

Lab 4: Python GUI Programming Report

學號：0811562 姓名：何祁恩

1. 請簡述 Python Tkinter GUI 程式主體架構 (如何引入模組.宣告主視窗及物件...等等)

首先，透過 `import tkinter as tk` 引入套件。而後使用 `tk.Tk()` 創立主視窗，在最後要加上 `mainloop` 以持續進程式。

建立物件，常見的物件有 `label`, `button`, `frame` 等等。

可以透過 `tk.Label(...)`, `tk.Button(...)`, `tk.Frame(...)` 建立相對的物件。

❑ `label = tk.Label([window], text, img, ...)`

可以在指定的 `window` 顯示文字或是圖像。

❑ `button = tk.Button([window], text, command)`

可以在 `command` 的地方使用 `lambda function`，將 `command` 變為執行特定的 `function`，較為方便。

❑ `frame = tk.Frame(window)`

用來在視窗中放入一個矩形，可以當成 `container` 放入各式不同的物件，在排版上也可以指定特定的 `frame`，如此一來也能夠比較精準的指定所建立的物件要放入視窗的哪個 `frame` 中。

物件 `layout` 排版，常見的有 `pack`, `place`, `grid` 這幾種方法。

❑ `pack(side = 'orientation')`

所指定的物件，會以使用者指定的方向開始堆疊。

❑ `place`

所指定的物件，會放置在使用者指定的 `x, y` 座標上。

❑ `grid`

所指定的物件會放在使用者指定的行列中。

2. 請解釋 Python 計算機的程式碼(將程式貼上並加上註解)。
(請詳細標註能實現的功能，如：/0 能顯示錯誤訊息，0/任何數字為 0，第一次計算完的結果能繼續做運算，負數計算…等等)

```
1  import tkinter as tk
2
3  def SetValue(var):
4      tk.Label(f1, textvariable= var, height = 3).grid()
5
6  def Click(num):
7
8      #deal with 0 +-x/ with other number
9      op = ['+', '-', '*', '/']
10     if var.get() == '0' and num in op:
11         var.set('0' + num)
12
13     # user keep click 0 → only 0 remain in expression
14     # if divide by 0 → expression display error
15     elif var.get() == '0' or var.get() == 'Error':
16         var.set(num)
17
18     # otherwise, concatenate the click character with expression
19     else:
20         print(var.get())
21         var.set(var.get() + num)
22
23     # When user click 'C' → expression = '0'
24     def Clear():
25         var.set('0')
```

```
27     # When user click '=' → evaluate the expression
28     def Calculate():
29         expression = var.get()
30
31         # find out what is the first / second number
32         # and get the operation
33         for i in range(1, len(expression)):
34             if expression[i] in ['-', '*', '+', '/']:
35                 operation = expression[i]
36                 a = int(expression[:i])
37                 b = int(expression[i+1:])
38                 break
39
40         #switch case to calculate the result
41         match operation:
42             case '+':
43                 var.set(a + b)
44             case '-':
45                 var.set(a - b)
46             case '*':
47                 var.set(a * b)
48             case '/':
49                 if b == 0:
50                     var.set('Error')
51                 else:
52                     var.set(a // b)
53
```

```

55 if __name__ == "__main__":
56     window = tk.Tk()
57     window.title("Lab4")
58
59     f1 = tk.Frame(window)
60     f2 = tk.Frame(window)
61     f1.pack()
62     f2.pack()
63
64     var = tk.StringVar()
65     var.set('0')
66     SetValue(var)
67
68     # button layout with grid
69     btn7 = tk.Button(f2, text = '7', borderwidth = 5, width = 6, height = 2, command = lambda: Click('7')).grid(row = 0, column = 0)
70     btn8 = tk.Button(f2, text = '8', borderwidth = 5, width = 6, height = 2, command = lambda: Click('8')).grid(row = 0, column = 1)
71     btn9 = tk.Button(f2, text = '9', borderwidth = 5, width = 6, height = 2, command = lambda: Click('9')).grid(row = 0, column = 2)
72     btn_times = tk.Button(f2, text = 'x', borderwidth = 5, width = 6, height = 2, command = lambda: Click('*')).grid(row = 0, column = 3)
73     btn4 = tk.Button(f2, text = '4', borderwidth = 5, width = 6, height = 2, command = lambda: Click('4')).grid(row = 1, column = 0)
74     btn5 = tk.Button(f2, text = '5', borderwidth = 5, width = 6, height = 2, command = lambda: Click('5')).grid(row = 1, column = 1)
75     btn6 = tk.Button(f2, text = '6', borderwidth = 5, width = 6, height = 2, command = lambda: Click('6')).grid(row = 1, column = 2)
76     btn_minus = tk.Button(f2, text = '-', borderwidth = 5, width = 6, height = 2, command = lambda: Click('-')).grid(row = 1, column = 3)
77     btn1 = tk.Button(f2, text = '1', borderwidth = 5, width = 6, height = 2, command = lambda: Click('1')).grid(row = 2, column = 0)
78     btn2 = tk.Button(f2, text = '2', borderwidth = 5, width = 6, height = 2, command = lambda: Click('2')).grid(row = 2, column = 1)
79     btn3 = tk.Button(f2, text = '3', borderwidth = 5, width = 6, height = 2, command = lambda: Click('3')).grid(row = 2, column = 2)
80     btn_plus = tk.Button(f2, text = '+', borderwidth = 5, width = 6, height = 2, command = lambda: Click('+')).grid(row = 2, column = 3)
81     btn0 = tk.Button(f2, text = '0', borderwidth = 5, width = 6, height = 2, command = lambda: Click('0')).grid(row = 3, column = 0)
82     btnC = tk.Button(f2, text = 'C', borderwidth = 5, width = 6, height = 2, command = lambda: Clear()).grid(row = 3, column = 1)
83     btn_equal = tk.Button(f2, text = '=', borderwidth = 5, width = 6, height = 2, command = lambda: Calculate()).grid(row = 3, column = 2)
84     btn_divide = tk.Button(f2, text = '/', borderwidth = 5, width = 6, height = 2, command = lambda: Click('/')).grid(row = 3, column = 3)
85
86     window.mainloop()

