

آموزش زبان برنامه نویسی جاوا

گرافیک در جاوا - پکیج Swing

جلسه هفتم

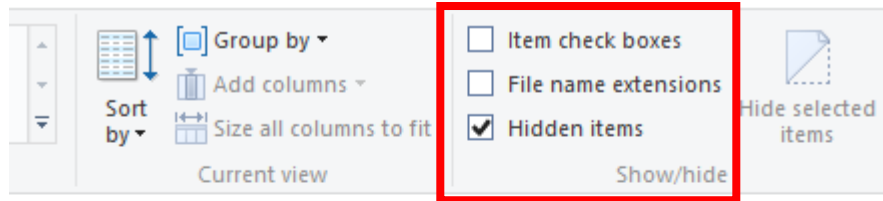
کلاس JCheckBox

نویسنده: رحمان زارعی

جاوا را ساده، آسان و شیرین بنوشید!!!!



برای ایجاد یک چک باکس از کلاس JCheckBox شی ایجاد می کنیم. حالا این چک باکس چه شکلی هست؟! و کاربردش چیه؟! گزینه های مربع شکل که به با تیک زدن آنها کار خاصی را برای ما انجام می دهند چک باکس هستند. مثال: تصویر (۱)



تصویر (۱)

در تصویر (۱) بخشی که با خطوط قرمز مشخص شده چک باکس (checkbox) می باشد.

- چک باکس ها دو گزینه روشن (turn) یا خاموش (off) دارند، با کلیک کردن روی یک چک باکس (checkbox) حالت روشن (on) و خاموش (off) آن تغییر می کند. گزینه روشن معادل مقدار true و گزینه خاموش معادل مقدار false می باشد.

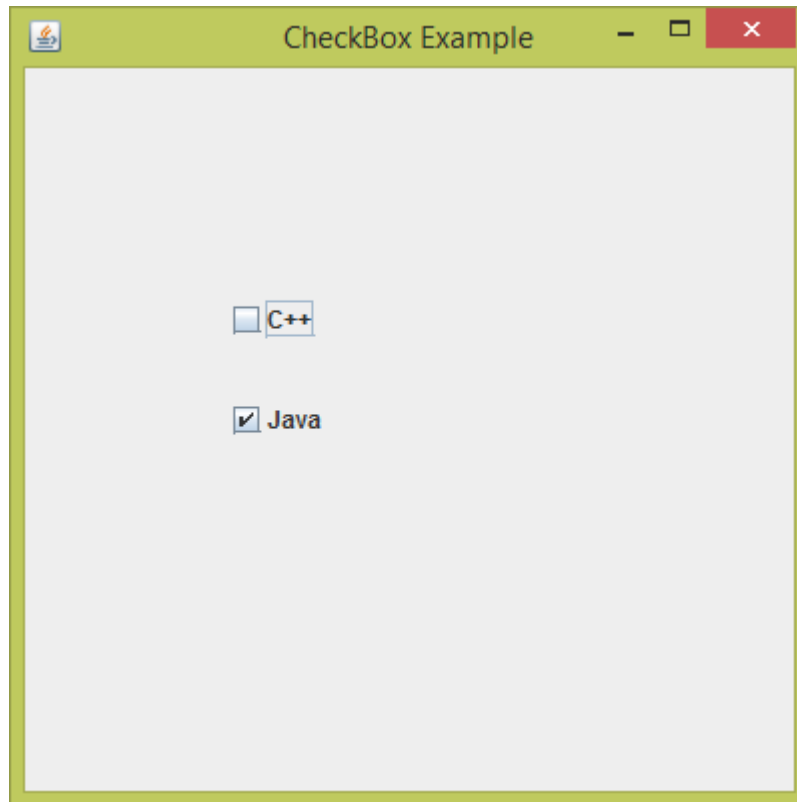
• سازنده های پر کاربرد کلاس JCheckBox:

سازنده	کاربرد
JCheckBox()	ایجاد یک دکمه چک باکس غیر انتخابی (unselected) بدون متن و آیکن منظور از غیر انتخابی یعنی on یا off بودن ، true یا false بود، تیک خوردن یا نخوردن چک باکس را مشخص نکرده ایم.
JChechBox(String s)	ایجاد یک دکمه چک باکس غیر انتخابی (unselected) همراه با متن
JCheckBox(String text, boolean selected)	ایجاد یک دکمه چک باکس همراه با متن که می توانیم در پارامتر selected خاموش یا روشن بودن ، true یا false بودن آن را مشخص کنیم.
JCheckBox(Action a)	ایجاد یک دکمه چک باکس که ویژگی گرفتن یک رویداد یا اکشن را دارا می باشد.

