

TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP
KHOA ĐIỆN TỬ
BỘ MÔN : CÔNG NGHỆ THÔNG TIN



BÀI TẬP KẾT THÚC MÔN HỌC
MÔN HỌC : LẬP TRÌNH PYTHON

Sinh viên : Ngô Thị Thùy Linh

Lớp : K58KTP

MSSV : K225480106038

GVHD : Ths . Nguyễn Văn Huy

Link Github : <https://github.com/linhngo2004/baitapketthucmonhoc.git>

THÁI NGUYÊN - 2024

TRƯỜNG ĐHKTCN CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM
KHOA ĐIỆN TỬ Độc lập – Tự do – Hạnh phúc

BÀI TẬP KẾT THÚC MÔN HỌC

Môn học : Lập trình Python

Bộ môn : Công nghệ thông tin

Sinh viên: Ngô Thị Thùy Linh

MSSV : K225480106038

Lớp: K58KTP

Ngành : Kỹ thuật máy tính

GVHD: Ths. Nguyễn Văn Huy

Ngày giao đề : 20/5/2025

Ngày hoàn thành : 10/6/2025

Tên đề tài : Tạo ứng dụng GUI cho phép chọn thư mục, liệt kê file theo từng loại (.txt, .py, .jpg), và cho phép mở file.

Yêu cầu :

- Chọn thư mục
- Hiện thị danh sách file có đuôi .txt, .py, .jpg
- Mở file bằng chương trình mặc định khi double-click.

GIÁO VIÊN HƯỚNG DẪN
(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

Xếp loại: Điểm :

Thái Nguyên, ngày....tháng.....năm 20....

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

MỤC LỤC

LỜI CAM ĐOAN.....	5
LỜI NÓI ĐẦU.....	6
DANH MỤC CÁC BẢNG VÀ HÌNH VẼ, ĐỒ THỊ.....	7
CHƯƠNG 1 : GIỚI THIỆU ĐẦU BÀI.....	8
CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT.....	9
2.1. Các nội dung chuyên môn sử dụng.....	9
2.2. Lý thuyết về giao diện GUI với Tkinter.....	9
CHƯƠNG 3 : THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH.....	10
3.1. Sơ đồ khối hệ thống.....	10
3.2. Sơ đồ khối thuật toán chính.....	10
3.3. Cấu trúc dữ liệu.....	11
3.4. Các hàm chính trong chương trình.....	11
CHƯƠNG 4 : THỬ NGHIỆM VÀ KẾT LUẬN.....	12
4.1. Thực nghiệm.....	12
4.2. Kết luận.....	15
KẾT LUẬN.....	16

LỜI CAM ĐOAN

Chúng em xin cam đoan “ Bài 9 :Trình quản lý thư mục GUI ” là kết quả do tôi trực tiếp thực hiện, dưới sự hướng dẫn và hỗ trợ từ các cá nhân và đơn vị liên quan.

Các nội dung trình bày trong báo cáo hoàn toàn trung thực, không sao chép từ bất kỳ tài liệu nào khác mà không trích dẫn rõ ràng nguồn gốc. Nếu phát hiện có bất kỳ hành vi gian lận nào, chúng em xin hoàn toàn chịu trách nhiệm t

Chúng em xin chân thành cảm ơn sự hỗ trợ và góp ý của thầy Ths.Nguyễn Văn Huy, cùng các đơn vị đã tạo điều kiện để chúng em hoàn thành đề tài này.

SINH VIÊN THỰC HIỆN
(Ký và ghi rõ họ tên)

LỜI NÓI ĐẦU

Trong lập trình ứng dụng hiện đại, việc cung cấp cho người dùng một giao diện đồ họa trực quan và thân thiện luôn là yếu tố quan trọng giúp nâng cao trải nghiệm sử dụng phần mềm. Python, với thư viện tích hợp sẵn Tkinter, mang đến cho lập trình viên khả năng xây dựng các ứng dụng GUI một cách dễ dàng và hiệu quả. Trong khi đó, thư viện os hỗ trợ thao tác với hệ thống tập tin, giúp việc truy xuất, liệt kê và mở file trở nên đơn giản.