```

Click()幾個值得注意的點:

在 9~11 先檢查針對第一個數字是 0 進行操作:

Ex: $0 + 2$, $0 - 5$, $0 * 9$, $0 / 4$ ，使其在螢幕上能夠正確顯示。

在 13~16 針對假如一開始顯示的是 0 及 error 做操作，將新的數字取代 0 或是 error。

在 18~21 行則是把新輸入的資料與當前顯示在螢幕上的字串串接在一起。

Clear():


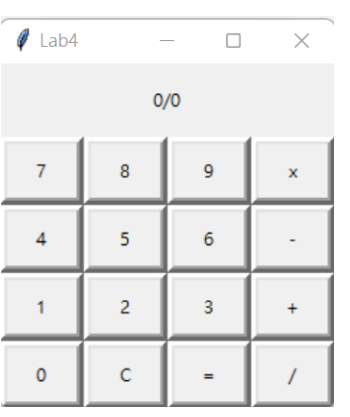

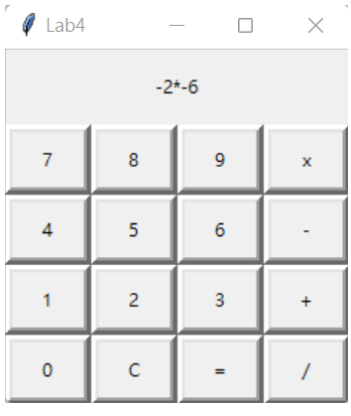


當使用者按下”C”，螢幕將會歸 0。

Calculate():

當使用者按下”=”，便會去執行當下 var.get()這個字串內部的運算。首先，先找到兩個數之間做的運算為何將其存在 operation 這個變數中，for loop 從 1 開始的原因是要避開第一個數是負的情形。而後將以 operation 這個 index 為切割線，前半部分為第一個數字，後半部分為第二個數字，透過 int()將字串轉為整數(負數同樣能這樣轉換)。而最後則是使用 switch case 針對不同的 operation 對兩個數字做不同的運算。值得注意的是除法的部分，當第二個數字為 0 的時候會顯示”Error”。

在 main 的 driven code 中，先初始化了主視窗，而後將其切割為兩個 frame，分別是 f1, f2。f1 是用來顯示數學運算式及運算結果的區域。而 f2 則是用來將數字按鈕即運算按鈕放置的矩形框架，而後將兩個 frame 上下 pack 在一起。接著，建立 16 個按鈕。並在最後加上 mainloop()以持續運行。

3. 結果:

<p>Button layout</p> 	<p>$0/0 = \text{Error}$</p> 	<p>$0/9 = 0$</p> 
<p>$-2 * -6 = 12$</p> 	<p>$-2 + 9 = 7$</p> 	<p>$-2 - -6 = 4$</p> 

4. 心得:

這次的 lab 看似簡單，但是要考慮各式各樣輸入的 pattern 並且都要確實做到還是需要花費一些時間。倘落要做到多數字多運算子的運算就必須使用到在資料結構學到的 stack，因為運算順序上還是要滿足數學的優先權。例如括號優先權最大，再來是乘除等等。就會複雜許多。但是也會讓這次 lab 增添不少難度。在上網查詢這次 lab 的相關資料時，看到有人甚至用 tkinter 做了一台工程用的計算機，非常瘋狂。如右圖。

<https://github.com/kostasthanos/Tkinter-Calculator>

