Dumangas, Hannizel

C 204

## Problem:

**Finals Lab Task 4. Python GUI using TKINTER**

**Note: Write your code following OOP code construct, you may use the attached simpleCalc.py program as guide.**

Instructions: READ AND UNDERSTAND THE PROBLEM FIRST BEFORE DOING THE ACTUAL PROGRAM.

1. Design the form below
2. Problem Statement: The cost of a long Distance call is based on the destination, the time of day the call was made, as well as the distance of the call. The rates as as follows:

| DAYTIME CALLS | | NIGHTIME CALLS | |
|---|---|---|---|
| 1. American Region | P 50 every 3 minutes | 1. American Region | P 45 every 3 minutes |
| 2. Asian Region | P 30 every 2 minutes | 2. Asian Region | P 27 every 2 minutes |
| 3. African Region | P 40 every 3 minutes | 3. African Region | P 36 every 3 minutes |
| 4. European Region | P 35 every 2 minutes | 4. European Region | P 30 every 2 minutes |

3. Make a program that will Allow the user to **Select Destination Code (between 1 – 4)** using ComboBox widget, A Time Code using radio buttons, And the Duration Of The Call in minutes and output the **TOTAL CHARGE**. – Validate user inputs by using **TRY EXCEPT block – Only numeric values are accepted.**

4. **Compute Button** should compute for the **TOTAL CHARGE**.

   4.1 Computations should be based on the table rates shown above. (The total charge is based on **Length of Calls**, **Destination Code** and **Time Code**)

   4.2. You may use the get () method of the comboBox to capture the selected option in your comboBox

5. **Reset Button** should clear the Radio Button Selection and the Text field entries should be cleared as well

6. **About button** should display a dialog with the message: "Hello I'm your Name"

7. See sample output below:

## Source Code:

```python
import tkinter as tk
from tkinter import ttk, messagebox


1 usage
class CallRateCalculator:
    def __init__(self):

        self.day_rates = {
            "American Region": (50, 3),
            "Asian Region": (30, 2),
            "African Region": (40, 3),
            "European Region": (35, 2)
        }

        self.night_rates = {
            "American Region": (45, 3),
            "Asian Region": (27, 2),
            "African Region": (36, 3),
            "European Region": (30, 2)
        }

    1 usage
    def compute(self, destination, minutes, time_code):

        # choose correct rate table
        rate_table = self.day_rates if time_code == "Day" else self.night_rates
        rate, per_min = rate_table[destination]

        cycles = minutes // per_min
        if minutes % per_min != 0:
            cycles += 1

        return cycles * rate
```

```python
class LongDistanceGUI:
    def __init__(self, root):
        self.root = root
        self.root.title("Long Distance Call Charge Calculator")
        self.root.geometry("600x420")
        self.root.resizable(False, False)

        self.calc = CallRateCalculator()

        self.create_widgets()


    1 usage
    def create_widgets(self):
        frame = tk.LabelFrame(self.root, text="User Inputs", padx=10, pady=10)
        frame.pack(padx=10, pady=10, fill="x")

        tk.Label(frame, text="Length of Call (minutes):").grid(row=0, column=0, sticky="w")
        self.entry_minutes = tk.Entry(frame, width=10)
        self.entry_minutes.grid(row=0, column=1, padx=10, pady=5)

        tk.Label(frame, text="Destination Code:").grid(row=1, column=0, sticky="w")
        self.combo_dest = ttk.Combobox(
            frame,
            values=["American Region", "Asian Region", "African Region", "European Region"],
            width=25,
            state="readonly"
        )
        self.combo_dest.grid(row=1, column=1, padx=10, pady=5)
        self.combo_dest.current(0)

        tk.Label(frame, text="Time Code").grid(row=2, column=0, sticky="w")
        self.time_var = tk.StringVar(value="Day")
        tk.Radiobutton(frame, text="Day Time", variable=self.time_var, value="Day").grid(row=2, column=1, sticky="w")
        tk.Radiobutton(frame, text="Night Time", variable=self.time_var, value="Night").grid(row=2, column=1)

        self.summary_box = tk.Text(self.root, width=55, height=8, state="disabled")
        self.summary_box.pack(pady=10)
```

```
    btn_frame = tk.Frame(self.root)
    btn_frame.pack()

    tk.Button(btn_frame, text="Compute Charge", width=15, command=self.compute_total).grid(row=0, column=0, padx=5)
    tk.Button(btn_frame, text="Reset", width=10, command=self.reset_all).grid(row=0, column=1, padx=5)
    tk.Button(btn_frame, text="About", width=10, command=self.about_msg).grid(row=0, column=2, padx=5)
    tk.Button(btn_frame, text="Close", width=10, command=self.root.destroy).grid(row=0, column=3, padx=5)


1 usage
def compute_total(self):
    try:
        minutes = int(self.entry_minutes.get())
        if minutes <= 0:
            raise ValueError

    except ValueError:
        messagebox.showerror( title: "Invalid Input",  message: "Please enter a valid positive integer for call length.")
        return

    destination = self.combo_dest.get()
    time_code = self.time_var.get()

    total = self.calc.compute(destination, minutes, time_code)

    summary = (
        f"Transaction Summary:\n"
        f"Length of Call: {minutes} minute(s)\n"
        f"Destination: {destination}\n"
        f"Time Code: {time_code} Time\n"
        f"Total Charge: Php {total:.2f}"
    )

    self.summary_box.config(state="normal")
    self.summary_box.delete( index1: "1.0", tk.END)
    self.summary_box.insert(tk.END, summary)
    self.summary_box.config(state="disabled")
```

```
    1 usage
    def reset_all(self):
        self.entry_minutes.delete( first: 0, tk.END)
        self.combo_dest.current(0)
        self.time_var.set("Day")

        self.summary_box.config(state="normal")
        self.summary_box.delete( index1: "1.0", tk.END)
        self.summary_box.config(state="disabled")


    1 usage
    def about_msg(self):
        messagebox.showinfo( title: "About",  message: "Hello! I'm your Name")

root = tk.Tk()
app = LongDistanceGUI(root)
root.mainloop()
```

Sample Output: