

# Biểu đồ hoạt động

---

- ❖ Khái niệm biểu đồ hoạt động
- ❖ Tập ký hiệu UML cho biểu đồ hoạt động
- ❖ Xây dựng biểu đồ hoạt động

# Khái niệm biểu đồ hoạt động

---

- ❖ Biểu đồ hoạt động
  - ❖ Mô tả một tiến trình bao gồm các hoạt động từ khi bắt đầu tới khi kết thúc
  - ❖ Biểu diễn các hoạt động và sự đồng bộ, chuyển tiếp các hoạt động của hệ thống trong một lớp hoặc kết hợp giữa các lớp với nhau trong một chức năng cụ thể
  - ❖ Là đồ thị có hướng, trong đó các nút (đỉnh) là các hoạt động và các cung là các dịch chuyển
- ❖ Hoạt động (activity):
  - ❖ Là công việc có thể được xử lý bằng tay như Điền mẫu, hoặc bằng máy như Hiện thị màn hình đăng nhập
  - ❖ Được biểu diễn trong biểu đồ bằng một hình chữ nhật tròn góc có mang tên của hoạt động

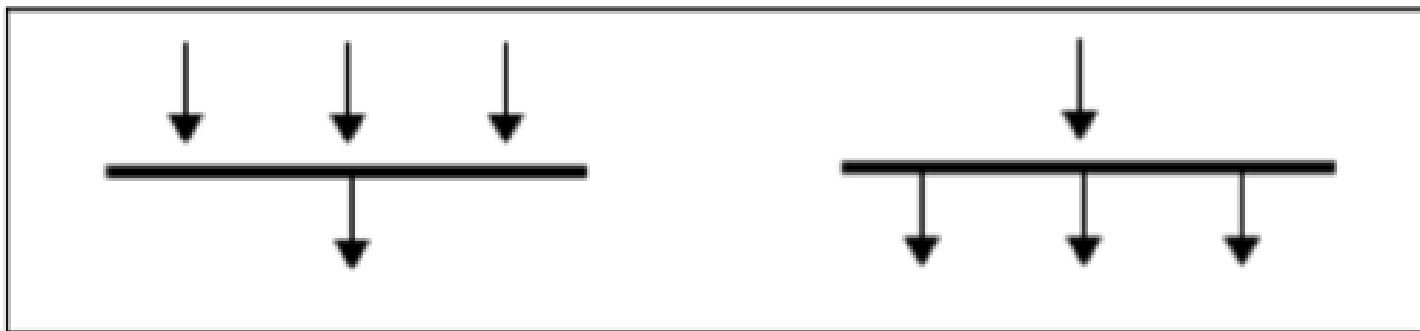
# Khái niệm biểu đồ hoạt động

---

- ❖ Chuyển dịch (Transition):
  - ❖ Là sự chuyển từ hoạt động này sang hoạt động khác được thể hiện bằng một mũi tên nối giữa hai hoạt động
- ❖ Biểu đồ hoạt động có thể được sử dụng cho nhiều mục đích khác nhau, ví dụ như
  - ❖ Để xác định các hành động phải thực hiện trong phạm vi một phương thức
  - ❖ Để xác định công việc cụ thể của một đối tượng
  - ❖ Để chỉ ra một nhóm hành động liên quan của các đối tượng được thực hiện như thế nào và chúng sẽ ảnh hưởng thế nào đến những đối tượng xung quanh
  - ❖ Mô tả tiến trình hoạt động của Usecase, của cả hệ thống

# Tập ký hiệu UML cho biểu đồ hoạt động

- ❖ Nút khởi đầu (start): thể hiện điểm bắt đầu của tiến trình hoạt động được ký hiệu bởi một hình tròn đặc
- ❖ Nút kết thúc (end): thể hiện điểm kết thúc các tiến trình hoạt động được ký hiệu hình tròn đặc có viền bao quanh
- ❖ Điều kiện chuyển dịch (transition condition): được ký hiệu bởi một hình thoi để thực hiện sự rẽ nhánh các hoạt động
- ❖ Thanh đồng bộ hóa (synchronization bars): để mở hay đóng các nhánh thực hiện song song



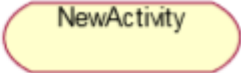






# Tập ký hiệu UML cho biểu đồ hoạt động

---

- ❖ Hoạt động (Activity):
  - ❖ Là một quy trình được định nghĩa rõ ràng
  - ❖ Có thể được thực hiện bởi một hàm hoặc một nhóm đối tượng
  - ❖ Được thể hiện bằng hình chữ nhật tròn cạnh
- ❖ Các luồng (swimlane):
  - ❖ Mỗi biểu đồ động có thể biểu diễn sự phối hợp hoạt động trong nhiều lớp khác nhau
  - ❖ Khi đó mỗi lớp được phân tách bởi một luồng (swimlane) riêng biệt
  - ❖ Các luồng này được biểu diễn đơn giản là các ô khác nhau trong biểu đồ

