

**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: Tạ Phạm Đình Hòa

Lớp: K58KTP.01

Giáo viên GIẢNG DẠY: TS Nguyễn Văn Huy

Link GitHub: <https://github.com/DinhHoa107/BTL-Python-De-9.git>



**Thái Nguyên-2025**

**TRƯỜNG ĐHKTCN**  
**KHOA ĐIỆN TỬ**

**CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**  
**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: Tạ Phạm Đình Hòa*

*Lớp: K58 KTP.01      Ngành: Kỹ Thuật Máy Tính*

*Giáo viên hướng dẫn: TS.Nguyễn Văn Huy*

*Ngày giao đề Ngày hoàn thành*

*Tên đề tài : Trình Quản Lý Thư Mục GUI*

*Yêu cầu : Sử dụng os để scam 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

GIÁO VIÊN HƯỚNG DẪN .....	3
<b>CHƯƠNG I GIỚI THIỆU ĐỀ TÀI .....</b>	<b>5</b>
<b>CHƯƠNG 2 CƠ SỞ LÝ THUYẾT .....</b>	<b>7</b>
2.1. Thư viện Python được sử dụng .....	7
2.2. Cấu trúc dữ liệu .....	7
2.3. Kỹ thuật lập trình .....	8
2.4. Giao diện người dùng (GUI) .....	9
2.5. Quản lý hệ thống file .....	9
<b>CHƯƠNG 3 THIẾT KẾ XÂY DỰNG CHƯƠNG TRÌNH .....</b>	<b>10</b>
3.1. Sơ đồ khối hệ thống .....	10
3.2. Sơ đồ khối các thuật toán chính .....	11
3.3. Cấu trúc dữ liệu .....	12
3.4. Chương trình .....	13
<b>CHƯƠNG 4 CHƯƠNG TRÌNH .....</b>	<b>15</b>
4.1. Chương trình đã làm được cái gì .....	15
4.2. Kết luận .....	15

# CHƯƠNG I GIỚI THIỆU ĐỀ TÀI

Trong thời đại công nghệ thông tin hiện nay, việc quản lý và xử lý dữ liệu tập trên máy tính là một nhu cầu phổ biến và thiết yếu. Đặc biệt, các ứng dụng có giao diện đồ họa người dùng (GUI) giúp người dùng tương tác trực quan và thuận tiện hơn so với giao diện dòng lệnh. Xuất phát từ yêu cầu thực tế đó, bài tập này đặt ra mục tiêu xây dựng một **trình quản lý thư mục dạng GUI** đơn giản, giúp người dùng chọn một thư mục bất kỳ, liệt kê các tệp theo từng loại định dạng (.txt, .py, .jpg, v.v.) và cho phép mở trực tiếp các tệp bằng phần mềm mặc định của hệ điều hành.

Chương trình được thiết kế với các tính năng chính sau:

- Cho phép người dùng **chọn thư mục** bằng hộp thoại đồ họa (askdirectory).

**Duyệt thư mục và liệt kê danh sách tệp** bằng cách sử dụng thư viện os.

- Hiện thị danh sách các tệp trong bảng Treeview thuộc thư viện tkinter.ttk, có thể hiển thị tên file và phân mở rộng (loại file).

- Hỗ trợ **mở file** được chọn bằng phần mềm mặc định của hệ điều hành thông qua thao tác nhấp đúp hoặc nút "Mở".

- **Xử lý lỗi** khi thư mục không tồn tại hoặc không thể truy cập, đảm bảo chương trình hoạt động ổn định và thân thiện với người dùng.

## Thách thức khi thực hiện bài toán

Một số thách thức khi xây dựng ứng dụng bao gồm:

- Phải xử lý đường dẫn và tệp một cách an toàn, đặc biệt là khi thư mục có chứa tệp bị lỗi, quyền hạn chế hoặc định dạng không chuẩn.

- Đảm bảo tương thích với nhiều hệ điều hành khác nhau (Windows, macOS, Linux) khi mở tệp từ chương trình mặc định.

- Giao diện cần dễ sử dụng nhưng vẫn đầy đủ tính năng cơ bản, đồng thời xử lý sự kiện như nhấn nút hoặc nhấp đúp chuột một cách chính xác.

## Kiến thức vận dụng

Để hoàn thành bài tập này, sinh viên cần vận dụng các kiến thức đã học bao gồm:

**- Lập trình giao diện đồ họa với Tkinter**, đặc biệt là cách sử dụng

Frame, Button, Treeview, và các hộp thoại chọn thư mục.

