

## ***Chương 7***

# ***MÔ HÌNH HÓA HÀNH VI***

1. Biểu đồ tuần tự
2. Biểu đồ cộng tác
3. Biểu đồ trạng thái
4. Biểu đồ hoạt động

# SDLC

## Lập kế hoạch hệ thống

Khởi tạo dự án

Quản lý dự án

## Giai đoạn phân tích

Thu Thập yêu cầu

Mô hình hóa chức năng

Mô hình hóa cấu trúc

**Mô hình hóa hành vi**

## Giai đoạn thiết kế

Thiết kế lớp và phương thức

Thiết kế giao diện người dùng

## Giai đoạn thiết kế

## Giai đoạn thiết kế

# Khái niệm mô hình động

- Biểu đồ lớp chỉ mô tả khía cạnh tĩnh của hệ thống
  - Hành vi của hệ thống được mô tả bằng mô hình động bao gồm
    - Tương tác giữa các đối tượng: cộng tác hay trình tự
    - Trạng thái của đối tượng/lớp
    - Quá trình hoạt động của lớp/đối tượng
- ➔ Thiết lập mô hình động (dynamic modeling) và chi tiết hoá mô hình tĩnh

# Biểu đồ tuần tự

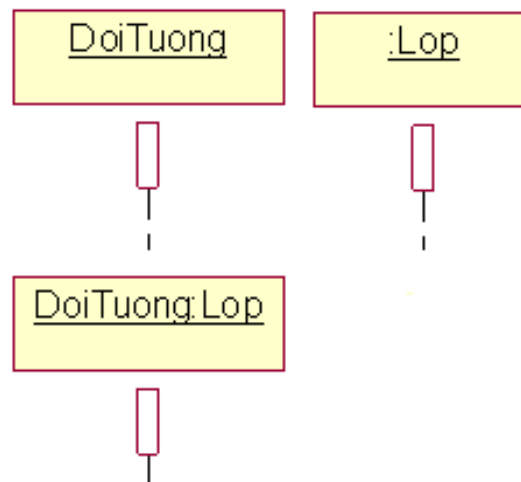
---

- Biểu đồ tuần tự (sequence diagram) biểu diễn sự tương tác giữa các đối tượng bằng việc nhấn mạnh thứ tự trao đổi thông điệp giữa các đối tượng
- Biểu đồ tuần tự gồm:
  - Đối tượng (Object)
  - Đường sinh tồn (Life Line)
  - Thông điệp (Message)
  - Vùng hoạt động (Activation)
  - Chú thích (Notes) và ràng buộc (Constraint)
  - Tác nhân kích hoạt biểu đồ (không phải là phần tử của biểu đồ này)

