

# بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

رخدادهای فرم در جاوا

مدرس: مهندس افشین رفوآ

## رخدادهای فرم در جاوا

در بخش قبل ماشین حساب خود را در جاوا طراحی کردید. در این بخش رویدادهایی برای آجکت های فرم جاوا را بررسی خواهیم کرد.

از لحاظ برنامه نویسی، یک رویداد به زمانی گفته می شود که یک اتفاق خاص بیفتد. برای فرم ها رویداد به معنای کلیک کردن دکمه ها، جابه جایی ماوس، وارد کردن متن به فیلد `text`، بستن برنامه و موارد بسیار دیگر.

رویدادها آجکت هایی در جاوا هستند. آنها از مجموعه ای از گروه ها گرفته می شوند که در `java.util.EventObject` ذخیره شده اند. وقتی که یک دکمه کلیک می شود، گفته می شود که آن دکمه منبع رویداد می باشد. اما عمل کلیک کردن یک آجکت رویداد را تولید می کند. سپس آجکت رویداد در جستجوی آجکتی است که منتظر کلیک کردن ماوس یا ضربه زدن به کلید یا هر اتفاق دیگری است که می تواند در یک فرم اتفاق بیفتد. به عنوان مثال یک فیلد `text` می تواند منتظر ضربه ی یک کلید در صفحه کلید باشد. یا باکس `drop down` می تواند منتظر آیتم انتخاب شده در باکس باشد.

آجکت های مختلفی رویدادهای متفاوتی به وجود می آورند. برای یک دکمه، رویدادی که شروع می شود `ActionListener` می باشد. برای یک فیلد `text` این رویداد `KeyEvent` می باشد. به موارد مشابه فکر کنید، آجکت رویداد مسئول انتقال پیام ها بین آجکت های فرم می باشد که رویدادی بر روی آنها اتفاق افتاده و یا آجکت هایی که منتظر رویداد هستند.

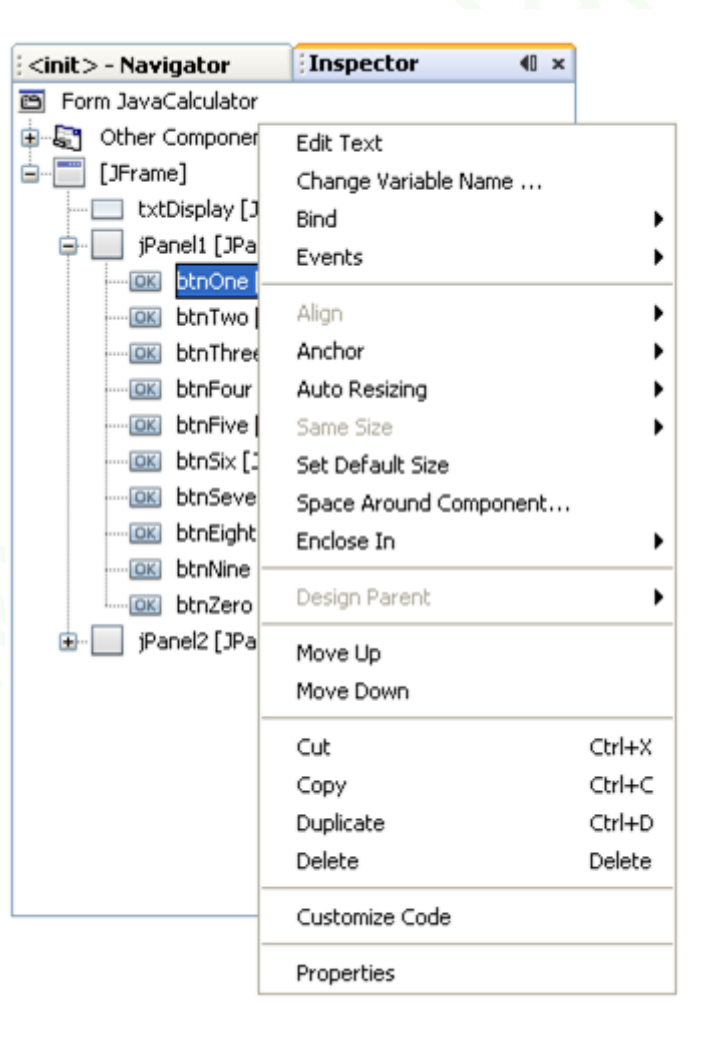
اگر همه ی اینها کمی پیچیده به نظر رسیدند، نگران نباشید. اما چند مثال برنامه نویسی می تواند تمام موارد را واضح سازد. ابتدا صحبتی در مورد کار کردن ماشین حساب.

