



## Báo cáo Design Pattern

Sinh viên: Nguyễn Hoàng Sơn    20206165

Lớp: Tài Năng Toán - Tin

Khoá: 65

Kì học: 2022-1

# Mục lục

<b>1</b>	<b>Design Pattern</b>	<b>1</b>
1.1	Visitor Pattern . . . . .	1
1.2	Mediator Pattern . . . . .	2
<b>2</b>	<b>Ứng dụng</b>	<b>3</b>

# 1. Design Pattern

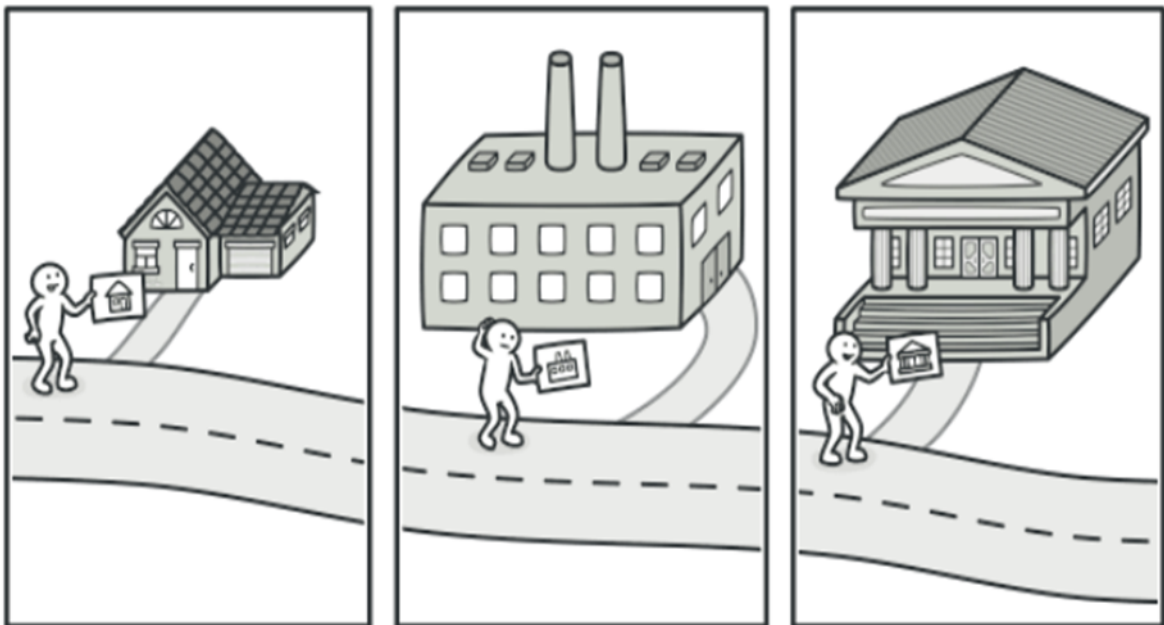
## 1.1 Visitor Pattern

Visitor Pattern thường được sử dụng khi bạn cần thêm các thao tác mới vào các lớp hiện có mà không cần phải thay đổi các lớp đó; hoặc khi bạn muốn tách rời các thao tác khỏi các đối tượng mà nó thao tác.

Visitor Pattern cũng giúp bạn tránh việc sử dụng các lớp có quá nhiều phương thức, giúp cho các lớp trở nên gọn gàng và dễ đọc hơn.

### Ví dụ minh họa

Hãy tưởng tượng một đại lý bảo hiểm dày dạn kinh nghiệm mong muốn có được khách hàng mới.



Hình 1.1: Minh họa Visitor Pattern

Anh ta ghé thăm mọi tòa nhà trong khu phố, cố gắng bán bảo hiểm cho mọi người anh ta gặp. Tùy thuộc vào loại tổ chức chiếm giữ tòa nhà, anh ta có thể đưa ra các chính sách bảo hiểm chuyên biệt:

- Nếu đó là một tòa nhà dân cư, anh ta bán bảo hiểm nhân thọ.
- Nếu đó là một nhà máy, anh ta bán bảo hiểm cháy nổ cho các thiết bị
- Nếu đó là một ngân hàng, anh ta bán bảo hiểm trộm cắp.