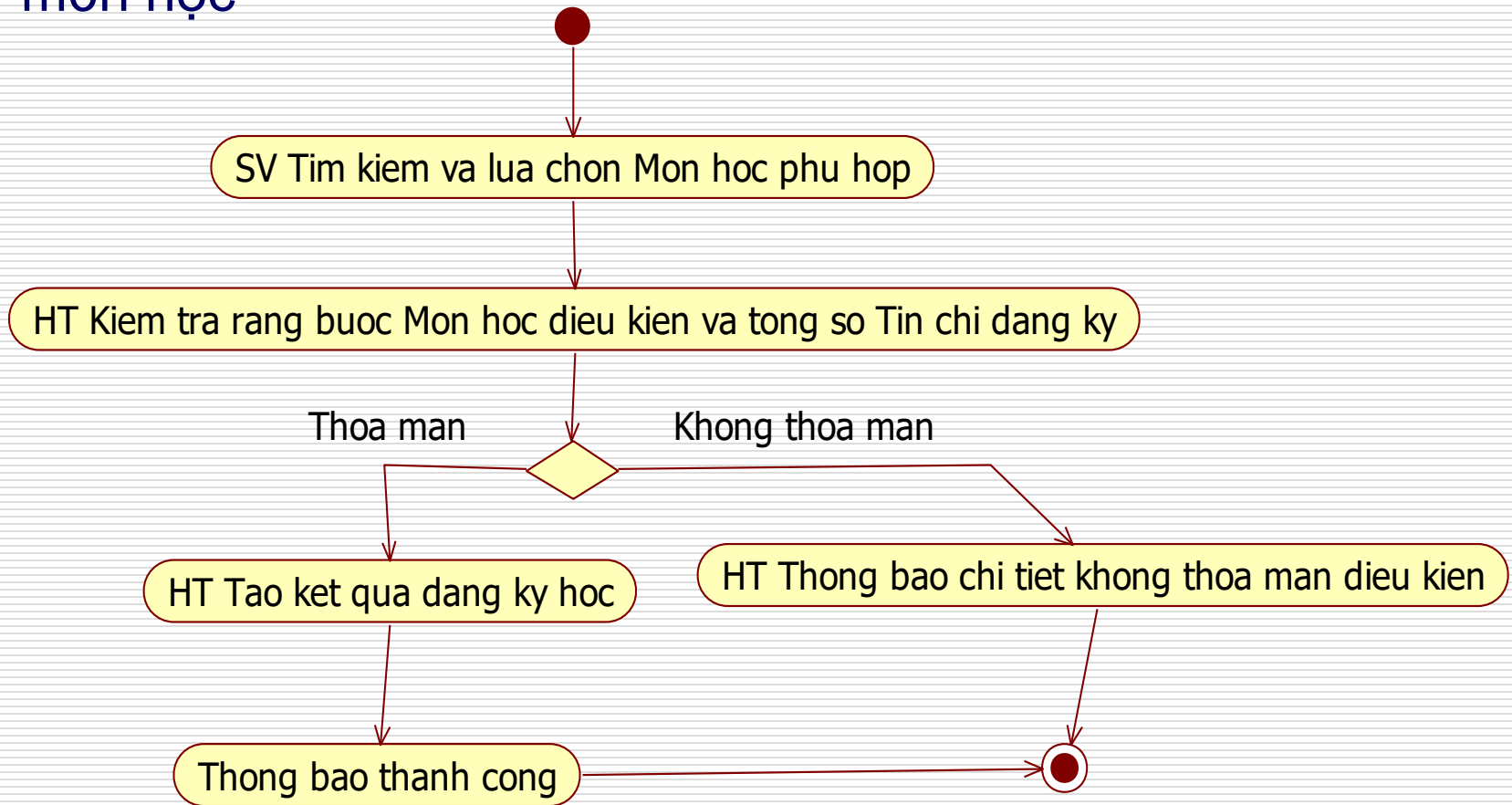
# Tập ký hiệu UML cho biểu đồ hoạt động

- ❖ Các ký hiệu UML cho biểu đồ hoạt động được tổng kết trong Bảng sau:

Phần tử mô hình	Ý nghĩa	Ký hiệu trong biểu đồ
<i>Hoạt động</i>	Mô tả một hoạt động gồm tên hoạt động và đặc tả của nó.	
<i>Trạng thái khởi đầu</i>		
<i>Trạng thái kết thúc</i>		
<i>Thanh đồng bộ ngang</i>	Mô tả thanh đồng bộ nằm ngang	
<i>Thanh đồng bộ hoá dọc</i>	Mô tả thanh đồng bộ theo chiều thẳng đứng	
<i>Chuyển tiếp</i>		
<i>Quyết định</i>	Mô tả một lựa chọn điều kiện.	
<i>Các luồng (swimlane)</i>	Phân tách các lớp đối tượng khác nhau tồn tại trong biểu đồ hoạt động	Phân cách nhau bởi một đường kẻ dọc từ trên xuống dưới biểu đồ

# Xây dựng biểu đồ hoạt động

- ❖ Ví dụ: Biểu đồ hoạt động cho usecase Sinh viên đăng ký môn học



# Xây dựng mô hình lớp phân tích

---

- ❖ Giới thiệu về biểu đồ lớp
- ❖ Mô hình 3 tầng
- ❖ Vấn đề tìm lớp trong hành vi của ca sử dụng
- ❖ Lớp phân tích là gì?
- ❖ Lớp Biên
- ❖ Lớp Thực thể
- ❖ Lớp Điều khiển



# Giới thiệu về biểu đồ lớp

---

- ❖ Trong phương pháp hướng đối tượng, một nhóm đối tượng có chung một số thuộc tính và phương thức tạo thành một lớp
- ❖ Mỗi tương tác giữa các đối tượng trong hệ thống sẽ được biểu diễn thông qua mối quan hệ giữa các lớp
- ❖ Các lớp (bao gồm cả các thuộc tính và phương thức) cùng với các mối quan hệ sẽ tạo thành biểu đồ lớp
- ❖ Biểu đồ lớp là một biểu đồ dạng mô hình tĩnh nhằm mô tả hướng nhìn tĩnh về một hệ thống bằng các khái niệm lớp, các thuộc tính, phương thức của lớp và mối quan hệ giữa chúng với nhau
- ❖ Trong pha phân tích, người phát triển xác định các lớp phân tích mà chưa quan tâm đến thuộc tính và phương thức chi tiết

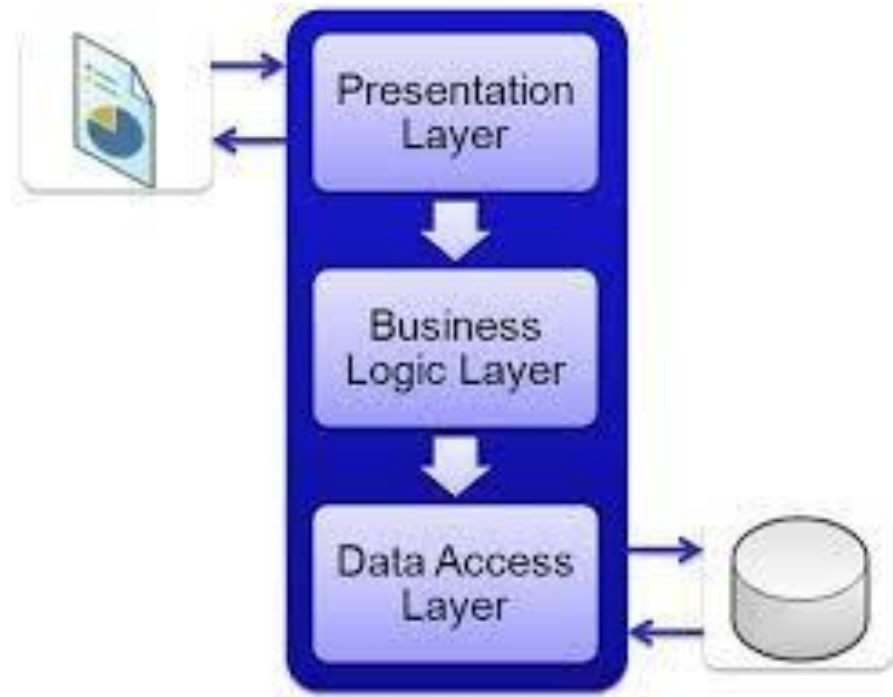
# Mô hình 3 tầng

❖ Gồm 3 phần chính:

❖ Presentation Layer (GUI):

❖ Layer này có nhiệm vụ chính giao tiếp với người dùng

❖ Bao gồm các thành phần giao diện (win form, web form...) và thực hiện các công việc như nhập liệu, hiển thị dữ liệu, kiểm tra tính đúng đắn dữ liệu trước khi gọi lớp Business Logic Layer (BLL)



# Mô hình 3 tầng

---

- ❖ Business Logic Layer (BLL): Layer này phân ra 2 thành nhiệm vụ
  - ❖ Đáp ứng các yêu cầu thao tác dữ liệu của GUI layer, xử lý chính nguồn dữ liệu từ Presentation Layer trước khi truyền xuống Data Access Layer và lưu xuống hệ quản trị CSDL
  - ❖ Kiểm tra các ràng buộc, tính toán vẹn và hợp lệ dữ liệu, thực hiện tính toán và xử lý các yêu cầu nghiệp vụ, trước khi trả kết quả về Presentation Layer
- ❖ Data Access Layer (DAL):
  - ❖ Layer này có chức năng giao tiếp với hệ quản trị CSDL như thực hiện các công việc liên quan đến lưu trữ và truy vấn dữ liệu (tìm kiếm, thêm, xóa, sửa...)