Bài thực hành số 9 có mục tiêu xây dựng một trình quản lý thư mục dạng GUI, trong đó người dùng có thể chọn một thư mục bất kỳ, liệt kê các tệp tin trong đó theo từng loại định dạng (.txt, .py, .jpg, .py, v.v), và mở các file bằng phần mềm mặc định của hệ điều hành. Thông qua bài tập này, em sẽ rèn luyện kỹ năng sử dụng Tkinter để tạo giao diện, sử dụng os để thao tác với thư mục và file, đồng thời xử lý các tình huống ngoại lệ như đường dẫn không hợp lệ hoặc không có quyền truy cập. Đây là bước chuẩn bị quan trọng trước khi làm việc với các ứng dụng phức tạp hơn trong tương lai.

Chúng em xin chân thành cảm ơn sự hướng dẫn tận tình của Thầy ThS. Nguyễn Văn Huy đã giúp cho chúng em hoàn thành đề tài này.

Nhóm rất mong nhận được ý kiến đóng góp từ quý thầy cô để hoàn thiện đề tài tốt hơn.

DANH MỤC CÁC BẢNG VÀ HÌNH VẼ, ĐỒ THỊ

STT	Hình ảnh và bảng	Trang
1	Hình 1 : Biểu đồ phân cấp chức năng	9
2	Bảng 1 : Các bước thử nghiệm	10
3	Hình 4.1 :Giao diện trình quản lý thư mục	13
4	Hình 4.2 : Kết quả của test1.jpg	14
5	Hình 4.3 : Kết quả của test1.txt	14
6	Hình 4.4 :Kết quả của test2.py	15

CHƯƠNG 1 : GIỚI THIỆU ĐẦU BÀI

Đề tài yêu cầu sinh viên xây dựng một ứng dụng giao diện người dùng (GUI) sử dụng ngôn ngữ Python, nhằm mục đích quản lý thư mục và các tệp tin bên trong thư mục đó. Ứng dụng cần cho phép chọn một thư mục bất kỳ trên máy tính, tự động quét và liệt kê các file có phần mở rộng .txt, .py, và .jpg, hiển thị danh sách file dưới dạng bảng có thể tương tác (Treeview), cho phép người dùng double-click để mở file bằng chương trình mặc định của hệ điều hành.

Chương trình được xây dựng với mục tiêu rèn luyện sinh viên khả năng sử dụng thư viện Tkinter để xây dựng giao diện người dùng, làm việc với hệ thống tập tin thông qua thư viện os, xử lý sự kiện GUI, bao gồm chọn thư mục và mở file, bắt lỗi liên quan đến đường dẫn hoặc file không tồn tại.

Một số thách thức chính trong bài toán gồm, tương thích với nhiều định dạng file, giao tiếp giữa giao diện người dùng và các thao tác hệ thống, bắt sự kiện đúng thời điểm (như double-click vào đúng dòng dữ liệu), đảm bảo chương trình hoạt động ổn định, không bị lỗi khi file bị xóa, đổi tên hoặc không tồn tại.

Thông qua bài tập này, sinh viên có cơ hội củng cố và áp dụng các kiến thức đã học về thiết kế GUI với Tkinter, làm việc với hệ thống file trong Python (os, os.path), xử lý ngoại lệ (try-except), tổ chức mã nguồn thành lớp (class) và phương thức (method) một cách rõ ràng.

CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT

2.1. Các nội dung chuyên môn sử dụng

Trong chương trình "Trình quản lý thư mục GUI", các kiến thức và kỹ thuật sau được áp dụng:

- **List (Danh sách):** Cấu trúc dữ liệu lưu trữ tập hợp các phần tử, được dùng để duyệt và xử lý danh sách file trong thư mục.
- **Dictionary (Từ điển):** Dạng dữ liệu ánh xạ khóa - giá trị, thích hợp để lưu thông tin chi tiết của từng file. (Dù không dùng trực tiếp, nhưng có thể dùng để mở rộng).
- **os module:** Dùng để làm việc với hệ thống tệp như quét thư mục, kiểm tra loại file, mở file.
- **Xử lý lỗi (try-except):** Đảm bảo chương trình không bị dừng đột ngột khi gặp lỗi.
- **Tkinter và ttk (Treeview):** Dùng để xây dựng giao diện người dùng, cho phép tương tác trực quan với thư mục và file.

2.2. Lý thuyết về giao diện GUI với Tkinter

- **Tkinter:** Thư viện giao diện người dùng tiêu chuẩn trong Python.
- **Treeview:** Thành phần giao diện thuộc tkinter.ttk dùng để hiển thị dữ liệu dạng bảng.
- **askdirectory():** Hàm cho phép người dùng chọn thư mục trên máy.
- **os.startfile():** Dùng để mở file bằng phần mềm mặc định (Windows).

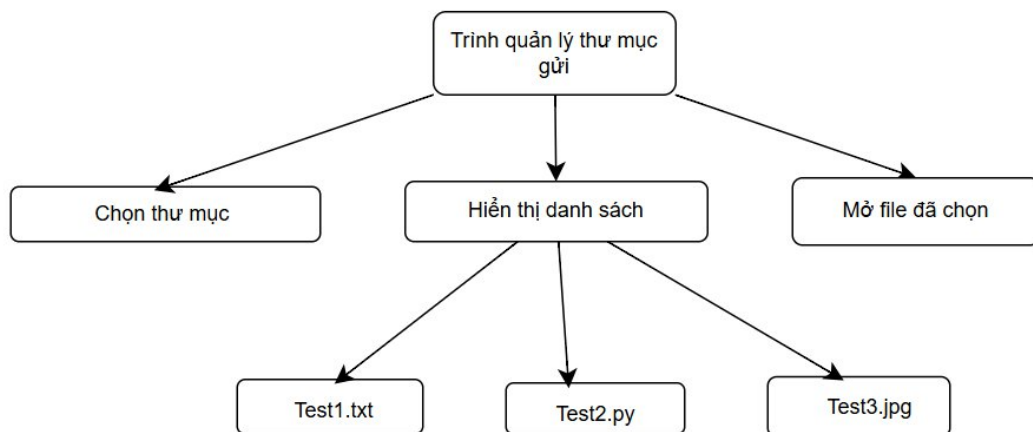
CHƯƠNG 3 : THIẾT KẾ VÀ XÂY DỰNG CHƯƠNG TRÌNH

3.1. Sơ đồ khối hệ thống

➤ **Các module chính:**

- Giao diện chính (Main GUI)
- Chọn thư mục (select_folder)
- Hiển thị file (show_files)
- Mở file (open_file)

➤ **Biểu đồ phân cấp chức năng :**



Hình 1 : Biểu đồ phân cấp chức năng

3.2. Sơ đồ khối thuật toán chính

Thuật toán chọn thư mục và hiển thị file

- Đầu vào: Đường dẫn thư mục từ người dùng.
- Xử lý:
- Quét file bằng `os.listdir()`
 - Kiểm tra định dạng file (.txt, .py, .jpg)
 - Thêm vào Treeview

- Đầu ra: Giao diện hiển thị danh sách file.

Thuật toán mở file

- Đầu vào: File được chọn trong Treeview.
- Xử lý: Dùng `os.startfile(path)` để mở.
- Đầu ra: File mở trong ứng dụng mặc định.

3.3. Cấu trúc dữ liệu

Không dùng cơ sở dữ liệu, chương trình sử dụng các biến tạm để lưu:

- **Danh sách file:** dạng list, chứa tên file lọc từ thư mục.
- **Đường dẫn file:** lưu qua tags trong Treeview để truy xuất lại khi cần mở.