Anh ta tượng trưng cho một đối tượng **Visitor**.

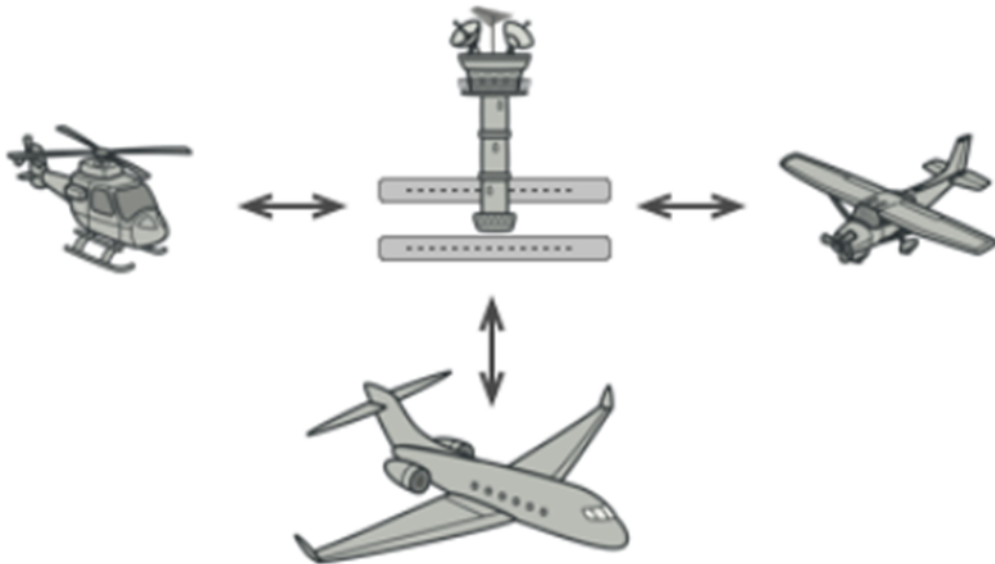
## 1.2 Mediator Pattern

Mediator Pattern (mô hình trung gian) được sử dụng để giảm sự phức tạp trong “giao tiếp” giữa các lớp và các đối tượng.

Mô hình này cung cấp một lớp trung gian có nhiệm vụ xử lý thông tin liên lạc giữa các lớp, hỗ trợ bảo trì mã code dễ dàng hơn, giúp thuận lợi cho việc mở rộng code sau này.

### Ví dụ minh họa

Các phi công của máy bay khi tiếp cận hoặc rời khỏi sân bay thì họ sẽ liên hệ với đài kiểm soát không lưu để biết rằng liệu hoạt động của mình có được cho phép hay không ?



Hình 1.2: Minh họa Mediator Pattern

Nếu không có đài kiểm soát không lưu, thì các phi công phải tự trao đổi với nhau. Điều này dẫn đến rủi ro tai nạn lớn khi số lượng máy bay tăng lên !!