# Biểu đồ tuần tự - Đối tượng

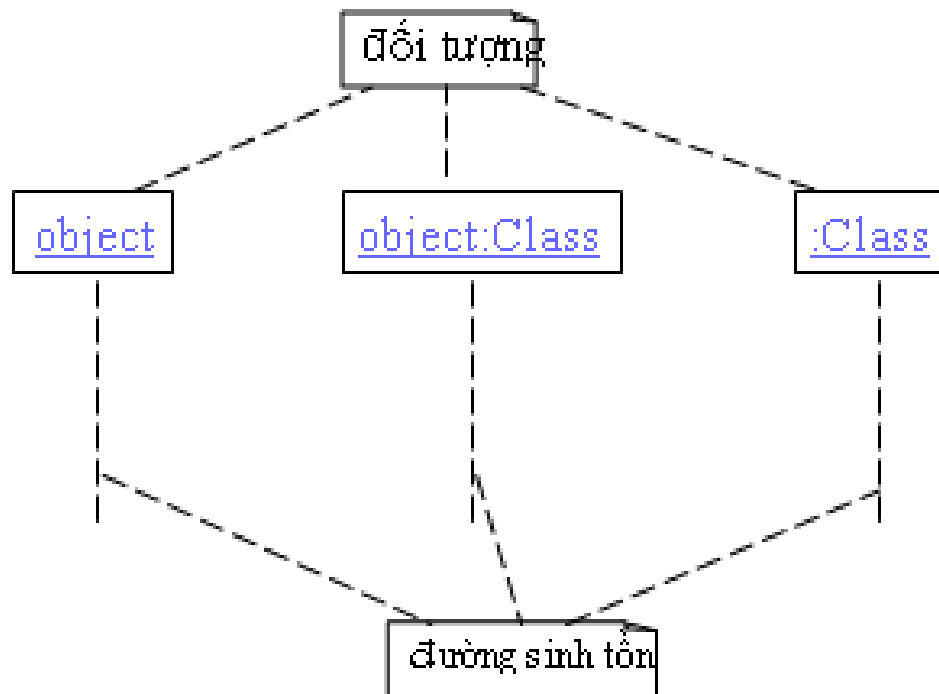
- **Đối tượng (Object)**

- Biểu diễn bởi hình chữ nhật, bên trong là tên đối tượng, có các cách viết tên đối tượng khác nhau
- Tham gia vào biểu đồ tuần tự bằng cách gọi/ nhận các thông điệp
- Được đặt trên đỉnh của biểu đồ



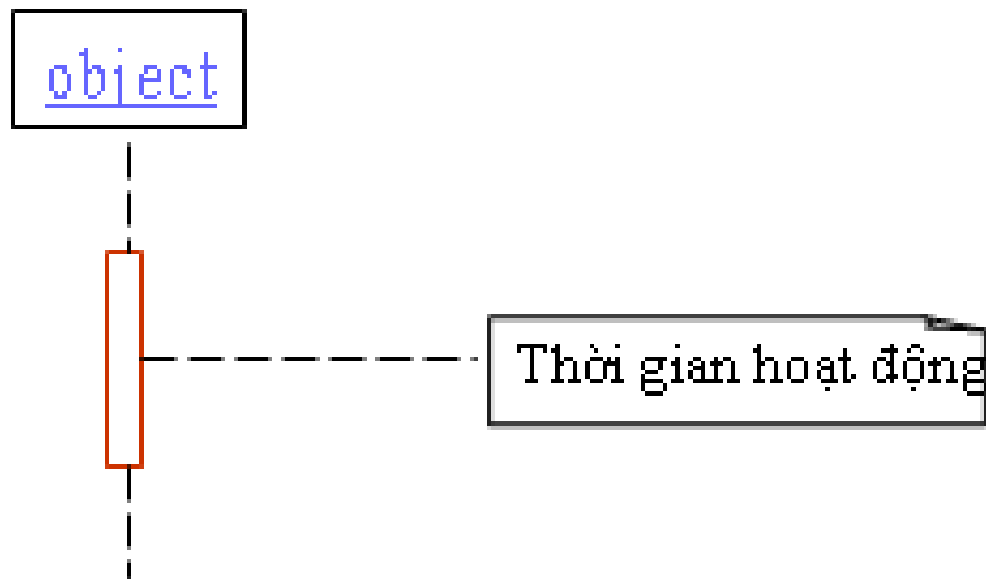
# Biểu đồ tuần tự - Đường sinh tồn

- Mỗi đối tượng có một đường sinh tồn (lifeline) biểu diễn thời gian tồn tại của nó.
- Kí hiệu



## Biểu đồ tuần tự (tt)

- Thời gian hoạt động (activation) là thời gian mà đối tượng đang thực hiện một thao tác



