

TRƯỜNG ĐẠI HỌC VĂN LANG
KHOA CÔNG NGHỆ THÔNG TIN



BÀI TẬP 1

MÔN HỌC LẬP TRÌNH PYTHON NÂNG CAO

Đề bài:

LẬP TRÌNH GIAO DIỆN GUI

SINH VIÊN THỰC HIỆN:

NGUYỄN TIỀN PHÚC 2274802010685

LỚP: 241_71ITSE31003_02

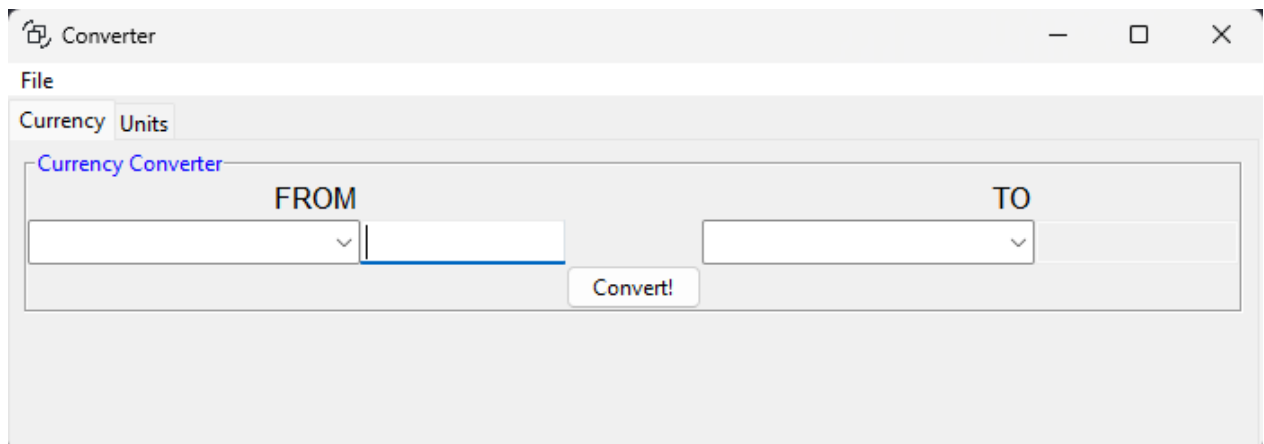
GVHD: HUỲNH THÁI HỌC

TP. Hồ Chí Minh – 08/10/2024

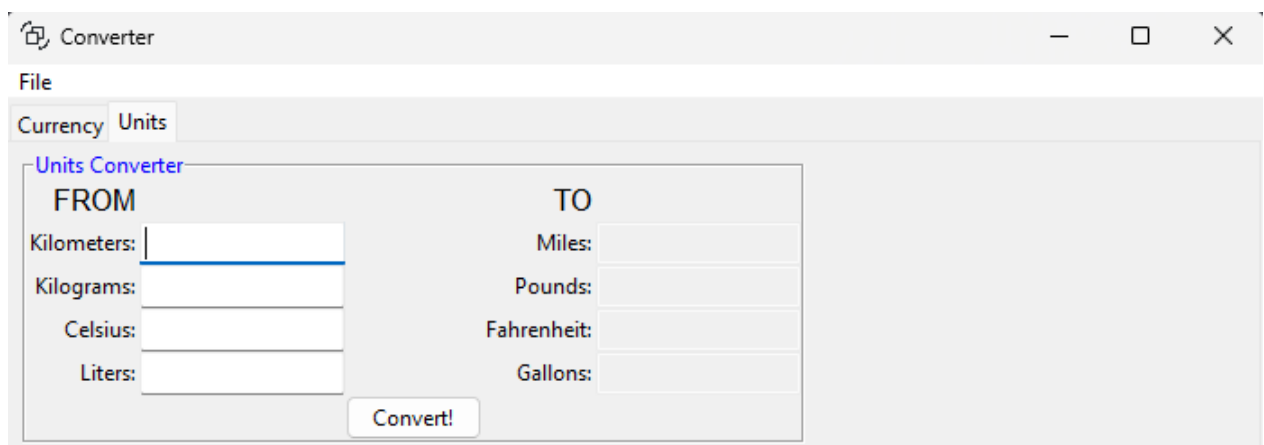
MỤC LỤC

I.	GIAO DIỆN	3
II.	CHỨC NĂNG.....	4
1.	Quy đổi tiền tệ.....	4
2.	Quy đổi đơn vị.....	6
3.	Menu File	8
III.	MÃ CHƯƠNG TRÌNH.....	10
IV.	GITHUB.....	18

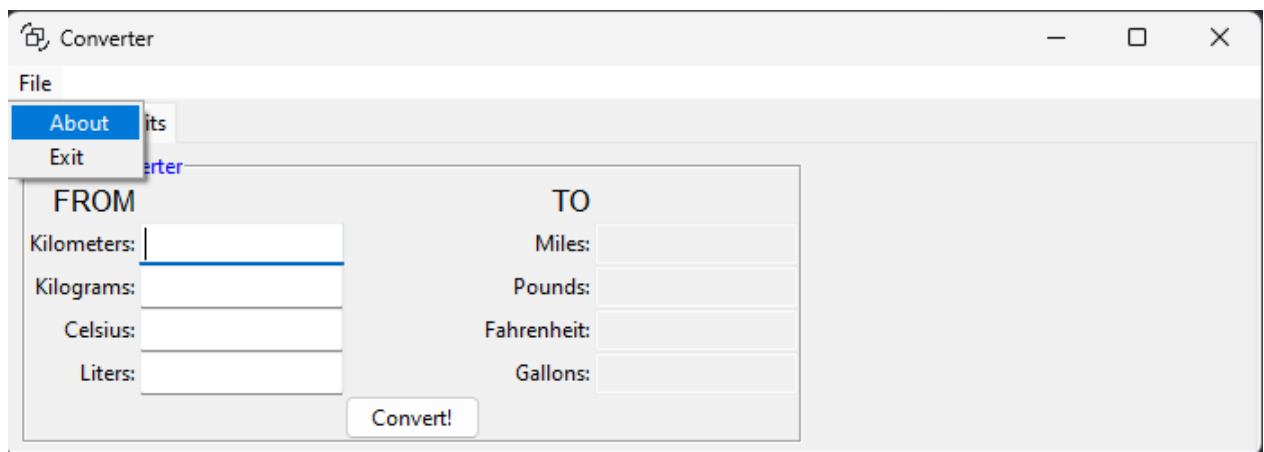
I. GIAO DIỆN



The screenshot shows the 'Converter' application window with the 'Currency' tab selected. The 'Currency Converter' section contains two dropdown menus labeled 'FROM' and 'TO', and a 'Convert!' button.