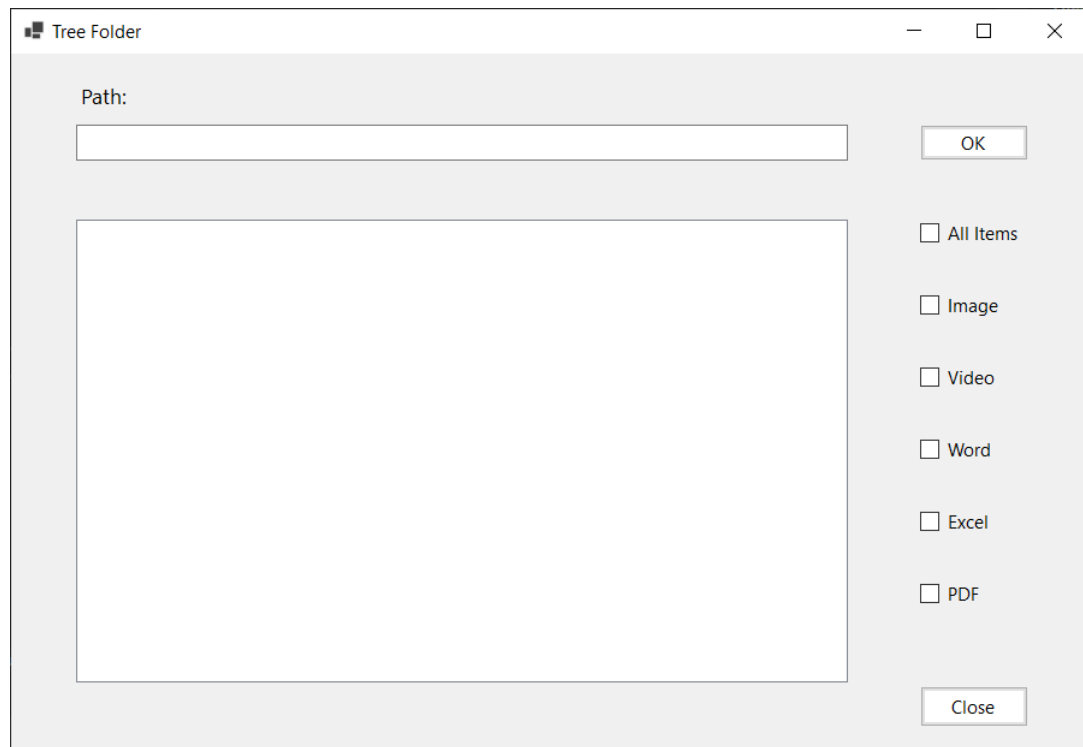
Tháp không lưu ở trường hợp này đóng vai trò là một **Mediator** điều tiết hoạt động của các chuyến bay.

## 2. Ứng dụng

### Mô tả ứng dụng

App có chức năng hiển thị cấu trúc của thư mục khi ta nhập vào đường dẫn của nó theo từng tùy chọn.