3.4. Các hàm chính trong chương trình

- `__init__()`: Tạo GUI, các nút, Treeview.
- `select_folder()`: Mở hộp thoại chọn thư mục.
- `show_files(folder)`: Quét thư mục, lọc file và hiển thị.
- `open_file()`: Mở file được chọn bằng ứng dụng mặc định

CHƯƠNG 4 : THỬ NGHIỆM VÀ KẾT LUẬN

4.1. Thử nghiệm

Các bước kiểm thử:

Tình huống	Kết quả mong đợi	Kết quả thực tế
Chọn thư mục chứa 3 file	Hiện thị 3 dòng Treeview	Đúng
Nhấn đút chuột vào file .txt	Mở Notepad với nội dung file	Mở được
Chọn thư mục trống	Không hiển thị gì	Không lỗi, danh sách rỗng
Chọn file không hợp lệ	Không hiển thị trong danh sách	Không hiển thị
Lỗi quyền truy cập	Báo lỗi bằng messagebox	Có xử lý ngoại lệ

Bảng 1 : Các bước thử nghiệm

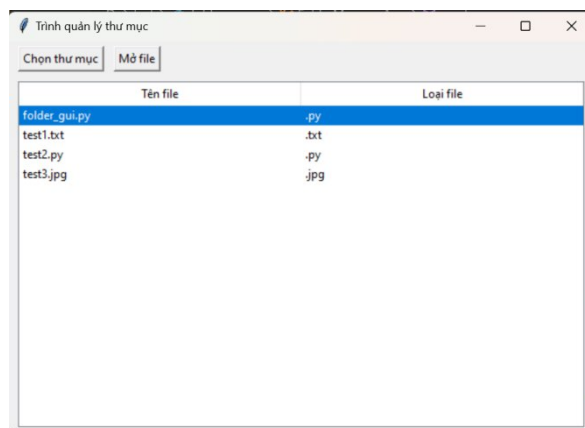
➤ Code

```
import os
import tkinter as tk
from tkinter import ttk, messagebox, filedialog
class FolderManagerApp(tk.Tk):
    def __init__(self):
        super().__init__()
        self.title("Trình quản lý thư mục")
        self.geometry("600x400")
        # Frame chứa nút chọn thư mục
        frame = tk.Frame(self)
        frame.pack(fill=tk.X, padx=10, pady=5)
        self.btn_choose = tk.Button(frame, text="Chọn thư mục",
command=self.choose_folder)
        self.btn_choose.pack(side=tk.LEFT)
        self.btn_open = tk.Button(frame, text="Mở file",
command=self.open_file)
        self.btn_open.pack(side=tk.LEFT, padx=10)
        # Treeview hiển thị file
        columns = ("Tên file", "Loại file", "Đường dẫn")
```

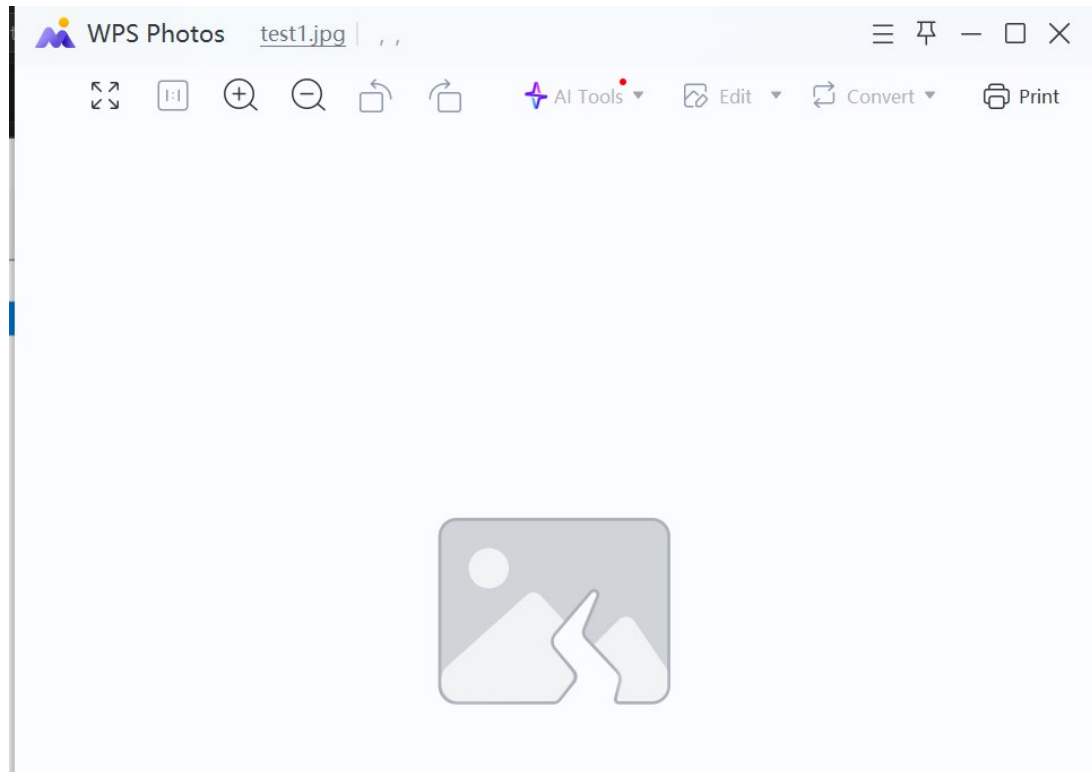
```
self.tree = ttk.Treeview(self, columns=columns, show="headings")
self.tree.heading("Tên file", text="Tên file")
self.tree.heading("Loại file", text="Loại file")
self.tree.heading("Đường dẫn", text="Đường dẫn")
self.tree.column("Đường dẫn", width=0, stretch=False)    # ẩn cột
đường dẫn
self.tree.pack(fill=tk.BOTH, expand=True, padx=10, pady=5)
# Bắt sự kiện double-click mở file
self.tree.bind("<Double-1>", self.on_double_click)
def choose_folder(self):
    folder_path = filedialog.askdirectory()
    if folder_path:
        self.show_files(folder_path)
def show_files(self, folder_path):
    # Xóa dữ liệu cũ
    for item in self.tree.get_children():
        self.tree.delete(item)
    try:
        # Lấy danh sách file trong thư mục
        files = os.listdir(folder_path)
    except FileNotFoundError:
        messagebox.showerror("Lỗi", "Không tìm thấy đường dẫn")
        return
    # Loại và thêm file có đuôi .txt, .py, .jpg
    for f in files:
        if f.lower().endswith(('.txt', '.py', '.jpg')):
            full_path = os.path.join(folder_path, f)
            ext = os.path.splitext(f)[1].lower()
            self.tree.insert("", tk.END, values=(f, ext, full_path))
def open_file(self):
    selected = self.tree.focus()
```

```
if not selected:
    messagebox.showwarning("Chú ý", "Vui lòng chọn file để mở")
    return
full_path = self.tree.item(selected, "values")[2]
if os.path.exists(full_path):
    try:
        os.startfile(full_path)
    except Exception as e:
        messagebox.showerror("Lỗi", f"Không thể mở file:\n{e}")
else:
    messagebox.showerror("Lỗi", "File không tồn tại")
def on_double_click(self, event):
    self.open_file()
if __name__ == "__main__":
    app = FolderManagerApp()
    app.mainloop()
```

