```
# Name: Trien Bang Huynh
# Assignment 2: numpy, matplotlib, tkinter
# gui.py: PlotWin class, DialogWin class, and MainWin class

import matplotlib
matplotlib.use('TkAgg')
import tkinter as tk
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
import matplotlib.pyplot as plt
import tkinter.messagebox as tkmb

from tuition import Tuition, FileNotFound


class PlotWin(tk.Toplevel):
    '''
    A class to create appropriate plots.
    '''
    def __init__(self, master, plotType, *args):
        super().__init__(master)
        self.transient(master)
        fig = plt.figure(figsize = (10, 6))
        plotType(*args)
        canvas = FigureCanvasTkAgg(fig, master = self)
        canvas.get_tk_widget().grid()
        canvas.draw()


class DialogWin(tk.Toplevel):
    '''
    A class to show a dialog and ask user for number of states they want to view.
    '''
    def __init__(self, master):
        super().__init__(master)
        self.grab_set()
        self.focus_set()
        self.transient(master)
        self._controlVar = tk.IntVar()
        self.protocol("WM_DELETE_WINDOW", self.on_close)
        self._onClick = False

        tk.Label(self, text="Choose the number of states").pack()
        tk.Radiobutton(self, text="5", variable=self._controlVar,
value=5).pack(anchor=tk.W)
        tk.Radiobutton(self, text="10", variable=self._controlVar,
value=10).pack(anchor=tk.W)
        tk.Radiobutton(self, text="15", variable=self._controlVar,
value=15).pack(anchor=tk.W)
        tk.Radiobutton(self, text="20", variable=self._controlVar,
value=25).pack(anchor=tk.W)

        self._controlVar.set(5)

        self.confirm_button = tk.Button(self, text="Click to lock in selection",
command=self.on_confirm)
        self.confirm_button.pack(pady=10)


    def on_confirm(self):
```

```
        '''
        A function to confirm user clicked on lock in button.
        '''
        self._onClick = True
        self.on_close()

    def on_close(self):
        '''
        A function to process closing Dialog Window
        '''
        if not self._onClick:
            self._controlVar.set(0)
        self.destroy()


    @property
    def getNumStates(self):
        return self._controlVar.get()

class MainWin(tk.Tk):
    '''
    The main class will display tution statistic of states and interact with users
to view the plots and process their choices
    '''
    def __init__(self):
        super().__init__()
        self.title("Tuition")

        try:
            self._tuition = Tuition()
        except FileNotFound as e:
            tkmb.showerror("File Not Found", str(e), parent=self)
            self.destroy()
            self.quit()
            return

        tk.Label(self, text="Yearly College Tuition", fg="blue",font=('Times',
17)).grid(row=0, column=0, columnspan=3, pady=10)

        # Create buttons

        tk.Button(self, text="Overview",
command=self.show_tuition_overview).grid(row=1, column=0, padx=10, pady=10)
        tk.Button(self, text="Lowest
Cost",command=self.show_lowest_cost_dialog).grid(row=1, column=1, padx=10, pady=10)
        tk.Button(self, text="Largest Change",
command=self.show_largest_change_plot).grid(row=1, column=2, padx=10, pady=10)


        # Display statistics

        min_tuition, max_tuition, mean_tuition, median_tuition =
self._tuition.get_tuition_statistics()

        tk.Label(self, text=f"Lowest tution: ${min_tuition}",
fg="green",font=('Times', 14)).grid(row=2, column=0, sticky=tk.W, pady=10,
padx=(10, 0))

        tk.Label(self, text=f"Highest tuition: ${max_tuition}",
```

```
fg="green",font=('Times', 14)).grid(row=2, column=2, sticky=tk.E, pady=10, padx=(0,
10))

        tk.Label(self, text=f"Mean tution: ${mean_tuition}",
fg="green",font=('Times', 14)).grid(row=3, column=0, sticky=tk.W, pady=10,
padx=(10, 0))

        tk.Label(self, text=f"Median tuition: ${median_tuition}",
fg="green",font=('Times', 14)).grid(row=3, column=2, sticky=tk.E, pady=10, padx=(0,
10))


        self.protocol("WM_DELETE_WINDOW", self.close_window)


    def show_tuition_overview(self):
        '''
        A function to call a PlotWin object for tuition distribution plot.
        '''
        PlotWin(self, self._tuition.plot_tuition_distribution)

    def show_lowest_cost_dialog(self):
        '''
        A function to call DialogWin object for processing user choice and a
PlotWin object for showing states that have loweset tuition.
        '''
        self.dialog_window = DialogWin(self)
        self.wait_window(self.dialog_window)
        num_states = self.dialog_window.getNumStates
        if num_states != 0:
            PlotWin(self,self._tuition.plot_lowest_tuition_states,num_states)

    def show_largest_change_plot(self):
        '''
        A function to call a PlotWin object for plotting 5 states with largest
increase and 1 state with smallest increase in tuition.
        '''
        PlotWin(self, self._tuition.plot_tuition_trend)

    def close_window(self):
        '''
        A method to ask and confirm closing main window.
        '''
        val = tkmb.askokcancel("Confirm close", "It will close your main and/or
other opening windows. Still wanna close?", parent=self)
        if val :
            self.destroy()
            self.quit()



app = MainWin()
app.mainloop()
```