# Biểu đồ tuần tự (tt)

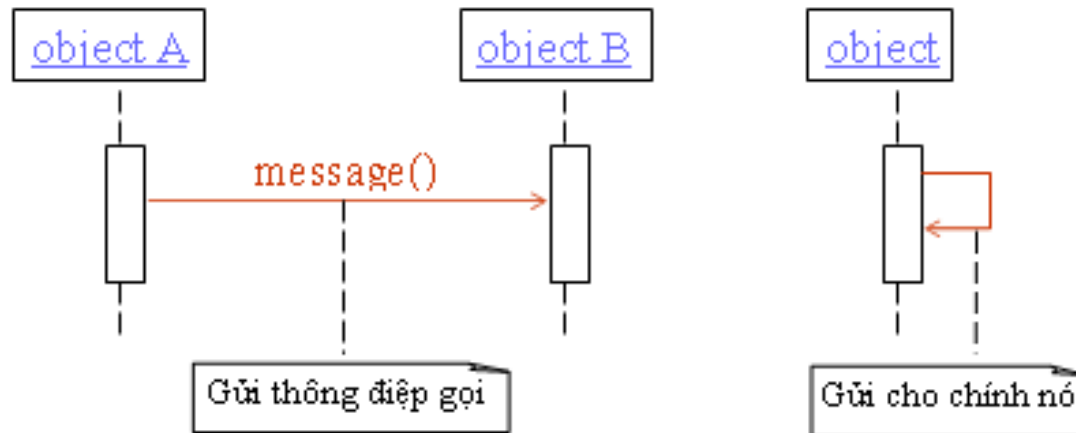
---

- Một thông điệp đặc tả trao đổi giữa các đối tượng
- Các loại thông điệp
  - Gọi (call)
  - Trả về (return)
  - Gửi (send)
  - Tạo (create)
  - Hủy (destroy)



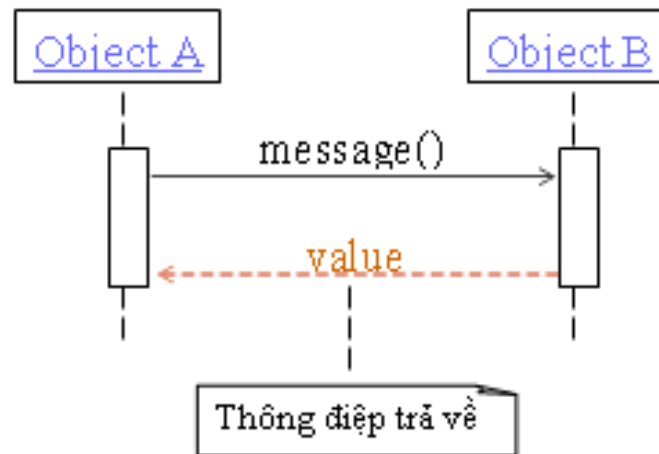
# Biểu đồ tuần tự (tt)

- **Thông điệp gọi** gọi một phương thức/thao tác trên đối tượng
  - đối tượng gọi phải đợi thông điệp được thực hiện kết thúc mới có thể thực hiện công việc khác (thông điệp đồng bộ)
- Một đối tượng có thể gửi thông điệp cho chính nó
- Kí hiệu



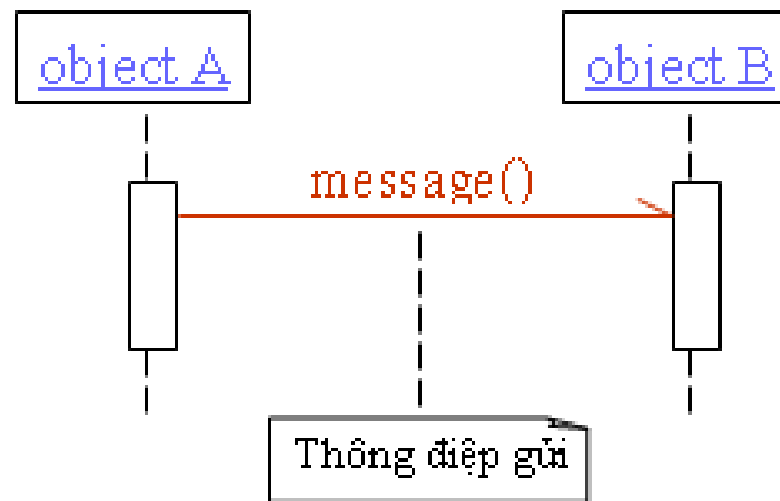
## Biểu đồ tuần tự (tt)