# Mô hình 3 tầng – Ưu điểm

---

- ❖ Việc phân chia thành từng lớp giúp cho code được tường minh hơn:
  - ❖ Việc chia ra từng lớp đảm nhận các chức năng khác nhau và riêng biệt như giao diện, xử lý, truy vấn thay vì để tất cả lại một chỗ
- ❖ Dễ bảo trì khi được phân chia, thì một thành phần của hệ thống sẽ dễ thay đổi
  - ❖ Việc thay đổi này có thể được cô lập trong 1 lớp, hoặc ảnh hưởng đến lớp gần nhất mà không ảnh hưởng đến cả chương trình

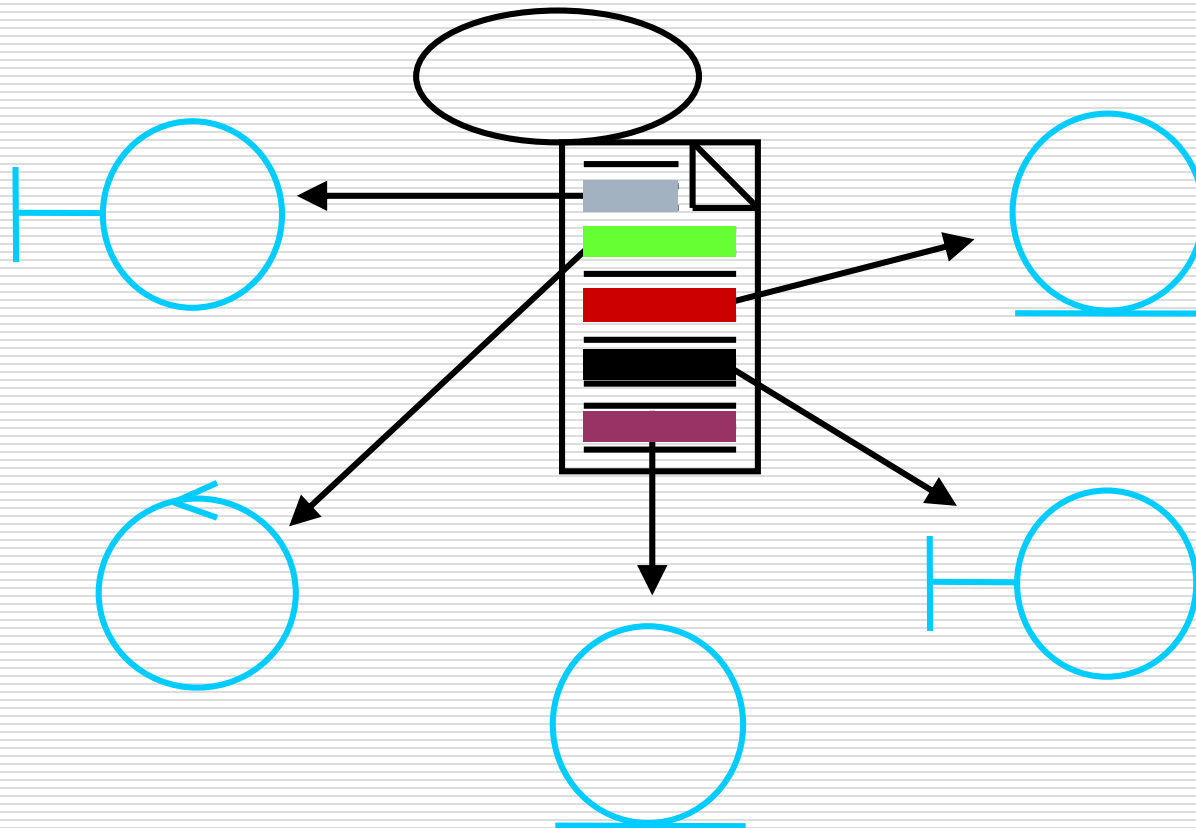
# Mô hình 3 tầng – Ưu điểm

---

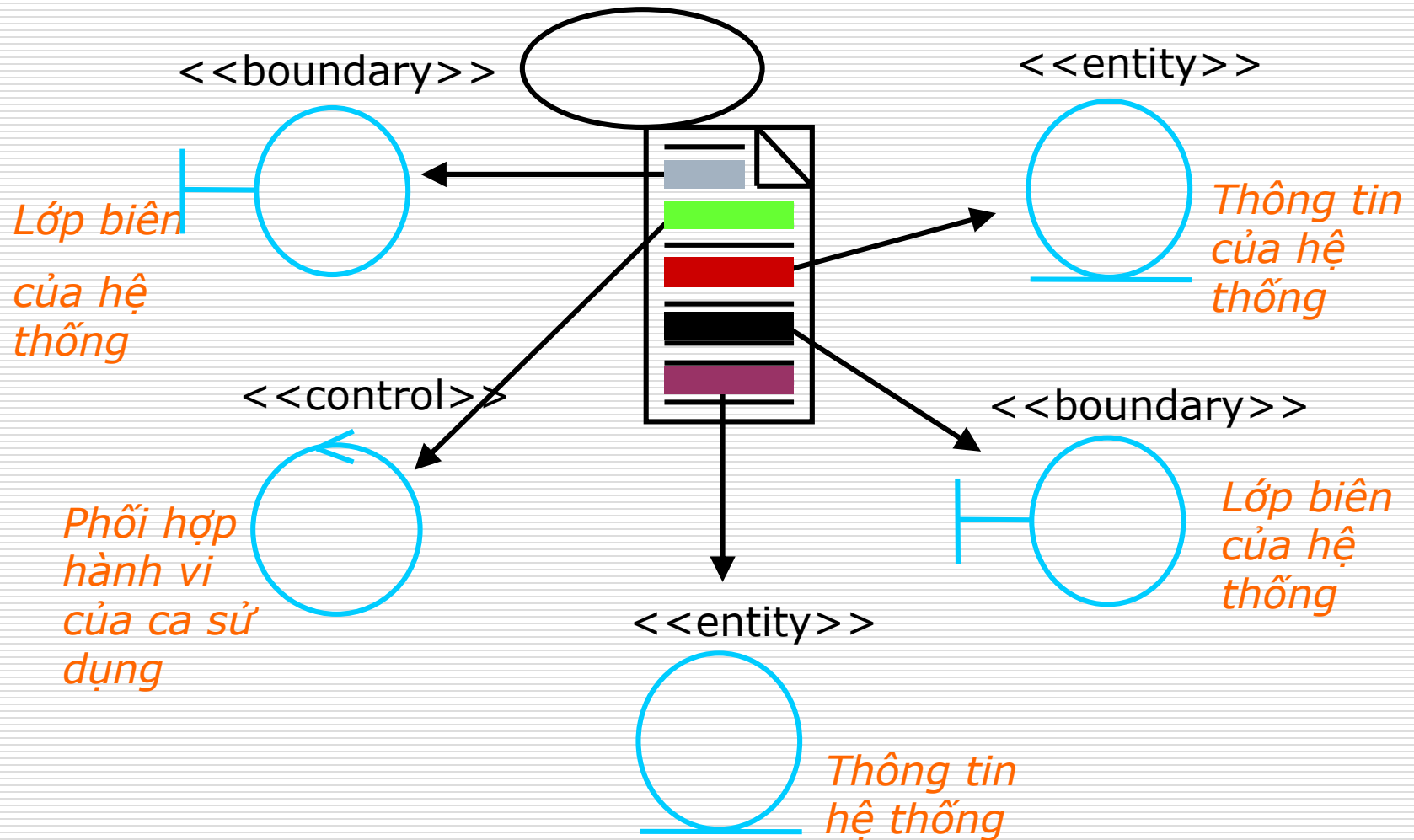
- ❖ Dễ phát triển, tái sử dụng:
  - ❖ Khi chúng ta muốn thêm một chức năng nào đó thì việc lập trình theo một mô hình sẽ dễ dàng hơn vì chúng ta đã có chuẩn để tuân theo
  - ❖ Việc sử dụng lại khi có sự thay đổi giữa hai môi trường (Winform sang Webfrom) thì chỉ việc thay đổi lại lớp GUI
- ❖ Dễ bàn giao
  - ❖ Nếu mọi người đều theo một quy chuẩn đã được định sẵn, thì công việc bàn giao, tương tác với nhau sẽ dễ dàng hơn và tiết kiệm được nhiều thời gian
- ❖ Dễ phân phối khối lượng công việc
  - ❖ Mỗi một nhóm, một bộ phận sẽ nhận một nhiệm vụ trong mô hình 3 lớp
  - ❖ Việc phân chia rõ ràng như thế sẽ giúp các lập trình viên kiểm soát được khối lượng công việc của mình