مثال:

```
package swing_javalike;
import javax.swing.*;
public class CheckBoxExample
{
    CheckBoxExample(){
        JFrame f= new JFrame("CheckBox Example");
        JCheckBox checkBox1 = new JCheckBox("C++");
        checkBox1.setBounds(100,100, 100,50);
        JCheckBox checkBox2 = new JCheckBox("Java", true);
        checkBox2.setBounds(100,150, 100,50);
        f.add(checkBox1);
        f.add(checkBox2);
        f.setSize(400,400);
        f.setLayout(null);
        f.setVisible(true);
    }
    public static void main(String args[])
    {
        new CheckBoxExample();
    }
}
```

خروجی: تصویر (۲)



تصویر (۲)

- در تصویر (۲) ما دو چک باکس داریم که متن اولی C++ ، تیک نخورده یا false و متن دومی Java ، تیک خورده یا true می باشد.

```
JCheckBox checkBox1 = new JCheckBox("C++");
checkBox1.setBounds(100,100, 100,50);
```

- ایجاد یک چک باکس با متن "C++"
- تعیین مختصات و ابعاد چک باکس

```
JCheckBox checkBox2 = new JCheckBox("Java", true);
checkBox2.setBounds(100,150, 100,50);
```

- ایجاد یک چک باکس با متن "Java" و انتخاب پارامتر true یعنی این که این چک باکس بصورت تیک خورده ایجاد می شود.
- تعیین مختصات و ابعاد چک باکس

```
f.add(checkBox1);  
f.add(checkBox2);  
f.setSize(400,400);
```

- اضافه کردن چک باکس ها به فریم
- تعیین ابعاد فریم

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

- چون فعلا از طرح بندی خاصی نمیخواهیم استفاده کنیم پارامتر این متد را null قرار می دهیم.

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

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

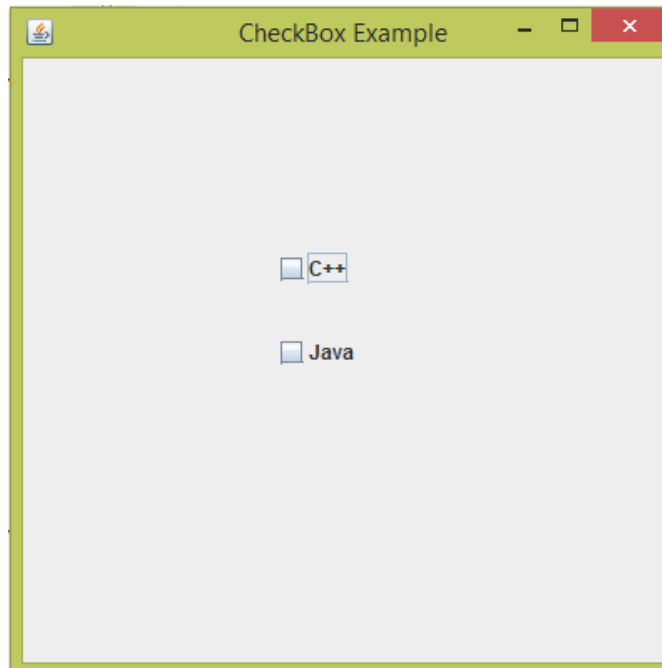
مثال:

```
package javalike;  
  
import javax.swing.*;  
  
import java.awt.event.*;  
  
public class CheckBoxExample {  
    JCheckBox checkBox1, checkBox2;  
    final JLabel label;  
  
    CheckBoxExample() {  
        JFrame f = new JFrame("CheckBox Example");  
        label = new JLabel();  
        final JLabel label2 = new JLabel();  
        label.setBounds(100, 20, 400, 100);  
        label2.setBounds(100, 180, 400, 100);  
        checkBox1 = new JCheckBox("C++");  
        checkBox1.setBounds(150, 100, 100, 50);  
        checkBox2 = new JCheckBox("Java");  
        checkBox2.setBounds(150, 150, 100, 50);  
        f.add(checkBox1);  
        f.add(checkBox2);  
        f.add(label);  
        f.add(label2);  
        checkBox1.addItemListener(new ItemListener() {  
            public void itemStateChanged(ItemEvent e) {  
                if (e.getStateChange() == 1)  
                    label.setText("C++ Checkbox: " + " checked");  
                else
```

```
        label.setText("C++ Checkbox: " + "unchecked");
    }
});
checkbox2.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        System.out.println(e.getStateChange());
        if (e.getStateChange() == 1)
            label2.setText("Java Checkbox: " + " checked");
        else
            label2.setText("Java Checkbox: " + "unchecked");
    }
});
f.setSize(400, 400);
f.setLayout(null);
f.setVisible(true);
}

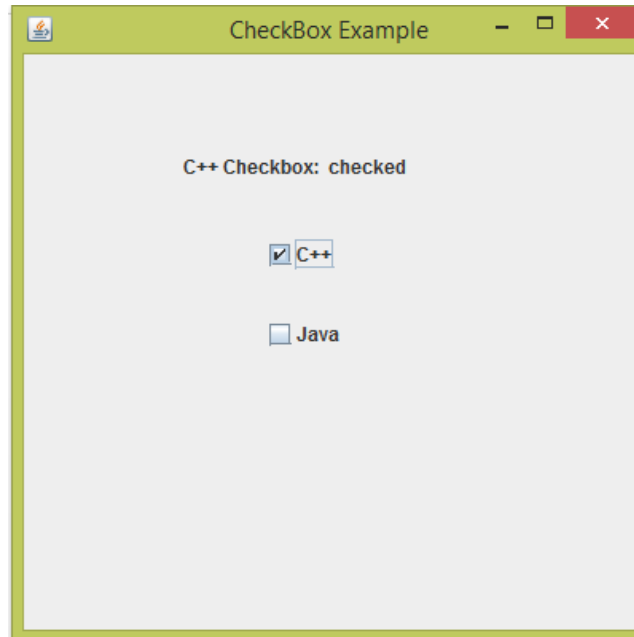
public static void main(String args[]) {
    new CheckBoxExample();
}
}
```

