2274802010310\_nguyễn công huy

Giao diện:

Ảnh có chứa văn bản, ảnh chụp màn hình, số, Phông chữ

Mô tả được tạo tự động

Giải pt bậc 1:

Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, số

Mô tả được tạo tự động

Giải pt bậc 2:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Kiểm tra số nguyên tố:

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

Code giao dien:

import tkinter as tk

from tkinter import ttk

from tkinter import Menu

from tkinter import messagebox as msg

from code\_pt import \*

# Tạo giao diện chính

win = tk.Tk()

win.title('Giải Phương Trình')

# khóa điều chỉnh GUI

win.resizable(False, False)

# Tạo menu bar

menu\_bar = Menu(win)

win.config(menu=menu\_bar)

def clear\_entries():

    # Reset các ô nhập hệ số a, b, c và kết quả phương trình

    a.set(0)

    b.set(0)

    c.set(0)

    kq.set("")

    # Reset ô nhập số kiểm tra nguyên tố và kết quả kiểm tra

    snt\_input.set(0)

    result\_output.set("")

# Tạo menu File và Help

file\_menu = Menu(menu\_bar, tearoff=0)

menu\_bar.add\_cascade(label="File", menu=file\_menu)

file\_menu.add\_command(label="Exit", command=win.quit)

file\_menu.add\_command(label="New", command=clear\_entries)

def \_msgBox():

    msg.showinfo('Python Message Info Box', 'Chương trình tính các phương trình và kiểm tra số nguyên tố \n')

help\_menu = Menu(menu\_bar, tearoff=0)

menu\_bar.add\_cascade(label="Help", menu=help\_menu)

help\_menu.add\_command(label="About", command=\_msgBox)

# Tạo tab

tabControl = ttk.Notebook(win)

Pg1 = ttk.Frame(tabControl)

tabControl.add(Pg1, text="Phương trình")

tabControl.pack(expand=1, fill="both")

Pg2 = ttk.Frame(tabControl)

tabControl.add(Pg2, text="Số Nguyên Tố")

tabControl.pack(expand=2, fill="both")

# Giải Phương Trình

# Tạo Frame cho hệ số

mighty = ttk.LabelFrame(Pg1, text='Các hệ số: ')

mighty.grid(column=0, row=0, padx=8, pady=4)

# Ô nhập liệu cho hệ số a

a\_label = ttk.Label(mighty, text="a:")

a\_label.grid(column=0, row=0, sticky='W')

a = tk.IntVar()

a\_entry = ttk.Entry(mighty, width=10, textvariable=a)

a\_entry.grid(column=0, row=1, padx=5, pady=5, sticky='W')

# Ô nhập liệu cho hệ số b

b\_label = ttk.Label(mighty, text="b:")

b\_label.grid(column=1, row=0, sticky='W')

b = tk.IntVar()

b\_entry = ttk.Entry(mighty, width=10, textvariable=b)

b\_entry.grid(column=1, row=1, padx=5, pady=5, sticky='W')

# Ô nhập liệu cho hệ số c

c\_label = ttk.Label(mighty, text="c:")

c\_label.grid(column=2, row=0, sticky='W')

c = tk.IntVar()

c\_entry = ttk.Entry(mighty, width=10, textvariable=c)

c\_entry.grid(column=2, row=1, padx=5, pady=5, sticky='W')

# Tạo Frame cho kết quả

Ketqua = ttk.LabelFrame(Pg1, text='Giải')

Ketqua.grid(column=0, row=1, columnspan=3, padx=8, pady=4, sticky='EW')

# Nhãn "Kết quả:"

ketqua\_lb = ttk.Label(Ketqua, text="Kết quả:")

ketqua\_lb.grid(column=0, row=0, sticky='W')

# Ô nhập liệu hiển thị kết quả

kq = tk.StringVar()

kq\_en = ttk.Entry(Ketqua, width=40, textvariable=kq, state='readonly')

kq\_en.grid(column=1, row=0, columnspan=2, padx=10, pady=5, sticky='EW')

# Tạo Frame cho danh sách chọn phương trình

Tinh = ttk.LabelFrame(Pg1, text='Phương Trình')

Tinh.grid(column=1, row=0, padx=8, pady=4)

# Hàm kiểm tra input

def Input():

    try:

        a\_value = a.get()

        b\_value = b.get()

        c\_value = c.get()

        return a\_value, b\_value, c\_value

    except tk.TclError:

        msg.showerror("Lỗi", "Vui lòng nhập vào các số hợp lệ.")

