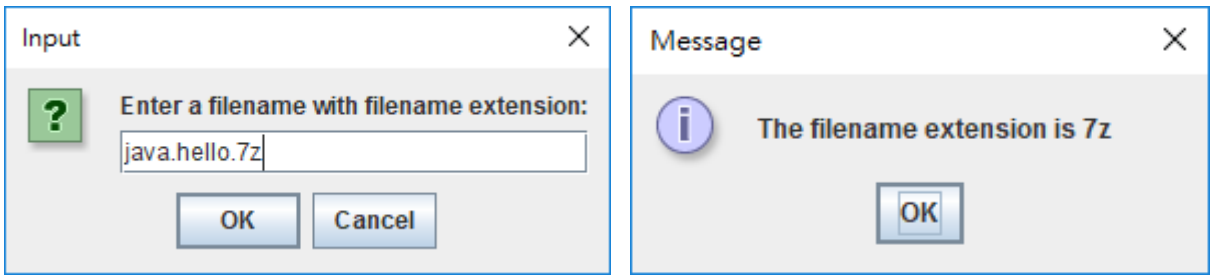
LAB 0307

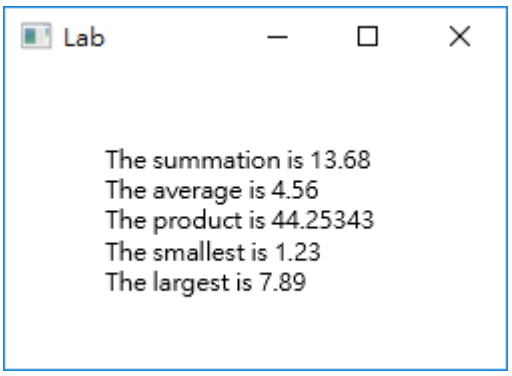
1. Write an application program that read a file name from the input dialog. The file name should has one . (dot) character in it, separating the file name from the file extension. Retrieve the file extension and display the result on the stage.
   For example:
   If the user inputs JavaTest1.html, you should output the file extension is html.
   If the user inputs Test2.java you should output the filet extension is java.

   Please write the program with following requirements.
   1) Define getFileExtension(…) to accept the filename and return the filename extension.
   2) Please do not use instance variable.
   3) The final result should be shown as follows.



2. Write a JavaFX program that inputs three float numbers from the user by using input dialog and displays the summation, average, product, maximum and minimum of the numbers on the stage.

   Please write the program with following requirements.
   4) Define “sum”, “average”, “product”, “max”, “min” functions in the program.
   5) The functions all have to return the results.
   6) The final result should be shown as follows.



3. Write a JavaFX program that inputs 5 data from the user by using input dialog and show the result with the inputting specification on the stage. The input data are 3 texts, font size, and the arrangement style.

   Note:

   1) To set the font size, you can refer to following example.

```
// The class: javafx.scene.text.Font
// text is an object of Text
text.setFont(Font.font(20));
```

   2) To compare two strings, it is suggested not to use "==" because the "==" operator checks to see if two objects are exactly the same object. It is suggested to use "equals()".

      Example:

```
String s = "test", t = "test";
if (s == t)           // Legal, but usually WRONG.
if (s.equals(t)) // RIGHT
if (s > t)        // ILLEGAL
if (s.compareTo(t) > 0) // CORRECT
```

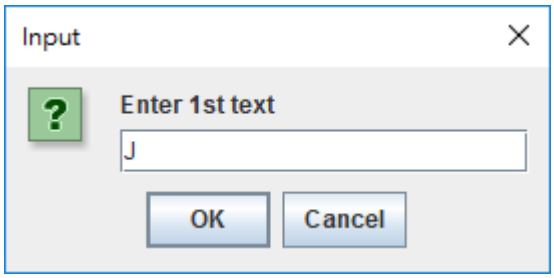
   3) To set the color of font, you can refer to following example.

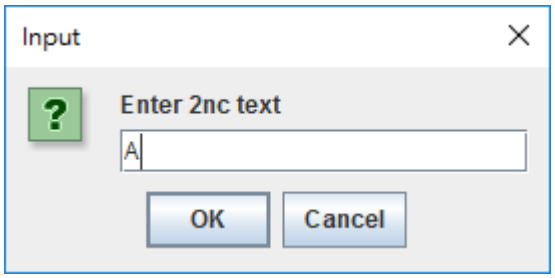
```
// The class: javafx.scene.paint.Color
// text is an object of Text
text.setFill(Color.RED);
```

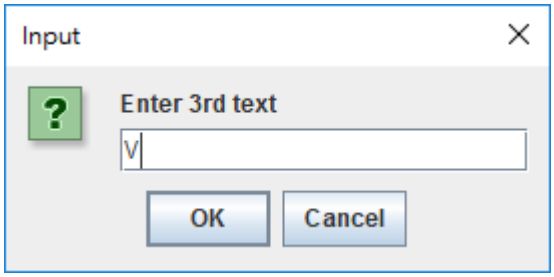
   4) To group multiple objects in one group, you can refer to following example.

