوطه را صدا زده ایم. خروجی این برنامه بعد از کامپایل و اجرا بصورت زیر خواهد بود: تصویر (۳)



تصویر (۳)

خب ما تا اینجا **frame** یا بقولاً تنه یا اسکلت برنامه خود را ایجاد کرده ایم. خب حالا میتونید سایر اجزای گرافیکی یا **component** های خود را به این **frame** خود اضافه کنیم. خب ما قصد داریم به **frame** خود یک دکمه یا باتن اضافه کنیم:

- برای اضافه کردن یک دکمه یا **Button** کافی است که از کلاس **JButton** شی ایجاد کنیم و شی ایجاد شده از آن را به **frame** خود اضافه کنیم:

```
package swing_javalike;

import javax.swing.*;
public class FirstSwingExample {
    public static void main(String[] args) {
        JFrame f = new JFrame();// creating instance of JFrame
        f.setSize(400, 500);
        f.setLayout(null);
        f.setVisible(true);
        JButton b=new JButton("click");//creating instance of JButton
    }
}
```


- ما از کلاس JButton شی ای با نام b ایجاد کرده ایم. مقدار پارامتر سازنده آن را "click" گذاشته ایم. حالا باید button خود را به frame اضافه کنیم که کد برنامه تا اینجا بصورت زیر خواهد بود:

```
package swing_javalike;

import javax.swing.*;

public class FirstSwingExample {
    public static void main(String[] args) {
        JFrame f = new JFrame();// creating instance of JFrame
        f.setSize(400, 500);
        f.setLayout(null);
        f.setVisible(true);
        JButton b=new JButton("click");//creating instance of JButton
        f.add(b);//adding button in JFrame
    }
}
```

- از طریق شی b متد add را برای اضافه شدن button به frame صدا زده ایم.
- آیا کارمون تمام شده و باید برنامه رو اجرا کنیم؟! زیاد عجله نکنید! یک کار کوچک دیگه مونده!!! ما باید عرض و ارتفاع و مختصات قرار گیری دکمه (button) خود در frame را نیز مشخص کنیم! برای تعیین عرض و ارتفاع و مختصات یک اجزای گرافیکی در جاوا از متد setBounds استفاده می کنیم:

```
setBounds(x, y, width, height) ;
```

- پارامتر X محور Xها و پارامتر y محور yها می باشد.
- width عرض اجزای گرافیکی
- height ارتفاع اجزای گرافیکی

کد بالا رو بصورت زیر تغییر می دهیم:

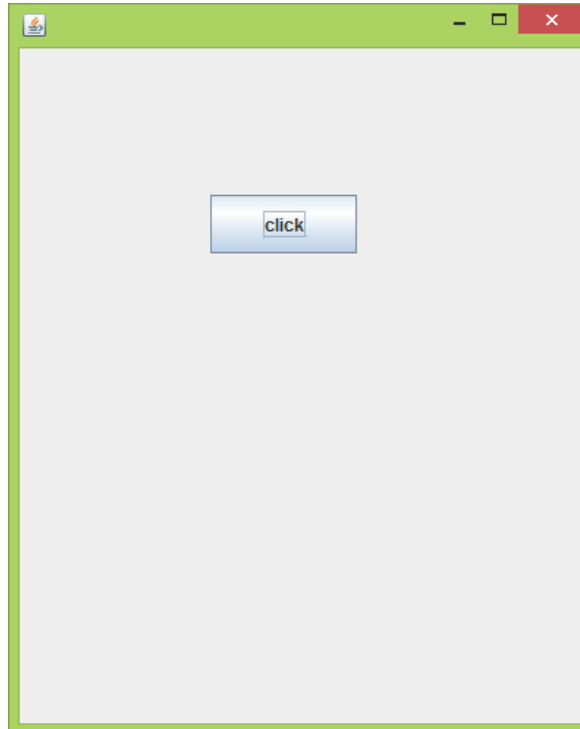
```
package swing_javalike;

import javax.swing.*;

public class FirstSwingExample {
    public static void main(String[] args) {
        JFrame f = new JFrame();// creating instance of JFrame
        f.setSize(400, 500);
        f.setLayout(null);
    }
}
```

```
f.setVisible(true);
JButton b=new JButton("click");//creating instance of JButton
f.add(b);//adding button in JFrame
b.setBounds(130,100,100, 40);//x axis, y axis, width, height
}
}
```

خروجی این برنامه بصورت زیر خواهد بود: تصویر (۴)



تصویر (۴)

- خوب ما در این مثال با نحوه ایجاد یک **frame** و اضافه کردن یک **component** نظیر **button** را به آن آشنا شدیم. چطور بود؟! فقط از راه مثال گرافیک رو یادبگیرید تئوری گیج کننده است.