**- Xử lý tệp và thư mục với thư viện os**: lấy danh sách file, tách phần mở rộng, xử lý đường dẫn.

**- Xử lý sự kiện GUI**: xử lý nhấn nút, chọn file trong Treeview, và nhấn đúp chuột.

**- Lập trình hướng đối tượng** để tổ chức mã nguồn dễ hiểu, dễ mở rộng và bảo trì.

## CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

Chương này trình bày các khái niệm và kỹ thuật chuyên môn được sử dụng trong chương trình "Trình Quản Lý Thư Mục" được viết bằng Python, dựa trên mã nguồn trong tài liệu Python\_Hoa.py. Các nội dung bao gồm các thư viện, cấu trúc dữ liệu, và các kỹ thuật lập trình liên quan.

### 2.1. Thư viện Python được sử dụng

#### 2.1.1. Thư viện tkinter

- **Mô tả:** tkinter là thư viện chuẩn của Python dùng để xây dựng giao diện đồ họa người dùng (GUI). Trong chương trình, tkinter được sử dụng để tạo cửa sổ chính, các nút bấm, và bảng hiển thị danh sách file (Treeview).
- **Ứng dụng trong chương trình:**
  - Tạo cửa sổ chính (tk.Tk()).
  - Tạo khung (Frame), nút bấm (Button), và bảng hiển thị (tk.Treeview).
  - Gắn sự kiện (bind) để xử lý hành động double-click trên Treeview.
  - Sử dụng filedialog và messagebox từ tkinter để mở hộp thoại chọn thư mục và hiển thị thông báo lỗi/cảnh báo.

#### 2.1.2. Thư viện os

- **Mô tả:** Thư viện os cung cấp các hàm để tương tác với hệ điều hành, như truy cập hệ thống file, kiểm tra thư mục, và mở file.
- **Ứng dụng trong chương trình:**
  - Hàm os.path.isdir(): Kiểm tra tính hợp lệ của thư mục.
  - Hàm os.listdir(): Liệt kê các file trong thư mục.
  - Hàm os.path.join(): Tạo đường dẫn đầy đủ của file.
  - Hàm os.startfile(): Mở file bằng chương trình mặc định của hệ điều hành.

### 2.2. Cấu trúc dữ liệu

#### 2.2.1. Danh sách (list)

- **Mô tả:** Danh sách trong Python là một cấu trúc dữ liệu động, cho phép lưu trữ một tập hợp các phần tử có thể thuộc nhiều kiểu dữ liệu khác nhau.
- **Ứng dụng trong chương trình:**
  - Danh sách các file trong thư mục được lấy từ os.listdir(self.current\_folder). Danh sách này được duyệt để lọc các file có phần mở rộng hợp lệ (.txt, .py, .jpg).
  - Danh sách các phần mở rộng hợp lệ valid\_extensions = (".txt", ".py", ".jpg") được sử dụng để kiểm tra loại file.

### 2.2.2. Từ điển (dict)

\* **Mô tả:** Từ điển là cấu trúc dữ liệu lưu trữ các cặp khóa-giá trị, cho phép truy xuất nhanh thông tin dựa trên khóa.

\* **Ứng dụng trong chương trình:**

Từ điển `file_type` ánh xạ phần mở rộng file với loại file:

```
python
Thu gọnBọc lạiChạy
Sao chép
file_type = { ".txt": "Text", ".py": "Python", ".jpg": "Image" }
```

Từ điển này được sử dụng để xác định loại file (Text, Python, Image) dựa trên phần mở rộng, hiển thị trong cột "Loại File" của Treeview.

## 2.3. Kỹ thuật lập trình

### 2.3.1. Lập trình hướng đối tượng

\* **Mô tả:** Chương trình sử dụng lập trình hướng đối tượng (OOP) thông qua việc định nghĩa lớp `FileManager`. Lớp này đóng gói các thuộc tính (như `self.current_folder`) và phương thức (như `select_folder`, `show_files`, `open_file`).

\* **Ứng dụng trong chương trình:**

- Lớp `FileManager` quản lý toàn bộ logic của ứng dụng, bao gồm giao diện và xử lý file.

- Các phương thức như `select_folder`, `show_files`, và `open_file` được gọi để thực hiện các chức năng cụ thể.