خروجی: هنگام اجرای برنامه خروجی بصورت تصویر (۳) خواهد بود:



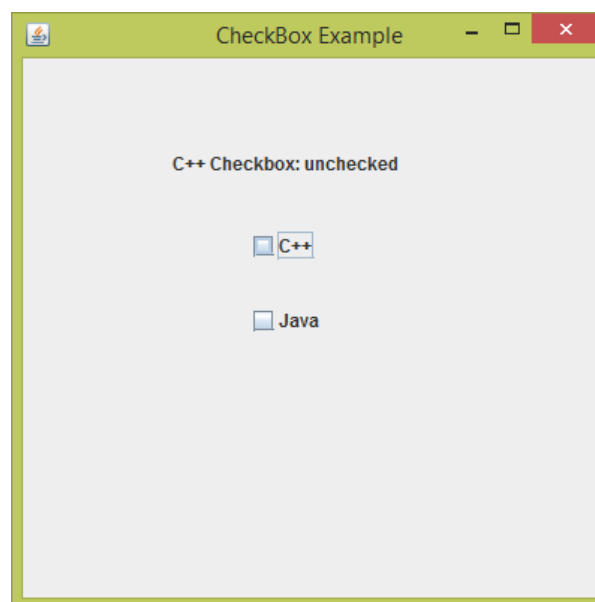
تصویر (۳)

حال با کلیک کردن یا تیک زدن چک باکس C++ خروجی بصورت تصویر (۴) خواهد شد:



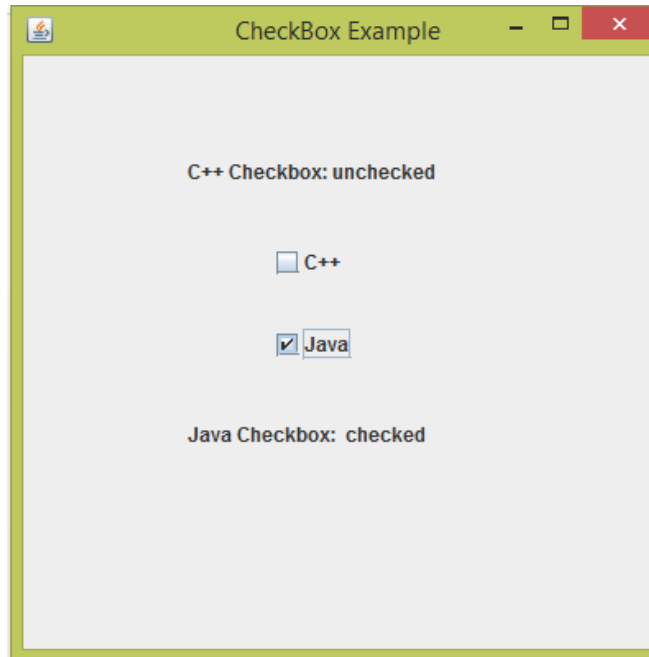
تصویر (۴)