در مثال قبل ما اجزای گرافیکی خود را درون متد **main** ایجاد کردیم حالا در مثال زیر قصد داریم اجزای گرافیکی خود را درون سازنده کلاس ایجاد کنیم:

```
package swing_javalike;
import javax.swing.*;
public class Simple {
    JFrame f;
    Simple(){
        f=new JFrame();//creating instance of JFrame
    }
}
```

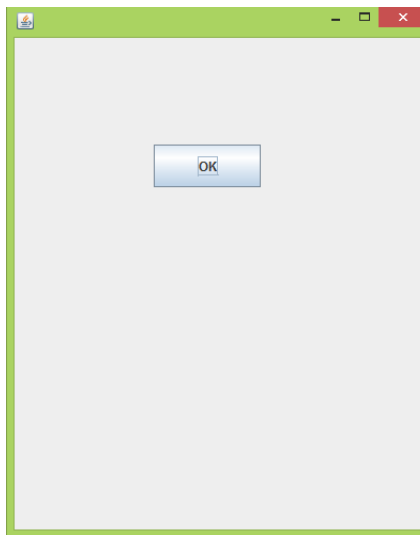
```
JButton b=new JButton("OK");//creating instance of JButton
b.setBounds(130,100,100, 40);

f.add(b);//adding button in JFrame

f.setSize(400,500);//400 width and 500 height
f.setLayout(null);//using no layout managers
f.setVisible(true);//making the frame visible
}

public static void main(String[] args) {
Simple s=new Simple();
}
}
```

خروجی: بعد از کامپایل و اجرای برنامه خروجی بصورت زیر خواهد بود، همان طور که مشاهده می کنید در برنامه های گرافیکی دیگر خبری از محیط کنسول نیست و همگی بصورت دکمه، لیبل و پنجره و... خواهد بود. تصویر (۵)



تصویر (۵)

- در این مثال ما اجزای گرافیکی خود یعنی فریم و باتن را درون سازنده کلاس ایجاد کرده و متدهای مربوطه هر کدام را صدا زده ایم. در نهایت با شی سازی از کلاس خود در متد **main**، دستورات درون سازنده کلاس اجرا می شود، خوب دستورات درون سازنده کلاس ما هم همون ایجاد اجزای گرافیکی ما هستند.
- متد **setBounds** در مثال بالا برای مشخص کردن مختصات و عرض و ارتفاع **button** ما می باشد.

یک مثال ساده از ایجاد یک **frame** با استفاده از به ارث بردن کلاس **JFrame** :

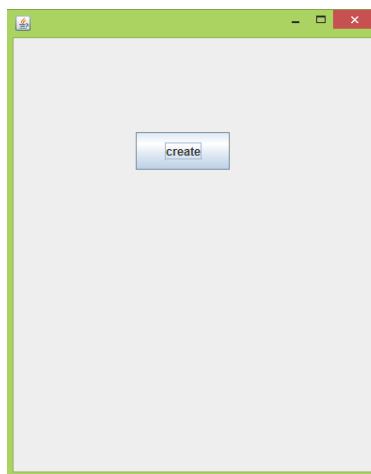
- در این روش کلاس ما ، کلاس **JFrame** را به ارث می برد ، در نتیجه به ویژگی ها و متدهای این کلاس دسترسی پیدا می کند ، به بیانی دیگر وقتی کلاس ما ، کلاس **JFrame** را به ارث می برد، کلاس ما می شود فرزند کلاس **JFrame** به نوعی کلاس ما تبدیل به یک **frame** می شود.
- در این روش دیگر نیازی به ایجاد یک شی از کلاس **JFrame** بصورت مجزا نیست.

مثال:

```
package swing_javalike;
import javax.swing.*;
public class Simple2 extends JFrame{//inheriting JFrame
Simple2(){
JButton b=new JButton("create");//create button
b.setBounds(130,100,100, 40);

add(b);//adding button on frame
setSize(400,500);
setLayout(null);
setVisible(true);
}
public static void main(String[] args) {
new Simple2();
}}
```

خروجی : تصویر (۵)



تصویر (۵)

```
public class Simple2 extends JFrame{
```

- کلاس ما با استفاده از کلمه کلیدی `extends` کلاس `JFrame` را به ارث برده است.

```
Simple2(){
    JButton b=new JButton("create");//create button
    b.setBounds(130,100,100, 40);

    add(b);//adding button on frame
    setSize(400,500);
    setLayout(null);
    setVisible(true);
}
```

- تمام اجزای گرافیکی یا همون `component` های خود را درون سازنده کلاس ایجاد کرده ایم.
- در اینجا یک `button` ایجاد کرده ، مختصات و ابعاد آن را مشخص کرده و سپس با متد `add` به `frame` خود آن را اضافه کرده ایم. از آنجایی که کلاس ما کلاس `JFrame` را به ارث برده است هنگام صدا زدن متدهای کلاس `JFrame` نیاز به شی سازی از کلاس `JFrame` نیست و می توانیم مستقیماً متدهای کلاس `JFrame` را در کلاس خود صدا بزنیم.

```
public static void main(String[] args) {
    new Simple2();
}}
```

- در پایان هم از در متد `main` از کلاس خود شی ایجاد می کنیم. در اینجا مستقیماً سازنده کلاس خود را بعد کلمه کلیدی `new` صدا زده ایم. خب بعد از اجرای دستورات درون سازنده کلاس ، اجزای گرافیکی ما ساخته و نمایش داده می شوند.

۱. همه این ها مقدمه ای برای معرفی گرافیک و پکیج `Swing` در جاوا بود. صبر داشته باشید! در جلسات آینده با سایر اجزای گرافیکی که ما در ایجاد یک برنامه کاربردی یا بازی در جاوا کمک می کنند آشنا می شویم.

۲. خب نظرتون در مورد نحوه بیان این جلسه آموزشی را برای ما از طریق تلگرام یا ایمیل ارسال کنید، قطعاً نظرات شما در بهبود کیفیت آموزش های آینده تاثیر گذار خواهد بود.

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.

عضویت در بزرگترین و با کیفیت ترین کانال تلگرام آموزش جاوا فراموش نشود 😊