### 2.3.2. Xử lý ngoại lệ

\* **Mô tả:** Xử lý ngoại lệ (try-except) được sử dụng để quản lý lỗi có thể xảy ra trong quá trình thực thi, đảm bảo chương trình không bị dừng đột ngột.

\* **Ứng dụng trong chương trình:**

Trong phương thức `show_files`, khối try-except xử lý lỗi khi liệt kê file trong thư mục:

```
python
Thu gọnBọc lạiChạy
Sao chép
```



```
try: # Lọc và hiển thị file except Exception as e:  
    messagebox.showerror("Lỗi", f"Không thể liệt kê file: {e}")
```

- Trong phương thức `open_file`, khối `try-except` xử lý lỗi khi mở file, bao gồm `FileNotFoundError` và các lỗi khác.

### 2.3.3. Gắn kết sự kiện (Event Binding)

\* **Mô tả:** Sự kiện trong GUI được gắn với các hàm xử lý để phản hồi hành động của người dùng, như nhấp chuột hoặc double-click.

\* **Ứng dụng trong chương trình:**

- Sự kiện double-click trên Treeview được gắn với phương thức `open_file` thông qua `self.tree.bind("<Double-1>", self.open_file)`.

## 2.4. Giao diện người dùng (GUI)

\* **Mô tả:** Giao diện người dùng được xây dựng bằng các thành phần của tkinter, bao gồm nút bấm, bảng Treeview, và hộp thoại.

\* **Ứng dụng trong chương trình:**

- **Nút "Chọn Thư Mục":** Kích hoạt hộp thoại chọn thư mục (`filedialog.askdirectory`).

- **Treeview:** Hiển thị danh sách file với hai cột "Tên File" và "Loại File".

- **Nút "Mở File":** Mở file được chọn bằng chương trình mặc định.

- **Hộp thoại thông báo:** Sử dụng `messagebox` để hiển thị lỗi hoặc cảnh báo khi cần.

## 2.5. Quản lý hệ thống file

\* **Mô tả:** Chương trình sử dụng các hàm của thư viện `os` để tương tác với hệ thống file, bao gồm liệt kê, kiểm tra, và mở file.

\* **Ứng dụng trong chương trình:**

- Lọc file dựa trên phần mở rộng (.txt, .py, .jpg).

- Tạo đường dẫn file đầy đủ bằng `os.path.join`.

- Mở file bằng `os.startfile`.

## CHƯƠNG 3 THIẾT KẾ XÂY DỰNG CHƯƠNG TRÌNH

### 3.1. Sơ đồ khối hệ thống

#### 3.1.1. Mô tả các module chính

Chương trình được thiết kế theo mô hình hướng đối tượng, với các module chính được tích hợp trong lớp FileManager. Các module chính bao gồm:

##### **Module Giao diện người dùng (GUI):**

**1. Chức năng:** Cung cấp giao diện đồ họa để người dùng tương tác, bao gồm cửa sổ chính, nút bấm, bảng hiển thị file, và hộp thoại.

##### **2. Thành phần:**

1. Cửa sổ chính (tk.Tk): Nơi chứa toàn bộ giao diện.
2. Khung chính (tk.Frame): Chứa các thành phần giao diện như nút và bảng.
3. Nút "Chọn Thư Mục" (tk.Button): Kích hoạt hộp thoại chọn thư mục.
4. Bảng Treeview (ttk.Treeview): Hiển thị danh sách file với các cột "Tên File" và "Loại File".
5. Nút "Mở File" (tk.Button): Mở file được chọn.
6. Hộp thoại (filedialog, messagebox): Chọn thư mục và hiển thị thông báo lỗi/cảnh báo.

##### **Module Quản lý thư mục:**

**1. Chức năng:** Xử lý các thao tác liên quan đến hệ thống file, như chọn thư mục, liệt kê file, và mở file.

##### **2. Thành phần:**

1. Hàm select\_folder: Mở hộp thoại để chọn thư mục.
2. Hàm show\_files: Liệt kê các file hợp lệ trong thư mục.
3. Hàm open\_file: Mở file bằng chương trình mặc định của hệ điều hành.

##### **Module Xử lý sự kiện:**

**1. Chức năng:** Gắn kết và xử lý các sự kiện người dùng, như nhấp chuột vào nút hoặc double-click trên bảng Treeview.

##### **2. Thành phần:**

Sự kiện double-click trên Treeview được gắn với hàm open\_file.