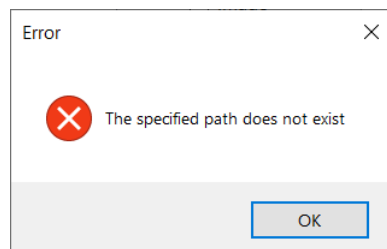
### Giao diện ứng dụng



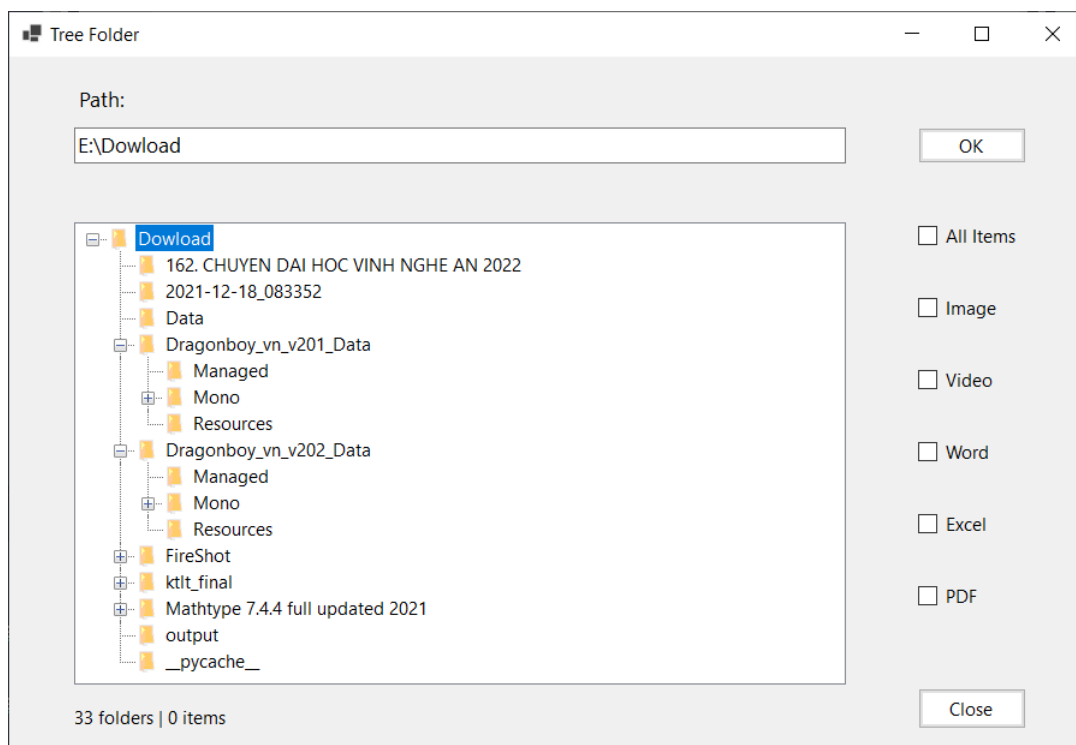
Hình 2.1: Giao diện App

### Cách sử dụng

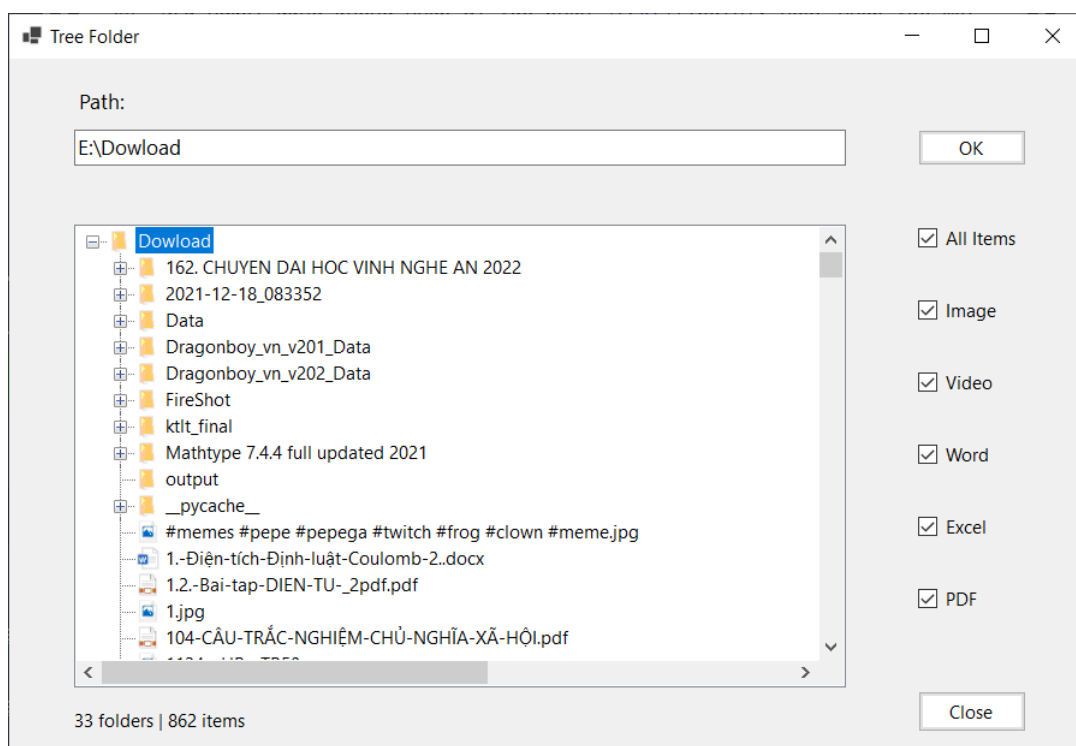
Nếu người dùng không nhập gì vào phần *path*, hoặc nhập vào một đường dẫn không tồn tại thì sẽ hiển thị một *MessageBox*.