# Vấn đề tìm lớp trong hành vi của ca sử dụng

- ❑ Toàn bộ hành vi của ca sử dụng cần phải được phân phối vào các lớp phân tích



# Lớp phân tích là gì?



# Lớp phân tích là gì?

---

- ❖ Lớp biên (boundary class):
  - ❖ Là lớp nằm ở ranh giới giữa hệ thống với môi trường bên ngoài
  - ❖ Thực hiện vai trò nhận yêu cầu trực tiếp từ các tác nhân và chuyển các yêu cầu đó cho các lớp bên trong hệ thống
- ❖ Lớp thực thể (entity class):
  - ❖ Là lớp đại diện cho các thực thể chứa thông tin về các đối tượng xác định nào đó.
  - ❖ Ví dụ, lớp Khách hàng, Hóa đơn
- ❖ Lớp điều khiển (controller class):
  - ❖ Thực hiện các chức năng điều khiển hoạt động của hệ thống tương ứng với các chức năng cụ thể nào đó của một nhóm các lớp biên hoặc nhóm các lớp thực thể



# Lớp Biên

---

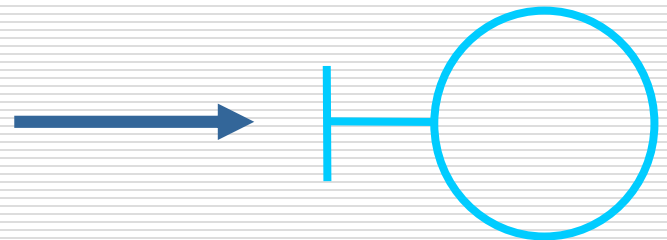
- ❖ Lớp biên là gì?
- ❖ Vai trò của lớp biên
- ❖ Cách xác định lớp biên

# Lớp biên là gì?

---

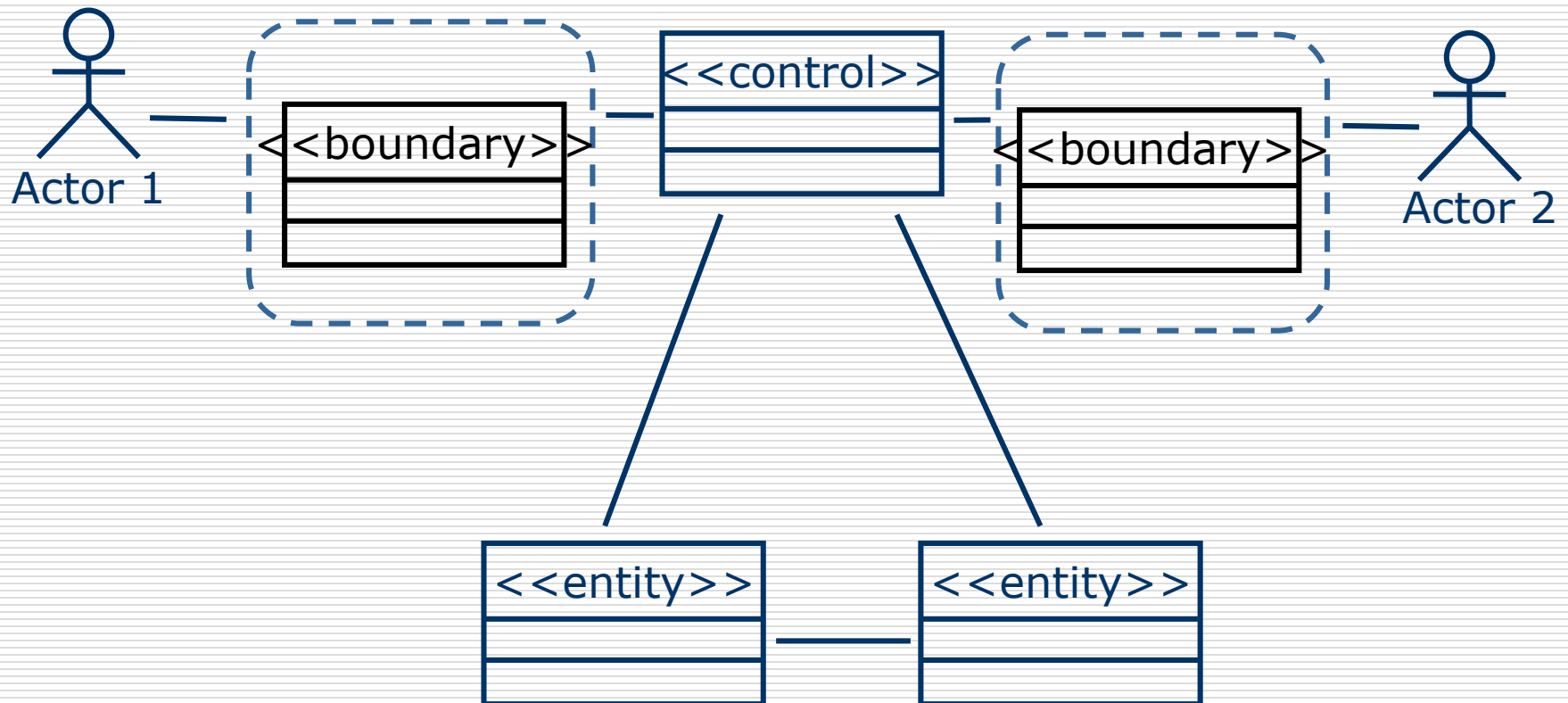
- ❑ Là cầu nối giữa hệ thống và những thứ bên ngoài hệ thống
  - Nằm ở ranh giới giữa hệ thống với môi trường bên ngoài
  - Thực hiện vai trò nhận yêu cầu trực tiếp từ các tác nhân và chuyển các yêu cầu đó cho các lớp bên trong hệ thống
- ❑ Một số kiểu lớp biên
  - Lớp giao diện người dùng
  - Lớp giao diện hệ thống
  - Lớp giao diện thiết bị