اگر بخواهیم 3 و 2 را بایکدیگر جمع کنیم (3+2)، ابتدا لازم است روی دکمه ی 3 کلیک کنیم. سپس عدد 3 روی فیلد `text` ظاهر خواهد شد. پس از آن دکمه ی به اضافه کلیک خواهد شد و این به برنامه در مورد این واقعیت که قصد جمع کردن چند مورد را داریم، پیغام خواهد داد و همچنین فیلد متن را برای عدد بعدی آماده خواهد کرد. عدد بعدی 2 می باشد و ما این مقدار را همراه با 3 ذخیره می کنیم. برای به دست آوردن

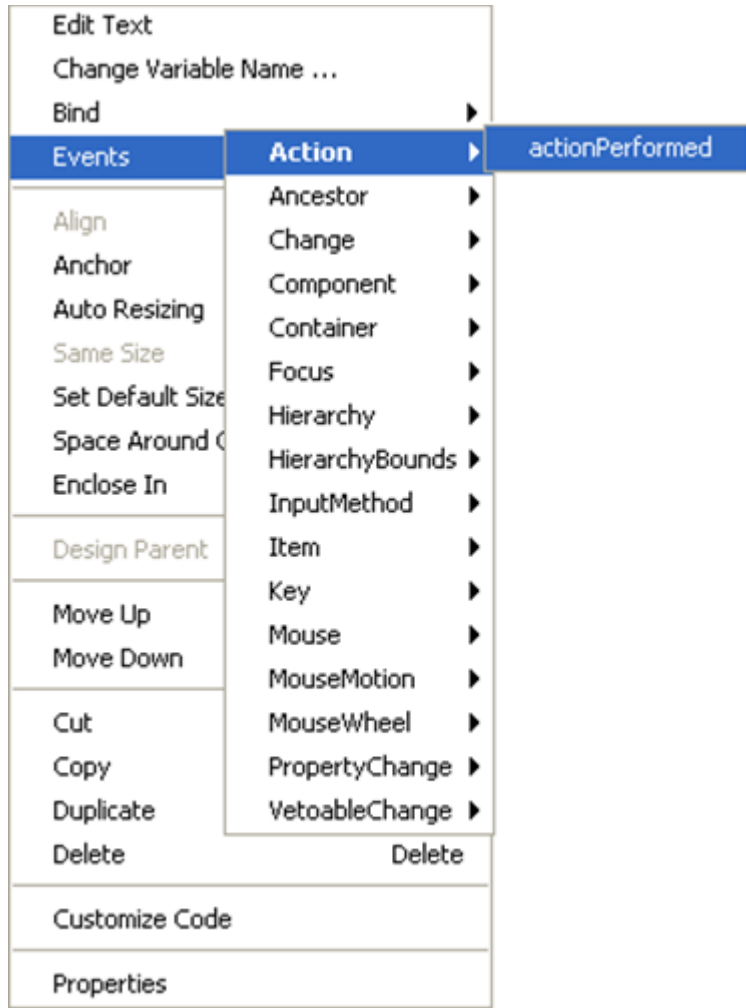
مجموع دکمه‌ی تساوی کلیک می‌شود و در واقع در اینجا اعداد ذخیره شده را به دست می‌آوریم (2, 3) و آنها را با یکدیگر جمع می‌کنیم. سرانجام پاسخ در فیلد متن ذخیره می‌شود.

اولین مشکل چگونگی دستیابی به اعداد روی دکمه‌ها می‌باشد. این کار را می‌توانیم با بازگرداندن پراپرتی متن مربوط به دکمه انجام دهیم. زمانی که متن را داشته باشیم می‌توانیم آن را در تکست باکس قرار دهیم. اما برای زمان کلیک کردن دکمه به یک **ActionEvent** احتیاج داریم.