- **Thông điệp trả về** trả về một giá trị cho đối tượng gọi
- Kí hiệu



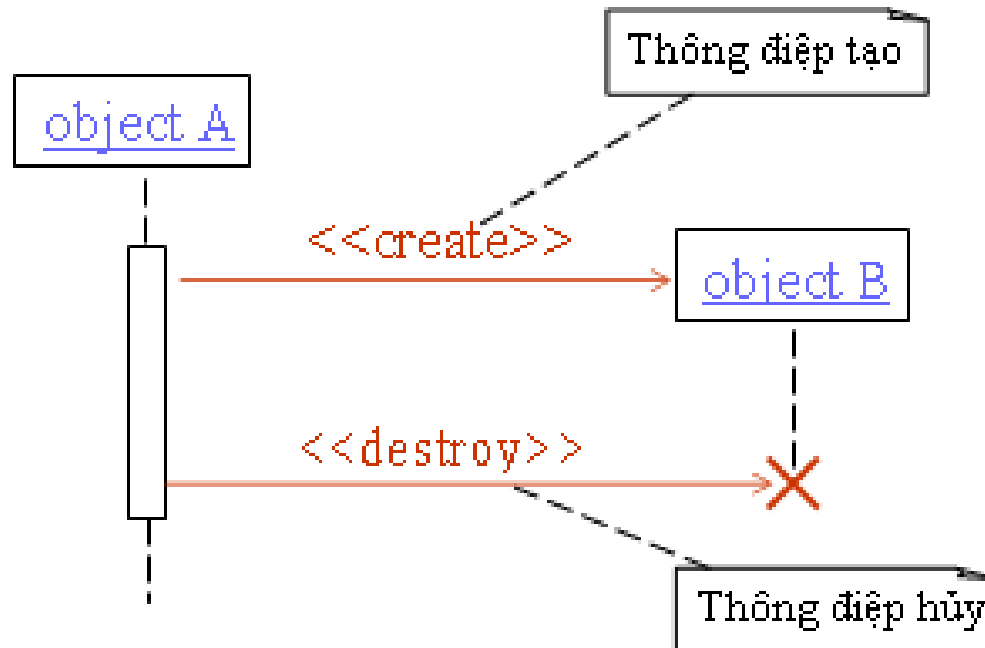
## Biểu đồ tuần tự (tt)

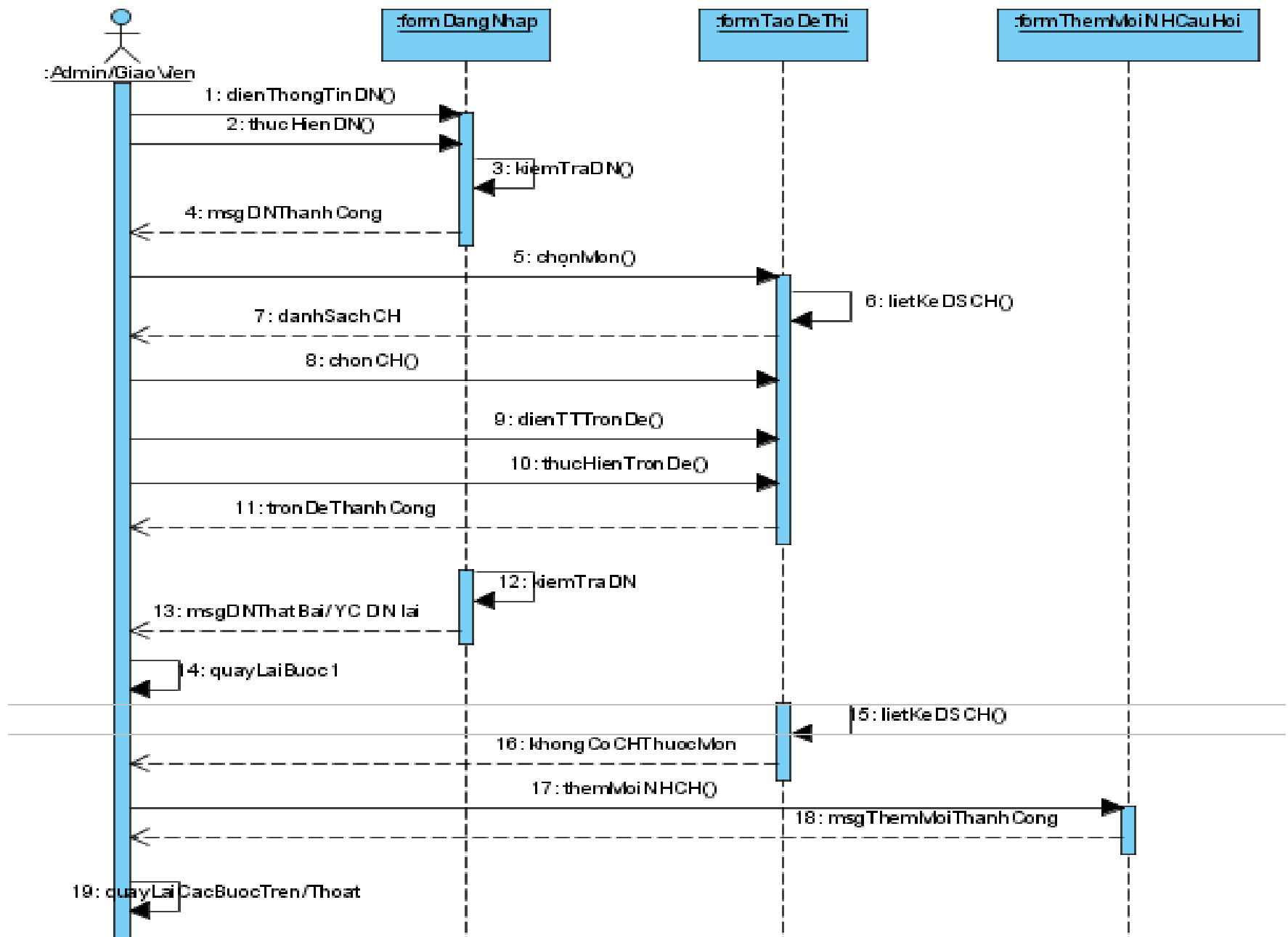
- **Thông điệp gửi** gửi một tín hiệu đến một đối tượng
- Khác với thông điệp gọi, khi đối tượng gửi thông điệp gửi nó không chờ đợi, mà tiếp tục thực hiện công việc khác (thông điệp không đồng bộ)
- Kí hiệu



