資訊程式原理報告

組別： 第三組

組員： 資科所碩一 黃俐瑗

資科所碩一 周軒正

統計所碩二 方俊財

統計所碩二 魏良育

企管所博二 郭家富

計算機plus+

創作理念：

由上堂課所學啟發，

希望能做出一台更多功能之

計算機。

使用工具：

Spyder(python3.7)

UI介面

UI介面

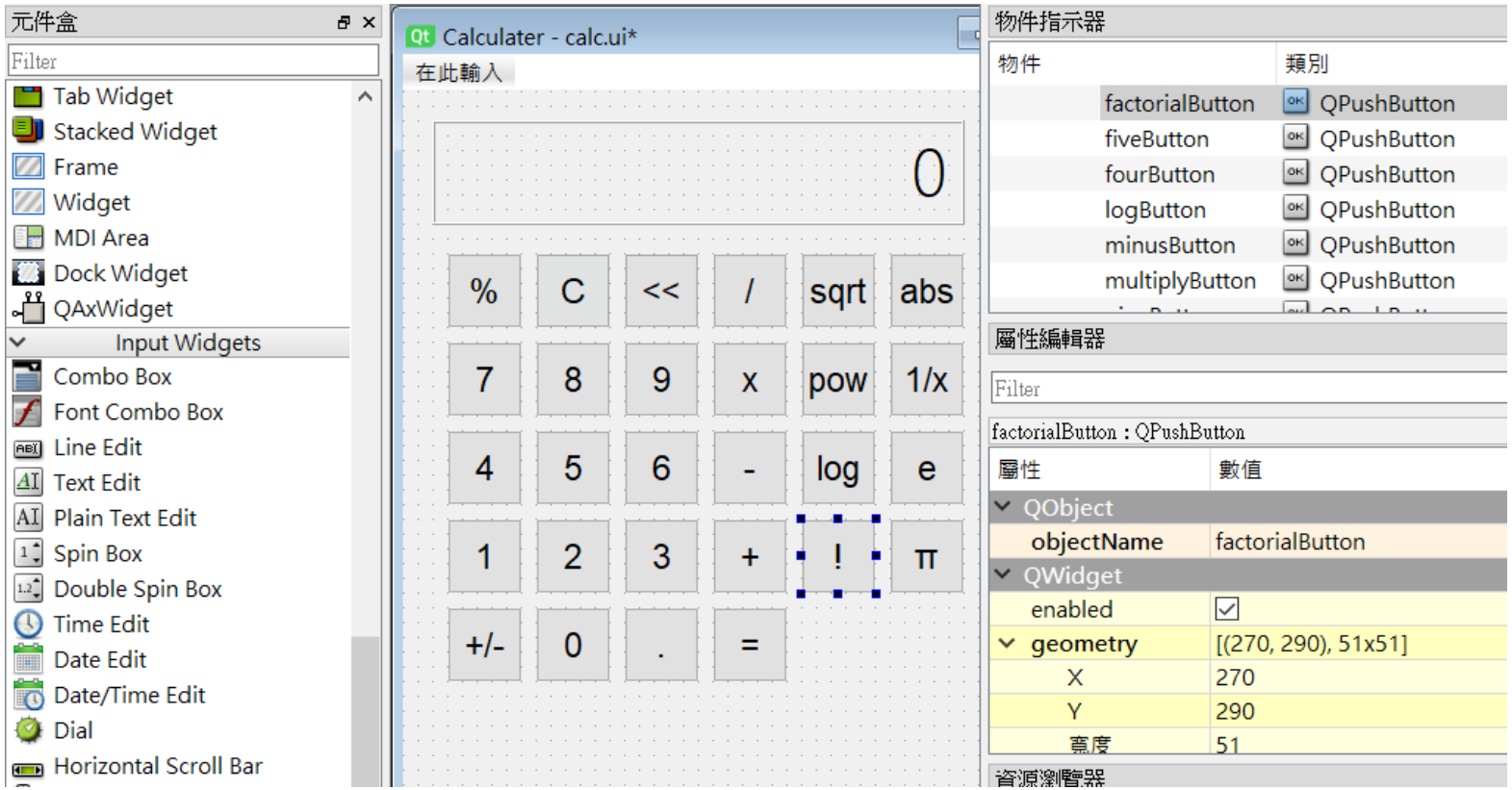
所需按鍵:

Vertical Layout

Horizontal Layout

Push Button

Label



轉檔

從ui檔轉成py檔

```
C:\Users\Lenovo>cd C:\Users\Lenovo\Desktop\上課啦\資訊課程原理\報告

C:\Users\Lenovo\Desktop\上課啦\資訊課程原理\報告>pyuic5 -o calculat.py    calc.ui
```

程式碼說明

主畫面對各按鍵的操控及排版指令

```
from PyQt5 import QtCore, QtGui, QtWidgets

class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(550, 559)

        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")

        self.outputLabel = QtWidgets.QLabel(self.centralwidget)
        self.outputLabel.setGeometry(QtCore.QRect(30, 20, 470, 71))
        font = QtGui.QFont()
        font.setPointSize(26)
        self.outputLabel.setFont(font)
        self.outputLabel.setFrameShape(QtWidgets.QFrame.Box)
        self.outputLabel.setFrameShadow(QtWidgets.QFrame.Raised)
        self.outputLabel.setAlignment(QtCore.Qt.AlignRight|QtCore.Qt.AlignTrailing|QtCore.Qt.AlignVCenter)
        self.outputLabel.setObjectName("outputLabel")

        self.percentButton = QtWidgets.QPushButton(self.centralwidget,clicked=lambda:self.press_it("%"))
        self.percentButton.setGeometry(QtCore.QRect(30, 110, 51, 51))
        font = QtGui.QFont()
        font.setPointSize(14)
        self.percentButton.setFont(font)
        self.percentButton.setObjectName("percentButton")
```

修改位置

程式碼說明

主畫面對各按鍵的操控及排版指令

```
self.absButton = QtWidgets.QPushButton(self.centralwidget,clicked=lambda:self.abs_it())
self.absButton.setGeometry(QtCore.QRect(330, 110, 51, 51))
font = QtGui.QFont()
font.setPointSize(14)
self.absButton.setFont(font)
self.absButton.setObjectName("absButton")
```

點選button會在介面產生的指令

按鍵的(x,y,w,h)

物件名稱

文字大小.字體

程式碼說明

def函式 :先import math套件，接著開始寫各判斷式

log、!(階層)介紹：

```
def math_it(self):
    global math
    screen = self.outputLabel.text()
    try:
        if "log" in screen:
            answer = round(math.log10(float(screen[3:])),4)
        elif "!" in screen:
            answer = round(math.factorial(float(screen[:-1])))
        else:
            answer=eval(screen)
        self.outputLabel.setText(str(answer))
    except:
        self.outputLabel.setText("ERROR")
```

程式碼說明

sqrt(開根號)、abs(絕對值)介紹：

```
def sqrt_it(self):
    try:
        screen = float(self.outputLabel.text())
        answer = round(math.sqrt(float(screen)),4)
        self.outputLabel.setText(str(answer))

    except:
        self.outputLabel.setText("ERROR")

def abs_it(self):
    try:
        screen = float(self.outputLabel.text())
        answer = abs(screen)
        self.outputLabel.setText(str(answer))

    except:
        self.outputLabel.setText("ERROR")
```

程式碼說明

pi(π)、exp(自然指數)介紹：

```
def pi_it(self):
    try:
        pi = round(math.pi,4)
        if self.outputLabel.text() == "0":
            self.outputLabel.setText("")
        self.outputLabel.setText(f'{self.outputLabel.text()}{str(pi)}')

    except:
        self.outputLabel.setText("ERROR")

def exp_it(self):
    try:
        exp = round(math.exp(1),4)
        if self.outputLabel.text() == "0":
            self.outputLabel.setText("")
        self.outputLabel.setText(f'{self.outputLabel.text()}{str(exp)}')
```

