**Sự Khác Nhau Giữa Folder**core**,**modules**, và**utils**Trong Tổ Chức Mã Nguồn**

Khi phát triển phần mềm, việc phân chia code vào các thư mục như core, modules, và utils giúp quản lý dự án dễ dàng hơn. Dưới đây là giải thích chi tiết về từng loại:

**1. Folder**core**(Lõi)**

**Định nghĩa**:

* Chứa **logic chính, thành phần "xương sống"** của ứng dụng.
* Thường bao gồm các class/function quan trọng nhất, không thể thiếu để chương trình hoạt động.
* Code trong core thường **ít bị thay đổi** trừ khi có cập nhật lớn.

**Ví dụ**:

text

/core

├── Engine.h // Game engine, hệ thống chính

├── Database.cpp // Kết nối CSDL cốt lõi

└── ConfigManager.h // Quản lý cấu hình toàn cục

**Đặc điểm**:

✅ **Tính chất**:

* Code **quan trọng, ổn định**, ít thay đổi.
* Thường được **dùng chung** bởi nhiều phần khác.

❌ **Không nên chứa**:

* Code UI, logic nghiệp vụ riêng lẻ.
* Tiện ích nhỏ (nên đặt trong utils).

**2. Folder**modules**(Các Module)**

**Định nghĩa**:

* Chứa **các thành phần độc lập**, có thể **tách rời** khỏi hệ thống chính.
* Mỗi module thường đại diện cho **một tính năng cụ thể** (Ví dụ: payment, auth, inventory).
* Có thể **bật/tắt** mà không ảnh hưởng đến core.

**Ví dụ**:

text

/modules

├── /auth // Xử lý đăng nhập

│ ├── AuthService.h

│ └── LoginController.cpp

├── /payment // Thanh toán

│ └── PaymentGateway.cpp

└── /inventory // Quản lý kho

└── ItemManager.h

**Đặc điểm**:

✅ **Tính chất**:

* **Độc lập**, có thể phát triển riêng.
* Dễ dàng **mở rộng/thay thế** (Ví dụ: thay StripePayment bằng PayPalPayment).

❌ **Không nên chứa**:

* Code dùng chung (nên đặt trong utils).
* Logic quá nhỏ, không đủ thành một module.

**3. Folder**utils**(Tiện ích)**

**Định nghĩa**:

* Chứa **các hàm/chức năng hỗ trợ** dùng chung trong dự án.
* Thường là **code ngắn, tái sử dụng nhiều lần** ở nhiều nơi.
* Không thuộc về logic nghiệp vụ cụ thể.

**Ví dụ**:

text

/utils

├── StringUtils.h // Xử lý chuỗi (trim, split)

├── MathHelper.cpp // Hàm toán học

└── Logger.h // Ghi log

**Đặc điểm**:

✅ **Tính chất**:

* **Không phụ thuộc** vào core hay modules.
* **Dễ kiểm thử** (unit test).

❌ **Không nên chứa**:

* Logic nghiệp vụ (Ví dụ: calculateTax() nên thuộc modules/payment).
* Code chỉ dùng ở **một nơi duy nhất**.

**Bảng So Sánh Chi Tiết**

| **Tiêu Chí** | core | modules | utils |
| --- | --- | --- | --- |
| **Mục đích** | Logic chính, hệ thống lõi | Tính năng độc lập | Hàm tiện ích dùng chung |
| **Phụ thuộc** | Được dùng bởi nhiều thành phần | Có thể phụ thuộc vào core | Không phụ thuộc vào đâu |
| **Khả năng tách rời** | Không thể thiếu | Có thể disable mà không lỗi | Luôn sẵn dùng |
| **Ví dụ** | Game Engine, Database Core | Payment, Auth, Inventory | Logger, String Manipulation |

**Khi Nào Dùng Folder Nào?**

1. **Dùng**core khi:
   * Code là **nền tảng** của ứng dụng (Ví dụ: hệ thống Render, Kernel).
   * Cần **truy cập từ nhiều nơi**.
2. **Dùng**modules khi:
   * Code đại diện cho **một tính năng riêng biệt** (Ví dụ: thanh toán, đăng nhập).
   * Có thể **tích hợp hoặc gỡ bỏ** mà không phá vỡ hệ thống.
3. **Dùng**utils khi:
   * Code là **helper function** (Ví dụ: formatDate(), generateRandomID()).
   * Được dùng **lặp lại ở nhiều chỗ**.

**Ví Dụ Thực Tế**

**Game Engine (C++)**

text

/src

/core

├── GameLoop.cpp

└── PhysicsEngine.h

/modules

├── /ai // Trí tuệ nhân tạo

└── /multiplayer // Kết nối mạng

/utils

├── VectorMath.h

└── DebugDraw.cpp

**Web App (Node.js)**

text

/src

/core

├── Server.js // Khởi tạo server

└── Database.js // Kết nối DB chính

/modules

├── /user // Quản lý người dùng

└── /cart // Giỏ hàng

/utils

├── jwtHelper.js // Xử lý JWT

└── logger.js // Ghi log

**Kết Luận**

* core = **Trái tim** của ứng dụng (bắt buộc phải có).
* modules = **Các tính năng độc lập** (có thể thêm/bớt tùy ý).
* utils = **Công cụ hỗ trợ** (dùng chung toàn dự án).

Việc phân chia rõ ràng giúp:  
🔹 **Dễ bảo trì** (fix bug ở utils không ảnh hưởng core).  
🔹 **Tăng tính tái sử dụng** (code trong utils dùng được ở mọi nơi).  
🔹 **Dễ mở rộng** (thêm module mới không cần sửa core).