


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	Nguyễn Tiến Thắng
Lớp	K58 KTPM
Giáo viên giảng dạy	Ths.Nguyễn Văn Huy
Link github	https://github.com/nTienthang03/L-p-tr-nhPYTHON
	

Thái nguyên 2025

TRƯỜNG ĐHKTCN

CỘNG HOÀ 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: Nguyễn Tiến Thắng

Lớp: K58KTPM

Ngành: Kỹ thuật máy tính

Giáo viên hướng dẫn: Ths. Nguyễn Văn Huy

Ngày giao đề 20/05/2025

Ngày hoàn thành : 06/06/2025

Tên đề tài : Trình quản lý thư mục GUI

Yêu cầu :

- Sử dụng os để scan folder.
- Bắt lỗi không tìm thấy đường dẫn.
- GUI với Treeview (tkinter.ttk).
- Mở file bằng chương trình mặc định

.....

.....

.....

.....

GIÁO VIÊN HƯỚNG DẪN

(Ký và ghi rõ họ tên)

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, ngày....tháng....năm 20....
GIÁO VIÊN HƯỚNG DẪN
(Ký ghi rõ họ tên)

Mục Lục

Danh Mục Hình Ảnh.....	5
Lời cảm ơn	6
LỜI NÓI ĐẦU	7
1.1 Mô tả đề tài.....	8
1.2 Thách thức của đề tài	8
1.3 Kiến thức vận dụng	9
CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	10
2.1 Danh sách list trong python	10
2.2 Widget Treeview	10
2.3 Thư viện Tkinter	11
2.4 Module os.....	12
Chương 3 Thiết kế và xây dựng chương trình	13
3.1 Sơ đồ khối hệ thống	13
3.1.1 Mô tả các module chính	13
3.1.2 Biểu đồ phân cấp chức năng	13
3.2 Sơ đồ khối các thuật toán chính	14
3.2.1 Thuật toán chọn thư mục:	14
3.2.2 Thuật toán hiển thị danh sách tệp:	14
3.2.3 Thuật toán mở tệp:	15
3.3 Cấu trúc dữ liệu	16
3.4 Chương trình	17
3.4.1 Hàm select_folder().....	17
3.4.2, hàm def show_files(folder_path):	17
3.4.3 Hàm def open_file(event):	18
Chương 4 Thực nghiệm và kết luận.....	20
4.1. Thực nghiệm	20
4.2 Kết Luận.....	21
4.2.1 Sản phẩm đã làm được :.....	21
4.2.2 Học được từ Sản Phẩm.....	22
4.2.3 Cải tiến Sản phẩm	22
Tài liệu tham khảo.....	23

Danh Mục Hình Ảnh

Hình 3.1 Sơ đồ phân cấp chức năng.....	13
Hình 3.2 Sơ đồ khối chọn thư mục	14
Hình 3.3 Sơ đồ khối hiện thị danh sách tệp	15
Hình 3.4 Sơ đồ khối mở tệp	16
Hình 3.5 Các trường thông tin	16
Hình 4.1 Giao diện thực nghiệm	20
Hình 4.2 kết quả thực nghiệm	21

Lời cảm ơn

Em xin gửi lời cảm ơn chân thành đến thầy Th.sNguyễn Văn Huy – người đã tận tình giảng dạy môn Lập trình Python và hướng dẫn em trong quá trình thực hiện bài tập.

Nhờ sự chỉ bảo tận tình, kiến thức sâu rộng và tinh thần hỗ trợ của thầy, em đã có thể tiếp cận ngôn ngữ lập trình Python một cách hiệu quả, hiểu rõ hơn về tư duy lập trình cũng như ứng dụng thực tế của ngôn ngữ này.

Em xin trân trọng cảm ơn thầy và kính chúc thầy sức khỏe, thành công trong công tác giảng dạy và nghiên cứu.

LỜI NÓI ĐẦU

Trong bối cảnh công nghệ thông tin ngày càng phát triển, việc xây dựng các ứng dụng giao diện người dùng (GUI) để hỗ trợ thao tác trực quan, tiện lợi với hệ thống tệp là một yêu cầu phổ biến và thực tiễn. Với mục tiêu rèn luyện kỹ năng lập trình Python kết hợp với thư viện giao diện đồ họa tkinter, đề tài "*Tạo ứng dụng GUI cho phép chọn thư mục, liệt kê file theo từng loại và cho phép mở file*" được lựa chọn nhằm giúp sinh viên vận dụng kiến thức đã học vào một bài toán cụ thể.

Ứng dụng được xây dựng với các chức năng chính như: cho phép người dùng chọn thư mục trên máy, tự động quét và liệt kê các tệp theo định dạng (.txt, .py, .jpg) trong một bảng Treeview, đồng thời cho phép mở tệp bằng chương trình mặc định của hệ thống. Trong quá trình phát triển, chương trình tận dụng thư viện os để xử lý đường dẫn và tệp, xử lý ngoại lệ khi không tìm thấy thư mục, và sử dụng tkinter.ttk để xây dựng giao diện thân thiện với người dùng.

Việc hoàn thành đề tài không chỉ giúp người thực hiện củng cố kiến thức về Python và lập trình hướng sự kiện với GUI, mà còn nâng cao tư duy thiết kế phần mềm ứng dụng thực tế.

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1 Mô tả đề tài

Đề tài yêu cầu phát triển một ứng dụng giao diện đồ họa (GUI) sử dụng ngôn ngữ lập trình Python và thư viện Tkinter trong môi trường Visual Studio Code. Ứng dụng cho phép người dùng chọn một thư mục, liệt kê các tệp tin có định dạng cụ thể (bao gồm .txt, .py, .jpg) trong một Treeview và hỗ trợ mở tệp tin bằng chương trình mặc định của hệ thống. Các thành phần chính bao gồm:

- Đầu vào: Một nút “Chọn thư mục” để người dùng chọn đường dẫn thư mục thông qua hộp thoại.

- Đầu ra: Một Treeview (hoặc Listbox) hiển thị danh sách các tệp tin với thông tin như tên tệp, loại tệp và đường dẫn; cùng với cơ chế mở tệp khi nhấp đúp chuột.

- Tính năng chính:

- Sử dụng module os để quét thư mục và lọc các tệp theo định dạng (.txt, .py, .jpg).
- Xử lý lỗi khi đường dẫn thư mục không hợp lệ (ví dụ: thư mục không tồn tại).
- Xây dựng giao diện người dùng với tkinter.ttk.Treeview để hiển thị danh sách tệp.
- Cho phép mở tệp bằng chương trình mặc định của hệ thống thông qua os.startfile.

- Kết quả mong đợi:

- Khi chọn một thư mục chứa ba tệp (ví dụ: file1.txt, script.py, image.jpg), Treeview hiển thị ba dòng tương ứng.

- Nhấp đúp vào tệp .txt sẽ mở tệp bằng Notepad (hoặc chương trình mặc định của hệ thống).

1.2 Thách thức của đề tài

Đề tài đặt ra một số thách thức kỹ thuật và logic như sau:

- Quản lý giao diện GUI: Sử dụng Tkinter để tạo giao diện trực quan, bao gồm bố trí các thành phần như nút, Entry và Treeview sao cho thân thiện với người dùng
- Xử lý lỗi: Cần bắt và xử lý các trường hợp ngoại lệ, chẳng hạn khi thư mục không tồn tại hoặc tệp không thể mở, để đảm bảo ứng dụng không bị lỗi nghiêm trọng.
- Quét và lọc tệp: Sử dụng module os để duyệt thư mục và lọc tệp theo định dạng yêu cầu, đòi hỏi hiểu biết về thao tác với hệ thống tệp.
- Tích hợp sự kiện: Kết nối sự kiện nhấp đúp chuột trên Treeview với hàm mở tệp, yêu cầu xử lý chính xác đường dẫn tệp được chọn.
- Tương thích hệ thống: Đảm bảo tính năng os.startfile hoạt động ổn định trên các hệ điều hành (chủ yếu là Windows) và xử lý lỗi nếu tệp không mở được

1.3 Kiến thức vận dụng

Để hoàn thành đề tài, cần vận dụng các kiến thức sau:

- Ngôn ngữ lập trình Python: Sử dụng cú pháp Python cơ bản, thao tác với chuỗi, danh sách và xử lý ngoại lệ (try-except).
- Thư viện Tkinter: Kiến thức về tạo giao diện GUI, sử dụng các widget như Frame, Entry, Button và Treeview từ tkinter.ttk.
- Module os: Hiểu cách sử dụng os.listdir(), os.path.join() và os.startfile() để tương tác với hệ thống tệp.
- Xử lý sự kiện: Gắn sự kiện (event binding) cho Treeview để phản hồi hành động nhấp đúp chuột.
- Quản lý môi trường phát triển: Sử dụng Visual Studio Code để viết, chạy và gỡ lỗi mã Python, bao gồm cấu hình môi trường ảo và tiện ích mở rộng như Pylance.
- Kỹ năng gỡ lỗi: Kiểm tra và xử lý lỗi trong quá trình phát triển, đảm bảo ứng dụng hoạt động đúng như yêu cầu.

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

2.1 Danh sách list trong python

Danh sách (List) là một trong những kiểu dữ liệu có cấu trúc quan trọng và được sử dụng phổ biến trong Python. List cho phép lưu trữ tập hợp các giá trị (gọi là phần tử), có thể thuộc bất kỳ kiểu dữ liệu nào, và có thể thay đổi (mutable). Mỗi phần tử trong list được đánh chỉ số (index) bắt đầu từ 0, giúp việc truy cập và xử lý dữ liệu trở nên linh hoạt.

VD Cú pháp khai báo

```
my_list = [1, 2, 3, 4, 5]
```

Trong chương trình, list được sử dụng một cách gián tiếp nhưng hiệu quả thông qua các hàm trả về danh sách hoặc làm việc với danh sách:

a , `os.listdir(path)`: Trả về một list chứa tên các tệp và thư mục trong thư mục được chỉ định. Danh sách này được duyệt bằng vòng lặp for để lọc và xử lý các tệp theo định dạng mong muốn

```
files = os.listdir(folder_path)
for file in files:
```

b, `tree.get_children()`: Trả về danh sách các phần tử đang hiển thị trong bảng Treeview. Danh sách này được sử dụng để xóa các phần tử cũ trước khi hiển thị danh sách tệp mới

```
for item in tree.get_children():
    tree.delete(item)
```

c, `allowed_extensions = ('.txt', '.py', '.jpg')`: Mặc dù được khai báo dưới dạng tuple, danh sách định dạng tệp này đóng vai trò tương tự một list, cho phép kiểm tra phần mở rộng của tệp khi lọc.

```
if file_name.lower().endswith(allowed_extensions):
```

2.2 Widget Treeview

Treeview là một widget đặc biệt thuộc thư viện `tkinter.ttk` (themed Tkinter), dùng để hiển thị dữ liệu theo dạng bảng có nhiều cột, hỗ trợ lựa chọn từng hàng, sắp xếp, tổ chức dữ liệu phân cấp, và gán các sự kiện tương tác như click hoặc double-click.

Trong chương trình, Treeview được sử dụng để hiển thị danh sách các tệp hợp lệ từ thư mục do người dùng chọn. Mỗi hàng (record) trong Treeview đại diện cho một tệp, bao gồm các thông tin: Tệp, link, đường dẫn ví dụ Đặt tiêu đề cho từng cột và cấu hình chiều rộng:

```
tree.heading("Tên file", text="Tên file")
tree.heading("Loại", text="Loại")
tree.heading("Link", text="Link")
tree.column("Tên file", width=200)
tree.column("Loại", width=100)
tree.column("Link", width=250)
```

Lợi ích của Treeview là giúp người dùng có thể duyệt dữ liệu theo cấu trúc rõ ràng, dễ quan sát, hỗ trợ thao tác lựa chọn và tương tác linh hoạt — đặc biệt phù hợp với các chương trình quản lý tệp hoặc cơ sở dữ liệu dạng bảng.

2.3 Thư viện Tkinter

Tkinter là một thư viện trong ngôn ngữ lập trình Python được sử dụng để tạo giao diện đồ họa người dùng (GUI). "Tkinter" là viết tắt của "Tk interface", một toolkit đồ họa cung cấp các công cụ để phát triển giao diện người dùng.

Tkinter là một phần của thư viện tiêu chuẩn của Python và đã được tích hợp sẵn trong hầu hết các cài đặt Python. Điều này giúp cho Tkinter trở thành một lựa chọn phổ biến cho việc phát triển ứng dụng với giao diện đồ họa đơn giản trong Python.

Một số đặc điểm của Tkinter bao gồm khả năng tạo các thành phần giao diện như cửa sổ, nút, ô văn bản, và các widget khác để tương tác với người dùng. Tkinter cung cấp cả các sự kiện và phương thức để xử lý tương tác người dùng và thay đổi trạng thái của ứng dụng

VD tạo cửa sổ chính

```
root = tk.Tk()
root.title("Trình quản lý thư mục GUI")
root.geometry("600x400")
```

2.4 Module os

Module os là một thư viện tiêu chuẩn trong Python, cung cấp các phương thức giúp lập trình viên tương tác với hệ điều hành, đặc biệt là hệ thống tệp (file system). Trong ứng dụng, os đóng vai trò quan trọng trong việc truy cập thư mục, kiểm tra và xử lý các tệp tin.

a, os.listdir(path): Trả về danh sách các tệp và thư mục nằm trong đường dẫn path. Đây là cơ sở để chương trình duyệt và liệt kê các tệp có trong thư mục được chọn.

b, os.path.join(path, file): Dùng để kết hợp đường dẫn thư mục và tên tệp, đảm bảo tạo ra đường dẫn đầy đủ và đúng định dạng (phù hợp với hệ điều hành).

c, os.path.isfile(path): Kiểm tra xem một đường dẫn cụ thể có phải là tệp tin hay không. Điều này rất quan trọng để phân biệt tệp với thư mục, chỉ hiển thị các tệp trong danh sách.

d, os.startfile(path): Mở tệp bằng ứng dụng mặc định của hệ điều hành. Lưu ý: hàm này hoạt động chủ yếu trên hệ điều hành Windows

Ví dụ minh họa

```
import os

folder_path = "C:/Users/Documents"      # Đường dẫn đến thư mục
files = os.listdir(folder_path)          # Lấy danh sách tệp/thư mục

for file in files:
    full_path = os.path.join(folder_path, file) # Kết hợp thành đường dẫn đầy đủ
    if os.path.isfile(full_path):              # Kiểm tra nếu là tệp
        print(full_path)                      # In đường dẫn tệp ra màn hình
```

Chương 3 Thiết kế và xây dựng chương trình

3.1 Sơ đồ khối hệ thống

3.1.1 Mô tả các module chính

Ứng dụng được chia thành ba module chính, mỗi module đảm nhiệm một chức năng cụ thể:

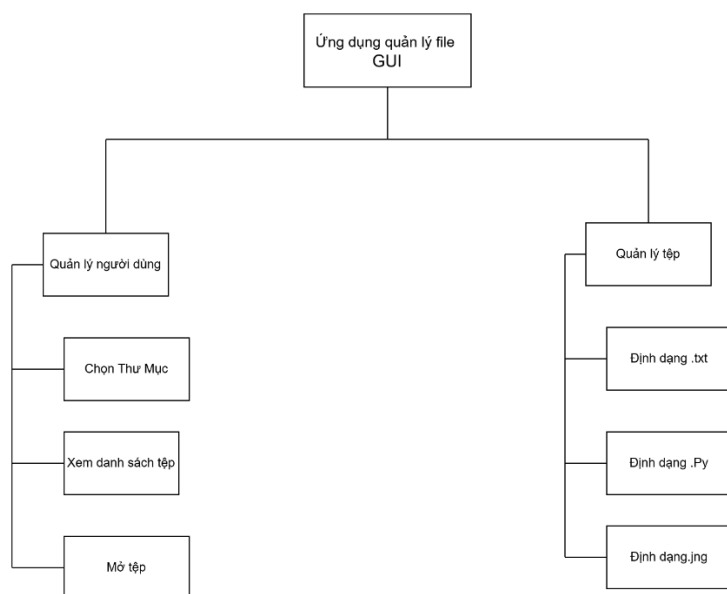
- Module giao diện người dùng (GUI): Sử dụng Tkinter để tạo cửa sổ chính, chứa các widget như Entry (ô nhập đường dẫn), Button (nút "Chọn thư mục"), và Treeview (bảng hiển thị danh sách tệp). Module này chịu trách nhiệm hiển thị giao diện và tương tác với người dùng.

- Module quản lý tệp: Sử dụng module os để quét thư mục, lọc tệp theo định dạng (.txt, .py, .jpg), và cung cấp đường dẫn đầy đủ của tệp. Module này xử lý các thao tác liên quan đến hệ thống tệp.

- Module xử lý hoạt động : Quản lý các hoạt động như nhấp nút "Chọn thư mục" hoặc nhấp đúp vào Treeview để mở tệp. Module này đảm bảo ứng dụng phản hồi chính xác với hành động của người dùng và xử lý lỗi (ví dụ: thư mục không tồn tại).

3.1.2 Biểu đồ phân cấp chức năng

Biểu đồ phân cấp chức năng mô tả các chức năng chính của chương trình, được trình bày như sau:



Hình 3.1 Sơ đồ phân cấp chức năng

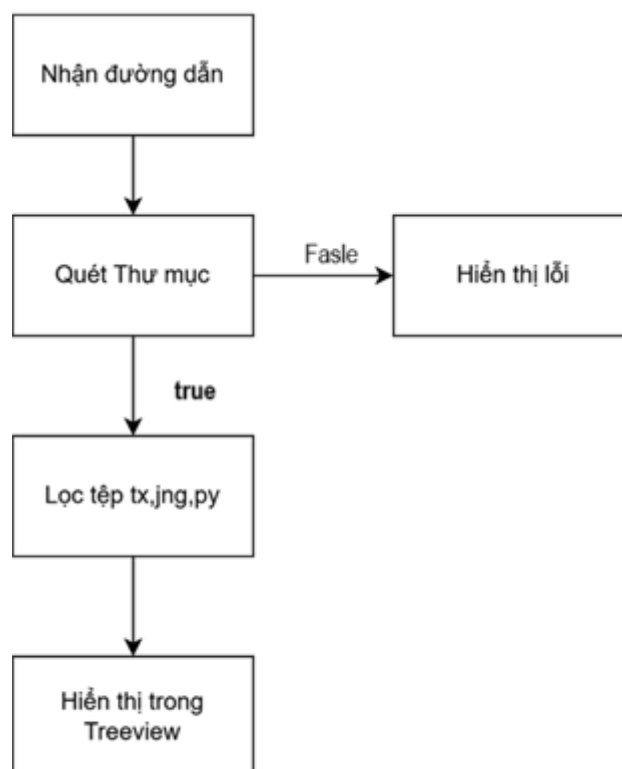
- Chọn thư mục: Cho phép người dùng chọn thư mục thông qua hộp thoại `filedialog.askdirectory`.
- Hiển thị danh sách tệp: Quét thư mục, lọc tệp theo định dạng, và hiển thị trong Treeview.
- Mở tệp: Mở tệp được chọn bằng chương trình mặc định thông qua `os.startfile`.

3.2 Sơ đồ khối các thuật toán chính

Các thuật toán chính trong chương trình bao gồm:

3.2.1 Thuật toán chọn thư mục:

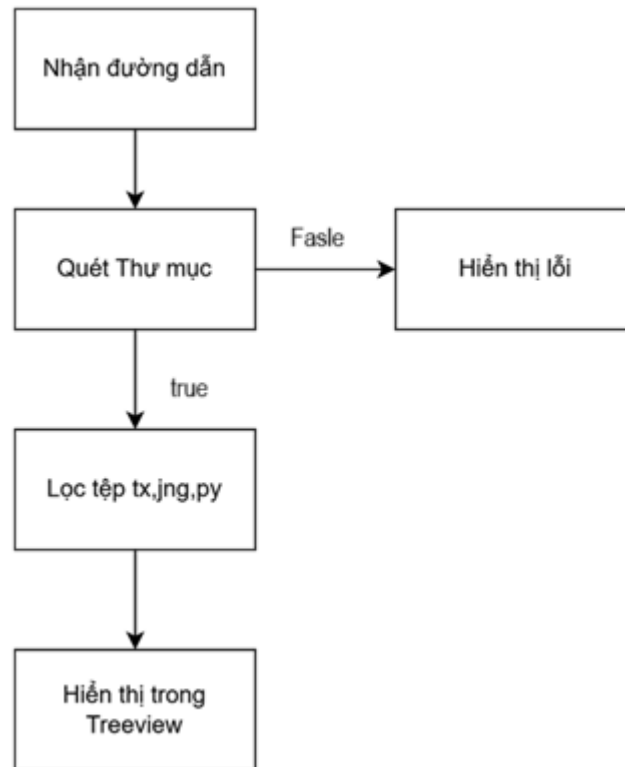
- Đầu vào: Hành động nhấp nút "Chọn thư mục".
- Đầu ra: Đường dẫn thư mục được chọn hoặc thông báo lỗi nếu không chọn thư mục.
- Chức năng: Mở hộp thoại chọn thư mục, lưu đường dẫn vào Entry, và gọi hàm hiển thị danh sách tệp.



Hình 3.2 Sơ đồ khối chọn thư mục

3.2.2 Thuật toán hiển thị danh sách tệp:

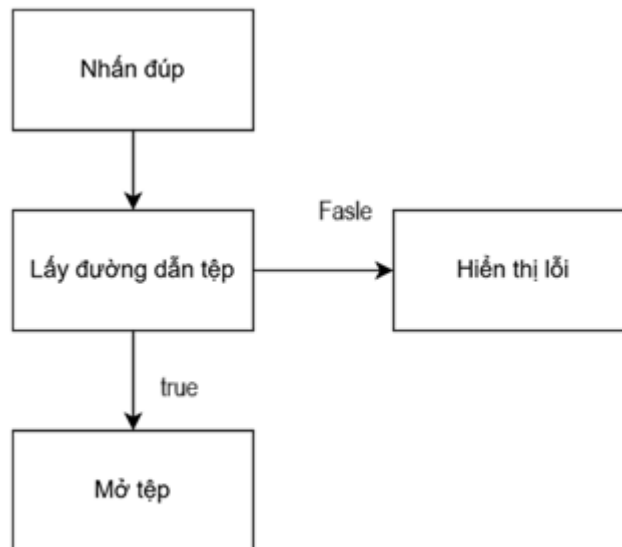
- Đầu vào: Đường dẫn thư mục.
- Đầu ra: Danh sách tệp hiển thị trong Treeview hoặc thông báo lỗi nếu thư mục không hợp lệ.
- Chức năng: Quét thư mục bằng `os.listdir`, lọc tệp theo định dạng (.txt, .py, .jpg), và thêm vào Treeview.
- Sơ đồ khối:



Hình 3.3 Sơ đồ khối hiển thị danh sách tệp

3.2.3 Thuật toán mở tệp:

- Đầu vào: Sự kiện nhấp đúp trên Treeview.
- Đầu ra: Tệp được mở bằng chương trình mặc định hoặc thông báo lỗi nếu không mở được.
- Chức năng: Lấy đường dẫn tệp từ Treeview và sử dụng `os.startfile` để mở.



Hình 3.4 Sơ đồ khối mở tệp

3.3 Cấu trúc dữ liệu

Dữ liệu trong chương trình được tổ chức để hiển thị trong Treeview. Bảng dữ liệu chính được mô tả như sau:

- Bảng danh sách tệp:

- Các trường thông tin:

- Tên tệp (chuỗi): Tên của tệp, ví dụ: example.txt.
- Loại (chuỗi): Phần mở rộng của tệp, ví dụ: .txt, .py, .jpg.
- Đường dẫn (chuỗi): Đường dẫn đầy đủ của tệp

- Lưu trữ: Dữ liệu được lưu tạm thời trong Treeview dưới dạng các hàng (row), mỗi hàng là một tuple chứa ba giá trị tương ứng với các trường dữ liệu

Ví dụ

Tên file	Loại	Đường dẫn
bai1.py	.py	C:/Users/ACER/Documents/Zalo Received Files/
Bai2.py	.py	C:/Users/ACER/Documents/Zalo Received Files/
Bai3.py	.py	C:/Users/ACER/Documents/Zalo Received Files/
espresso.jpg	.jpg	C:/Users/ACER/Documents/Zalo Received Files/
hop-3-banh-pha-co-trang-ram.jpg	.jpg	C:/Users/ACER/Documents/Zalo Received Files/
latte.jpg	.jpg	C:/Users/ACER/Documents/Zalo Received Files/
New Tài liệu văn bản.txt	.txt	C:/Users/ACER/Documents/Zalo Received Files/
tra-thanh-dao.jpg	.jpg	C:/Users/ACER/Documents/Zalo Received Files/
tra-xanh-dau-do.jpg	.jpg	C:/Users/ACER/Documents/Zalo Received Files/

Hình 3.5 Các trường thông tin

3.4 Chương trình

Chương trình trình bao gồm các hàm sau :

3.4.1 Hàm select_folder()

```
def select_folder(): # Hàm xử lý sự kiện khi người dùng nhấn nút "Chọn thư mục"
    folder_path = filedialog.askdirectory() # Hiển thị hộp thoại chọn thư mục và lưu
    đường dẫn đã chọn
    if folder_path: # Kiểm tra nếu người dùng đã chọn một thư mục (không để trống)
        folder_entry.delete(0, tk.END) # Xóa nội dung cũ trong ô nhập đường dẫn
        folder_entry.insert(0, folder_path) # Hiển thị đường dẫn vừa chọn vào ô nhập
        show_files(folder_path) # Gọi hàm show_files để hiển thị danh sách file trong
        thư mục đó
    else:
        messagebox.showwarning("Cảnh báo", "Vui lòng chọn một thư mục!")
# Thông báo nếu người dùng chưa chọn thư mục
```

Chức năng : có nhiệm vụ mở hộp thoại để người dùng chọn một thư mục từ máy tính. Sau khi người dùng chọn xong, hàm sẽ cập nhật đường dẫn thư mục được chọn vào ô nhập liệu (Entry) trên giao diện và hiển thị danh sách các tệp trong thư mục đó thông qua hàm show_files(). Nếu người dùng không chọn thư mục nào, hệ thống sẽ hiển thị thông báo cảnh báo yêu cầu người dùng chọn thư mục.

Đầu vào : Hàm không nhận tham số truyền vào trực tiếp. Dữ liệu đầu vào được cung cấp gián tiếp thông qua hành động người dùng chọn thư mục từ hộp thoại filedialog.askdirectory().

Đầu ra: Không có giá trị trả về.

Kết quả được thể hiện thông qua:Việc cập nhật nội dung trong widget Entry (folder_entry) với đường dẫn thư mục được chọn.

Gọi hàm show_files(folder_path) để xử lý và hiển thị các tệp trong thư mục.

- Hiển thị hộp thoại cảnh báo nếu người dùng không chọn thư mục.

3.4.2, hàm def show_files(folder_path):

```
def show_files(folder_path): # Hàm hiển thị danh sách các file trong thư mục được chọn
    try:
```

```

# Xóa dữ liệu cũ trong Treeview trước khi thêm dữ liệu mới
for item in tree.get_children():
    tree.delete(item)

# Định nghĩa các phần mở rộng file được phép hiển thị
allowed_extensions = ('.txt', '.py', '.jpg')
# Duyệt qua tất cả các tên file và thư mục trong folder_path
for file_name in os.listdir(folder_path):
    # Kiểm tra file có phần mở rộng nằm trong allowed_extensions không
    if file_name.lower().endswith(allowed_extensions):
        file_path = os.path.join(folder_path, file_name) # Tạo đường dẫn đầy đủ đến
file
        if os.path.isfile(file_path): # Kiểm tra chắc chắn đây là file, không phải thư mục
            ext = os.path.splitext(file_name)[1].lower() # Lấy phần mở rộng file và
chuyển thành chữ thường
            # Thêm dòng mới vào Treeview với các cột: tên file, phần mở rộng, đường
dẫn
            tree.insert("", tk.END, values=(file_name, ext, file_path))
        except FileNotFoundError: # Xử lý trường hợp thư mục không tồn tại hoặc không tìm
thấy
            messagebox.showerror("Lỗi", "Không tìm thấy thư mục!")
        except Exception as e: # Bắt các lỗi khác và hiển thị thông báo lỗi
            messagebox.showerror("Lỗi", f"Đã xảy ra lỗi: {str(e)}")

```

- Chức năng: Quét thư mục, lọc tệp theo định dạng (.txt, .py, .jpg), và hiển thị danh sách các tệp này trong bảng Treeview.
- Đầu vào: Đường dẫn thư mục (folder_path) – kiểu chuỗi. Đây là thư mục cần quét và xử lý.
- Đầu ra: Danh sách các tệp hợp lệ được hiển thị trong bảng Treeview, với các thông tin gồm: Tên file, loại, đường dẫn

3.4.3 Hàm def open_file(event):

Chức năng: Mở tệp được chọn trong Treeview bằng chương trình mặc định.

Đầu vào: Sự kiện nhấp đúp (event).

Đầu ra: Tệp được mở hoặc thông báo lỗi.

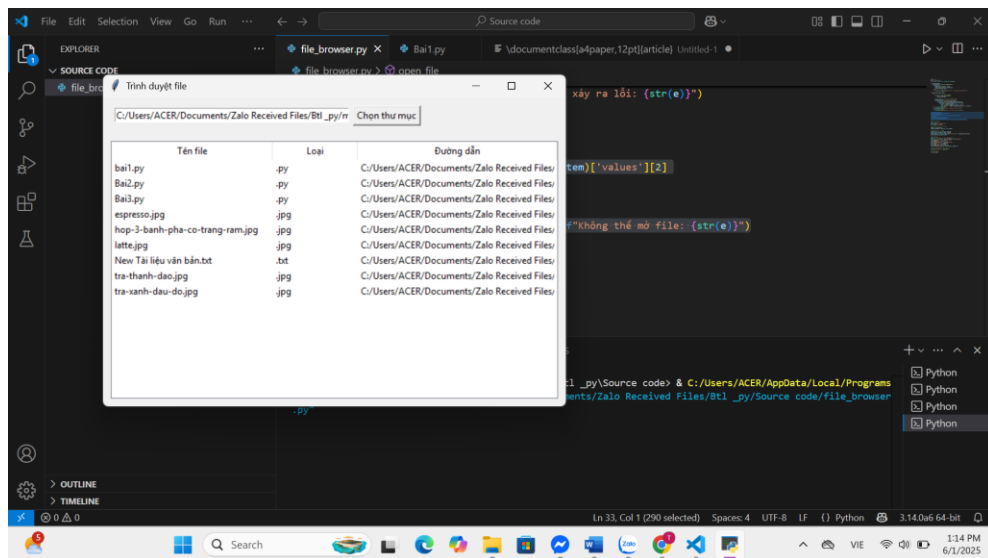
```
def open_file(event): # Hàm xử lý sự kiện khi người dùng double-click  
vào một hàng trong Treeview  
    selected_item = tree.selection() # Lấy danh sách các mục (items) được  
chọn trong Treeview  
    if selected_item: # Nếu có ít nhất một mục được chọn  
        # Lấy đường dẫn file từ cột thứ 3 (index 2) của mục được chọn  
        file_path = tree.item(selected_item)['values'][2]  
        try:  
            os.startfile(file_path) # Mở file bằng ứng dụng mặc định của hệ  
điều hành (chỉ trên Windows)  
        except Exception as e: # Nếu xảy ra lỗi khi mở file (ví dụ file bị xóa  
hoặc không có quyền truy cập)  
            messagebox.showerror("Lỗi", f"Không thể mở file: {str(e)}")  
# Hiển thị thông báo lỗi cho người dùng
```

Chương 4 Thực nghiệm và kết luận

4.1. Thực nghiệm

Chạy và ghi lại kết quả các bài test các tính năng của sản phẩm, đánh giá chất lượng

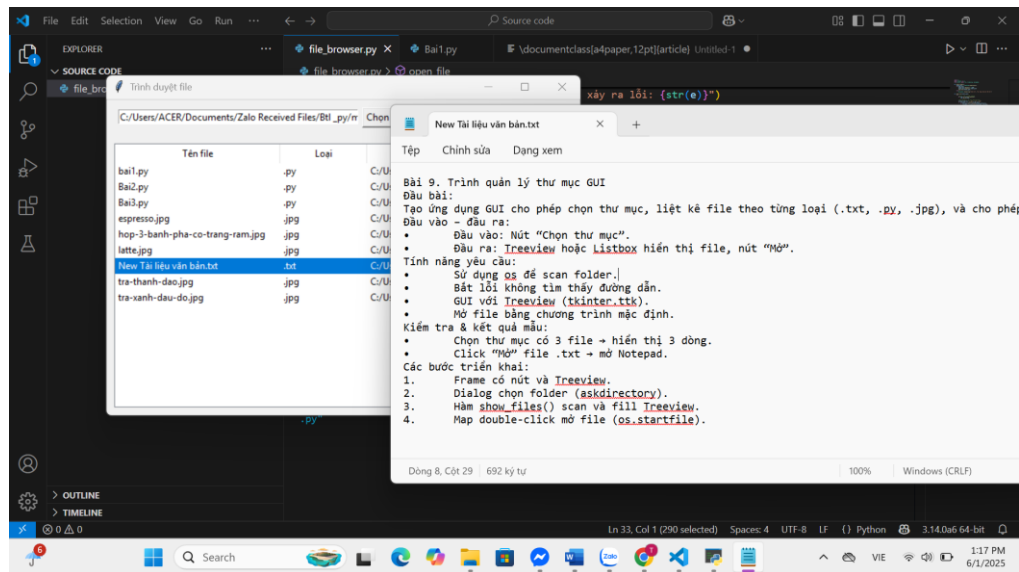
- Giao diện hoạt động đúng yêu cầu: chọn thư mục, lọc file đúng định dạng cho phép và hiển thị trong bảng.
- Hiển thị đầy đủ thông tin theo 3 cột: tên, định dạng và đường dẫn.
- Có thể mở file trực tiếp bằng cách nhấp đúp chuột (như đã cài đặt với hàm `open_file(event)`).