همان طور که در تصویر (۴) مشاهده می کنید، در لیبل بالای فریم متن نمایش داده شده که بیانگر آن است که چک باکس C++ تیک خورده یا on می باشد. اگر تیک چک باکس C++ را برداریم خروجی بصورت تصویر (۵) خواهد شد:

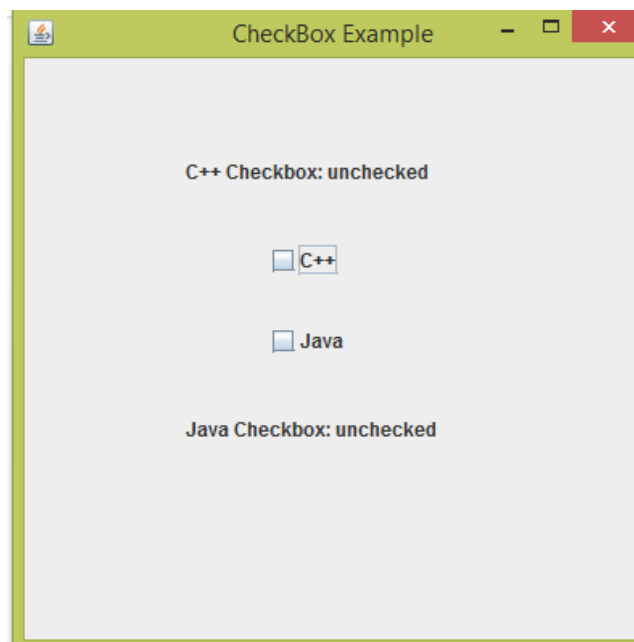


تصویر (۵)

همان طور که در تصویر(۵) مشاهده می کنید هنگامی که تیک چک باکس C++ را بر می دارید در لیبل بالای صفحه پیامی چاپ میشه که بیانگر این است که چک باکس off یا تیکش برداشته شده است. برای چک باکس Java نیز همین طور اتفاق می افتد و محل نمایش پیام در لیبل دیگر پایین صفحه می باشد. تصویر(۶) و (۷)



تصویر(۶)



تصویر(۷)

توضیحات برنامه:

```
checkbox1.addItemListener(new ItemListener() {
    public void itemStateChanged(ItemEvent e) {
        if (e.getStateChange() == 1)
            label.setText("C++ Checkbox: " + " checked");
        else
            label.setText("C++ Checkbox: " + "unchecked");
    }
});
```

- ما وقتی روی یک چک باکس کلیک می کنیم یک رویداد یا **event** رخ می دهد، نظیر اکشن ها در **Button** ..
- ما با استفاده از متد **addItemListener** این رویدادها را به چک باکس خود اضافه می کنیم.
- پارامتر متد **addItemListener** شی ای از نوع اینترفیس **ItemListener** می باشد.
- ما بصورت مستقیم شی از نوع اینترفیس **ItemListener** را ساخته و جایگزین پارامتر متد **addItemListener** می کنیم.
- بعد شی سازی از اینترفیس **ItemListener** ، بعد از سازنده این اینترفیس دو آکولاد بازو بسته می کنیم و میان این دو آکولاد متد **itemStateChanged** که دارای پارامتری از نوع **ItemEvent** می باشد را **Override** می کنیم.
- بعد از آکولاد اینترفیس **ItemListener** پرانتز متد **addItemListener** را قرار داده و علامت " ; " قرار می دهیم.
- چی شد؟ چی شد؟! جپوری شد؟! 😊
- احتمالا شما رو گیج کرده باشم! در کل شما اگر میخواید رویدادی که با کلیک کردن روی چک باکس شما رخ می دهد را دریافت کنید و عمل مورد نظر را در واکنش به آن رویداد رخ داده اجرا کنید باید با شی ایجاد شده از چک باکس خود متد **addItemListener** را صدا بزنید:

```
checkbox1.addItemListener(ItemListener arg)
```