程式碼說明

三角函數介紹：

```
def sin_it(self):
    try:
        screen = float(self.outputLabel.text())
        sin = round(math.sin(screen*math.pi/180),4)
        self.outputLabel.setText(str(sin))
    except:
        self.outputLabel.setText("ERROR")

def cos_it(self):
    try:
        screen = float(self.outputLabel.text())
        cos = round(math.cos(screen*math.pi/180),4)
        self.outputLabel.setText(str(cos))
    except:
        self.outputLabel.setText("ERROR")

def tan_it(self):
    try:
        screen = float(self.outputLabel.text())
        tan = round(math.tan(screen*math.pi/180),4)
        self.outputLabel.setText(str(tan))
    except:
        self.outputLabel.setText("ERROR")
```

程式碼說明

三角函數介紹：

```
def cot_it(self):
    try:
        screen=float(self.outputLabel.text())
        tan=round(math.tan(screen*math.pi/180),4)
        cot=round(1/tan,4)
        self.outputLabel.setText(str(cot))
    except:
        self.outputLabel.setText("ERROR")

def sec_it(self):
    try:
        screen=float(self.outputLabel.text())
        cos=round(math.cos(screen*math.pi/180),4)
        sec=round(1/cos,4)
        self.outputLabel.setText(str(sec))
    except:
        self.outputLabel.setText("ERROR")

def csc_it(self):
    try:
        screen=float(self.outputLabel.text())
        sin=round(math.sin(screen*math.pi/180),4)
        csc=round(1/sin,4)
        self.outputLabel.setText(str(csc))
    except:
        self.outputLabel.setText("ERROR")
```

程式碼說明

pow(次方) 、plus_minus(正負號對調) 、remove(移除)介紹：

```
def pow_it(self):
    try:
        screen=self.outputLabel.text()
        self.outputLabel.setText(f'{screen}**')
    except:
        self.outputLabel.setText("ERROR")

def plus_minus_it(self):
    screen=self.outputLabel.text()
    if "-" in screen:
        self.outputLabel.setText(screen.replace("-",""))
    else:
        self.outputLabel.setText(f'-{screen}')

def remove_it(self):
    screen = self.outputLabel.text()
    screen = screen[:-1]
    self.outputLabel.setText(screen)
```

程式碼說明

dot(小數點) 、reciprocal(倒數) 介紹：

```
def dot_it(self):
    screen = self.outputLabel.text()
    if screen[-1]==".":
        pass
    else:
        self.outputLabel.setText(f'{screen}.')

def reciprocal_it(self):
    screen = self.outputLabel.text()
    answer = float(screen)
    reciprocal=round((1/answer),4)
    self.outputLabel.setText(str(reciprocal))
```

程式碼說明

C(歸零)、Binary(二進位轉換)介紹：

```
def press_it(self,pressed):
    if pressed == "C":
        self.outputLabel.setText("0")
    else:
        if self.outputLabel.text() == "0":
            self.outputLabel.setText("")
        self.outputLabel.setText(f'{self.outputLabel.text()}{pressed}')

def Binary_it(self):
    try:
        screen=int(self.outputLabel.text())
        Binary=format(screen, '08b')
        self.outputLabel.setText(str(Binary))
    except:
            self.outputLabel.setText("ERROR")
```

