import tkinter as Tk

from tkinter import ttk

import tkinter.messagebox as messagebox

from tkinter.filedialog import askopenfilename

from tkinter.messagebox import showerror

########################################################################

class ResultFrame(Tk.Toplevel):

""""""

#----------------------------------------------------------------------

def \_\_init\_\_(self):

"""Constructor"""

Tk.Toplevel.\_\_init\_\_(self)

self.geometry("320x240")

self.title("resultFrame")

########################################################################

class ManualFrame(Tk.Toplevel):

""""""

#----------------------------------------------------------------------

def \_\_init\_\_(self):

"""Constructor"""

Tk.Toplevel.\_\_init\_\_(self)

self.geometry("320x240")

self.title("manualFrame")

########################################################################

########################################################################

class MyApp(object):

""""""

#----------------------------------------------------------------------

def \_\_init\_\_(self, parent):

"""Constructor"""

self.root = parent

self.frame = Tk.Frame(parent, width=320, height=240)

self.frame.pack()

self.createWidgets()

self.fname = None

self.contenu = None

#----------------------------------------------------------------------

def createWidgets(self):

self.fileLabel = Tk.Label(self.frame, text='Please enter a file path or use the load button to find the file')

self.fileLabel.pack()

self.fileInput = Tk.Entry(self.frame)

self.fileInput.pack(expand=None)

self.loadButton = Tk.Button(self.frame, text="load",fg="black",command = self.load\_file)

self.loadButton.pack()

self.indicationLabel = Tk.Label(self.frame, text='Please enter the text in the box below')

self.indicationLabel.pack()

self.textInput = Tk.Text(self.frame)

self.textInput.pack()

self.checkButton = Tk.Button(self.frame, text="Check", command=self.openResultFrame)

self.checkButton.pack()

#----------------------------------------------------------------------

def file(self):

filePath = self.fileInput.get()

#tkMessageBox.showinfo('File Path','The selected file path is : ' %filePath)

file = open(filePath,"r")

#----------------------------------------------------------------------

def hide(self):

""""""

self.root.withdraw()

#----------------------------------------------------------------------

def openResultFrame(self):

""""""

self.hide()

if self.fname is None:

self.contenu = self.textInput.get(1.0, "end")

resultFrame = ResultFrame()

resultFrame.title(self.fname)

handler = lambda: self.onCloseResultFrame(resultFrame)

#self.resultLabel = Tk.Label(self.subFrame, text='Result :')

#self.resultLabel.pack()

#self.similarityLabel = Tk.Label(self.subFrame, text='Similarity :')

#self.similarityLabel.pack()

#self.judgementLabel = Tk.Label(self.subFrame, text='Judgement :')

#self.judgementLabel.pack()

similarity = 99.99

resultLabel = Tk.Label(resultFrame, text='Result :')

resultLabel.pack()

similarityLabel = Tk.Label(resultFrame, text='Similarity : %.2f%%'%similarity )

similarityLabel.pack()

judgementLabel = Tk.Label(resultFrame, text='Judgement : ')

judgementLabel.pack()

manualCheckButton = Tk.Button(resultFrame, text="Transfer to check", command=self.openManualCheckFrame)

manualCheckButton.pack()

finishButton = Tk.Button(resultFrame, text="OK", command=self.OK)

finishButton.pack()

def openManualCheckFrame(self):

""""""

self.hide()

manualCheckFrame = ManualFrame()

handler = lambda: self.onCloseManulCheckFrame(manualCheckFrame)

similarityLabel = Tk.Label(manualCheckFrame, text='Similarity :')

similarityLabel.pack()

chooseJudgementLabel = Tk.Label(manualCheckFrame, text='Choose judgement :')

chooseJudgementLabel.pack()

enterJudgementText = Tk.StringVar()

levelChoosed = ttk.Combobox(manualCheckFrame, width=12, textvariable=enterJudgementText, state="readonly")

levelChoosed['values'] = ("Safe", "By check", "Dangerous")

#levelChoosed.current(0)

levelChoosed.pack()

watchTextButton = Tk.Button(manualCheckFrame, text="Watch text", command=self.showText)

watchTextButton.pack()

okButton = Tk.Button(manualCheckFrame, text="OK", command=self.Success)

okButton.pack()

#----------------------------------------------------------------------

def showText(self):

messagebox.showinfo('Message', self.contenu)

def onCloseResultFrame(self, resultFrame):

""""""

resultFrame.destroy()

self.show()

def onCloseManualCheckFrame(self, manualCheckFrame):

""""""

manualCheckFrame.destroy()

self.show()

def OK(self):

messagebox.showinfo('Message', 'OK' )

self.root.destroy()

def Success(self):

messagebox.showinfo('Message', 'You have added the new template into the dataset' )

self.root.destroy()

#----------------------------------------------------------------------

def show(self):

""""""

self.root.update()

self.root.deiconify()

def load\_file(self):

self.fname = askopenfilename(filetypes=(("Text files", "\*.txt"),

("Word files", "\*.doc;\*.docx") ),

title = "Choose a file.")

if self.fname:

try:

with open(self.fname, 'r') as f:

self.contenu = f.read()

except: # <- naked except is a bad idea

showerror("Open Source File", "Failed to read file\n'%s'" % self.fname)

return 0

#----------------------------------------------------------------------

if \_\_name\_\_ == "\_\_main\_\_":

root = Tk.Tk()

root.geometry("800x600")

app = MyApp(root)

root.mainloop()