- حالا شما باید از اینترفیس **ItemListener** شی ایجاد کنید و جایگزین پارامتر **arg** کنید. چرا؟! چون پارامتر **arg** از نوع اینترفیس **ItemListener** می باشد.
- هنگام شی سازی مستقیم از اینترفیس **ItemListener** تمام موارد بازنویسی یا **override** متد **itemStateChanged** و آکولادها و... به صورت خودکار ایکلیپس براتون انجام میده نگران نباشید . 😊

```
public void itemStateChanged(ItemEvent e) {
    if (e.getStateChange() == 1)
        label.setText("C++ Checkbox: " + " checked");
    else
        label.setText("C++ Checkbox: " + "unchecked");
}
```

- این متد مثل سرباز آماده ای ایستاده که اگر شما کوچک ترین کلیک بر روی چک باکس خود کنید ، رویداد رخ داده که همان کلیک بر روی چک باکس هستش را دریافت کند.

```
if (e.getStateChange() == 1)
```

- این دستور حالت رویدادی که چک باکس شما تولید کرده را بررسی می کند.
- کلیک کردن بر روی یک چک باکس از دو حالت خارج نیست :
 - اگر کلیک انجام شده روی چک باکس موجب روشن شدن (on) یا true شدن یا همون تیک خوردن چک باکس شود رویدادی که تولید می شود عدد " ۱ " می باشد.
 - در غیر این صورت یعنی اگر کلیک صورت گرفته روی چک باکس موجب برداشته شدن تیک یا false شدن انتخاب چک باکس شود ، رویدادی که تولید می شود برابر عدد " ۲ " می شود.

```
if (e.getStateChange() == 1)
    label.setText("C++ Checkbox: " + " checked");
else
    label.setText("C++ Checkbox: " + "unchecked");
}
```

- خب در کد بالا هم گفته شده اگر حالت رویداد رخ دادن در اثر کلیک کردن روی چک باکس برابر عدد ۱ بود دستور زیر اجرا شود:

```
label.setText("C++ Checkbox: " + " checked");
```

- این دستور متن درون پرانتز را در label ما نمایش می دهد.

```
else
    label.setText("C++ Checkbox: " + "unchecked");
```

- در غیر این صورت یعنی اگر حالت رویداد رخ داده برابر عدد ۲ بود متن درون پرانتز را در label نمایش بده.
- سایر دستورات درون کد هم مشابه هستش که درموردشون در این جلسه و جلسات قبل توضیح داده شده است. تنها راه یادگیری این مفاهیم اینه که به خودتون سخت نگیرید!! شکل قالب دریافت رویدادهای این چک باکس بد قیافه همینه! در هر مکان و هر زمان خواستید در برنامه تون چک باکس بسازید و براش عمل خاصی رو تعریف کنید باید از این

دستورات ثابت استفاده کنید که نیاز به حفظ کردنشون هم نیست!! نیاز داشتید کافیه با ی جست و جوی کوتاه این کدها رو پیدا کنید و ازشون استفاده کنید! تنها شما فقط باید یادبگیرید که دقیقا چه میخواید!!!!

