

Initial Version

Asks the user what calculating mode they want

Creates initial integer variables

Checks what mode the user picked

Asks user for two numbers

Calls the right method of calculation based upon the mode

Writes the full calculation

Adding Method

Returns result

Subtraction Method

Returns result

Multiplication Method

Returns result

#Division Method

Returns result

Simplification (Removal of Duplication)

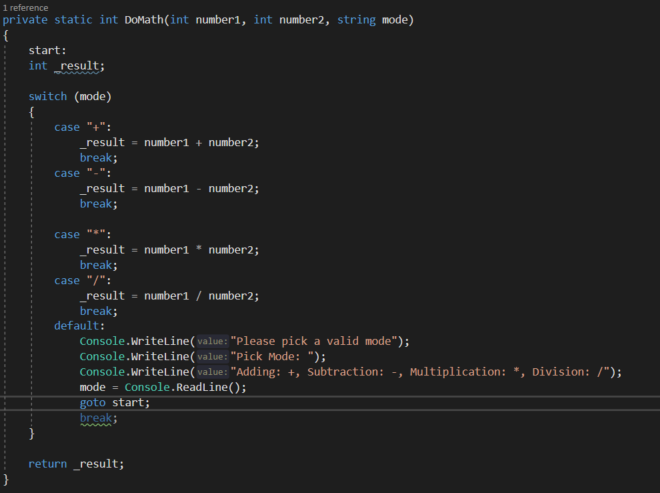


Moved case statement into a method below

Asks user once for two numbers

Sends the mode and the two numbers to the method

Writes full calculation

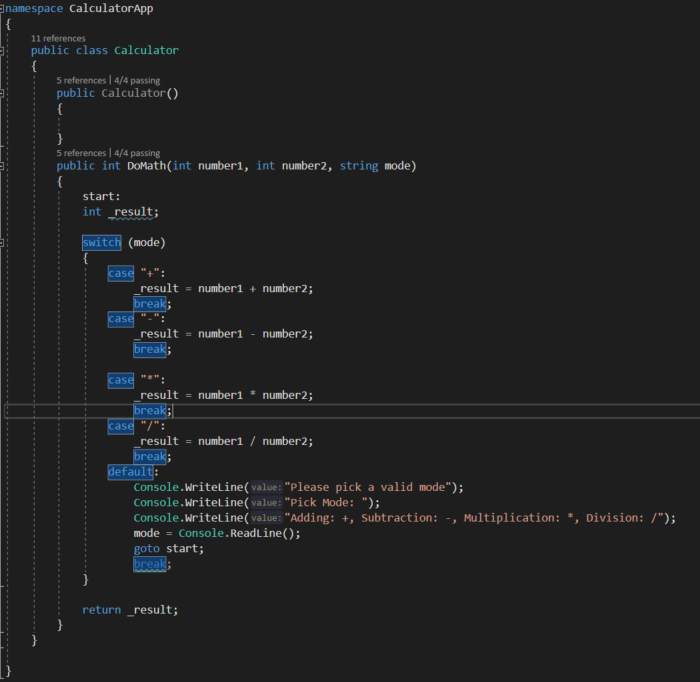


Case statement checks mode the user picked

Does sum based on the mode

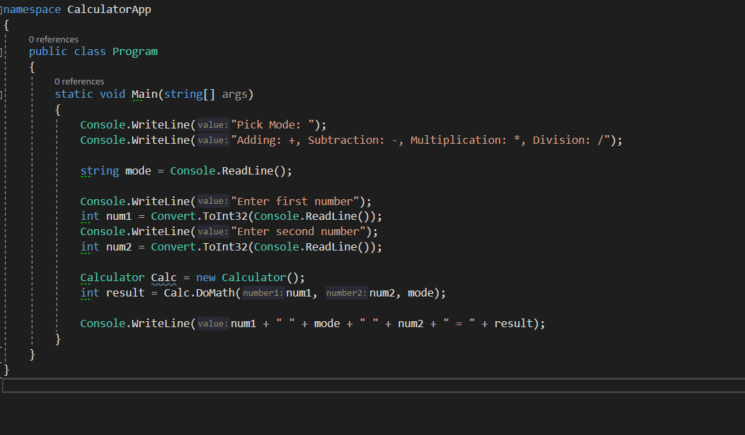
If user picks a mode which isn’t +,-,\* or / it will ask the user again until a valid mode is picked

Returns the answer of the sum

Initial Testing

For the testing to work I had to move the ‘DoMath’ method into its on class, which means when ever I call ‘DoMath’

It has to be called as shown below

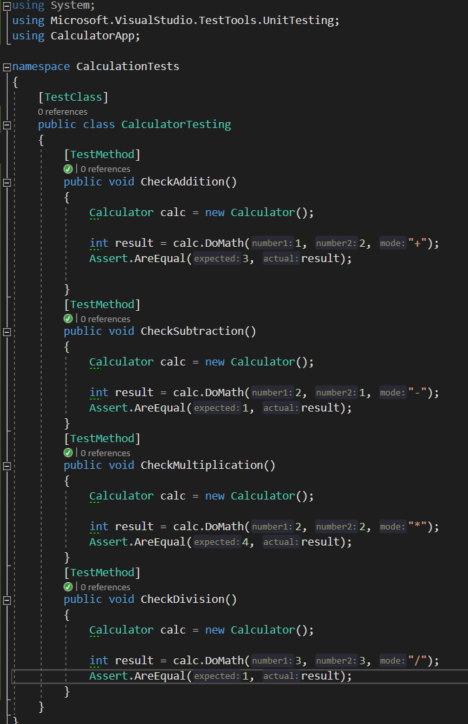
Here as ‘DoMath()’ is its own class, it has been called by making a new calculator instance

‘Calculator calc = new Calculator();”

Which creates a new instance, so now when I call ‘DoMath()’

Its called Calc.DoMath();

Testing (Checking for any errors)



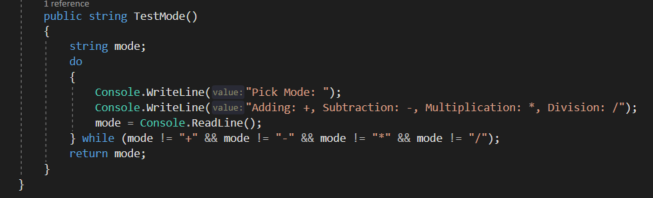
Testing Addition works

Testing Subtraction Works

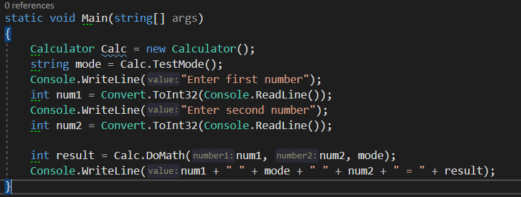
Testing Multiplication works

Testing Division works

Modified Version



As before, it would only check the mode the user had entered after the user entered two numbers. I have switched the order by adding this as a method in the new class file



As I changed how it checks the mode the user picked, I can now use the returned mode from the method to show the equation at the end, as otherwise if the method was Boolean it would return a null for ‘mode’

