In this video, we are going to continue the implementation of the view class by adding the text display to our simple calculator example.

0:00-0.07

Recall that in the second video, we created a JPanel that contained the JButtons for the calculator application.

0:07-0.17

We now want to add a numeric display to the calculator.

The component that we will use for the display is called a JTextField. A text field provides the user with a single line of text input. If you want more than one line of text input then you need a different text component.

0:17-0.32

This UML diagram shows the relationship between the View and its JTextField. The View has one JTextField that represents the calculator display. The JTextField has a public method that sets the alignment of the text. In our calculator, the display text is aligned on the right side of the text field.

The JTextField is a child class of JTextComponent. JTextComponent is the parent class for all of the components that are designed for text entry and editing. A text component provides accessor methods to get the text stored in the component, and mutator methods to set the text in the component.

0:32-1:24

We can create the JTextField by specifying its width as the number of columns. The calculator display aligns its text on the right hand side of the text field, so we need to set the horizontal alignment. When the view is created, there is no result to display, so we can set the text of the display to the empty string.

1:24-1:56

ECLIPSE
In the View constructor, we can create the JTextField. Once we have the JTextField, we can add it to the View by using the Container method add.

When we run the Main app, the text field does not appear. The reason this happens is because our View is using the default layout manager of JFrame, which is called a BorderLayout. We could use the BorderLayout, but instead, let’s use a different layout called a FlowLayout.

1:56-3:37

A FlowLayout tries to organize components in a container into a single row. If the width of a container is too small to fit all of the components then the layout will use multiple rows.

3:37-3:47

We can create the FlowLayout manager using the default constructor. There are other constructors which you can look up in the API of FlowLayout.

3:47-3:57

ECLIPSE
Let’s create the FlowLayout and set the layout manager of the view to the newly created FlowLayout. Now when we run the Main app we see the textfield above the buttons. If we make the calculator wider, we see that the flowlayout arranges the textfield and buttons horizontally. If we make the calculator narrower, we see that the flowlayout arranges the textfield and buttons vertically.

That brings us to the end of this video. In the next video, we’ll see how to create the menus for the calculator.

3:57-4:56