ی برنامه ساده سفارش غذا!!!!

```
package swing_javalike;

import javax.swing.*;

import java.awt.event.*;

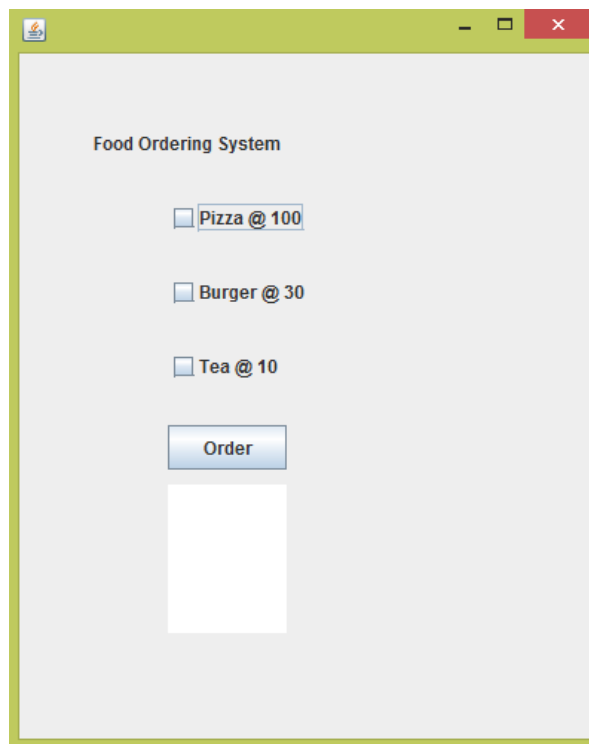
public class FoodOrder extends JFrame implements ActionListener {
    JLabel l;
    JCheckBox cb1, cb2, cb3;
    JButton b;
    JTextArea show;

    FoodOrder() {
        show = new JTextArea();
        show.setBounds(100, 290, 80, 100);
        show.setEditable(false);

        l = new JLabel("Food Ordering System");
        l.setBounds(50, 50, 300, 20);
        cb1 = new JCheckBox("Pizza @ 100");
        cb1.setBounds(100, 100, 150, 20);
        cb2 = new JCheckBox("Burger @ 30");
        cb2.setBounds(100, 150, 150, 20);
        cb3 = new JCheckBox("Tea @ 10");
        cb3.setBounds(100, 200, 150, 20);
        b = new JButton("Order");
        b.setBounds(100, 250, 80, 30);
        b.addActionListener(this);
        add(l);
        add(cb1);
        add(cb2);
        add(cb3);
        add(b);
        add(show);
        setSize(400, 500);
        setLayout(null);
        setVisible(true);
        setDefaultCloseOperation(EXIT_ON_CLOSE);
    }
}
```

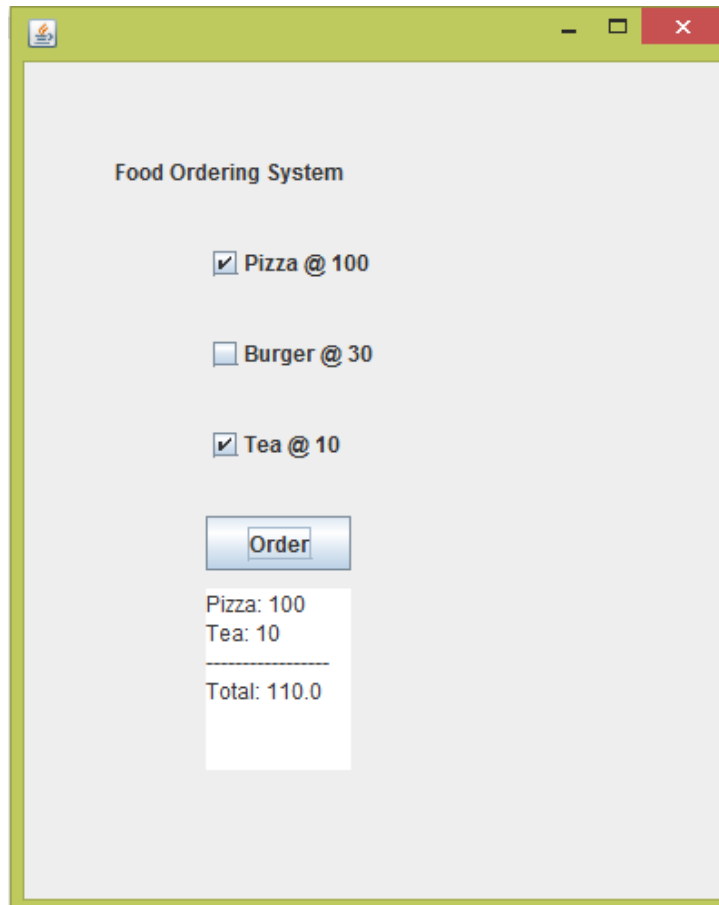
```
}  
  
public void actionPerformed(ActionEvent e) {  
    float amount = 0;  
    String msg = "";  
    if (cb1.isSelected()) {  
        amount += 100;  
        msg = "Pizza: 100\n";  
    }  
    if (cb2.isSelected()) {  
        amount += 30;  
        msg += "Burger: 30\n";  
    }  
    if (cb3.isSelected()) {  
        amount += 10;  
        msg += "Tea: 10\n";  
    }  
    msg += "-----\n";  
  
    show.setText(msg + "Total: " + amount);  
}  
  
public static void main(String[] args) {  
    new FoodOrder();  
}  
}
```

خروجی: خروجی هنگام اجرای بصورت تصویر (۸) خواهد بود:



تصویر(۸)

