Physics Department
Faculty of Science

search...

Home Profile Teaching Links Contact Me My Work

Newsflash

Joomla! 1.0 - 'Experience the Freedom'!. It has never been easier to create your own dynamic site. Manage all your content from the best CMS admin interface.

10th Step

Written by Mohd Khalid Kasmin

Monday, 17 September 2007

Understanding basics of Input/Output via GUI

Getting the input

Say we want to get two numbers, A and B, from the user of our application. The now very familiar interface would look somewhat like Figure 10.1.

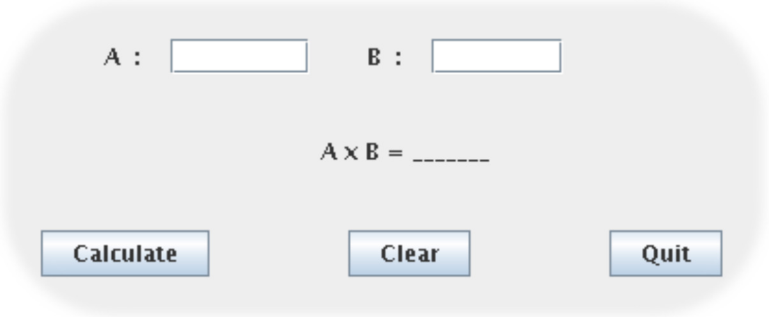
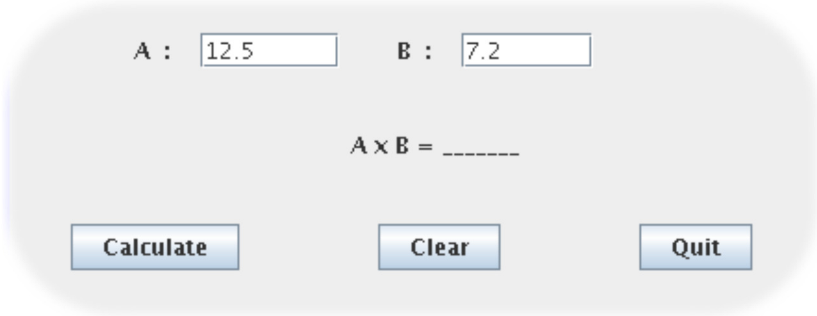


Figure 10.1

The letters A and B labeling the white boxes were made from JLabel components and the white boxes (made from JTextField components) are the input boxes where user of the application will typed in the values of A and B respectively. Let say that we have the situation like in Figure 10.2.


A : 12.5

B : 7.2

Figure 10.2

Here input box A has “12.5” in it and input box B has “7.2”. When the user click on button “Calculate”, execution of the application is passed to the btnCalculateActionPerformed() method. Thus in this method we need to put some codes to perform the required Calculation. What we need to do, in essence, is to multiply two numbers a and b and assign the result to c. In Java, we can do that multiplication if a, b, and c were first declared as double or float or int variables. So the code of the calculation should look like the following

```
    double a, b, c;

    a = ............ ;
    b = ............ ;
    c = a * b;
```

Clearly, variables a and b need to be assigned with values of typed double. Now looking back at inputboxes A and B, we made those boxes using the JTextField component, meaning that any data or value written in a JTextField is of typed Text or String even if the value displayed in the box is “12.5”. But first here, there is this task of getting to that Text (or String) values “12.5” and “7.2”. In Java, we can get to a text in JTextField component using the getText() method. So, to retrieve values in jTextField1 and jTextField2 we can use

```
    jTextField1.getText()
```

and

```
    jTextField2.getText().
```

But those values are still in text mode, and we need another method to convert the data mode from text to double. For that we can use the method Double.parseDouble(). Thus the whole process of retrieve text -> convert text to double -> assign to double variable can be done using

```
    a = Double.parseDouble( jTextField1.getText() );
```

and

```
    b = Double.parseDouble( jTextField2.getText() );
```

Hence, the code for performing the multiplication becomes

```
double a, b, c;

a = Double.parseDouble(jTextField1.getText());
b = Double.parseDouble(jTextField1.getText());
c = a * b;
```

Up to this point, we have the variable c as the result of the multiplication of a and b. So the next thing to do is to display this result on the JLabel that we have already prepared before:

A x B = _____

The text "A x B = ______" was made using JLabel component and its variable name is jLabel3. Since the variable c is of type double, and text property of JLabel is of type String, we need to convert it first to String data type. The method to convert a double to String is Double.toString(). So we can write the code as Double.toString(c). After getting the text, we can now use it to be the text displayed on jLabel3. To do that, we use the method setText(). Hence the appropriate code for displaying the result is

```
jLabel3.setText( "A x B = " + Double.toString(c) );
```

And the full code for our calculation becomes

```
double a, b, c;

a = Double.parseDouble(jTextField1.getText());
b = Double.parseDouble(jTextField1.getText());
c = a * b;
jLabel3.setText( "A x B = " + Double.toString(c) );
```

Last Updated (Tuesday, 18 September 2007)

< Prev **Next >**

[Back]

Home | Contact Us | News | Links |