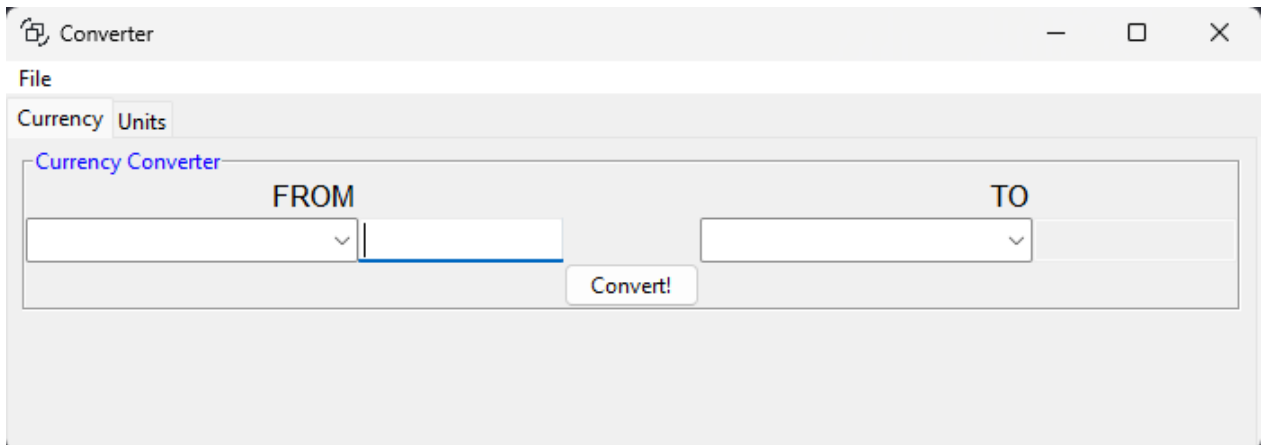
The screenshot shows the 'Converter' application window with the 'Units' tab selected. The 'Units Converter' section contains two columns of input fields labeled 'FROM' and 'TO'. The 'FROM' column includes Kilometers, Kilograms, Celsius, and Liters. The 'TO' column includes Miles, Pounds, Fahrenheit, and Gallons. A 'Convert!' button is located at the bottom.



The screenshot shows the 'Converter' application window with the 'Units' tab selected. The 'File' menu is open, showing options: 'About', 'Exit', and 'Converter'. The 'Units Converter' section is visible in the background, showing the 'FROM' and 'TO' columns and the 'Convert!' button.

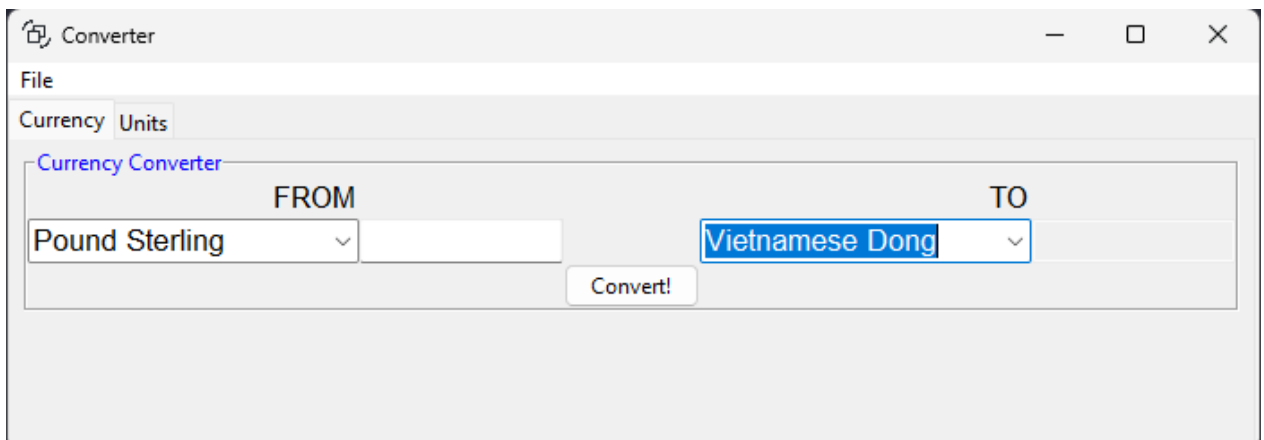
II. CHỨC NĂNG

1. Quy đổi tiền tệ



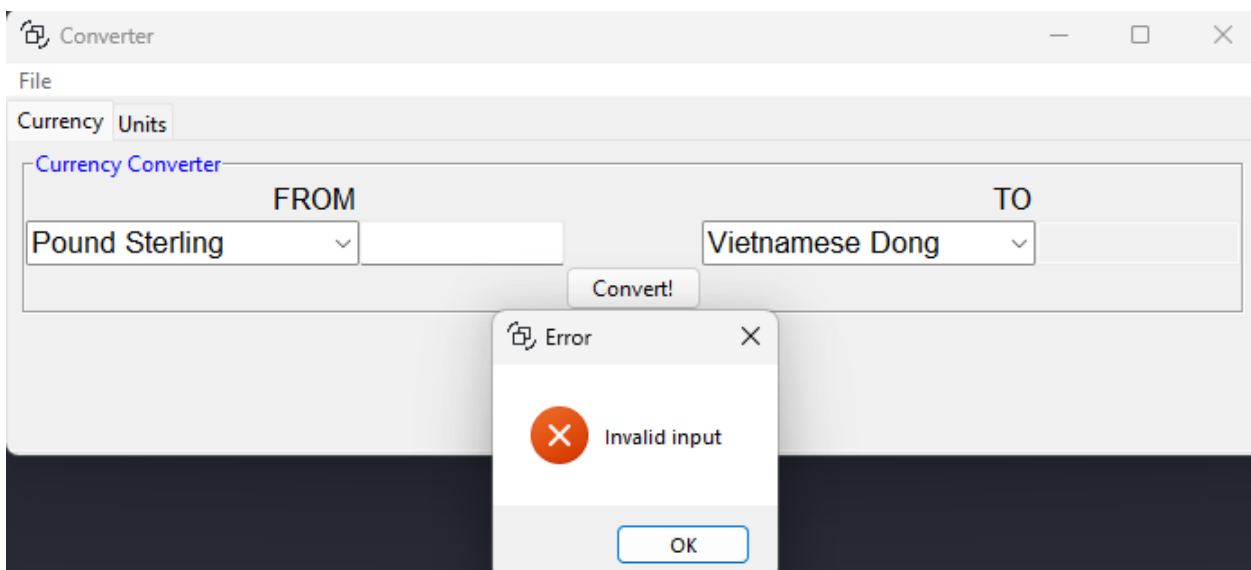
The screenshot shows a window titled "Converter" with a menu bar containing "File". Below the menu bar are two tabs: "Currency" (selected) and "Units". The main area is titled "Currency Converter" and contains two dropdown menus labeled "FROM" and "TO". Both dropdown menus are currently empty, showing only a downward arrow. A "Convert!" button is located between the two dropdown menus.