در ویو **Design** در **NetBeans** دکمه‌ی شماره 1 خود را انتخاب کنید. اکنون نگاهی به پنجره **Inspector** در پایین سمت چپ، داشته باشید. دکمه‌ی **btnOne** خود را جایگذاری کنید. حالا کلیک راست کنید تا منوی زیر را مشاهده کنید.



از منوی پیش رو **Event** را انتخاب کرده و باز از منوی جدیدتر روی **Action** و سپس روی **actionPerformed** کلیک کنید.



وقتی که روی **actionPerformed** کلیک می کنید، محل کد برای **btnOne** ایجاد خواهید کرد.

```
private void btnOneActionPerformed(java.awt.event.ActionEvent  
evt) {  
    // TODO add your handling code here:  
}
```

اولین خط کمی بلند می باشد. اما این خط فقط یک متود با یک آبجکت **ActionEvent** بین پرانتزها می باشد.

وقتی دکمه کلیک می شود، هر آنچه بین پرانتزها نوشته ایم، اجرا خواهند شد.

Writing code **for** the numbers buttons on our Java Calculator

برای به دست آوردن متن از آبجکت فرم، می توانید از متود `getText` استفاده کنید. بنابراین برای دریافت متن از `btnOne`، می توانیم از کد زیر استفاده کنیم.

```
String btnOneText = btnOne.getText( );
```

برای دریافت متن از فیلد متن نیز می توان از کد زیر استفاده کرد.

```
String textFieldText = txtDisplay.getText( );
```

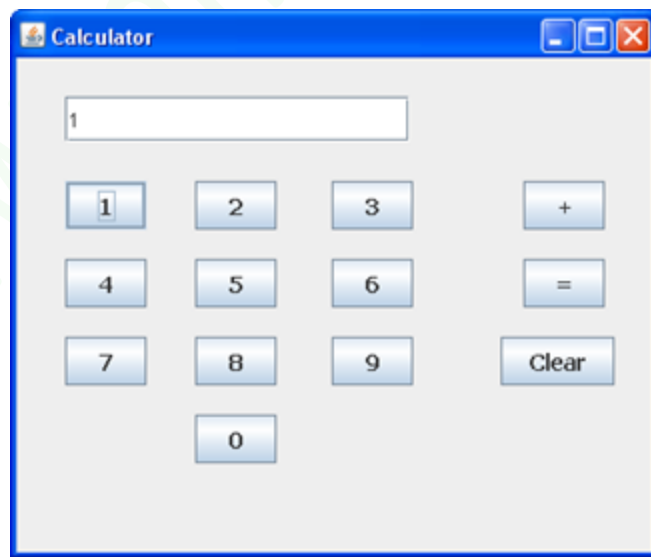
به هر حال اگر بخواهیم چیزی را در داخل فیلد متن قرار دهیم، متود مورد استفاده `setText` می باشد.

```
txtDisplay.setText( btnOneText );
```

آن را امتحان کنید. این کد را به محل کد اضافه کنید.

```
private void btnOneActionPerformed(java.awt.event.ActionEvent  
evt) {  
    String btnOneText = btnOne.getText( );  
    txtDisplay.setText(btnOneText);  
}
```

برنامه ی خود را اجرا کرده و آن را تست کنید. روی دکمه ی 1 کلیک کنید و پس از آن باید عدد 1 را در قسمت فیلد متن مشاهده کنید.



