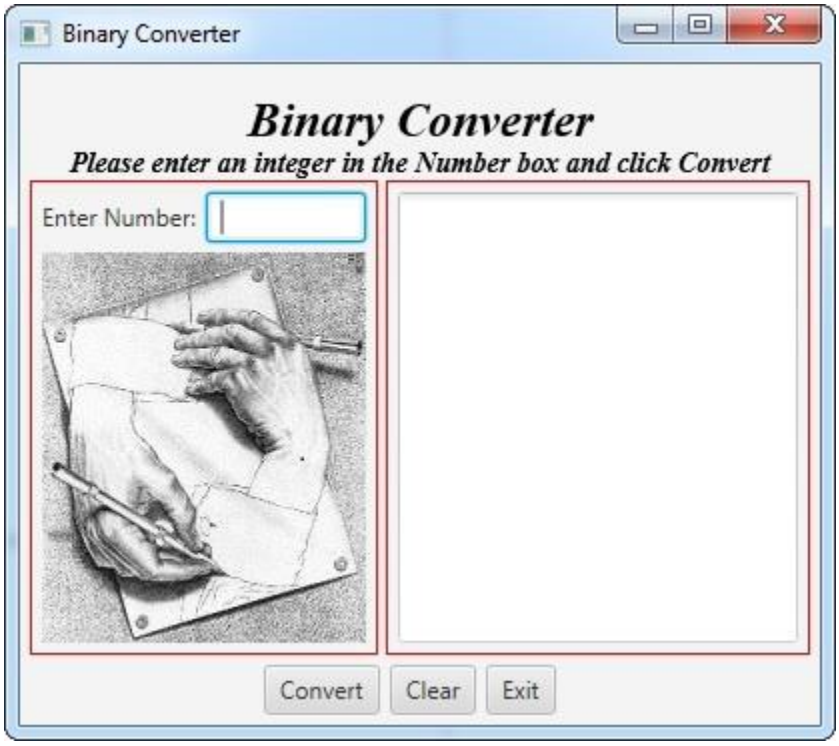
JavaFX Recursion

Assignment

This week, we will create a Binary Converter to convert base-10 integers to base-2 binary numbers.



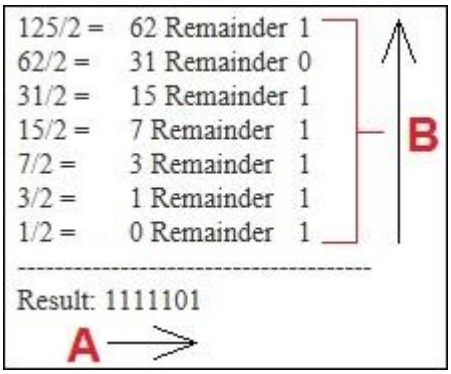
Requirements

- Your GUI may not appear exactly as the sample above. However, it should have all of the components/nodes shown in the image above
- The results area (*on right*) is a TextArea control
- The Convert button initiates the program to use the number entered in the "Enter Number" TextField to call a recursive method that determines the binary equivalent
- The Clear button clears the "Enter Number" TextField and the TextArea controls and requests focus back (*places cursor in*) to the "Enter Number" TextField
- The Exit button closes the program
- Use entry verification and alert popup messages to alert the user of:
  - No entry in the "Enter Number" TextField
  - Improper entry (*characters or strings instead of integer*)
- Include an icon/picture somewhere in your GUI. The sample shown is provided on the assignment page
- This assignment covers subjects from Chapter 1 - 18

Hints

Binary numbers are found by dividing a base 10 number by 2 continually until 0 is reached. Each iteration will produce a remainder of 1 or a 0. For example:

125 / 2 = 62 with a remainder of 1
62 / 2 = 31 with a remainder of 0
Etc.



Notice the order of the outcome (**A**) is the reverse order of the remainder outcomes (**B**) in our program.