*Analysis class  
stereotype*



*Phụ thuộc môi trường*

# Vai trò của các lớp biên



*Mô hình hóa tương tác giữa hệ thống với môi trường của nó*

# Cách xác định lớp biên

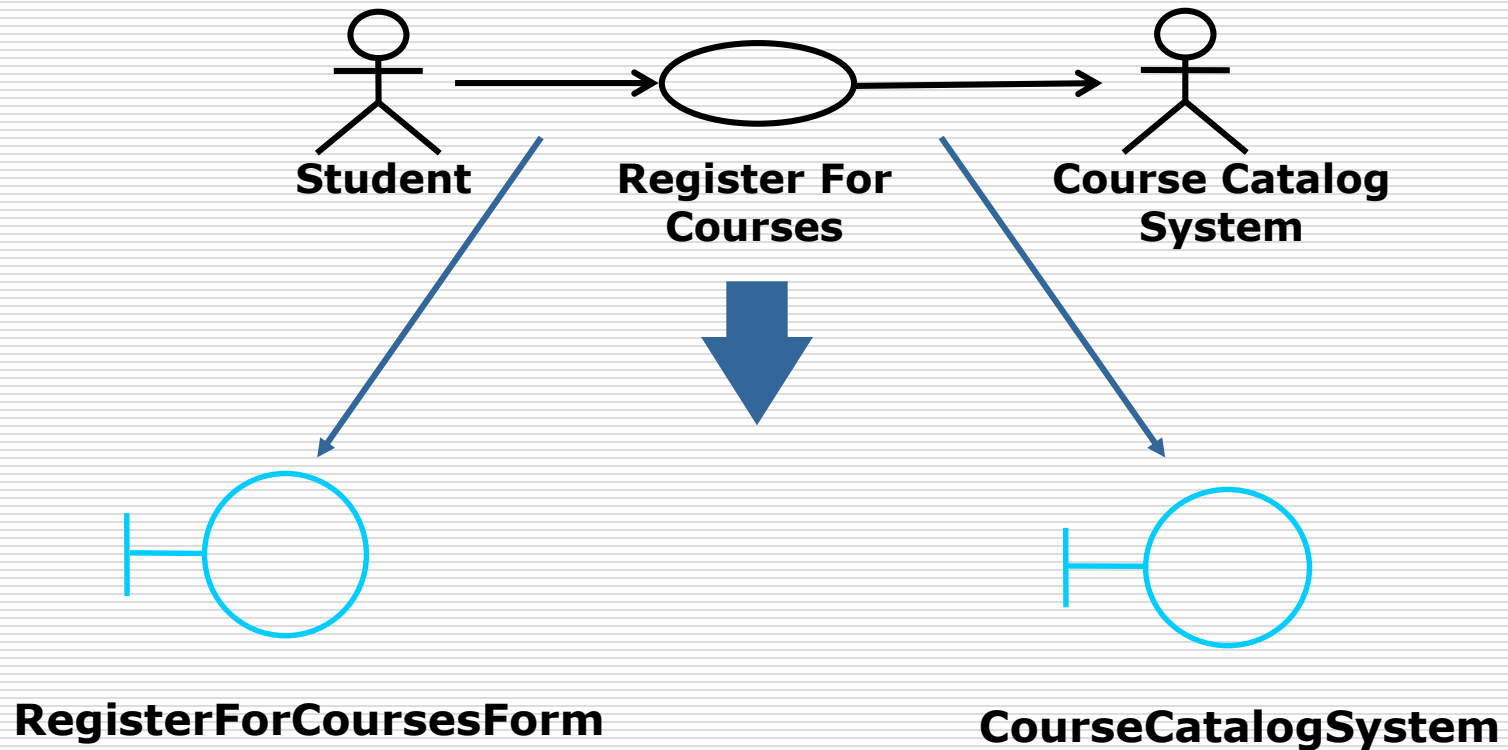
---

- Thông thường, xác định được một lớp biên trên mỗi cặp tác nhân/ca sử dụng