Chức năng này được tạo ra nhằm mục đích kiểm tra giá tiền tệ khi quy đổi giữa 6 đơn vị tiền tệ khác nhau (Vietnamese Dong VND, US Dollar USD, Pound Sterling GBP, Japanese Yen JPY, South Korean Won KRW và Chinese Yuan CNY).

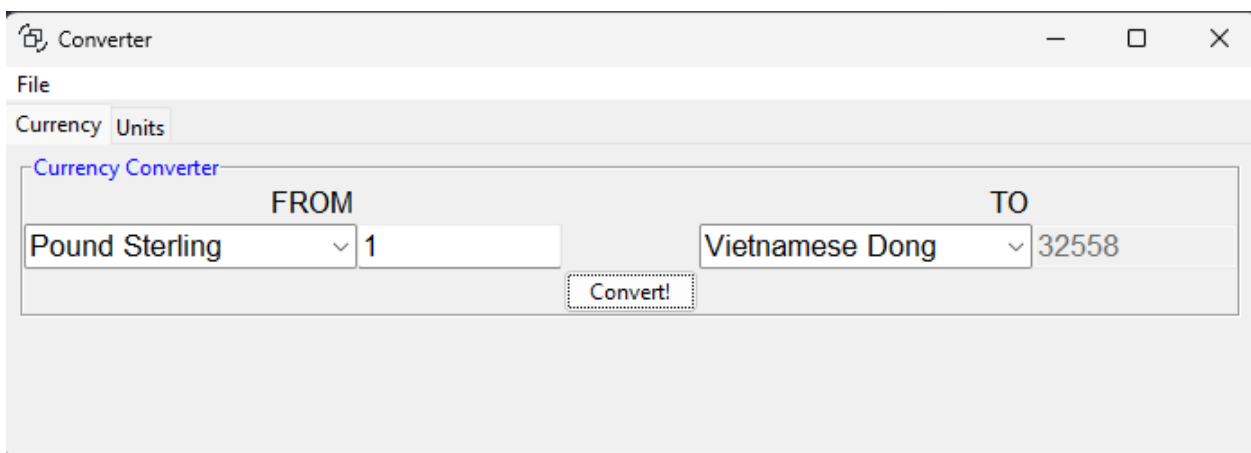


The screenshot shows the same "Converter" window. In this state, the "FROM" dropdown menu is set to "Pound Sterling" and the "TO" dropdown menu is set to "Vietnamese Dong". The "Convert!" button remains visible between the two dropdown menus.

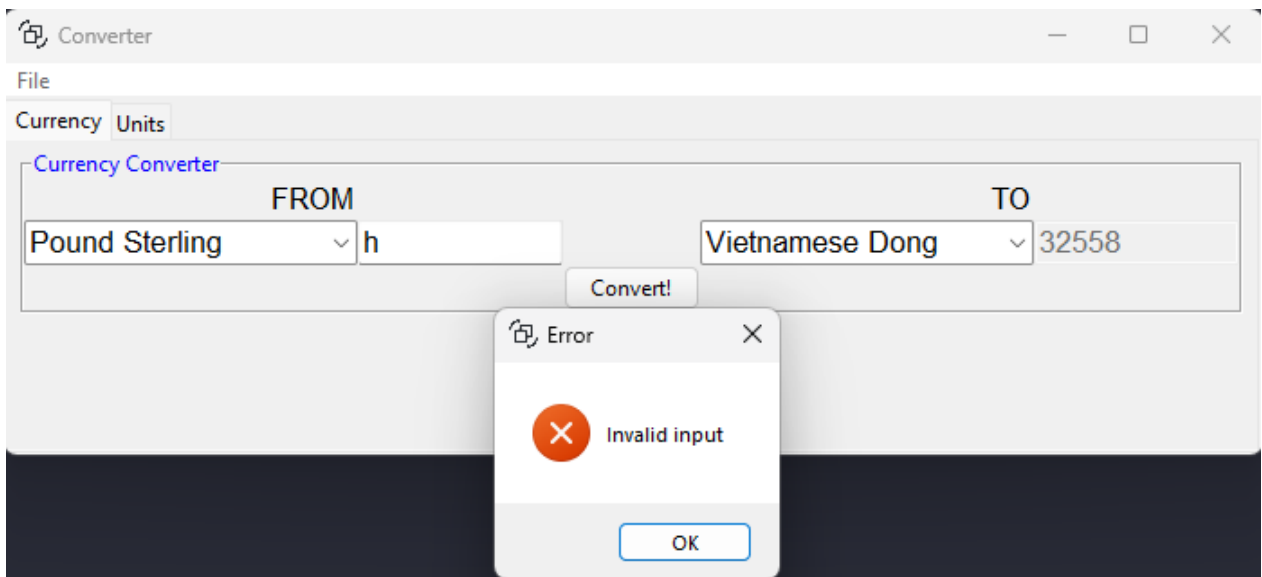
Người dùng có thể chọn 2 loại tiền tệ mà bản thân muốn đổi bằng cách chọn các giá trị trong 2 ComboBox.



