

```
python-05

#

# Multi-Function Calculator with GUI Window

#

import tkinter as tk # GUI interface

from tkinter import ttk # themed widgets interface of tkinter

from tkinter import messagebox

#####

# Input Validation Functions

#####

def VAL_NUMBER(input_value):

    """Validate numeric input"""

    if input_value == "" or input_value == "-":

        return True

    try:

        Value = float(input_value)

        if Value < 0:

            messagebox.showerror("Error",

                                "Please enter positive numbers only")

            return False

        return True

    except ValueError:

        messagebox.showerror("Error",

                                "Please enter numbers only")

        return False
```

```
def VAL_YEARS(input_value):  
    """Validate years input (must be integer)"""  
    if input_value == "" or input_value == "-":  
        return True  
    try:  
        Value = int(input_value)  
        if Value <= 0:  
            messagebox.showerror("Error",  
                                  "Years must be a positive integer")  
            return False  
        return True  
    except ValueError:  
        messagebox.showerror("Error",  
                                  "Please enter a valid integer for years")  
        return False
```

```
#####  
# Calculation Functions  
#####
```

```
def CONVERT_SPEED():  
    """Convert speed from km/h to m/s"""  
    try:  
        speed_input = I_SpeedKmh.get()  
        if speed_input == "":
```

```
        messagebox.showwarning("Warning", "Please enter a speed value")
    return
    speed_kmh = float(speed_input)
    if speed_kmh == 0:
        messagebox.showwarning("Warning", "Please enter a speed value")
        return
    speed_ms = speed_kmh / 3.6
    R_SpeedMs.set(round(speed_ms, 2))
except:
    messagebox.showerror("Error", "Calculation error")
```

```
def CALCULATE_MORTGAGE():
    """Calculate monthly mortgage payment"""
    try:
        principal_input = I_Principal.get()
        years_input = I_Years.get()
        rate_input = I_Rate.get()

        if principal_input == "" or years_input == "" or rate_input == "":
            messagebox.showwarning("Warning", "Please enter all values")
            return

        principal = float(principal_input)
        years = int(years_input)
        rate = float(rate_input)
```

```
if principal == 0 or years == 0:
    messagebox.showwarning("Warning", "Please enter all values")
    return

# Calculate monthly payment
monthly_rate = (rate / 100) / 12
num_payments = years * 12

if monthly_rate == 0:
    monthly_payment = principal / num_payments
else:
    monthly_payment = principal * (monthly_rate * (1 + monthly_rate) **
num_payments) / \
    ((1 + monthly_rate) ** num_payments - 1)

total_amount = monthly_payment * num_payments

R_Monthly.set(round(monthly_payment, 2))
R_Total.set(round(total_amount, 2))

except:
    messagebox.showerror("Error", "Calculation error")
```

```
def CLEAR_SPEED():
    """Clear speed conversion fields"""
    I_SpeedKmh.set("")
    R_SpeedMs.set("")
```

```
def CLEAR_MORTGAGE():  
    """Clear mortgage calculation fields"""  
    I_Principal.set("")  
    I_Years.set("")  
    I_Rate.set("")  
    R_Monthly.set("")  
    R_Total.set("")  
  
#####  
# MAIN LINE Window Management  
#####  
MainWindow = tk.Tk() # Create main window (tk)  
MainWindow.geometry("450x400") # Resize window  
MainWindow.title("Multi-Function Calculator") # Window title  
# Create notebook (tabbed interface)  
notebook = ttk.Notebook(MainWindow)  
notebook.pack(fill='both', expand=True, padx=10, pady=10)  
  
#####  
# TAB 1: Speed Converter  
#####  
tab1 = ttk.Frame(notebook)  
notebook.add(tab1, text="Speed Converter")
```

```
# Define data objects for speed converter

I_SpeedKmh = tk.StringVar()

R_SpeedMs = tk.StringVar()

# Widgets for speed converter

Row = 0

tk.Label(tab1, text="SPEED CONVERTER: km/h to m/s",
         font=("Arial", 12, "bold")).grid(row=Row, column=0, columnspan=2,