# Cách xác định lớp biên

- Ví dụ: Lớp biên cho usecase Đăng ký học



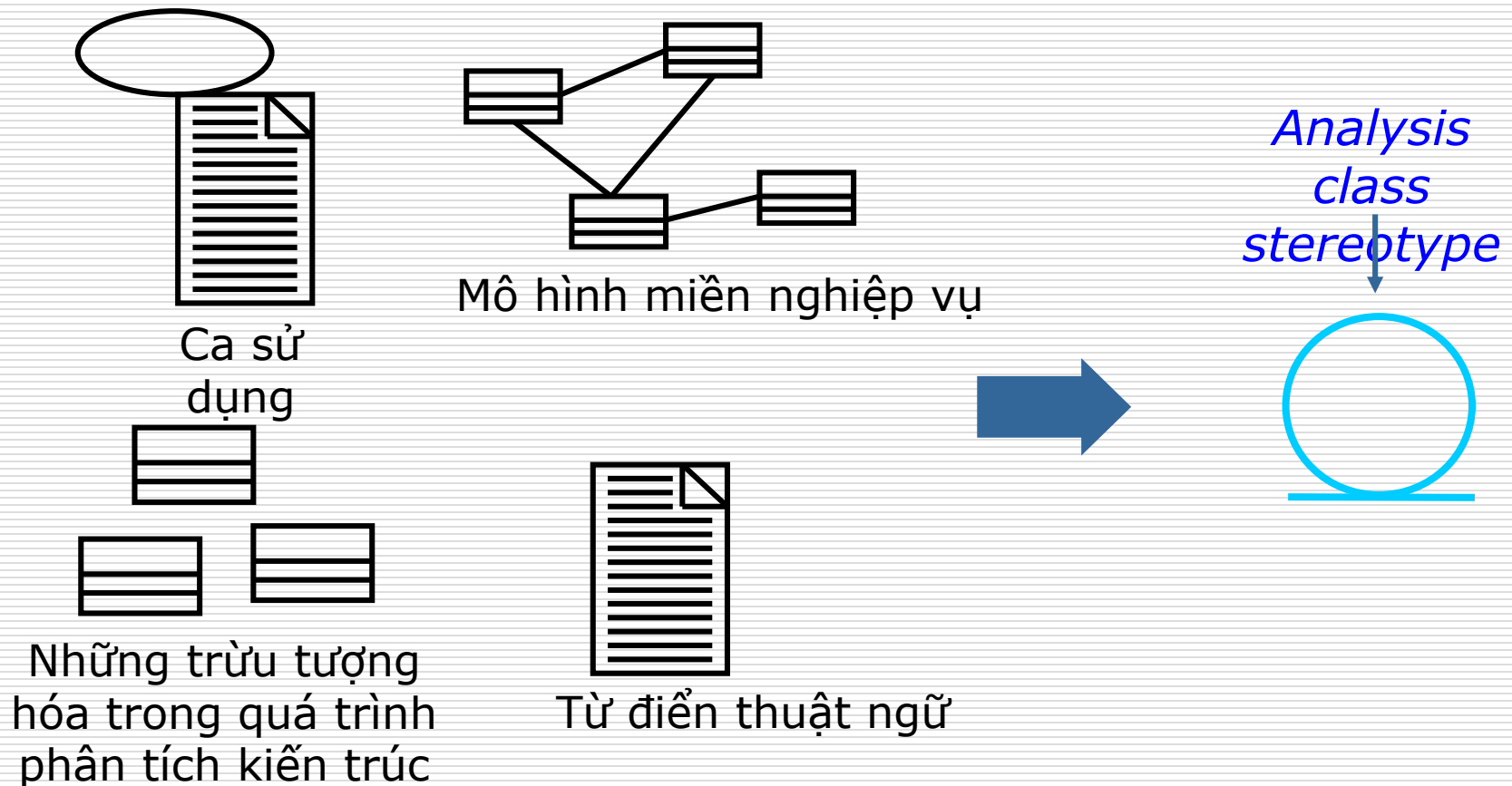
# Lớp thực thể

---

- ❖ Lớp thực thể là gì?
- ❖ Vai trò của lớp thực thể
- ❖ Cách xác định lớp thực thể

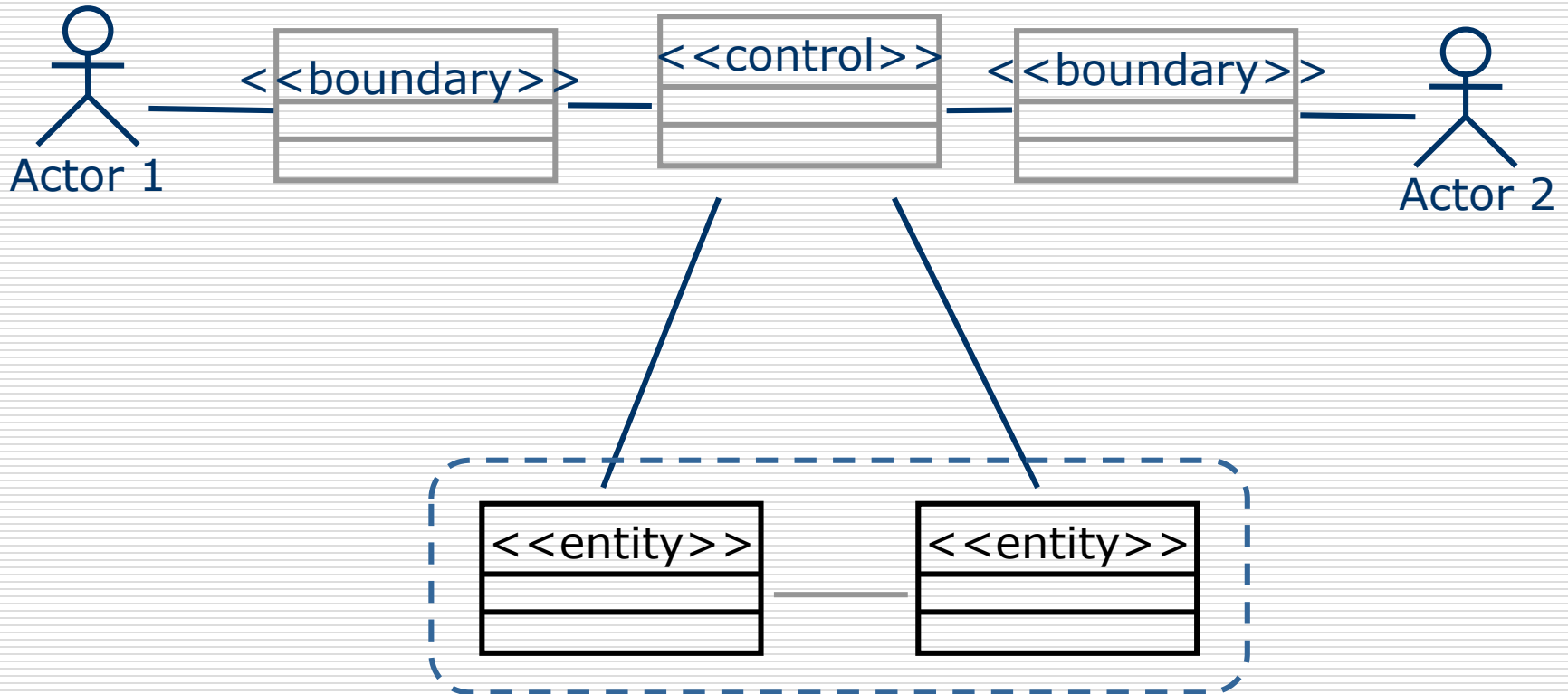
# Lớp thực thể là gì?