#### 3.1.2. Biểu đồ phân cấp chức năng

Dưới đây là biểu đồ phân cấp chức năng, mô tả các chức năng chính của chương trình:

\* **Chức năng chính:** Quản lý thư mục

\* **Chọn thư mục:**

- Mở hộp thoại chọn thư mục.
- Lưu đường dẫn thư mục được chọn.

\* **Hiển thị danh sách file:**

- Liệt kê các file trong thư mục.
- Lọc file theo phần mở rộng (.txt, .py, .jpg)
- Hiển thị thông tin file (tên, loại) trong bảng Treeview.

\* **Mở file:**

- Xác định file được chọn từ Treeview.
- Mở file bằng chương trình mặc định của hệ điều hành.

\* **Xử lý lỗi:**

- Hiển thị thông báo lỗi khi thư mục không hợp lệ, file không tồn tại, hoặc xảy ra lỗi khác.

## 3.2. Sơ đồ khối các thuật toán chính

### 3.2.1. Mô tả các khối thuật toán chính

**Thuật toán chọn thư mục (select\_folder):**

- **Đầu vào:** Không có.
- **Đầu ra:** Đường dẫn thư mục được lưu vào self.current\_folder.
- **Chức năng:** Mở hộp thoại chọn thư mục và gọi hàm show\_files để hiển thị danh sách file.
- **Luồng xử lý:**
  1. Mở hộp thoại chọn thư mục (filedialog.askdirectory).
  2. Nếu thư mục được chọn, lưu đường dẫn vào self.current\_folder.
  3. Gọi show\_files để cập nhật danh sách file.

**Thuật toán hiển thị danh sách file (show\_files):**

- **Đầu vào:** Đường dẫn thư mục (self.current\_folder).
- **Đầu ra:** Danh sách file được hiển thị trong bảng Treeview.

- **Chức năng:** Liệt kê các file có phần mở rộng hợp lệ (.txt, .py, .jpg) và hiển thị trong Treeview.

- **Luồng xử lý:**

1. Xóa nội dung hiện tại của Treeview.
2. Kiểm tra tính hợp lệ của thư mục bằng `os.path.isdir`.
3. Liệt kê file trong thư mục bằng `os.listdir`.
4. Lọc file theo phần mở rộng hợp lệ.
5. Xác định loại file bằng từ điển `file_type`.
6. Thêm thông tin file (tên, loại) vào Treeview.

**Thuật toán mở file (`open_file`):**

- **Đầu vào:** Tên file được chọn từ Treeview.
- **Đầu ra:** File được mở bằng chương trình mặc định.
- **Chức năng:** Mở file được chọn bằng chương trình mặc định của hệ điều hành.

- **Luồng xử lý:**

1. Lấy tên file từ item được chọn trong Treeview.
2. Tạo đường dẫn đầy đủ của file bằng `os.path.join`.
3. Mở file bằng `os.startfile`.
4. Xử lý lỗi nếu file không tồn tại hoặc không thể mở.

### 3.2.2. Quan hệ đầu vào-ra giữa các khối

- **Chọn thư mục → Hiển thị danh sách file:** Đường dẫn thư mục từ `select_folder` được truyền vào `show_files` để liệt kê file.
- **Hiển thị danh sách file → Mở file:** Thông tin file được chọn từ Treeview trong `show_files` được truyền vào `open_file` để mở file.
- **Xử lý lỗi:** Các thuật toán sử dụng messagebox để thông báo lỗi nếu đầu vào không hợp lệ hoặc xảy ra sự cố.

## 3.3. Cấu trúc dữ liệu

### 3.3.1. Các bảng dữ liệu

Chương trình sử dụng bảng Treeview để hiển thị danh sách file, với các trường thông tin sau:

- **Bảng Treeview:**

- **Trường thông tin:**

1. **Tên File** (string): Tên của file (bao gồm phần mở rộng).
2. **Loại File** (string): Loại file (Text, Python, Image) dựa trên phần mở rộng.

- **Mô tả:**

- Bảng được tạo bởi ttk.Treeview với hai cột: "Tên File" và "Loại File".
- Dữ liệu được điền động từ danh sách file trong thư mục, lọc theo phần mở rộng (.txt, .py, .jpg).