         pady=15)

Row += 1

tk.Label(tab1, text="Speed (km/h):").grid(row=Row, column=0, sticky='e', padx=10,
         pady=10)

vcmd = (MainWindow.register(VAL_NUMBER), '%P')

tk.Entry(tab1, textvariable=I_SpeedKmh, width=20, justify="right", validate="focusout",
         validatecommand=vcmd).grid(row=Row, column=1, sticky='w', padx=10,
         pady=10)

Row += 1

tk.Label(tab1, text="=", font=("Arial", 10)).grid(row=Row, column=0, sticky='e',
         padx=10, pady=10)

Row += 1

tk.Label(tab1, text="Speed (m/s):").grid(row=Row, column=0, sticky='e', padx=10,
         pady=10)

tk.Label(tab1, textvariable=R_SpeedMs, relief="sunken", width=20,
         anchor="e").grid(row=Row, column=1, sticky='w', padx=10, pady=10)
```

```
Row += 1
```

```
ttk.Button(tab1, text="Convert", command=CONVERT_SPEED).grid(row=Row,  
column=0, sticky="e", padx=10, pady=15)
```

```
ttk.Button(tab1, text="Clear", command=CLEAR_SPEED).grid(row=Row, column=1,  
sticky="w", padx=10, pady=15)
```

```
#####
```

```
# TAB 2: Mortgage Calculator
```

```
#####
```

```
tab2 = ttk.Frame(notebook)
```

```
notebook.add(tab2, text="Mortgage Calculator")
```

```
# Define data objects for mortgage calculator
```

```
L_Principal = tk.StringVar()
```

```
L_Years = tk.StringVar()
```

```
L_Rate = tk.StringVar()
```

```
R_Monthly = tk.StringVar()
```

```
R_Total = tk.StringVar()
```

```
# Widgets for mortgage calculator
```

```
Row = 0
```

```
tk.Label(tab2, text="MORTGAGE PAYMENT CALCULATOR",  
font=("Arial", 12, "bold")).grid(row=Row, column=0, colspan=2,  
pady=15)
```

```
Row += 1
```

```
tk.Label(tab2, text="Loan Amount (€):").grid(row=Row, column=0, sticky='e', padx=10,
```

```
pady=8)
vcmd = (MainWindow.register(VAL_NUMBER), '%P')
tk.Entry(tab2, textvariable=l_Principal, width=20, justify="right", validate="focusout",
         validatecommand=vcmd).grid(row=Row, column=1, sticky='w', padx=10,
pady=8)
```

```
Row += 1
```

```
tk.Label(tab2, text="Duration (years:)").grid(row=Row, column=0, sticky='e', padx=10,
pady=8)
```

```
vcmd = (MainWindow.register(VAL_YEARS), '%P')
```

```
tk.Entry(tab2, textvariable=l_Years, width=20, justify="right", validate="focusout",
         validatecommand=vcmd).grid(row=Row, column=1, sticky='w', padx=10,
pady=8)
```

```
Row += 1
```

```
tk.Label(tab2, text="Interest Rate (%:)").grid(row=Row, column=0, sticky='e', padx=10,
pady=8)
```

```
vcmd = (MainWindow.register(VAL_NUMBER), '%P')
```

```
tk.Entry(tab2, textvariable=l_Rate, width=20, justify="right", validate="focusout",
         validatecommand=vcmd).grid(row=Row, column=1, sticky='w', padx=10,
pady=8)
```

```
Row += 1
```

```
tk.Label(tab2, text="", font=("Arial", 10)).grid(row=Row, column=0, sticky='e',
padx=10, pady=8)
```

```
Row += 1
```

```
tk.Label(tab2, text="Monthly Payment (€:)").grid(row=Row, column=0, sticky='e',
padx=10, pady=8)
```

```
tk.Label(tab2, textvariable=R_Monthly, relief="sunken", width=20,
        anchor="e").grid(row=Row, column=1, sticky='w', padx=10, pady=8)
```

```
Row += 1
```

```
tk.Label(tab2, text="Total Amount (€):").grid(row=Row, column=0, sticky='e', padx=10,
pady=8)
```

```
tk.Label(tab2, textvariable=R_Total, relief="sunken", width=20,
        anchor="e").grid(row=Row, column=1, sticky='w', padx=10, pady=8)
```

```
Row += 1
```

```
ttk.Button(tab2, text="Calculate", command=CALCULATE_MORTGAGE).grid(row=Row,
column=0, sticky="e", padx=10, pady=15)
```

```
ttk.Button(tab2, text="Clear", command=CLEAR_MORTGAGE).grid(row=Row, column=1,
sticky="w", padx=10, pady=15)
```

```
#####
```

```
# Quit button at the bottom of main window
```

```
#####
```

```
ttk.Button(MainWindow, text="Quit",
command=MainWindow.quit).pack(side="bottom", pady=10)
```

```
#####
```

```
# Main Loop of Tkinter: handles events until window is closed
```

```
#####
```

```
MainWindow.mainloop()
```

```
#####
```

```
# Exit and close windows
```

```
#####
```

MainWindow.destroy()