Binary numbers are either 1s or 0s with their base 10 equivalents as such:

Position:	8	7	6	5	4	3	2	1
Value 10_{10}:	128	64	32	16	8	4	2	1

Therefore, as seen in the example above, we have the following result:

Value 10_2:	NA	1	1	1	1	1	0	1
Value 10_{10}:	NA	64	32	16	8	4	2	1

For every 1, we have a value so that:

1111101 => 64 + 32 + 16 + 8 + 4 + 1 = 125

The code snippets included may help but you are not required to utilize them as they are.

Your output should align similar to the output shown in the screen shot below. Sometimes with larger numbers, your results may skew slightly and that is acceptable.

You must use integer arithmetic. You cannot use long, double or floats with this exercise. Consider using class-wide declarations.

The resulting binary value(s) can be placed into an ArrayList. You may need to reverse that list before outputting the result.

Use .appendText() to add lines to the TextArea control.

Expected Output

You have creative freedom with this assignment. Your result is not required to appear exactly like the sample above or screen shot below. However; it must contain all of the same components and include the functionality described above and below in the screen shot example. Your GUI must also be visually appealing with consistent spacing/gapping.

Deliverables

Please zip your program and submit the zip file by the due date listed in the requirements.

Code Snippets

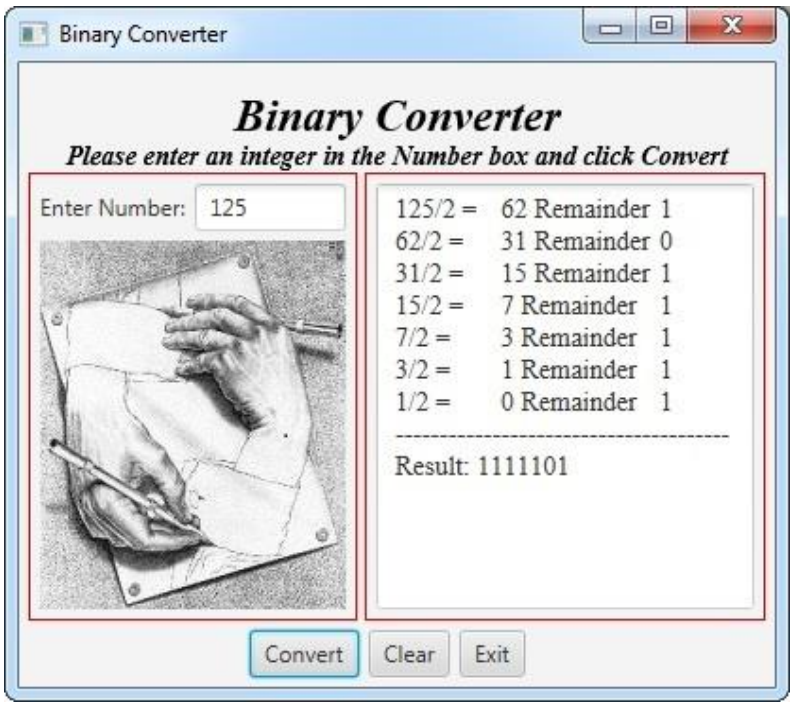
Pseudo Code for Recursive Binary Conversion:

```
public int ToBinary(int Number) {
   /* Define variable required in method */
   if (Number equals 0) {
      return;
   } else {
      Remainder = Number - 2 * (Number / 2);
      Convert remainder to a string
      Add remainder String to an ArrayList<String>
      Call a print method to print first result. Send values Number & Remainder
      ToBinary(Number / 2);      //Recurrsive call to method
   }
   return Number;
}
```

Alert Code Sample:

```
Alert alert = new Alert(Alert.AlertType.WARNING);
alert.setTitle("Warning");
alert.setHeaderText("Zip Code Format Error");
alert.setContentText("Zip codes should be in this format\n"
      + "nnnnn or nnnnn-nnnn\nPlease try again.");
alert.showAndWait();
```

Screen Shot:



Result using 125