### 3.3.2. Các cấu trúc dữ liệu khác

- **Biến** self.current\_folder (string): Lưu đường dẫn thư mục được chọn.
- **Danh sách** valid\_extensions (tuple): Chứa các phần mở rộng hợp lệ (.txt, .py, .jpg).
- **Từ điển** file\_type (dict): Ánh xạ phần mở rộng file với loại file:

```
file_type = {  
    ".txt": "Text",  
    ".py": "Python",  
    ".jpg": "Image"  
}
```

## 3.4. Chương trình

### 3.4.1. Mô tả các hàm trong chương trình

**Hàm** \_\_init\_\_(self, root):

- **Chức năng:** Khởi tạo giao diện và các thuộc tính của lớp FileManager.
- **Tham số:**
  - root: Cửa sổ chính (tk.Tk).
- **Hoạt động:**
  - Thiết lập tiêu đề cửa sổ: "Trình Quản Lý Thư Mục".
  - Tạo khung chính (tk.Frame) và các thành phần giao diện (nút, Treeview).
  - Gắn sự kiện double-click trên Treeview với hàm open\_file.
  - Khởi tạo biến self.current\_folder.

**Hàm** select\_folder(self):

- **Chức năng:** Mở hộp thoại chọn thư mục và cập nhật danh sách file.
- **Tham số:** Không có.
- **Hoạt động:**
  - Mở hộp thoại chọn thư mục bằng filedialog.askdirectory.
  - Lưu đường dẫn thư mục vào self.current\_folder.
  - Gọi show\_files để hiển thị danh sách file.

### Hàm show\_files(self):

- **Chức năng:** Liệt kê và hiển thị các file hợp lệ trong thư mục.
- **Tham số:** Không có.
- **Hoạt động:**
  - Xóa nội dung hiện tại của Treeview.
  - Kiểm tra tính hợp lệ của thư mục (os.path.isdir).
  - Liệt kê file bằng os.listdir và lọc theo phần mở rộng hợp lệ.
  - Xác định loại file bằng từ điển file\_type.
  - Thêm thông tin file vào Treeview.
  - Xử lý lỗi bằng try-except.

### Hàm open\_file(self, event=None):

- **Chức năng:** Mở file được chọn từ Treeview.
- **Tham số:**
  - event: Sự kiện double-click (mặc định là None nếu gọi từ nút).
- **Hoạt động:**
  - Lấy thông tin file được chọn từ Treeview.
  - Tạo đường dẫn đầy đủ bằng os.path.join.
  - Mở file bằng os.startfile.
  - Hiển thị thông báo lỗi nếu file không tồn tại hoặc không thể mở.

## 3.4.2. Chương trình chính

### • Khởi tạo:

```
root = tk.Tk()
app = FileManager(root)
root.geometry("450x400") # Kích thước cửa sổ
root.mainloop()
```

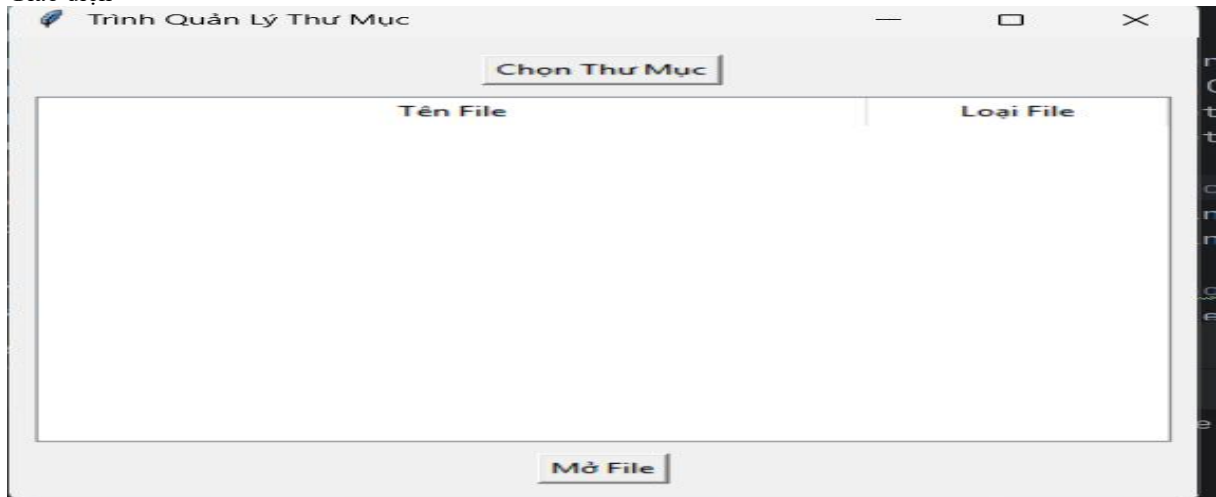
### • Mô tả:

- Tạo cửa sổ chính bằng tk.Tk.
- Khởi tạo đối tượng FileManager với cửa sổ chính.
- Thiết lập kích thước cửa sổ (450x400 pixel).
- Chạy vòng lặp chính của ứng dụng (mainloop) để xử lý các sự kiện giao diện.