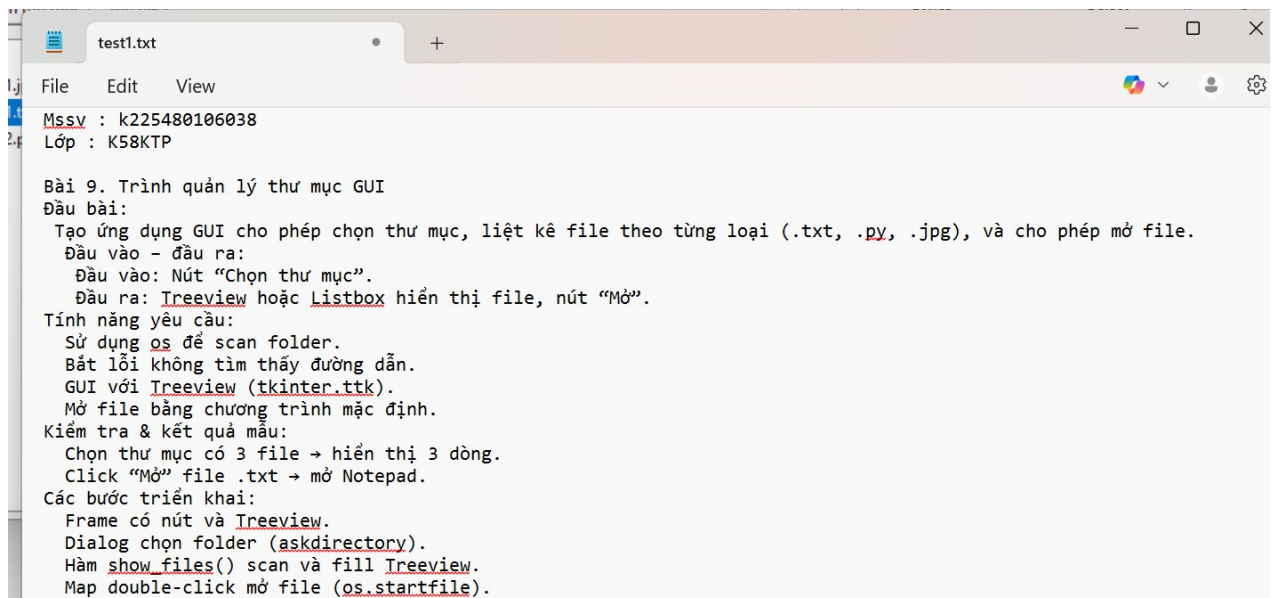
➤ **Ảnh minh họa**



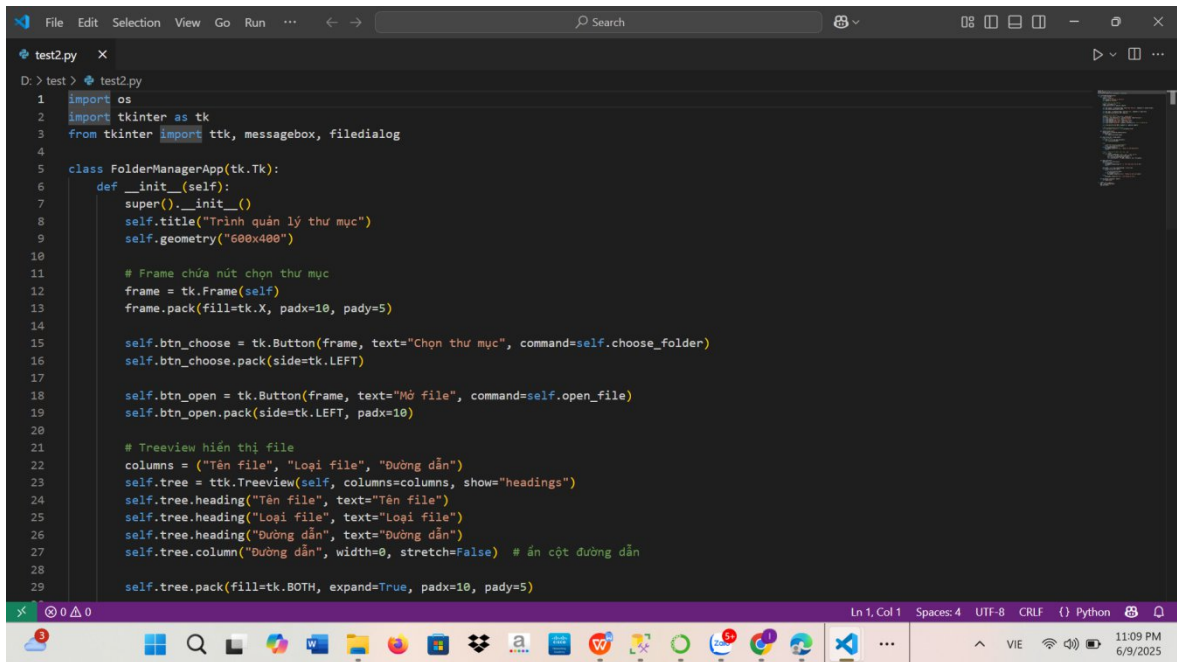
Hình 4.1 : Giao diện trình quản lý thư mục