Khi ta nhập vào một đường dẫn khả dụng, ở chế độ mặc định thì app sẽ chỉ hiển thị các thư mục con.

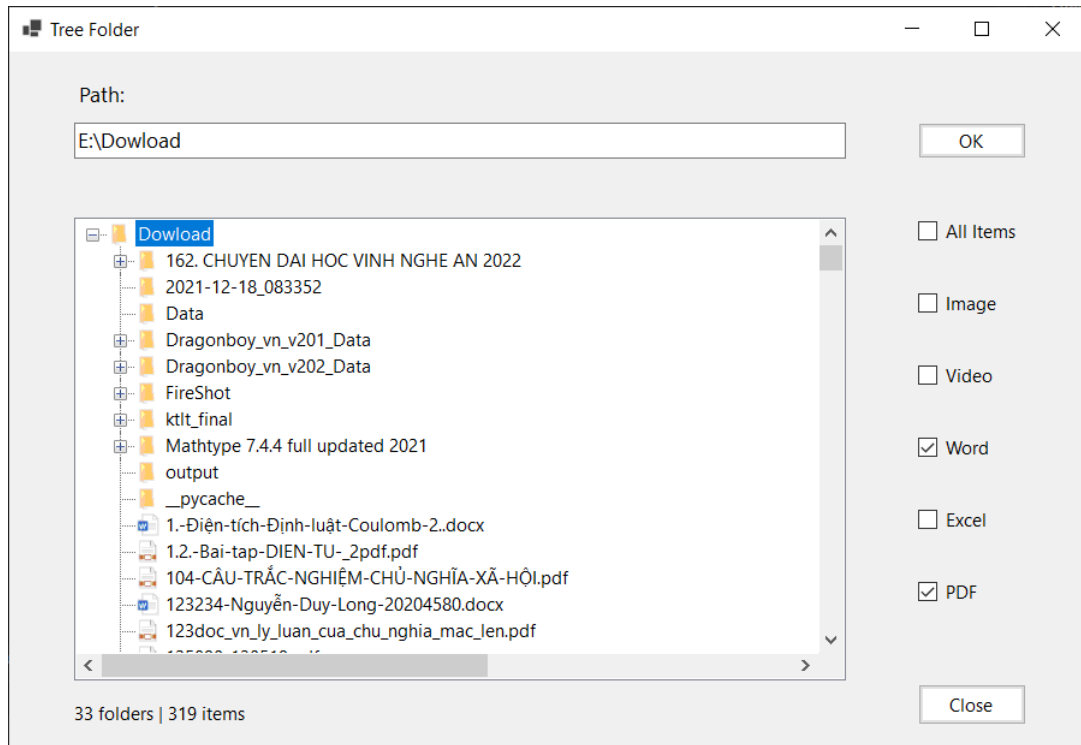


Khi ta *click* vào tùy chọn *All Items* thì tất cả các tùy chọn còn lại cũng được chọn theo. Ở chế độ này sẽ hiển thị mọi tệp có trong thư mục và các thư mục con.