Khi người dùng bỏ trống ô nhập và nhấn nút “Convert!”, hệ thống sẽ hiển thị 1 cửa sổ MessageBox thông báo lỗi.

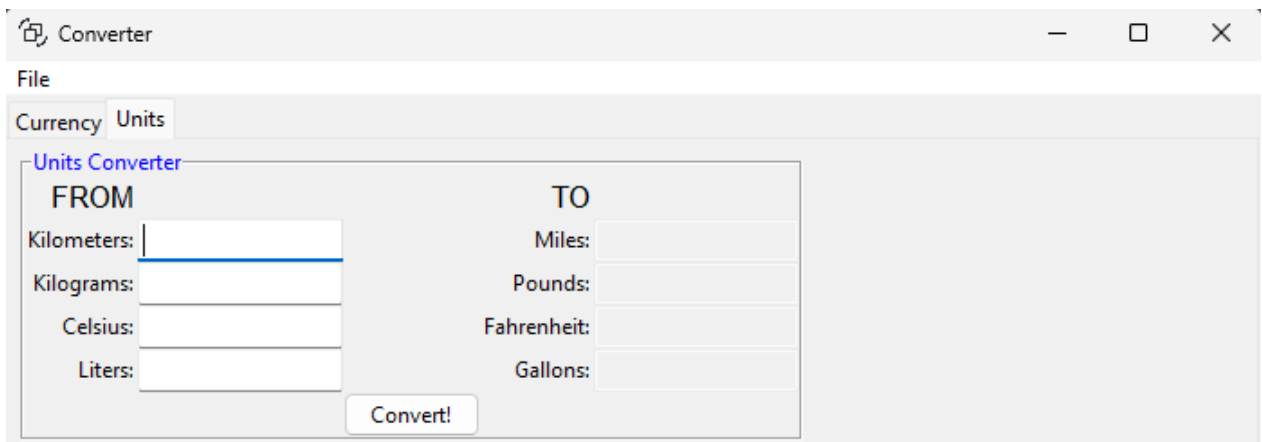


Khi người dùng nhập giá trị vào ô nhập và nhấn nút “Convert!”, hệ thống sẽ tính và hiển thị giá trị được quy đổi của đơn vị tiền tệ tương ứng. Nhược điểm của dữ liệu tính toán này là chưa sử dụng tỷ giá tiền tệ theo thời gian thực mà chỉ dựa trên dữ liệu được cài đặt sẵn trong chương trình.

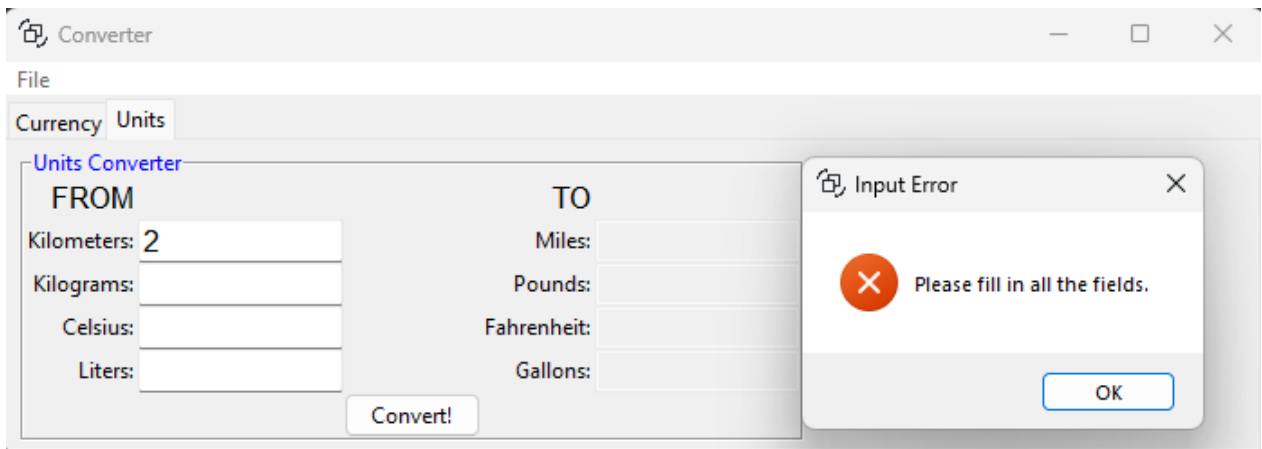
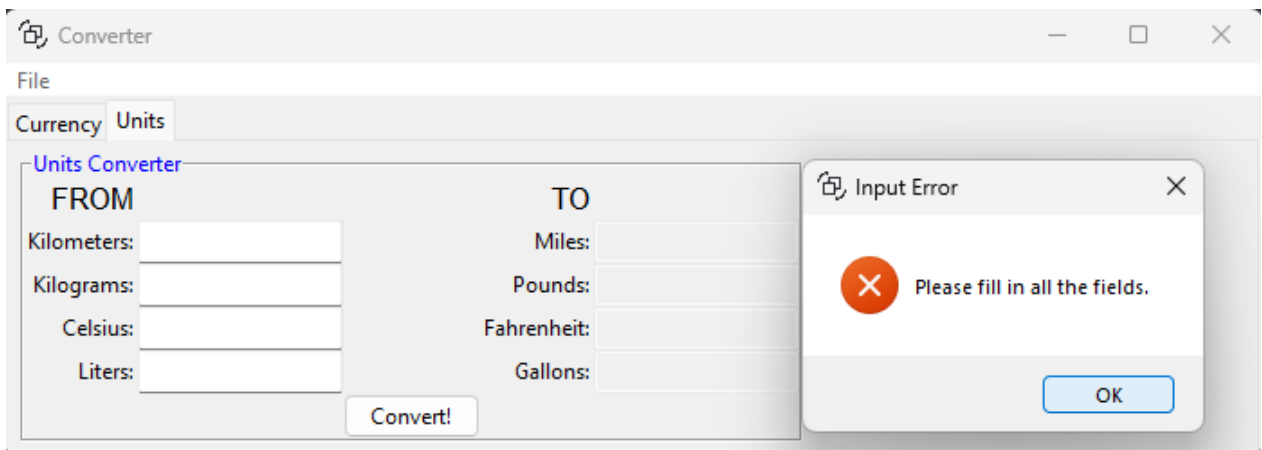


Khi người dùng nhập ký tự khác với ký tự số, hệ thống sẽ hiển thị 1 cửa sổ MessageBox thông báo lỗi.

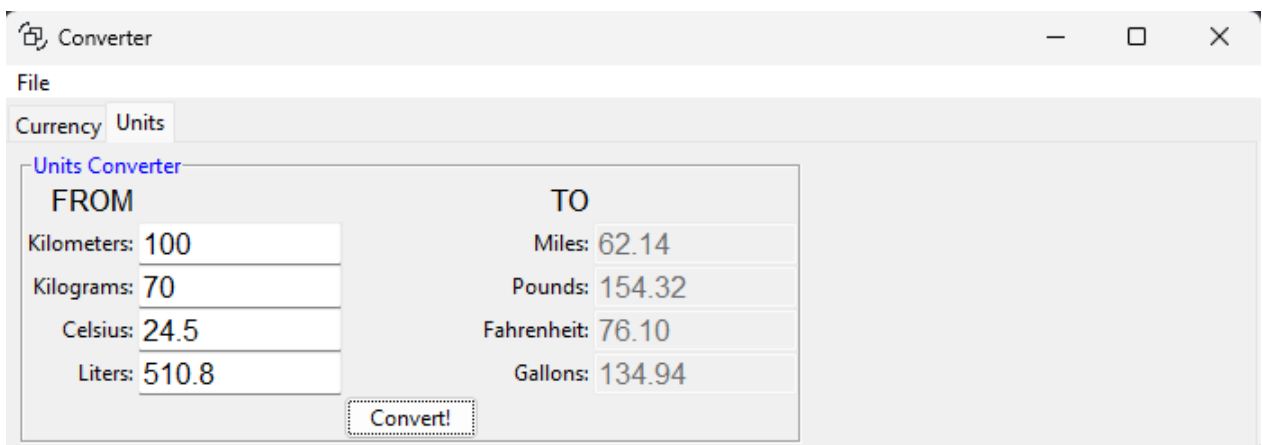
2. Quy đổi đơn vị



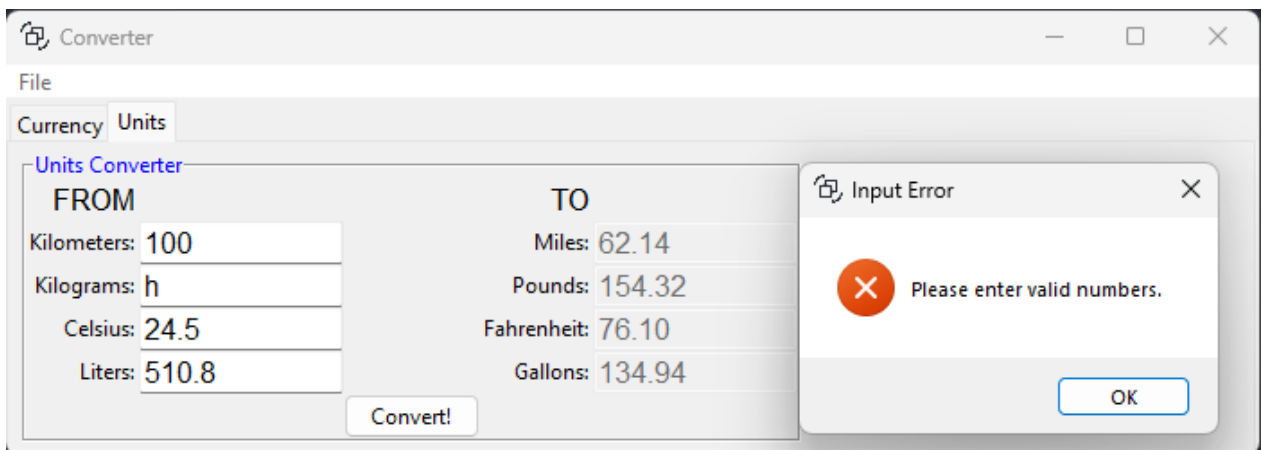
Chức năng này được tạo ra nhằm mục đích kiểm tra giá trị khi quy đổi giữa 4 cặp đơn vị đo khác nhau (Kilometers – Miles, Kilograms – Pounds, Celsius – Fahrenheit, Liters - Gallons).



Khi người dùng bỏ trống ít nhất 1 ô nhập và nhấn nút “Convert!”, hệ thống sẽ hiển thị 1 cửa sổ MessageBox thông báo lỗi.

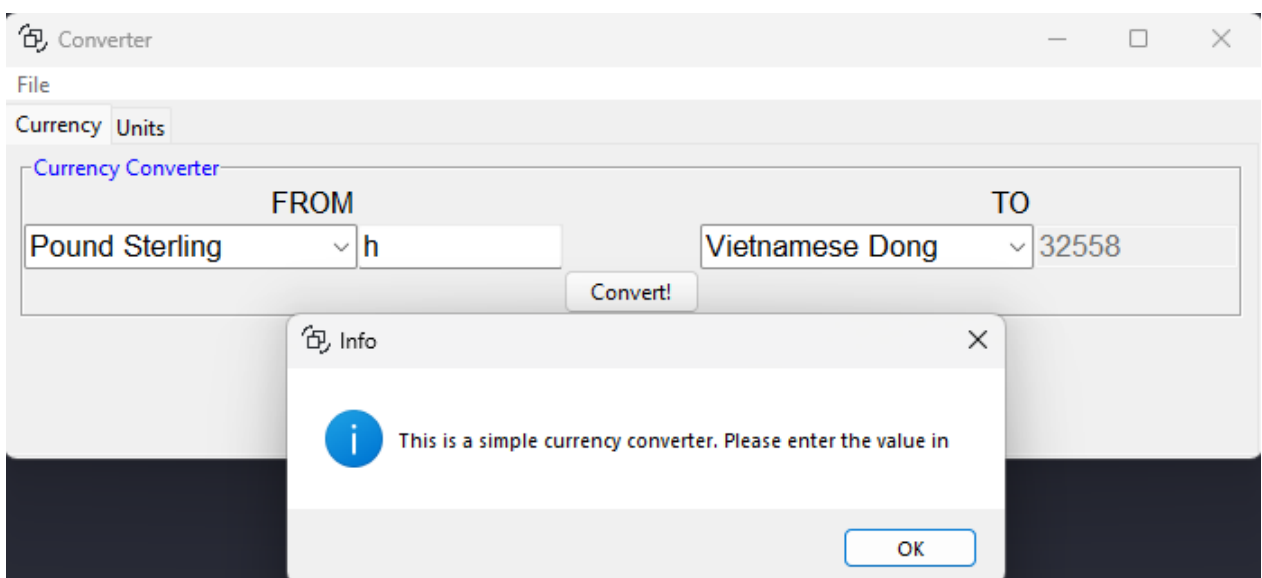
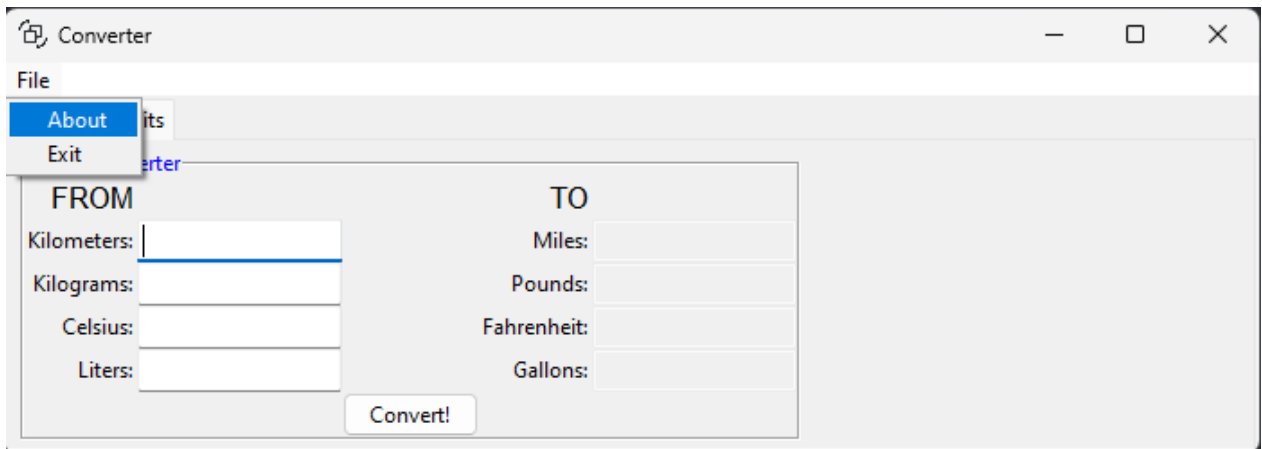


Khi người dùng nhập đầy đủ tất cả các giá trị vào ô nhập và nhấn nút “Convert!”, hệ thống sẽ tính và hiển thị giá trị được quy đổi của đơn vị đo tương ứng.

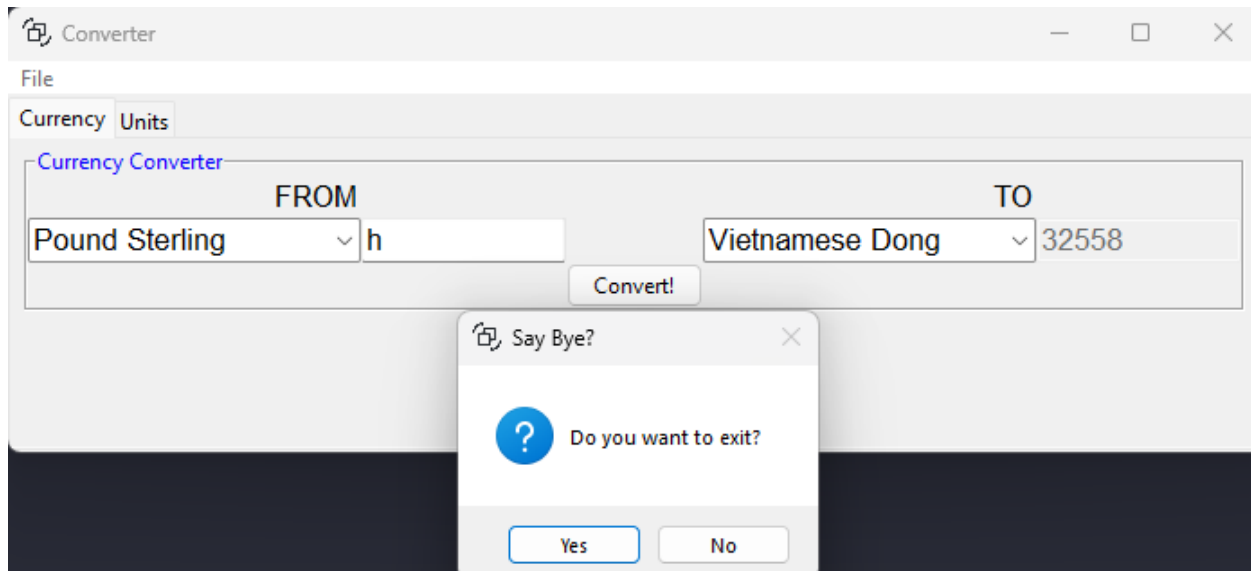


Khi người dùng nhập ký tự khác với ký tự số, hệ thống sẽ hiển thị 1 cửa sổ MessageBox thông báo lỗi.

3. Menu File



Khi người dùng chọn “About”, hệ thống sẽ hiển thị 1 cửa sổ MessageBox thông báo thông tin của giao diện.



Khi người dùng chọn “Exit”, hệ thống sẽ hiển thị 1 cửa sổ MessageBox câu hỏi Yes - No. Nếu người dùng chọn “Yes” thì sẽ thoát giao diện, ngược lại chọn “No” thì sẽ quay trở lại giao diện.

III. MÃ CHƯƠNG TRÌNH

```
import tkinter as tk
from tkinter import messagebox as msg
from tkinter import Menu, ttk

win = tk.Tk()

win.title("Converter")
win.iconbitmap('Baitap1/convert.ico')

tabControl = ttk.Notebook(win)
tab1 = ttk.Frame(tabControl)
tabControl.add(tab1, text="Currency")
tabControl.pack(expand=1, fill="both")

tab2 = ttk.Frame(tabControl)
tabControl.add(tab2, text="Units")
tabControl.pack(expand=1, fill="both")

#Currency Converter Frame
curFrame = tk.LabelFrame(tab1, text='Currency Converter', fg='blue')
curFrame.grid(column=0, row=0, padx=8, pady=4)

ttk.Label(curFrame, text="FROM", font=("Helvetica",
13)).grid(column=0, row=0, sticky=tk.E)
ttk.Label(curFrame, text="TO", font=("Helvetica",
13)).grid(column=3, row=0, sticky=tk.E)

money1 = tk.StringVar()
money1_chosen = ttk.Combobox(curFrame, width=18,
textvariable=money1, font=("Helvetica", 13))
money1_chosen['values'] = ("Vietnamese Dong", "US Dollar", "Pound
Sterling", "Japanese Yen", "South Korean Won", "Chinese Yuan")
money1_chosen.grid(column=0, row=1)

convertValue = tk.StringVar()
convertValue_entered = ttk.Entry(curFrame, width=12,
textvariable=convertValue, font=("Helvetica", 13))
convertValue_entered.grid(column=1, row=1)
```

```

money2 = tk.StringVar()
money2_chosen = ttk.Combobox(curFrame, width=18,
textvariable=money2, font=("Helvetica", 13))
money2_chosen['values'] = ("Vietnamese Dong", "US Dollar", "Pound
Sterling", "Japanese Yen", "South Korean Won", "Chinese Yuan")
money2_chosen.grid(column=3, row=1)

convertedValue = tk.StringVar()
convertedValue_entered = ttk.Entry(curFrame, width=12,
textvariable=convertedValue, font=("Helvetica", 13),
state="disabled")
convertedValue_entered.grid(column=4, row=1)

def format_value(value):
    if value.is_integer():
        return "{:.0f}".format(value)
    else:
        return "{:.6f}".format(value)

def convert():
    try:
        convert_value = float(convertValue.get())
        if money1_chosen.get() == "Vietnamese Dong":
            if money2_chosen.get() == "Vietnamese Dong":
                msg.showwarning("Warning", "Same currency")
            elif money2_chosen.get() == "US Dollar":
                usd_value = convert_value * 0.000040
                convertedValue.set(format_value(usd_value))
            elif money2_chosen.get() == "Pound Sterling":
                gbp_value = convert_value * 0.000031
                convertedValue.set(format_value(gbp_value))
            elif money2_chosen.get() == "Japanese Yen":
                jpy_value = convert_value * 0.0060
                convertedValue.set(format_value(jpy_value))
            elif money2_chosen.get() == "South Korean Won":
                krw_value = convert_value * 0.054
                convertedValue.set(format_value(krw_value))
            elif money2_chosen.get() == "Chinese Yuan":
                cny_value = convert_value * 0.00028

```

```

        convertedValue.set(format_value(cny_value))

elif money1_chosen.get() == "US Dollar":
    if money2_chosen.get() == "Vietnamese Dong":
        vnd_value = convert_value * 24923
        convertedValue.set(format_value(vnd_value))
    elif money2_chosen.get() == "US Dollar":
        msg.showwarning("Warning", "Same currency")
    elif money2_chosen.get() == "Pound Sterling":
        gbp_value = convert_value * 0.76
        convertedValue.set(format_value(gbp_value))
    elif money2_chosen.get() == "Japanese Yen":
        jpy_value = convert_value * 147
        convertedValue.set(format_value(jpy_value))
    elif money2_chosen.get() == "South Korean Won":
        krw_value = convert_value * 1348
        convertedValue.set(format_value(krw_value))
    elif money2_chosen.get() == "Chinese Yuan":
        cny_value = convert_value * 7.05
        convertedValue.set(format_value(cny_value))

elif money1_chosen.get() == "Pound Sterling":
    if money2_chosen.get() == "Vietnamese Dong":
        vnd_value = convert_value * 32558
        convertedValue.set(format_value(vnd_value))
    elif money2_chosen.get() == "US Dollar":
        usd_value = convert_value * 1.31
        convertedValue.set(format_value(usd_value))
    elif money2_chosen.get() == "Pound Sterling":
        msg.showwarning("Warning", "Same currency")
    elif money2_chosen.get() == "Japanese Yen":
        jpy_value = convert_value * 193.7
        convertedValue.set(format_value(jpy_value))
    elif money2_chosen.get() == "South Korean Won":
        krw_value = convert_value * 1765
        convertedValue.set(format_value(krw_value))
    elif money2_chosen.get() == "Chinese Yuan":
        cny_value = convert_value * 9.24
        convertedValue.set(format_value(cny_value))

```

```

elif money1_chosen.get() == "Japanese Yen":
    if money2_chosen.get() == "Vietnamese Dong":
        vnd_value = convert_value * 168.5
        convertedValue.set(format_value(vnd_value))
    elif money2_chosen.get() == "US Dollar":
        usd_value = convert_value * 0.007
        convertedValue.set(format_value(usd_value))
    elif money2_chosen.get() == "Pound Sterling":
        gbp_value = convert_value * 0.005
        convertedValue.set(format_value(gbp_value))
    elif money2_chosen.get() == "Japanese Yen":
        msg.showwarning("Warning", "Same currency")
    elif money2_chosen.get() == "South Korean Won":
        krw_value = convert_value * 9.12
        convertedValue.set(format_value(krw_value))
    elif money2_chosen.get() == "Chinese Yuan":
        cny_value = convert_value * 0.05
        convertedValue.set(format_value(cny_value))

elif money1_chosen.get() == "South Korean Won":
    if money2_chosen.get() == "Vietnamese Dong":
        vnd_value = convert_value * 18.46
        convertedValue.set(format_value(vnd_value))
    elif money2_chosen.get() == "US Dollar":
        usd_value = convert_value * 0.00074
        convertedValue.set(format_value(usd_value))
    elif money2_chosen.get() == "Pound Sterling":
        gbp_value = convert_value * 0.00057
        convertedValue.set(format_value(gbp_value))
    elif money2_chosen.get() == "Japanese Yen":
        jpy_value = convert_value * 0.11
        convertedValue.set(format_value(jpy_value))
    elif money2_chosen.get() == "South Korean Won":
        msg.showwarning("Warning", "Same currency")
    elif money2_chosen.get() == "Chinese Yuan":
        cny_value = convert_value * 0.0052
        convertedValue.set(format_value(cny_value))

elif money1_chosen.get() == "Chinese Yuan":
    if money2_chosen.get() == "Vietnamese Dong":

