

Lab 4: Python GUI Programming Report

學號：109511286 姓名：蔡佩蓉

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）

a) 引入模組：

```
import tkinter as tk
```

b) 宣告主視窗：

```
window = tk.Tk()  
window.title("Lab4") # 設定視窗標題
```

c) 建立 Widget 物件：

- `label = tk.Label(window, text='Hello World!')`
建立文字或圖形標籤
- `button = tk.Button(window, text='hit me', command='xxx')`
建立按鈕元件，讓使用者在按下後執行某一些行為
- `frame= tk.Frame(window)`
建立框架用於組織和擺放視窗中的元件

d) 元件配置方法：

- `pack()/pack(side='top/bottom/left/right')`
將 widget 包裝到視窗上(或指定的方位)
[由 上/下/左/右 往 下/上/右/左]
- `grid(column=0, row=1)`
將視窗分成格狀，依據設定不同的 `row` 和 `column` 索引值，將元件放入格子中，一個元件可使用一格或多格，索引值預設從 0 開始，但也可以從其它數字開始計數。
- `place(x=10, y=30)`
可直接設定元件的左上方位位置要顯示在視窗上的座標，視窗左上角的座標為 `x=0`、`y=0`，向右 `x` 座標遞增，向下 `y` 座標遞增。

e) 主程式迴圈 (Main Loop)：

```
window.mainloop()  
# 啟動主程式迴圈，使 GUI 一直處於運行狀態，等待用戶操作
```

2. 請解釋 Python 計算機的程式碼(將程式貼上並加上註解)。
(請詳細標注能實現的功能，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)

```
# -*- coding: utf-8 -*-
"""
Created on Tue Nov 28 21:35:01 2023

@author: 109511286
"""

import tkinter as tk

def SetValue():
    # 設定label上顯示的字為var，把label放到第一個frame上面
    tk.Label(f1, textvariable=var, height=3).grid(column=0, row=1)

def Click(num): # 當按下按鈕時，在label上顯示相應的數字或運算符
    current = var.get() # 存label上顯示的數字或運算符，因此第一次計算完的結果能繼續做運算
    if current == 'ERROR':
        Clear()
        var.set(num)
    else:
        if current == '0': # 如果之前顯示的是0，可以直接更新為新數字或運算符
            if num in ['x', '/']:
                var.set(current + num)
            else:
                var.set(num)
        else:
            # 如果之前有數字，新輸入的數字或運算符會被追加在後面
            var.set(current + num)

def Clear(): # 清除label的內容
    var.set('0')

def Calculate():
    expression = var.get()

    try:
        num1 = ""
        num2 = ""
        op = ""
        for char in expression:
            # 判斷輸入的是數字或運算符
            if char.isdigit() or ((char == expression[0] or op != "") and char == '-'):
                # 讀到負號(不為運算符)或數字
                if op == "": # 且還沒讀到運算符號(可以是負數)
                    num1 += char
                else: # 讀到運算符號後(可以是負數)
                    num2 += char
            elif char in ['+', '-', 'x', '/']:
                op = char

        # 執行計算
        num1 = int(num1) # num1 str轉成int
        num2 = int(num2) if num2 else 0 # num2 str轉成int

        if op == '+': # 加法運算
            result = num1 + num2
        elif op == '-': # 減法運算
            result = num1 - num2
        elif op == 'x': # 乘法運算
            result = num1 * num2
        elif op == '/': # 除法運算
            if num2 == 0: # X/0，顯示錯誤訊息
                var.set('ERROR')
                return
            result = num1 // num2 # 包含0/X，會顯示為0

        # 將計算結果顯示在標籤上
        var.set(str(result))

    except ValueError:
        # 處理非整數輸入的情況
        var.set('ERROR')
```

```

if __name__ == "__main__":
    # 創建主視窗
    window = tk.Tk()
    window.title("Lab4")

    # 創建兩個frame用於放置標籤和按鈕
    f1 = tk.Frame(window) # 在上面放label顯示結果
    f2 = tk.Frame(window) # 在下面放button
    f1.pack()
    f2.pack()

    # 將最初的顯示數字設定為0
    var = tk.StringVar()
    var.set('0')

    SetValue()

    # 按鈕的配置
    buttons = [
        ('7', 0, 2), ('8', 1, 2), ('9', 2, 2), ('x', 3, 2),
        ('4', 0, 3), ('5', 1, 3), ('6', 2, 3), ('-', 3, 3),
        ('1', 0, 4), ('2', 1, 4), ('3', 2, 4), ('+', 3, 4),
        ('0', 0, 5), ('C', 1, 5), ('=', 2, 5), ('/', 3, 5)
    ]

    # 創建按鈕並設置對應的功能
    for (text, x, y) in buttons:
        if text == '=':
            btn = tk.Button(f2, text=text, borderwidth=5, width=6, height=2, command=Calculate)
        elif text == 'C':
            btn = tk.Button(f2, text=text, borderwidth=5, width=6, height=2, command=Clear)
        else:
            btn = tk.Button(f2, text=text, borderwidth=5, width=6, height=2, command=lambda t=text: Click(t))
            btn.grid(column=x, row=y)

    window.mainloop()

```

3. 心得

這段時間學習了 **Python** 中的 **Tkinter** 模組，這是一個強大的 **GUI** 工具包，讓我可以輕鬆建立和設計各種視窗應用程式。透過了解它的基本架構，我學會了如何建立主視窗、框架和各種 **GUI** 元件，以及如何配置它們的屬性和位置。透過事件處理函數的使用，我可以輕鬆地為按鈕和其他元件添加功能，讓它們能夠回應使用者的操作。

這段學習讓我更了解到 **GUI** 程式設計的流程，從設計視窗到配置元件和事件處理，整個過程需要仔細考慮用戶體驗和程式邏輯。我也發現了 **Tkinter** 提供的許多實用功能，例如變數監聽、元件排版等，這些讓程式碼更加靈活和易於管理。

經過這段學習，我更有信心去開發自己的應用程式，無論是簡單的小工具還是更複雜的應用，**Tkinter** 都為我提供了一個很好的起點。這段學習還讓我更加尊重 **GUI** 設計的複雜性，但同時也讓我感到興奮，因為我現在能夠用程式碼來實現自己的創意和想法。