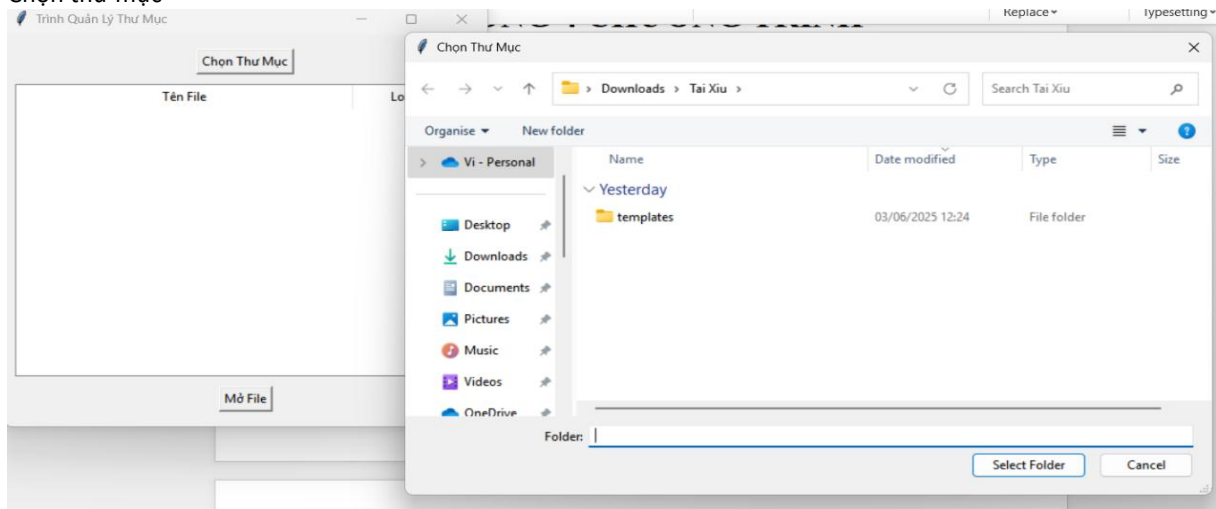
# CHƯƠNG 4 CHƯƠNG TRÌNH

## 4.1. Chương trình đã làm được cái gì

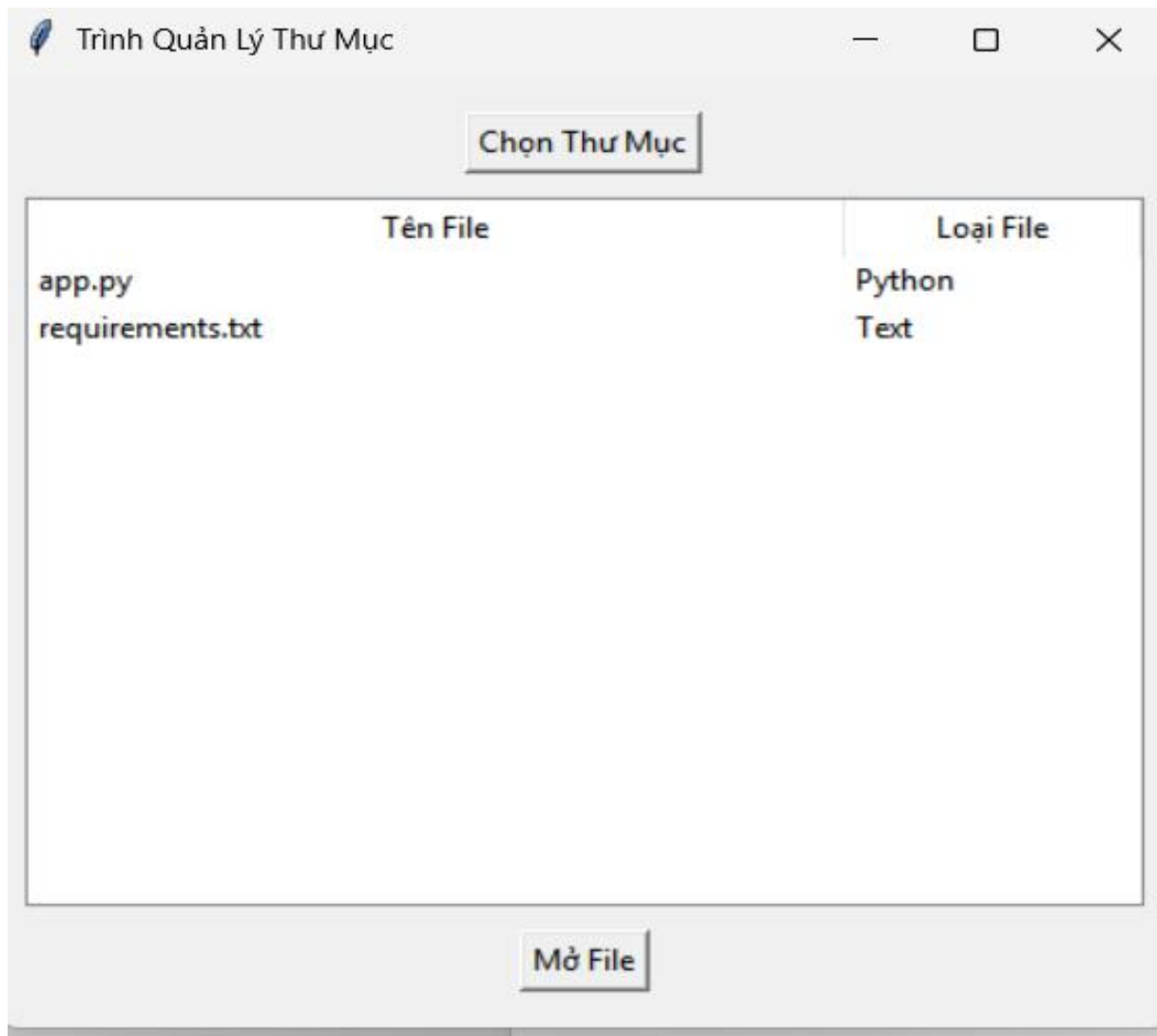
Giao diện



Chọn thư mục



Đọc folder



Mở File

```
pandas==2.2.0  
numpy==1.26.4
```

Ln 4, Col 1 29 characters

100%

Windows (CRLF)

UTF-8

## 4.2. Kết luận

### Sản phẩm đã thực hiện được:

Xây dựng thành công một giao diện quản lý thư mục đơn giản với Tkinter.

Cho phép người dùng chọn thư mục bất kỳ trên máy tính.

Hiển thị danh sách các file có định dạng .txt, .py, .jpg trong thư mục đã chọn bằng bảng Treeview.



Phân loại file theo từng loại: văn bản (Text), mã Python (Python), hình ảnh (Image).

Mở file trực tiếp bằng phần mềm mặc định khi người dùng nhấn nút “Mở File” hoặc nhấp đúp chuột.

Có xử lý lỗi và hiển thị thông báo khi gặp sự cố như: không chọn thư mục, file không tồn tại, lỗi mở file.

### **Những điều đã học được:**

Cách sử dụng thư viện tkinter để xây dựng giao diện người dùng thân thiện.

Sử dụng Treeview từ ttk để hiển thị danh sách có cột và phân loại.

Cách dùng filedialog để chọn thư mục, và messagebox để hiển thị cảnh báo/lỗi.

Áp dụng các kiến thức xử lý file và thư mục với os.path, os.listdir(), và os.startfile().

Biết cách gắn kết sự kiện chuột (double-click) với chức năng tương ứng trong giao diện.

### **Hướng cải tiến trong tương lai:**

Mở rộng chức năng để hiển thị thêm nhiều loại file như .pdf, .docx, .png, .mp3...

Thêm tính năng tìm kiếm file theo từ khóa trong thư mục hiện tại.

Hiển thị thông tin chi tiết như kích thước file, ngày sửa đổi cuối cùng.

Cho phép người dùng xóa, sao chép, đổi tên file trực tiếp từ giao diện.

Cải tiến giao diện bằng cách sử dụng ttk.Style hoặc kết hợp với thư viện giao diện hiện đại hơn như customtkinter.