- ❑ Những trừu tượng hóa chính của hệ thống
- ❑ Là lớp đại diện cho các thực thể chứa thông tin về các đối tượng xác định nào đó. Ví dụ: lớp Khách hàng, Hóa đơn



# Vai trò của lớp thực thể

- ❑ Lưu trữ và quản lý thông tin trong hệ thống





# Cách xác định thực thể

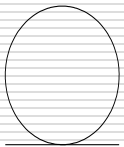
---

- ❖ Sử dụng luồng sự kiện của ca sử dụng như đầu vào
- ❖ Xác định các trừu tượng hóa chính của ca sử dụng
  - Gạch chân các cụm danh từ
- ❖ Thông thường, sử dụng phương pháp lọc danh từ
  - Gạch chân các mệnh đề danh từ trong luồng sự kiện của ca sử dụng
  - Xóa bỏ những cụm danh từ dư thừa
  - Xóa bỏ những cụm mơ hồ, không rõ ràng
  - Xóa bỏ tác nhân (nằm ngoài phạm vi hệ thống)
  - Xóa bỏ những cấu trúc cài đặt
  - Xóa bỏ thuộc tính (sử dụng trong giai đoạn sau)
  - Xóa bỏ hoạt động (operations)

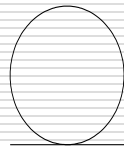
# Cách xác định thực thể

---

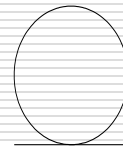
- ❖ Ví dụ: Bài toán quản lý thư viện
  - ❖ Từ các use case và scenario, có thể liệt kê các danh từ như sau: bạn đọc, tên bạn đọc, địa chỉ bạn đọc, thủ thư, username, password, thẻ mượn, sách, ngày mượn sách, ngày trả sách, số lượng sách...
  - ❖ Dựa vào tập danh từ này, bước đầu ta có thể xác định một số lớp như: bạn đọc, thủ thư, thẻ mượn, đầu sách



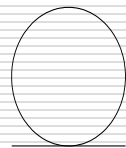
Entity\_ThuThuc



Entity\_BanDoc



Entity\_TaiLieu



Entity\_TheMuon

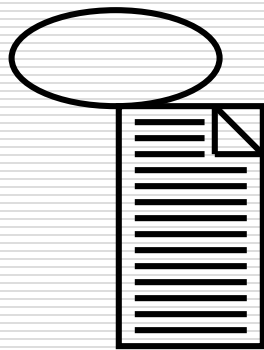
# Lớp điều khiển

---

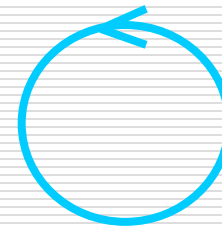
- ❖ Lớp điều khiển là gì?
- ❖ Vai trò của lớp điều khiển
- ❖ Cách xác định lớp điều khiển
- ❖ Đặc điểm các hành vi của lớp điều khiển

# Lớp điều khiển là gì?

- ❑ Thực hiện các chức năng điều khiển hoạt động của hệ thống tương ứng với các chức năng cụ thể nào đó của một nhóm các lớp biên hoặc nhóm các lớp thực thể
- ❑ Phối hợp hành vi của ca sử dụng
  - Các ca sử dụng phức tạp thường đòi hỏi một hoặc nhiều lớp điều khiển