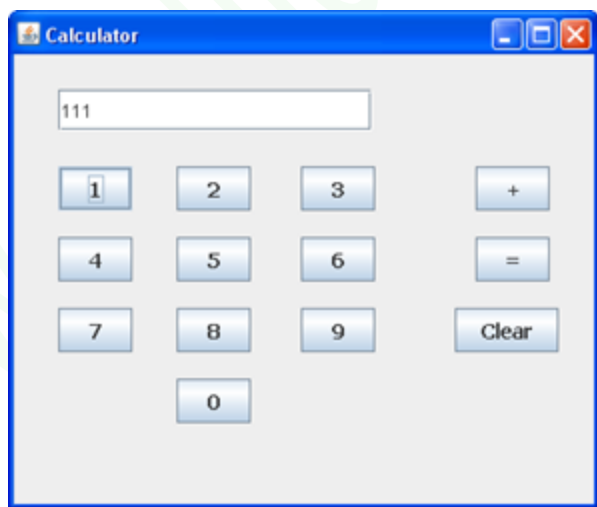
به هر حال مشکلی در اینجا وجود دارد. اگر بخواهید عدد 11 یا عدد 111 را وارد کنید چطور؟ وقتی که به طور مکرر روی یک دکمه کلیک می کنید، اتفاق دیگری نمی افتد. همیشه یک بار عدد 1 تایپ می شود و مهم نیست که چند بار این عدد را کلیک کردید.

دلیل آن این است که کد ما آنچه را از قبل در فیلد متن بوده، حفظ نمی کند و فقط عدد 1 را در آنجا قرار می دهد. اگر می خواستیم عدد 11 را تایپ کنیم، می بایست 1 اول را حفظ کنیم. برای انجام چنین کار باید متن را از فیلد متن گرفته و آن را با دکمه ی text ترکیب کنید. اولین خط کد را به این شکل اصلاح کنید.

```
String btnOneText = txtDisplay.getText() + btnOne.getText();
```

حالا می گوییم که متن را از فیلد متن گرفته و آن را با دکمه ی text ترکیب کنید. نتیجه را در متغیری به نام btnOneText ذخیره کنید.

برنامه ی خود را دوباره اجرا کرده و چند بار روی دکمه ی 1 کلیک کنید. پس از آن باید مشاهده کنید که فیلد متن مجموعه ای از 1 ها را نمایش خواهد داد.



می توانیم همان کد را برای همه ی اعداد روی ماشین حساب اضافه کنیم. به ویو design برگردید. روش دیگر اضافه کردن یک Action به یک دکمه این است که به جای راست کلیک کردن در بخش Inspector، بر روی دکمه دابل کلیک کنید. بنابراین روی دکمه ی عدد 2 دابل کلیک کنید. یک کد ایجاد خواهد شد، درست مانند دفعه ی قبل.

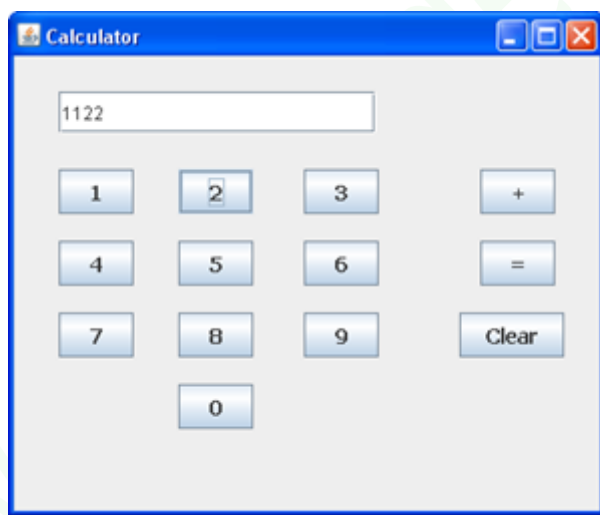
کد زیر را به آن اضافه کنید.

```
private void btnTwoActionPerformed( java.awt.event.ActionEvent
evt ) {
    String btnTwoText = txtDisplay.getText() +
btnTwo.getText();
    txtDisplay.setText( btnTwoText );
}
```

تنها تفاوت با کد قبل، نام متغیر **String** می باشد(اکنون **btnTwoText** نامیده می شود) و این واقعیت که ما

متن را از **btnTwo** دریافت می کنیم. بین پرازنهای **setText**، نام متغیر **String** را انتقال می دهیم.

دوباره برنامه ی خود را اجرا کنید. اکنون باید قادر به کلیک کردن روی دکمه های **1** و **2** باشید و همچنین متنی در فیلد متن ظاهر شود.