## Biểu đồ tuần tự (tt)

- **Thông điệp tạo** gọi phương thức tạo một đối tượng
- **Thông điệp hủy** gọi phương thức hủy một đối tượng
- Kí hiệu





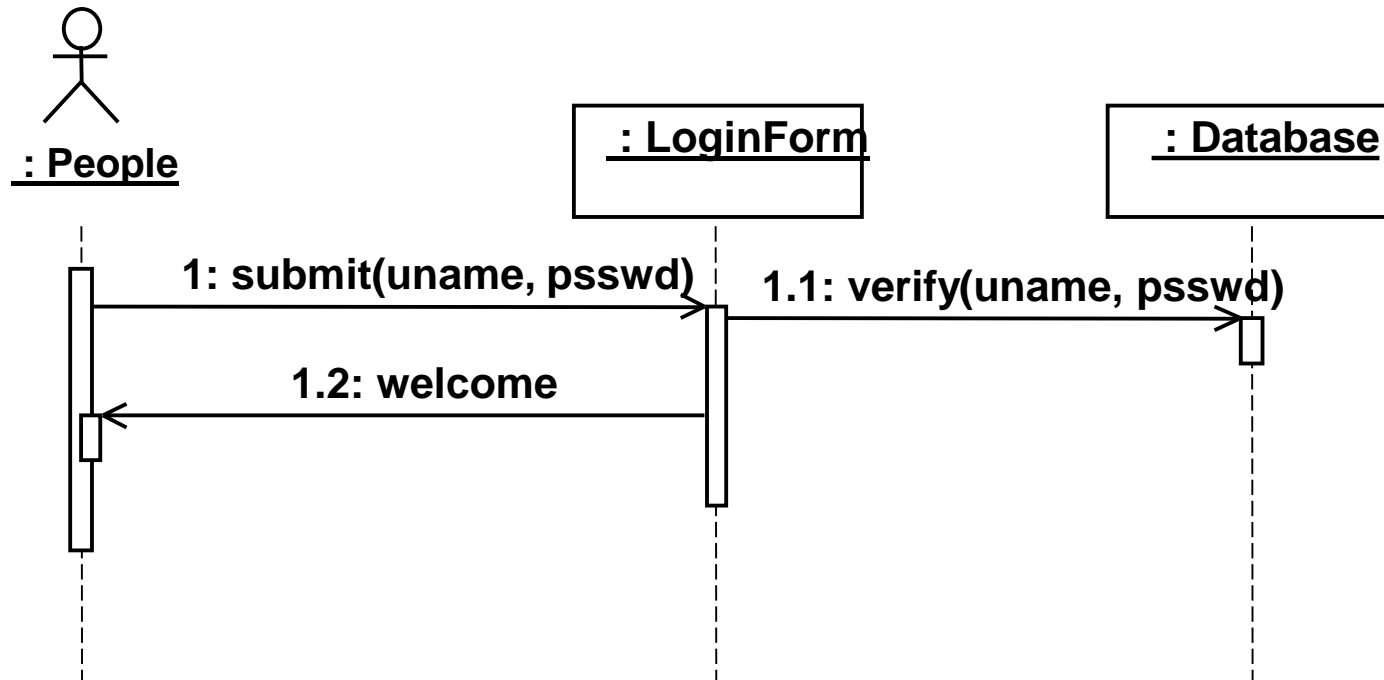
## Bài tập

---

- Ví dụ: Biểu đồ tuần tự cho use-case Login của hệ thống đăng ký môn học tín chỉ qua WEB.
- Luồng sự kiện:
  - SV chọn chức năng đăng nhập từ trang chính của hệ thống. Hệ thống hiển thị form đăng nhập. SV đăng nhập vào hệ thống bằng cách nhập username password. Hệ thống kiểm tra username và password trong csdl và nếu đăng nhập thành công, hệ thống mở màn hình welcome.

# Biểu đồ tuần tự - Ví dụ

- Ví dụ: Biểu đồ tuần tự cho use-case Login của hệ thống đăng ký môn học tín chỉ qua WEB. SV đăng nhập vào hệ thống bằng cách nhập username password



# Biểu đồ tuần tự - Thông điệp (tt)

- Cú pháp tổng quát của một thông điệp có dạng như sau:

– *Predecessor* *Guard-Condition* *Sequence-Expression*  
*Return-Value*  $:=$  *Message-Name*(*Argument-List*)

Thành phần  
bắt buộc

– *Predecessor*:

- Các thông điệp tiên quyết của một thông điệp, bao gồm một danh sách các số thứ tự cách nhau bởi dấu phẩy và kết thúc bằng một dấu chéo xuôi.
- Ví dụ: 2.1a, 2.1b / 2.2: drawLine(x1, y1, x2, y2)



# Biểu đồ tuần tự - Thông điệp (tt)

- *Predecessor*    *Guard-Condition*    *Sequence-Expression*  
*Return-Value*  $:=$  *Message-Name*(*Argument-List*)
  - *Guard-Condition*:
    - Điều kiện phải thỏa trước khi thông điệp được gửi đi
    - Ví dụ: [percentageComplete = 100]
  - *Sequence-Expression*:
    - Biểu thức tuần tự định nghĩa thứ tự xảy ra của các thông điệp. Nó là các mục thứ tự (Sequence-Term) theo sau là dấu hai chấm như 1:, 2:, 3:

# Biểu đồ tuần tự - Thông điệp (tt)

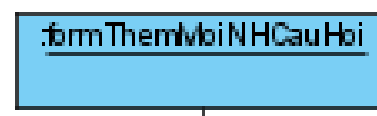
- *Predecessor Guard-Condition Sequence-Expression Return-Value* *:= Message-Name(Argument-List)*
  - *Return-Value*:
    - Giá trị trả về của thông điệp hay thao tác
    - Ví dụ: `journeyName := geJourneyName (i)`
  - *Message-Name*:
    - Tên thông điệp gọi đến đối tượng nhận
  - *Argument-List*:
    - Danh sách tham đối của thao tác
    - Ví dụ: `setArranged(dateArranged)`