        return None, None, None

# Hàm giải phương trình bậc 1

def PTB1():

    a\_value, b\_value, \_ = Input()

    if a\_value is not None and b\_value is not None:

        pt = Giai\_Phuong\_Trinh(a\_value, b\_value)

        result = pt.giai\_phuong\_trinh\_bac\_1()

        kq.set(str(result))

# Hàm giải phương trình bậc 2

def PTB2():

    a\_value, b\_value, c\_value = Input()

    if a\_value is not None and b\_value is not None and c\_value is not None:

        pt = Giai\_Phuong\_Trinh(a\_value, b\_value, c\_value)

        result = pt.giai\_phuong\_trinh\_bac\_2()

        kq.set(str(result))

# Hàm xử lý lựa chọn từ Listbox

def handle\_selection(event):

    selected\_option = listbox.get(listbox.curselection())

    if selected\_option == "Bậc 1":

        PTB1()

    elif selected\_option == "Bậc 2":

        PTB2()

# Tạo Listbox để chọn phương trình

listbox = tk.Listbox(Tinh, height=2)

listbox.insert(1, "Bậc 1")

listbox.insert(2, "Bậc 2")

listbox.grid(column=0, row=0, padx=5, pady=10)

listbox.bind("<<ListboxSelect>>", handle\_selection)

# Kiểm tra số nguyên tố

# Tạo Frame cho kiểm tra số nguyên tố

mighty2 = ttk.LabelFrame(Pg2, text='Kiểm tra số nguyên tố: ')

mighty2.grid(column=0, row=0, padx=8, pady=4)

# Ô nhập liệu cho số nguyên tố

snt\_label = ttk.Label(mighty2, text="Số:")

snt\_label.grid(column=0, row=0, sticky='W')

snt\_input = tk.IntVar()

snt\_entry = ttk.Entry(mighty2, width=10, textvariable=snt\_input)

snt\_entry.grid(column=0, row=1, padx=5, pady=5, sticky='W')

# Nhãn kết quả cho kiểm tra số nguyên tố

result\_label = ttk.Label(mighty2, text="Kết quả:")

result\_label.grid(column=1, row=0, sticky='W')

result\_output = tk.StringVar()

result\_entry = ttk.Entry(mighty2, width=35, textvariable=result\_output, state='readonly')

result\_entry.grid(column=1, row=1, padx=10, pady=5, sticky='W')

def kiem\_tra\_snt():

    try:

        number = snt\_input.get()

        kt\_snt = KT\_SNT(number)

        if kt\_snt.kiem\_tra\_so\_nguyen\_to():

            result\_output.set(f"{number} là số nguyên tố.")

        else:

            result\_output.set(f"{number} không phải là số nguyên tố.")

    except:

        msg.showerror("Lỗi", "Vui lòng nhập một số nguyên hợp lệ.")

# Nút kiểm tra số nguyên tố

check\_prime\_btn = ttk.Button(mighty2, text="Kiểm tra", command=kiem\_tra\_snt)

check\_prime\_btn.grid(column=0, row=3, padx=5, pady=10)

win.mainloop()

code pt:

from math import sqrt

from tkinter import messagebox as msg

class Giai\_Phuong\_Trinh:

    def \_\_init\_\_(self, a, b, c=0):

        self.a = a

        self.b = b

        self.c = c

    # Giải phương trình bậc 1

    def giai\_phuong\_trinh\_bac\_1(self):

        if self.a == 0:

            if self.b == 0:

                return "Phương trình có vô số nghiệm"

            else:

                return "Phương trình vô nghiệm"

        else:

            nghiem = -self.b / self.a

            return f"Nghiệm x = {round(nghiem, 2)}"

    # Giải phương trình bậc 2

    def giai\_phuong\_trinh\_bac\_2(self):

        try:

            delta = self.b\*\*2 - 4\*self.a\*self.c

            if delta < 0:

                return "Phương trình vô nghiệm"

            elif delta == 0:

                x = -self.b / (2\*self.a)

                return f"Nghiệm kép: x = {x:.2f}"

            else:

                x1 = (-self.b + sqrt(delta)) / (2\*self.a)

                x2 = (-self.b - sqrt(delta)) / (2\*self.a)

                return f"Nghiệm x1 = {x1:.2f}, x2 = {x2:.2f}"

        except ZeroDivisionError:

            return "Hệ số a không thể bằng 0. Vui lòng nhập lại."

        except Exception as e:

            return f"Lỗi: {str(e)}. Vui lòng nhập lại."

class KT\_SNT:

    def \_\_init\_\_(self, n):

        self.n = n

    # Kiểm tra số nguyên tố

    def kiem\_tra\_so\_nguyen\_to(self):

        if self.n <= 1:

            return False

        for i in range(2, int(self.n\*\*0.5) + 1):

            if self.n % i == 0:

                return False

        return True