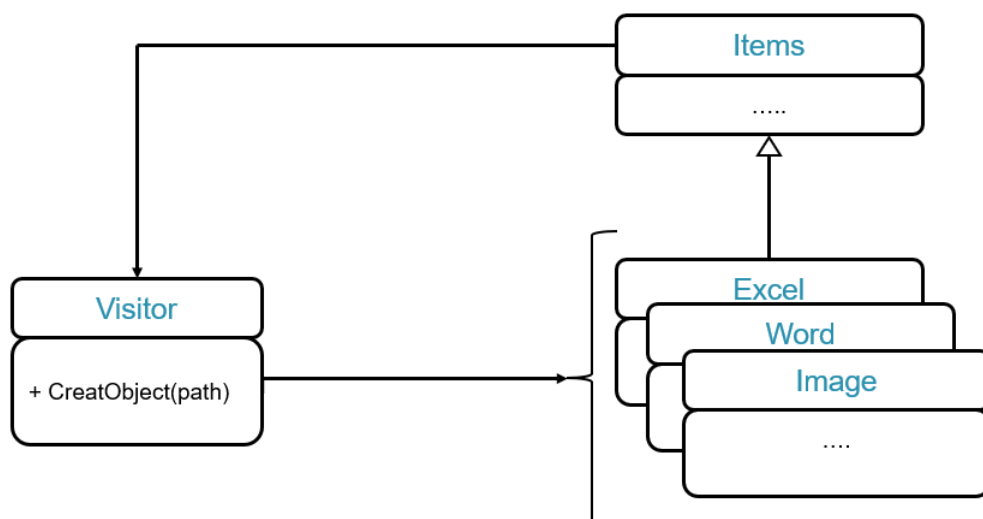
Khi ta bỏ chọn ở tùy chọn *All Items* thì các tùy chọn khác cũng được bỏ chọn theo. Trong khi *All Items* vẫn được chọn thì các tùy chọn khác không thể bỏ chọn được !!

Khi cần chỉ hiển thị loại tệp nào đó, ta cần chọn vào tùy trong tương ứng của nó trên giao diện



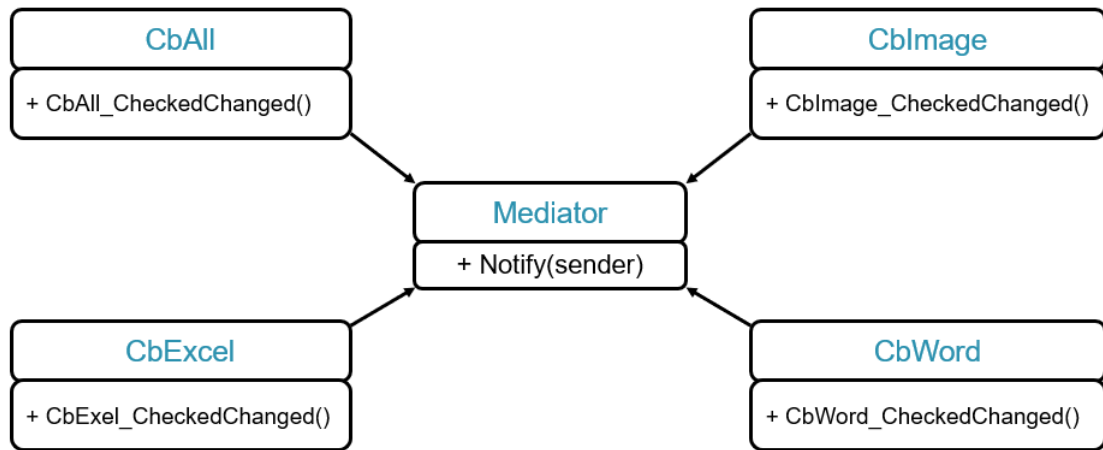
## Áp dụng design pattern

Với **Visitor Pattern**, em sử dụng để trả về đối tượng tương ứng với loại tệp khi ta có đường dẫn của nó. Việc này thể hiện tính linh hoạt của đối tượng **Visitor** đối với từng đối tượng khác nhau đúng với ý tưởng ban đầu.



Hình 2.2: Class Diagram Visitor Pattern

Còn **Mediator Pattern**, em sử dụng để quản lý quá trình người dùng thao tác với *CheckBox*. Mỗi khi người dùng bấm chọn các tùy chọn, sẽ có một tín hiệu được gửi đến đối tượng **Mediator**. Thông qua đó, ta có thể sử dụng để lọc các tệp chỉ định.



Hình 2.3: Class Diagram Mediator Pattern

Thông tin chi tiết xin vui lòng thầy xem trong file code em gửi kèm ạ!