```

```

        vnd_value = convert_value * 3529
        convertedValue.set(format_value(vnd_value))
    elif money2_chosen.get() == "US Dollar":
        usd_value = convert_value * 0.14
        convertedValue.set(format_value(usd_value))
    elif money2_chosen.get() == "Pound Sterling":
        gbp_value = convert_value * 0.11
        convertedValue.set(format_value(gbp_value))
    elif money2_chosen.get() == "Japanese Yen":
        jpy_value = convert_value * 21
        convertedValue.set(format_value(jpy_value))
    elif money2_chosen.get() == "South Korean Won":
        krw_value = convert_value * 191
        convertedValue.set(format_value(krw_value))
    elif money2_chosen.get() == "Chinese Yuan":
        msg.showwarning("Warning", "Same currency")
except ValueError:
    msg.showerror("Error", "Invalid input")

btnConvert = ttk.Button(curFrame, text="Convert!", command=convert)
btnConvert.grid(column=2, row=6)

#Units Converter Frame
uniFrame = tk.LabelFrame(tab2, text='Units Converter', fg='blue')
uniFrame.grid(column=0, row=0, padx=8, pady=4)

ttk.Label(uniFrame, text="FROM", font=("Helvetica",
13)).grid(column=0, row=0, sticky=tk.E)
ttk.Label(uniFrame, text="TO", font=("Helvetica",
13)).grid(column=3, row=0, sticky=tk.E)
ttk.Label(uniFrame, text="Kilometers:").grid(column=0, row=1,
sticky=tk.E)
ttk.Label(uniFrame, text="Miles:").grid(column=3, row=1,
sticky=tk.E)
ttk.Label(uniFrame, text="Kilograms:").grid(column=0, row=2,
sticky=tk.E)
ttk.Label(uniFrame, text="Pounds:").grid(column=3, row=2,
sticky=tk.E)
ttk.Label(uniFrame, text="Celsius:").grid(column=0, row=3,
sticky=tk.E)

```

```

ttk.Label(uniformFrame, text="Fahrenheit:").grid(column=3, row=3,
sticky=tk.E)
ttk.Label(uniformFrame, text="Liters:").grid(column=0, row=4,
sticky=tk.E)
ttk.Label(uniformFrame, text="Gallons:").grid(column=3, row=4,
sticky=tk.E)

km = tk.StringVar()
km_entered = ttk.Entry(uniformFrame, width=12, textvariable=km,
font=("", 12))
km_entered.grid(column=1, row=1)
km_entered.focus()

mile = tk.StringVar()
mile_entered = ttk.Entry(uniformFrame, width=12, textvariable=mile,
font=("", 12), state="disabled")
mile_entered.grid(column=4, row=1)

kg = tk.StringVar()
kg_entered = ttk.Entry(uniformFrame, width=12, textvariable=kg,
font=("", 12))
kg_entered.grid(column=1, row=2)

lbs = tk.StringVar()
lbs_entered = ttk.Entry(uniformFrame, width=12, textvariable=lbs,
font=("", 12), state="disabled")
lbs_entered.grid(column=4, row=2)

cel = tk.StringVar()
cel_entered = ttk.Entry(uniformFrame, width=12, textvariable=cel,
font=("", 12))
cel_entered.grid(column=1, row=3)

fah = tk.StringVar()
fah_entered = ttk.Entry(uniformFrame, width=12, textvariable=fah,
font=("", 12), state="disabled")
fah_entered.grid(column=4, row=3)

lit = tk.StringVar()

```

```

lit_entered = ttk.Entry(uniFrame, width=12, textvariable=lit,
font=(" ", 12))
lit_entered.grid(column=1, row=4)

gal = tk.StringVar()
gal_entered = ttk.Entry(uniFrame, width=12, textvariable=gal,
font=(" ", 12), state="disabled")
gal_entered.grid(column=4, row=4)

def uConvert():
    km = km_entered.get()
    kg = kg_entered.get()
    celsius = cel_entered.get()
    liters = lit_entered.get()

    if not km or not kg or not celsius or not liters:
        msg.showerror("Input Error", "Please fill in all the
fields.")
        return
    try:
        miles = float(km) * 0.621371
        pounds = float(kg) * 2.20462
        fahrenheit = (float(celsius) * 9/5) + 32
        gallons = float(liters) * 0.264172

        mile_entered.config(state="normal")
        mile_entered.delete(0, tk.END)
        mile_entered.insert(0, f"{miles:.2f}")
        mile_entered.config(state="disabled")

        lbs_entered.config(state="normal")
        lbs_entered.delete(0, tk.END)
        lbs_entered.insert(0, f"{pounds:.2f}")
        lbs_entered.config(state="disabled")

        fah_entered.config(state="normal")
        fah_entered.delete(0, tk.END)
        fah_entered.insert(0, f"{fahrenheit:.2f}")
        fah_entered.config(state="disabled")

```



```

        gal_entered.config(state="normal")
        gal_entered.delete(0, tk.END)
        gal_entered.insert(0, f"{gallons:.2f}")
        gal_entered.config(state="disabled")
    except ValueError:
        msg.showerror("Input Error", "Please enter valid numbers.")

btnConvert = ttk.Button(uniFrame, text="Convert!", command=uConvert)
btnConvert.grid(column=2, row=5)

menu_bar = Menu(win)
win.config(menu=menu_bar)

def _msgBoxInfo():
    msg.showinfo("Info", "This is a simple currency converter.
Please enter the value in")

def _msgBoxAsk():
    answer = msg.askyesno("Say Bye?", "Do you want to exit?")
    if answer:
        win.destroy()
    else:
        pass

file_menu = Menu(menu_bar, tearoff=0)
file_menu.add_command(label="About", command=_msgBoxInfo)
file_menu.add_command(label="Exit", command=_msgBoxAsk)

menu_bar.add_cascade(label="File", menu=file_menu)

money1_chosen.focus()

win.mainloop()

```

IV. GITHUB

Nguồn: <https://github.com/YueTruong/pythonnc>

HẾT