- این برنامه از سه چک باکس که نام غذا به همراه قیمت، کنار آنها درج شده است و یک دکمه **button** که با کلیک کردن روی آن غذا سفارش داده می شود و یک ناحیه سفید رنگ **TextArea** که فاکتور غذا را نمایش می دهد، تشکیل شده است.
- با کلیک کردن روی هر چک باکس فاکتور آن که شامل نام غذا و مجموع قیمت هستش نمایش داده می شود ما برای تست روی گزینه **pizaa** و **Tea** تیک میزنیم و روی دکمه **order** کلیک می کنیم، خروجی برنامه بصورت تصویر(۹) می باشد:



تصویر(۹)

توضیحات:

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

۱. برای استفاده از کلاس های آماده گرافیک این پکیج را `import` کرده ایم.

```
import java.awt.event.*;
```

۲. برای استفاده از رویدادها و اکشن ها در جاوا از این کتابخانه استفاده کرده ایم.

```
public class FoodOrder extends JFrame implements ActionListener {
```

۳. برای دسترسی به ویژگی ها و متدهای کلاس `JFrame` و همچنین پیاده سازی متدهای اینترفیس `ActionListener` کلاس خود `extends` به کلاس `JFrame` و `implements` به کلاس `ActionListener` کرده ایم.

```
JLabel l;
JCheckBox cb1, cb2, cb3;
JButton b;
JTextArea show;
```

۴. اجزای گرافیکی خود را در بدنه کلاس تعریف کرده ایم. چرا در سازنده کلاس مستقیم تعریف و از آنها شی ایجاد نکردیم؟ چون قصد داریم در تمامی مکان های کلاس از آنها استفاده کنیم و اگر تنها در سازنده کلاس آنها را تعریف میکردیم تنها دسترسی به این اجزای گرافیکی محدود به سازنده کلاس میشد.

1.	FoodOrder() {
2.	show = new JTextArea();
3.	show.setBounds(100, 290, 80, 100);
4.	show.setEditable(false);
5.	
6.	l = new JLabel("Food Ordering System");
7.	l.setBounds(50, 50, 300, 20);
8.	cb1 = new JCheckBox("Pizza @ 100");
9.	cb1.setBounds(100, 100, 150, 20);
10.	cb2 = new JCheckBox("Burger @ 30");
11.	cb2.setBounds(100, 150, 150, 20);
12.	cb3 = new JCheckBox("Tea @ 10");
13.	cb3.setBounds(100, 200, 150, 20);
14.	b = new JButton("Order");
15.	b.setBounds(100, 250, 80, 30);

۱. سازنده کلاس که اجزای گرافیکی را در آن ایجاد کرده ایم.

۲. ساخت شی از کلاس JTextArea

۳. تعیین مختصات و ابعاد شی ایجاد شده از کلاس JTextArea

۴. قابلیت ویرایش TextArea خود را غیرفعال یا قفل کرده ایم.

۶. ایجاد شی از کلاس JLabel و قرار دادن یک متن به عنوان پارامتر سازنده آن که هنگام اجرای برنامه این متن در لیبل نمایش داده می شود.

۷. تعیین مختصات و ابعاد شی label خود

۸. ایجاد یک شی از کلاس JCheckBox و یک متن به عنوان پارامتر سازنده آن داده ایم. این متن در کنار چک باکس ما نمایش داده می شود.

۹. تعیین مختصات چک باکس cb1

۱۰. ساخت شی دوم از کلاس JCheckBox و تعیین متن سازنده آن

۱۱. مشخص کردن مختصات و ابعاد شی cb2

۱۲. ساخت شی سوم از کلاس JCheckBox و تعیین متن سازنده آن

۱۳. مشخص کردن ابعاد و مختصات شی cb3

۱۴. ایجاد شی از کلاس JButton و جایگزین کردن یک متن به عنوان پارامتر سازنده آن که نقش نام باتن ما را بازی خواهد کرد.