همان کد را برای تمام دکمه های ماشین حساب خود اضافه کنید. سپس برای دریافت **stub code** روی روی هر دکمه دابل کلیک کنید و کد دوخطی را به هر دکمه اضافه کنید. لازم است که نام متغیر **String** را تغییر دهید. در اینجا کد مربوط به دکمه ی **3** را مشاهده می کنید.

```
String btnThreeText = txtDisplay.getText() + btnThree.getText();
txtDisplay.setText( btnThreeText );
```

می توانید کدی را که دارید کپی و پیست کنید. سپس فقط نام متغیر **String** را تغییر دهید، نام دکمه بعد از بخش **btn** و بخش بین پرانتزهای **setText**. اگر موردی را مشاهده کردید که زیر آن خط کشیده شده، باید بدانید که در جایی اشتباهی رخ داده است.

وقتی که کار شما تمام شد، کد مربوط به همه ی دکمه ها باید مانند زیر باشد.

www.tahlildadeh.com



```

private void btnOneActionPerformed(java.awt.event.ActionEvent evt) {

    String btnOneText = txtDisplay.getText() + btnOne.getText();
    txtDisplay.setText(btnOneText);
}

private void btnTwoActionPerformed(java.awt.event.ActionEvent evt) {

    String btnTwoText = txtDisplay.getText() + btnTwo.getText();
    txtDisplay.setText(btnTwoText);
}

private void btnThreeActionPerformed(java.awt.event.ActionEvent evt) {

    String btnThreeText = txtDisplay.getText() + btnThree.getText();
    txtDisplay.setText(btnThreeText);
}

private void btnFourActionPerformed(java.awt.event.ActionEvent evt) {

    String btnFourText = txtDisplay.getText() + btnFour.getText();
    txtDisplay.setText(btnFourText);
}

private void btnFiveActionPerformed(java.awt.event.ActionEvent evt) {

    String btnFiveText = txtDisplay.getText() + btnFive.getText();
    txtDisplay.setText(btnFiveText);
}

private void btnSixActionPerformed(java.awt.event.ActionEvent evt) {

    String btnSixText = txtDisplay.getText() + btnSix.getText();
    txtDisplay.setText(btnSixText);
}

private void btnSevenActionPerformed(java.awt.event.ActionEvent evt) {

    String btnSevenText = txtDisplay.getText() + btnSeven.getText();
    txtDisplay.setText(btnSevenText);
}

private void btnEightActionPerformed(java.awt.event.ActionEvent evt) {

    String btnEightText = txtDisplay.getText() + btnEight.getText();
    txtDisplay.setText(btnEightText);
}

private void btnNineActionPerformed(java.awt.event.ActionEvent evt) {

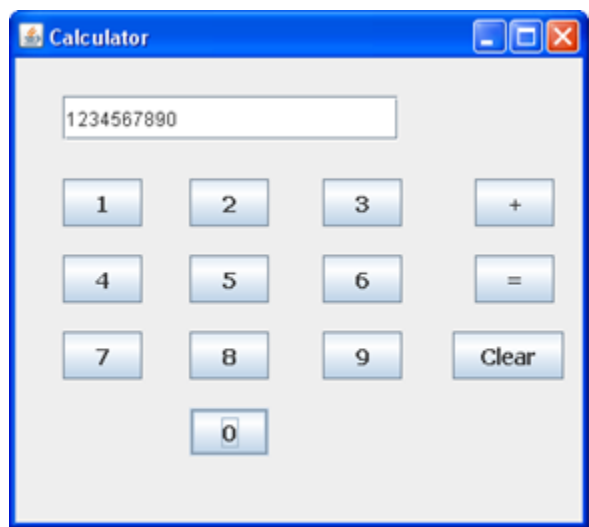
    String btnNineText = txtDisplay.getText() + btnNine.getText();
    txtDisplay.setText(btnNineText);
}

private void btnZeroActionPerformed(java.awt.event.ActionEvent evt) {

    String btnZeroText = txtDisplay.getText() + btnZero.getText();
    txtDisplay.setText(btnZeroText);
}

```

ماشین حساب خود را اجرا کرده و آن را تست کنید. باید بتوانید تمام اعداد را از 0 تا 9 وارد کنید.



در بخش بعد کد جاوا را برای دکمه ی به اضافه در ماشین حساب خواهید کرد.