Hình 4.1 Giao diện thực nghiệm

Kiểm tra & kết quả mẫu:

- Chọn thư mục có 3 file → hiển thị 3 dòng.
- Click “Mở” file .txt → mở Notepad.



Hình 4.2 kết quả thực nghiệm

4.2 Kết Luận

4.2.1 Sản phẩm đã làm được :

- Chọn thư mục: Người dùng có thể chọn một thư mục thông qua nút “Chọn thư mục”, sử dụng hộp thoại `filedialog.askdirectory`. Đường dẫn được hiển thị trong ô nhập (Entry) và được kiểm tra tính hợp lệ.

- Hiển thị danh sách tệp: Ứng dụng quét thư mục bằng module `os`, lọc các tệp có định dạng `.txt`, `.py`, và `.jpg`, sau đó hiển thị thông tin (tên tệp, loại, đường dẫn) trong Treeview với giao diện trực quan.

- Mở tệp: Người dùng có thể nhấp đúp vào một hàng trong Treeview để mở tệp bằng chương trình mặc định của hệ thống (ví dụ: tệp `.txt` mở bằng Notepad).

- Xử lý lỗi: Ứng dụng bắt và hiển thị các lỗi như thư mục không tồn tại (`FileNotFoundError`) hoặc không chọn thư mục, thông qua các thông báo lỗi thân thiện với người dùng (`messagebox`).

Kết quả kiểm tra cho thấy ứng dụng hoạt động ổn định: khi chọn một thư mục chứa ba tệp (ví dụ: `file1.txt`, `script.py`, `image.jpg`), Treeview hiển thị đúng ba dòng, và nhấp đúp vào tệp `.txt` sẽ mở bằng chương trình mặc định.

4.2.2 Học được từ Sản Phẩm

- Lập trình Python: Nắm vững cách sử dụng các cấu trúc dữ liệu như danh sách (list) để lọc tệp và xử lý ngoại lệ (try-except) để đảm bảo ứng dụng không bị lỗi.
- Thư viện Tkinter: Học cách xây dựng giao diện đồ họa với các widget như Frame, Entry, Button, và Treeview, cũng như gắn sự kiện (bind) để xử lý hành động người dùng
- Module os: Hiểu cách tương tác với hệ thống tệp thông qua các hàm như : `os.listdir`, `os.path.join`, `os.path.isfile`, và `os.startfile`
- Môi trường Visual Studio Code: Thành thạo cách cấu hình môi trường phát triển, sử dụng tiện ích mở rộng (như Pylance), chạy mã, và gỡ lỗi với breakpoint.
- Kỹ năng thiết kế phần mềm: Học cách phân chia chương trình thành các module (giao diện, quản lý tệp, xử lý sự kiện) và thiết kế sơ đồ khối để tổ chức chức năng.

4.2.3 Cải tiến Sản phẩm

- Lọc tệp linh hoạt hơn: Thêm tùy chọn cho phép người dùng chọn loại tệp để hiển thị (ví dụ: chỉ hiển thị `.txt` hoặc `.jpg`) thông qua menu thả xuống hoặc checkbox.
- Hiển thị chi tiết tệp: Bổ sung các cột trong Treeview để hiển thị thêm thông tin như kích thước tệp, ngày sửa đổi, hoặc quyền truy cập.
- Tương thích đa nền tảng: Thay `os.startfile` (chỉ hỗ trợ Windows) bằng `subprocess.run` để đảm bảo khả năng mở tệp trên macOS và Linux.
- Giao diện nâng cao: Cải thiện giao diện với các nút bổ sung (như “Xóa”, “Sao chép tệp”) và hỗ trợ kéo-thả thư mục trực tiếp vào ô nhập.
- Lưu lịch sử thư mục: Thêm tính năng lưu các thư mục đã chọn trước đó để người dùng dễ dàng quay lại.

Tài liệu tham khảo

- 1.Lutz, M. (2013). *Learning Python* (5th ed.). O'Reilly Media.
- 2.Sweigart, A. (2015). *Automate the Boring Stuff with Python: Practical Programming for Total Beginners*. No Starch Press.
<https://automatetheboringstuff.com>
- 3.Python Software Foundation. *The Python Tutorial*. Truy cập tại:
<https://docs.python.org/3/tutorial/>
- 4.Beazley, D., & Jones, B. (2013). *Python Cookbook* (3rd ed.). O'Reilly Media.
- 5.Martelli, A., Ravenscroft, A., & Ascher, D. (2005). *Python in a Nutshell* (2nd ed.). O'Reilly Media.