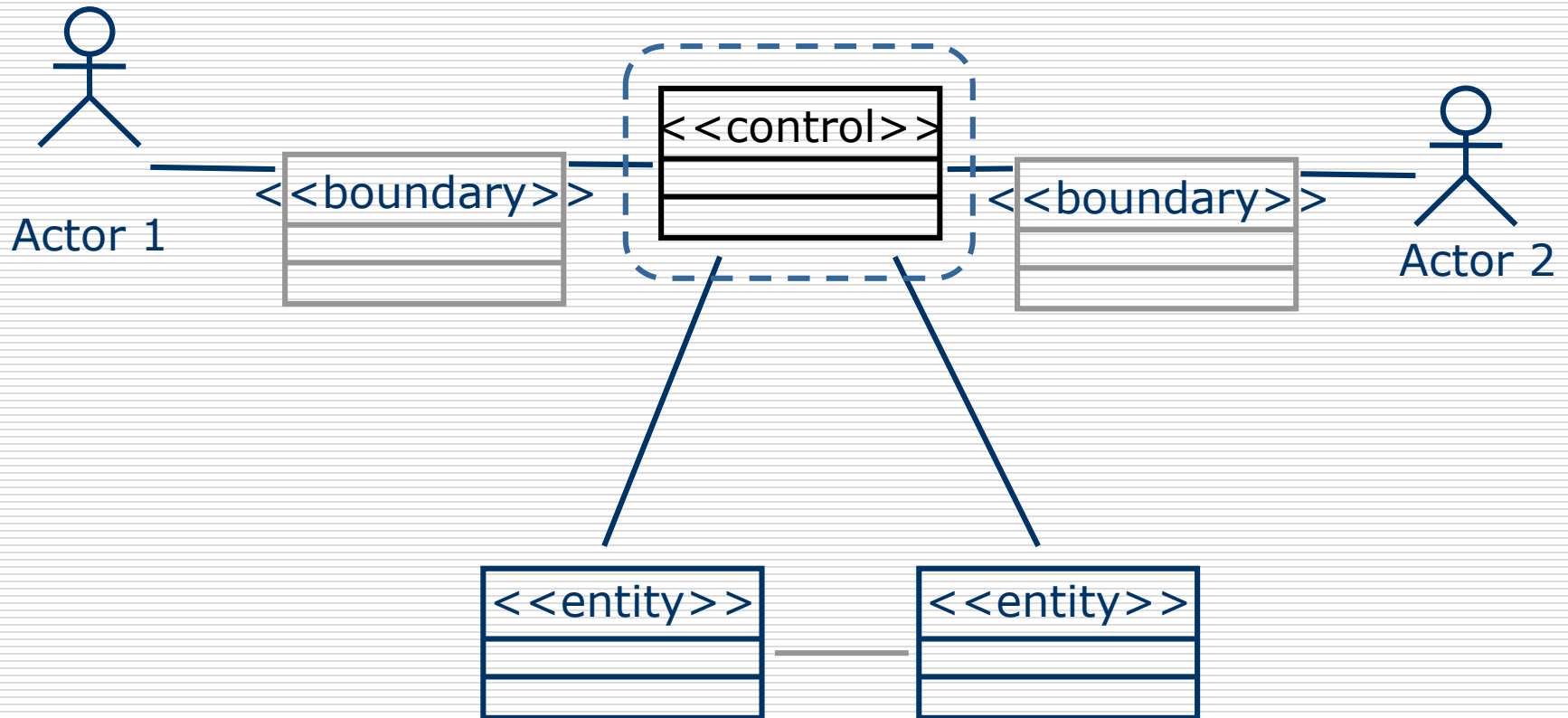


Ca sử dụng



***Analysis  
class  
stereotype***

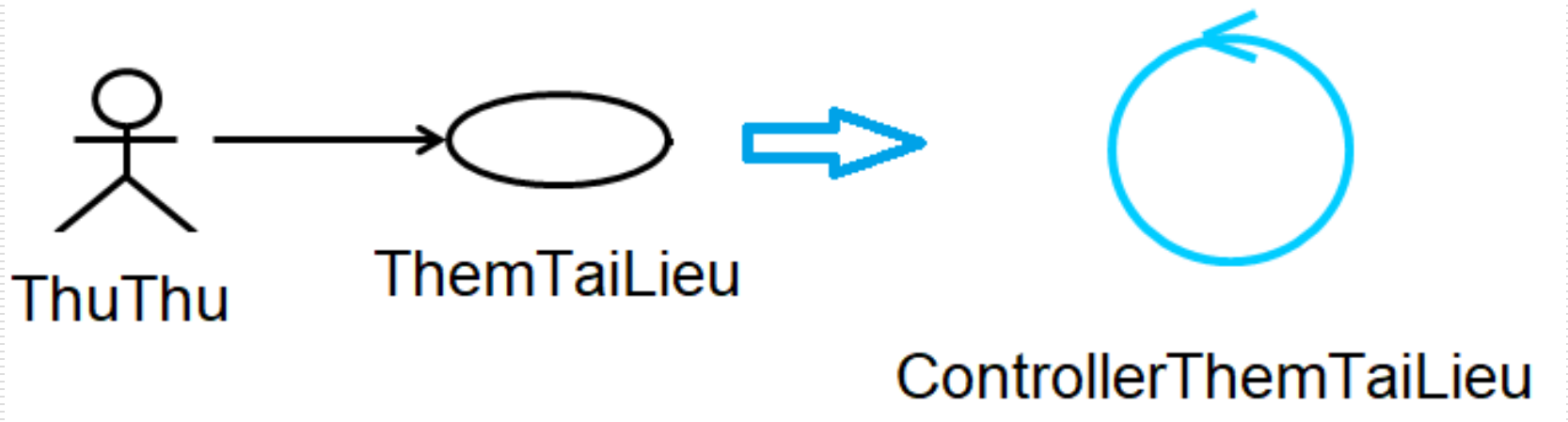
# Vai trò của một lớp điều khiển



*Phối hợp hành vi của ca sử dụng*

# Cách xác định lớp điều khiển

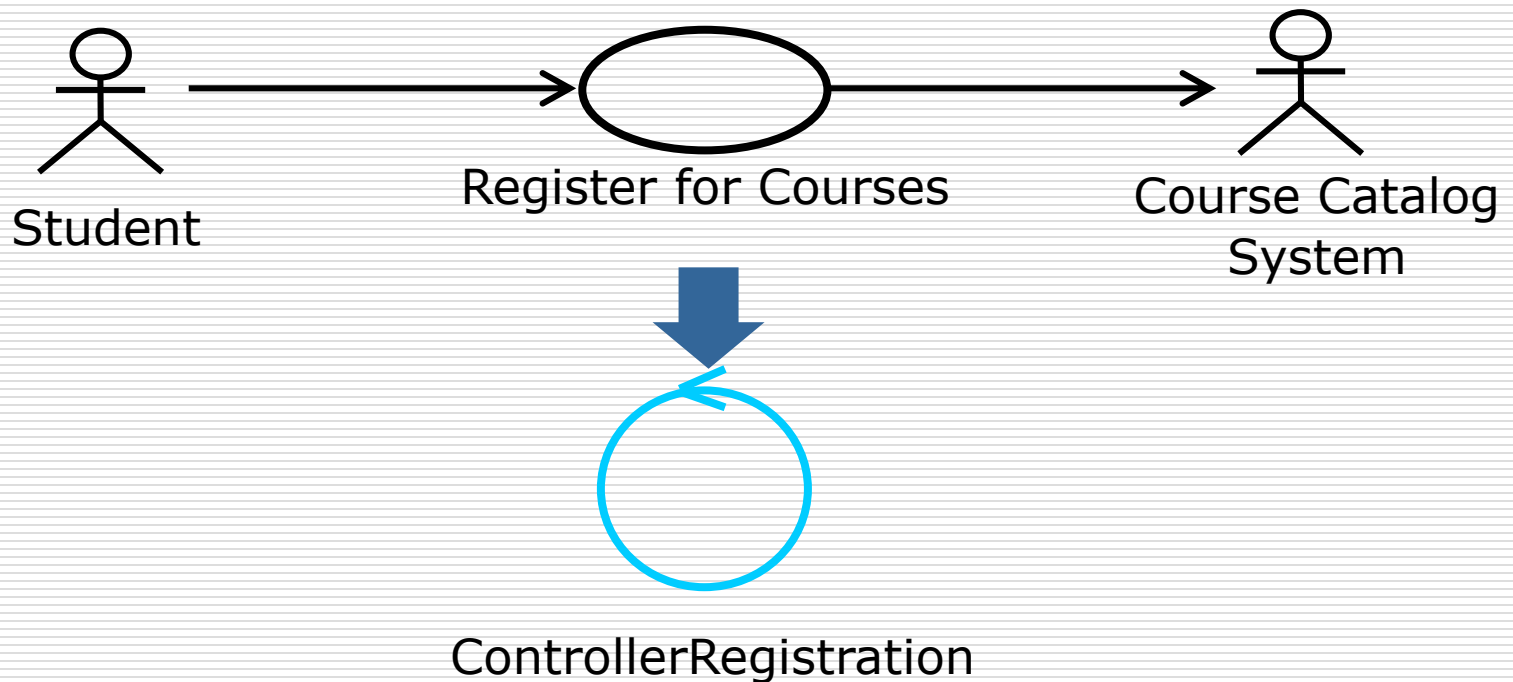
- Thông thường, xác định một lớp điều khiển trên mỗi ca sử dụng
  - Khi việc phân tích vẫn tiếp tục, một lớp điều khiển của một ca sử dụng phức tạp có thể được phát triển thành nhiều hơn một lớp
- Ví dụ: Lớp điều khiển cho usecase Thêm Tài liệu



# Cách xác định lớp điều khiển

---

- Ví dụ: Lớp điều khiển cho usecase Đăng ký học



# Đặc điểm các hành vi của lớp điều khiển

---

- ❑ Hành vi thường độc lập với bên ngoài (môi trường hệ thống)
- ❑ Hành vi xác định logic điều khiển và tổ chức các giao dịch trong ca sử dụng
- ❑ Hành vi ít bị thay đổi khi cấu trúc và hành vi bên trong lớp thực thể thay đổi.
- ❑ Hành vi sử dụng hay thiết lập nội dung của các lớp thực thể.
- ❑ Hành vi thường không là duy nhất, tùy theo từng tình huống, kịch bản ca sử dụng.



# Biểu đồ lớp phân tích cho usecase

- Biểu đồ lớp phân tích cho usecase Thêm tài liệu, Sửa tài liệu