程式碼說明

呼叫每個button所對應到介面之名稱

```
def retranslateUi(self, MainWindow):
    _translate = QtCore.QCoreApplication.translate
    MainWindow.setWindowTitle(_translate("MainWindow", "Calculater"))
    self.outputLabel.setText(_translate("MainWindow", "0"))
    self.percentButton.setText(_translate("MainWindow", "%"))
    self.cButton_2.setText(_translate("MainWindow", "C"))
    self.sqrtButton.setText(_translate("MainWindow", "√"))
    self.divideButton_4.setText(_translate("MainWindow", "÷"))
    self.arrowButton_3.setText(_translate("MainWindow", "<<"))
    self.sevenButton.setText(_translate("MainWindow", "7"))
    self.eightButton.setText(_translate("MainWindow", "8"))
    self.powButton.setText(_translate("MainWindow", "pow"))
    self.multiplyButton.setText(_translate("MainWindow", "×"))
    self.nineButton.setText(_translate("MainWindow", "9"))
    self.fourButton.setText(_translate("MainWindow", "4"))
    self.fiveButton.setText(_translate("MainWindow", "5"))
    self.logButton.setText(_translate("MainWindow", "log"))
```

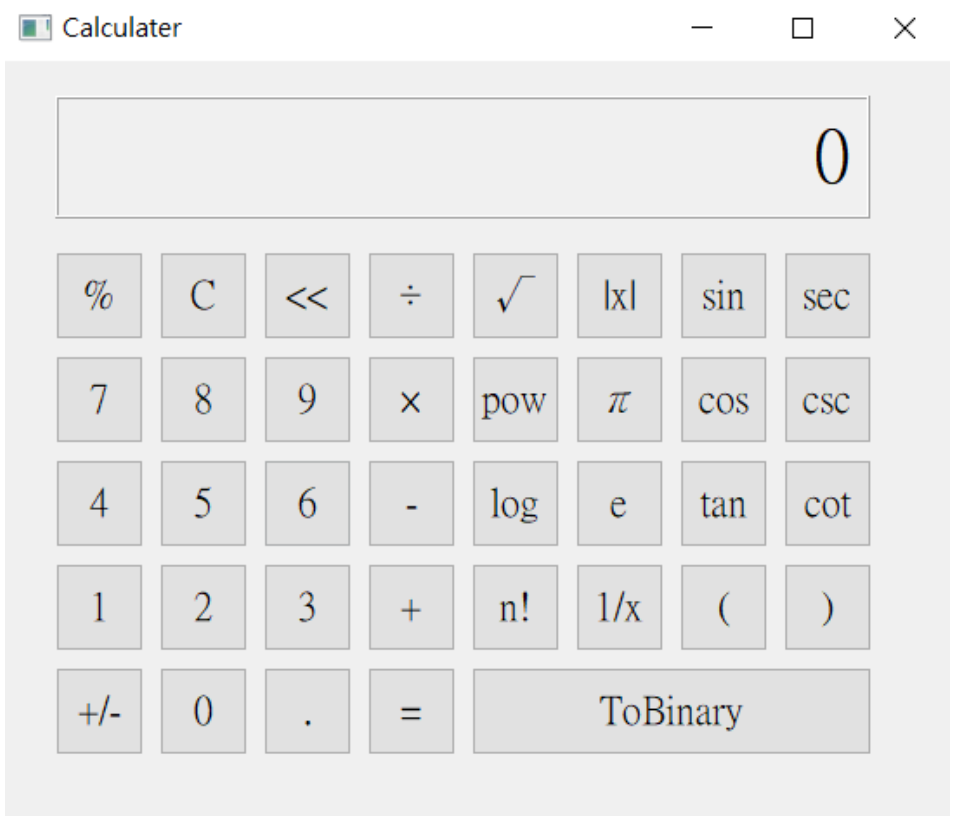
程式碼說明

展示

```
if __name__ == '__main__':
    import sys
    import math
    app = QtWidgets.QApplication(sys.argv)
    MainWindow=QtWidgets.QMainWindow()
    ui=Ui_MainWindow()
    ui.setupUi(MainWindow)
    MainWindow.show()
    sys.exit(app.exec_())
```

完成品

成品!!!


Calculater
0
% C << ÷ √ |x| sin sec
7 8 9 × pow π cos csc
4 5 6 - log e tan cot
1 2 3 + n! 1/x ( )
+/- 0 . = ToBinary

完成品

Demo

Q&A

結束

謝謝大家!!