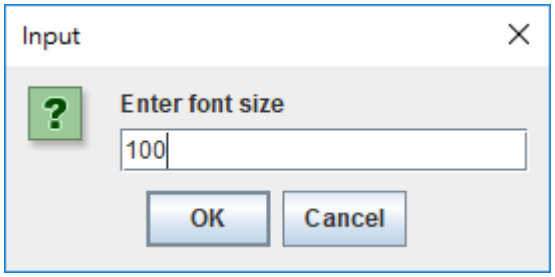
```
Group root = new Group(t1, t2, …);
```

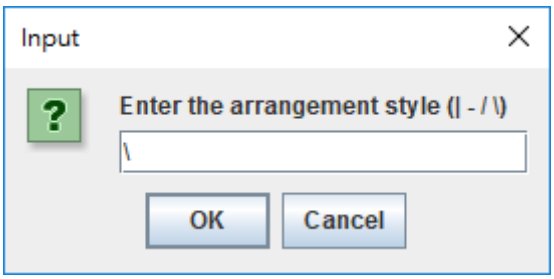
   5) The final result should be shown as follows.

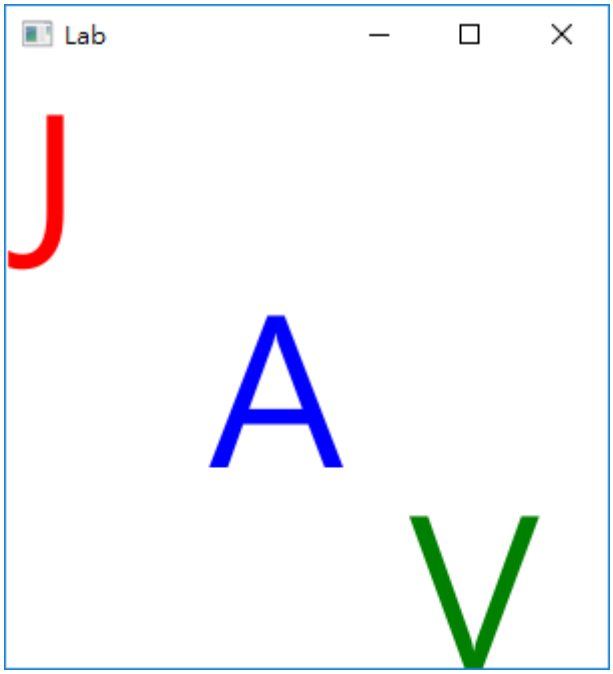




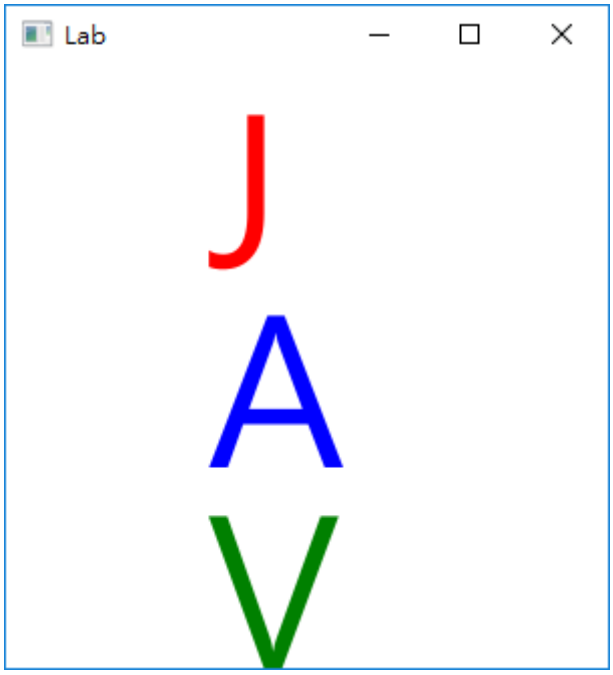








If the arrangement style is “|”, the result is as follows.



4. Please modify the code which you implemented in problem 3, let user input the data using only one input dialog and then output the result as follows. The input data in the one-line text area are 3 texts, font size, color, and arrangement style sequentially, which separated by one space.