Hình 4.2 : Kết quả của test1.jpg



Hình 4.3 : Kết quả của test1.txt



```
test2.py
D:\> test > test2.py
1 import os
2 import tkinter as tk
3 from tkinter import ttk, messagebox, filedialog
4
5 class FolderManagerApp(tk.Tk):
6     def __init__(self):
7         super().__init__()
8         self.title("Trình quản lý thư mục")
9         self.geometry("600x400")
10
11         # Frame chứa nút chọn thư mục
12         frame = tk.Frame(self)
13         frame.pack(fill=tk.X, padx=10, pady=5)
14
15         self.btn_choose = tk.Button(frame, text="Chọn thư mục", command=self.choose_folder)
16         self.btn_choose.pack(side=tk.LEFT)
17
18         self.btn_open = tk.Button(frame, text="Mở file", command=self.open_file)
19         self.btn_open.pack(side=tk.LEFT, padx=10)
20
21         # Treeview hiển thị file
22         columns = ("Tên file", "Loại file", "Đường dẫn")
23         self.tree = ttk.Treeview(self, columns=columns, show="headings")
24         self.tree.heading("Tên file", text="Tên file")
25         self.tree.heading("Loại file", text="Loại file")
26         self.tree.heading("Đường dẫn", text="Đường dẫn")
27         self.tree.column("Đường dẫn", width=0, stretch=False) # ẩn cột đường dẫn
28
29         self.tree.pack(fill=tk.BOTH, expand=True, padx=10, pady=5)
```

Hình 4.4 : Kết quả của test2.py

4.2. Kết luận

➤ Sản phẩm đã làm được:

- Xây dựng ứng dụng giao diện đơn giản, trực quan.
- Quét thư mục, lọc định dạng file, hiển thị và mở file.
- Có xử lý ngoại lệ, giúp chương trình ổn định.

➤ Học được:

- Cách dùng tkinter, Treeview, os để tương tác hệ thống.
- Kỹ năng xử lý lỗi, xây dựng giao diện người dùng cơ bản.

➤ Hướng cải tiến:

- Thêm tính năng xóa, đổi tên file.
- Cho phép lọc thêm các định dạng khác.
- Hỗ trợ mở file trên hệ điều hành khác (Linux, macOS)

KẾT LUẬN

Sau khi hoàn thành bài thực hành, chương trình quản lý thư mục GUI đã đáp ứng đầy đủ các yêu cầu được đặt ra: có thể chọn thư mục thông qua hộp thoại, hiển thị danh sách file dưới dạng bảng Treeview, phân loại file theo định dạng, và mở file bằng phần mềm mặc định khi người dùng nhấp đúp chuột vào tên file. Giao diện thân thiện, cách sử dụng đơn giản và phản hồi nhanh chóng là những điểm mạnh của ứng dụng này.

Bài học không chỉ giúp em nắm vững cách sử dụng Tkinter để xây dựng giao diện, mà còn rèn luyện kỹ năng thao tác hệ thống tập tin với thư viện os, xử lý sự kiện trong GUI, và đảm bảo chương trình hoạt động ổn định thông qua việc xử lý ngoại lệ. Đây là một bước đệm quan trọng trong việc phát triển các phần mềm thực tế, nơi mà giao diện trực quan và khả năng tương tác với hệ thống là hai yếu tố không thể thiếu.

Thông qua bài thực hành, em có cơ hội áp dụng kiến thức lập trình vào tình huống cụ thể, từ đó nâng cao kỹ năng tư duy logic, phân tích yêu cầu và hiện thực hóa ý tưởng thành sản phẩm phần mềm có thể sử dụng được.

TÀI LIỆU THAM KHẢO

- [1] Mark Lutz, Programming Python, 4th Edition, O'Reilly Media, 2011.*
- [2] John V. Guttag, Introduction to Computation and Programming Using Python, The MIT Press, 2016.*
- [3] <https://tkdocs.com/> — Trang hướng dẫn chi tiết và thực hành Tkinter.*
- [4] <https://www.pythonguis.com/> — Tài nguyên học tạo giao diện GUI với Python.*

MÃ QR GITHUB VÀ YOUTUBE

➤ Mã QR Github



➤ Mã QR Youtube

☐ bài tập kết thúc môn học python 



<https://qr.me-qr.com/unV3nWs9>

Kiểu: Liên kết

Tạo: 09.06.25
