

HEXADECIMAL CALCULATOR (QT AND C++)

Ali Alperen Sönmez, Ramazan Burak Sarıtaş

2020400354, 2020400321

Project description: The program is a gui calculator taking hexadecimal input and producing hexadecimal output. The program is written in C++ language and uses QT software to create gui. (display, buttons for: numbers from 0 to F, +, -, = and Clr)

Classes

main:

QApplication is created in main class and program starts there.

calculator:

Calculator class is the class that includes the functional implementation of the calculator.

mainwindow:

Mainwindow class implements the layout of the calculator. (Buttons, Display and their alignment)

Methods

- 1) **exponential:** This method is used in the process of calculating the decimal value of a hexadecimal number input. Since C++ standart library does not contain exponential function, we implemented it manually.
- 2) **ConvertNumber:** This method is used to convert a hexadecimal number to its decimal value when it is required to calculate a result.
- 3) **OperationAdd:** This method takes 2 hexadecimal numbers in QString format as argument, converts them to decimal by the help of ConvertNumber method mentioned above, performs an addition operation, converts the result to its hexadecimal representation and returns the result in the QString format.
- 4) **OperationSub:** Similarly, this method takes 2 hexadecimal numbers in QString format as argument, converts them to decimal by the help of ConvertNumber method mentioned above, performs a subtraction operation, converts the result to its hexadecimal representation and returns the result in the QString format.
- 5) **NumPressed:**
This method detects if any number button (0, 1, ..., D, E, F) is pressed and displays

the buttons assigned value on display. Basically adds the pressed buttons value to end of the present value on the display by string concatenation.

- 6) **OperationPressed**: This method detects if an operation button is pressed and performs necessary actions depending on the operation that is pressed. Both addition and subtraction operations are handled under this method. Distinction between the operations are detected by global boolean variables `addPressed` and `subPressed`. These variables take the value `true` if the operation button related to them is pressed.

If the user decides to calculate a result of more than 2 numbers, the method performs the operation and sets the displayed value to the result of first operation when the second operation button is pressed.

- 7) **EqualsPressed**: This method detects if the equals button is pressed, performs the last operation and displays the result value on display.
- 8) **ClearPressed**: This method detects if the `Clr` button is pressed. In the implementation, pressing `Clr` terminates the whole operation prior to the `Clr` button. Therefore, when the `Clr` button is pressed, both `displayValue` and `stored currentValue` resets along with `addPressed` and `subPressed` variables taking the value `false`. User has to input numbers and operations from the beginning in order to make calculations.

Design Choices:

1) An input like `"1+2=="` results in: 3. Program does not redo the operation when equals button is pressed more than once.

2) When there is more than one operation, program calculates the results along the way. For example, when it takes `"1+2+3="` as an input, the program displays the result of `1+2=3` as soon as the second `+` operation is pressed. It does not wait until the equals button in order to do all of the calculations.

3) The program does not operate when it comes across operations one after another. As an example, input `"1+2+-+--+-="` will only result in 3.

Moreover, In order for operations to perform, there must be a number input after them. For example, "input `"1+2---++-+-----+3="` will result in 6. Because the operations bolded out are not taken into account.

4) For negative numbers, if the user inputs `"10 + -12="` or `"10 + -12 + ..."`, negative sign in front of the number 10 will not be displayed in the process of input. However, result will be displayed after the equals or operation button as `"-2"`. This is because program processes input `"10 + -12"` as `"10 + - 12"`, which results in the operation of `"10 - 12"` as mentioned above in the 3rd design choice explanation.