۱۵. تعیین مختصات و ابعاد شی b

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

۵. این متد برای اضافه کردن یک رویداد و اکشن به دکمه button ما می باشد. این متد یک شی از کلاسی که implements به اینترفیس ActionListener شده است را به عنوان پارامتر می پذیرد نظیر کلاس FoodOrder ، خب ما به جای شی سازی از کلاس FoodOrder با کلمه کلیدی this اشاره به شی کلاس FoodOrder می کنیم.

```
add(l);
add(cb1);
add(cb2);
add(cb3);
add(b);
add(show);
```

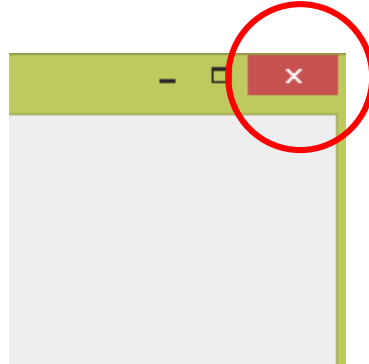
۶. با متد add اجزای گرافیکی خود را به فریم اضافه می کنیم. چرا از شی از نوع JFrame این متد را صدا نزدیم؟! چون کلاس ما کلاس JFrame را به ارث برده است و به تمام ویژگی ها و متدهای کلاس JFrame دسترسی دارد ، پس دیگر نیازی به صدا زدن این متد با شی از نوع کلاس JFrame نیست.

```
1. setSize(400, 500);
2. setLayout(null);
3. setVisible(true);
```

۱. برای تعیین ابعاد فریم ما استفاده شده است.
 ۲. چون از طرح بندی خاصی نمیخواهیم استفاده کنیم مقدار پارامتر این متد را null قرار داده ایم.
 ۳. برای نمایش فریم و تمام اجزای گرافیکی پارامتر این متد را true قرار دادیم. برای نمایش فریم و تمام اجزای گرافیکی صدا زدن این متد الزامی می باشد.


```
setDefaultCloseOperation(EXIT_ON_CLOSE);
```

- اگر قصد دارید با زدن دکمه ضربدر برنامه به طور کامل بسته شود از این دستور ثابت استفاده می کنیم. تصویر (۱۰)



تصویر (۱۰)

اگر از این دستور استفاده نکنیم! با زدن دکمه **close** یا همون شکل ضربدر موجود در فریم ، **frame** ما بسته می شود اما اجرای برنامه متوقف نمی شود و همچنان برنامه ما در کامپیوتر در حال اجرا می باشد.

1. public void actionPerformed(ActionEvent e) {
2. float amount = 0;
3. String msg = "";
4. if (cb1.isSelected()) {
5. amount += 100;
6. msg = "Pizza: 100\n";
7. }
8. if (cb2.isSelected()) {
9. amount += 30;
10. msg += "Burger: 30\n";
11. }
12. if (cb3.isSelected()) {
13. amount += 10;
14. msg += "Tea: 10\n";
15. }
16. msg += "-----\n";
17. show.setText(msg + "Total: " + amount);
18. }

توضیحات کد بالا براساس شماره هر خط داده شده است:

۱. این متد وظیفش دریافت رویدادها و اکشن هاست و با اجرای دستورات درون بدنه خود به این رویدادها و اکشن ها واکنش نشان می دهد.
 ۲. تعریف یک متغیر از نوع اعشار (float) که مجموع قیمت های غذا را در خود جای می دهد.
 ۳. تعریف یک متغیر از نوع رشته که اطلاعات و قیمت غذا ها را در خود جای میدهد.
 ۴. با شی ایجاد شده از کلاس چک باکس خود متد `isSelected` را صدا زده ایم. این متد یکی از مقادیر `true` یا `false` را برای ما برمی گرداند. اگر `true` بود یعنی چک باکس ما تیک خورده و در غیر این صورت چک باکس ما تیک نخورده است.
 ۵. اگر شرط `if` ما برقرار بود مقدار ۱۰۰ که قیمت `pizza` هست به مجموع قیمت ما اضافه می شود. چک باکس `cb1` مربوط به غذای `pizza` است.
 ۶. نام غذا به همراه قیمت نیز در متغیر `msg` ریخته می شود. هنگام نمایش فاکتور ، برای نشان داده این که مشتری چه غذایی و یا چه قیمتی سفارش داده به همچنین متغیری نیاز داریم.
- موارد ۸ تا ۱۵ نیز مثل موارد ۵ تا ۶ می باشد و فقط برای سایر غذا ها می باشد.

```
msg += "-----\n";
```

- برای تمایز بین قیمت ها و نام غذا با مجموع قیمت از این دستور استفاده کردیم.

```
show.setText(msg + "Total: " + amount);
```

- این دستور لیست غذاهای سفارش داده شده به همراه مجموع قیمت غذاها را در `textArea` ما نمایش می دهد.

امیدوارم از این آموزش راضی بوده باشید. اگر از این آموزش راضی بودید لینک کانال یا

سایت ما را به اشتراک بگذارید 😊

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.