Design Overview for Command-Line Calculator

Name: Luong Trac Duc Anh
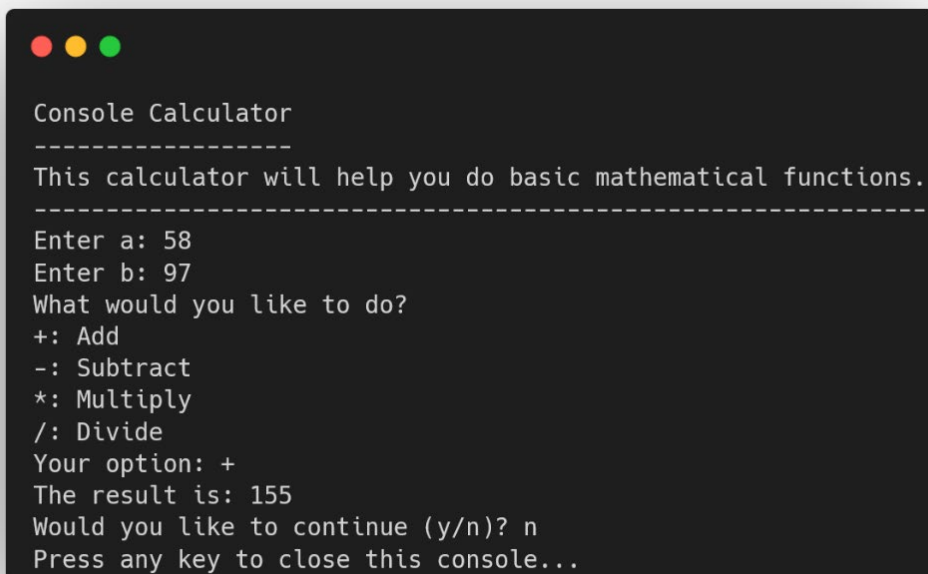Student ID: 103488117

Summary of Program

The program will take 2 number inputs from the user and do the basic mathematical functions, including addition, subtraction, multiplication, and division. After that, it will ask the user whether they want to do an equation again and loop through the program.

In the process of taking inputs from the user, the program can detect errors from wrong inputs to mathematical errors, with the use of exception and validation Boolean values to convert the input string to a double data type.

Sketch:

```
Console Calculator
------------------
This calculator will help you do basic mathematical functions.
----------------------------------------------------------------
Enter a: 58
Enter b: 97
What would you like to do?
+: Add
-: Subtract
*: Multiply
/: Divide
Your option: +
The result is: 155
Would you like to continue (y/n)? n
Press any key to close this console...
```

Required Roles

Describe each of the classes, interfaces, and any enumerations you will create. Use a different table to describe each role you will have, using the following table templates.

Table 1: <<role name>> details – duplicate

Responsibility	Type Details	Notes
a	Double	Placeholder for the 1st input

b	Double	Placeholder for the 2nd input
result <Calculator>	Double	Default value is not a number
option	String	Take in the command from the user and is used for the switch statement
endApp	Boolean	User's decision to loop through the program or not
input_a	String	Get the 1st input from the user
input_b	String	Get the 2nd input from the user
result <Program>	Double	The placeholder value is 0
a_valid	Double	The 1st input after validation of the correct data type
b_input	Double	The 2nd input after validation of the correct data type
ex	Exception	The error that arrives after the program execution
Program	Class	Gets user inputs, handles errors, loops through the program
Calculator	Class	Do the calculations based on the user's options

Table 2: <<enumeration name>> details

// no enumeration needed for now

Value	Notes