## Biểu đồ tuần tự - ví dụ

- Xây dựng biểu đồ tuần tự cho use case “**Tạo đề thi**” trong hệ thống HTTN:
  - **Bước 1:** Xác định các **tác nhân** tham gia vào hệ thống:

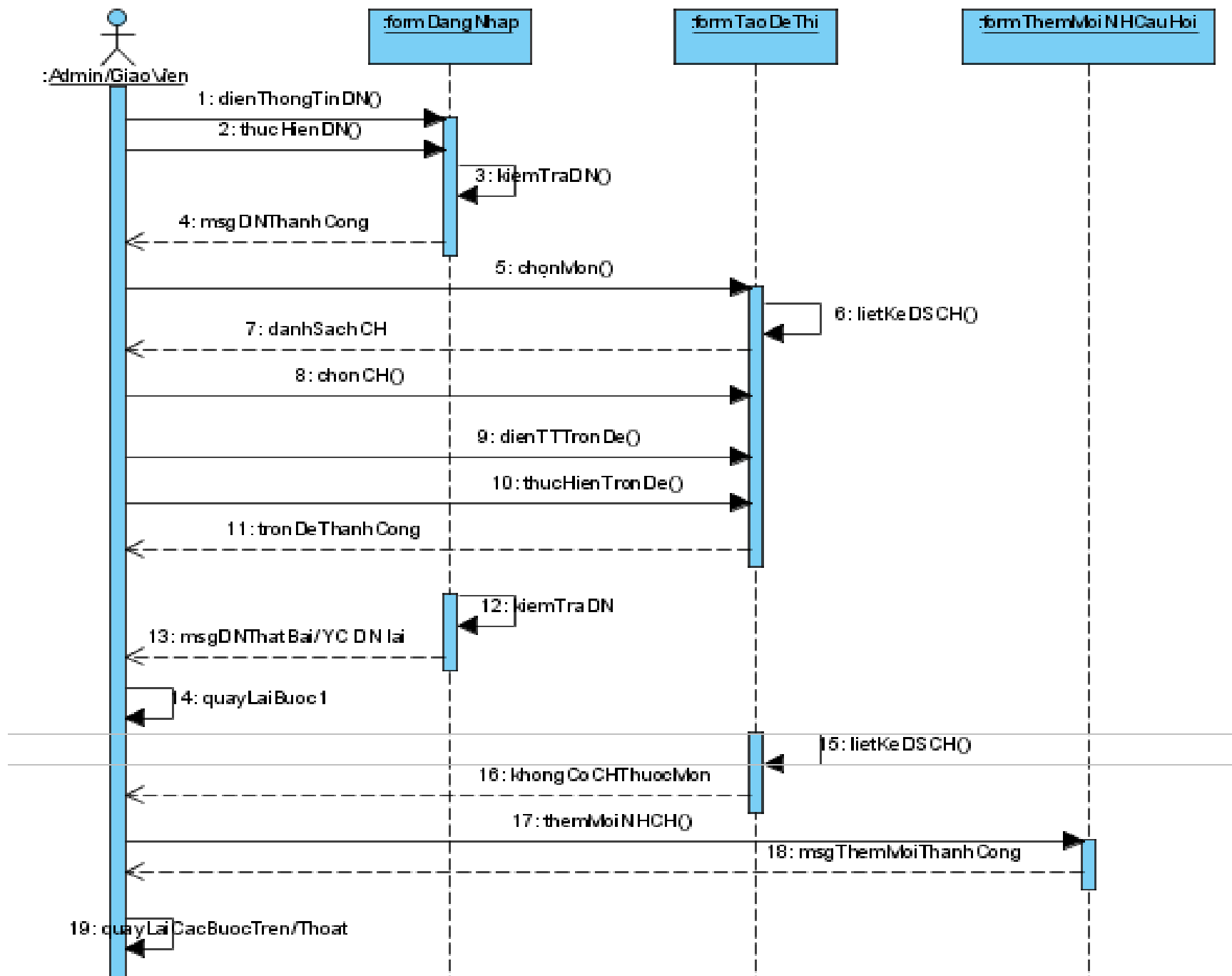


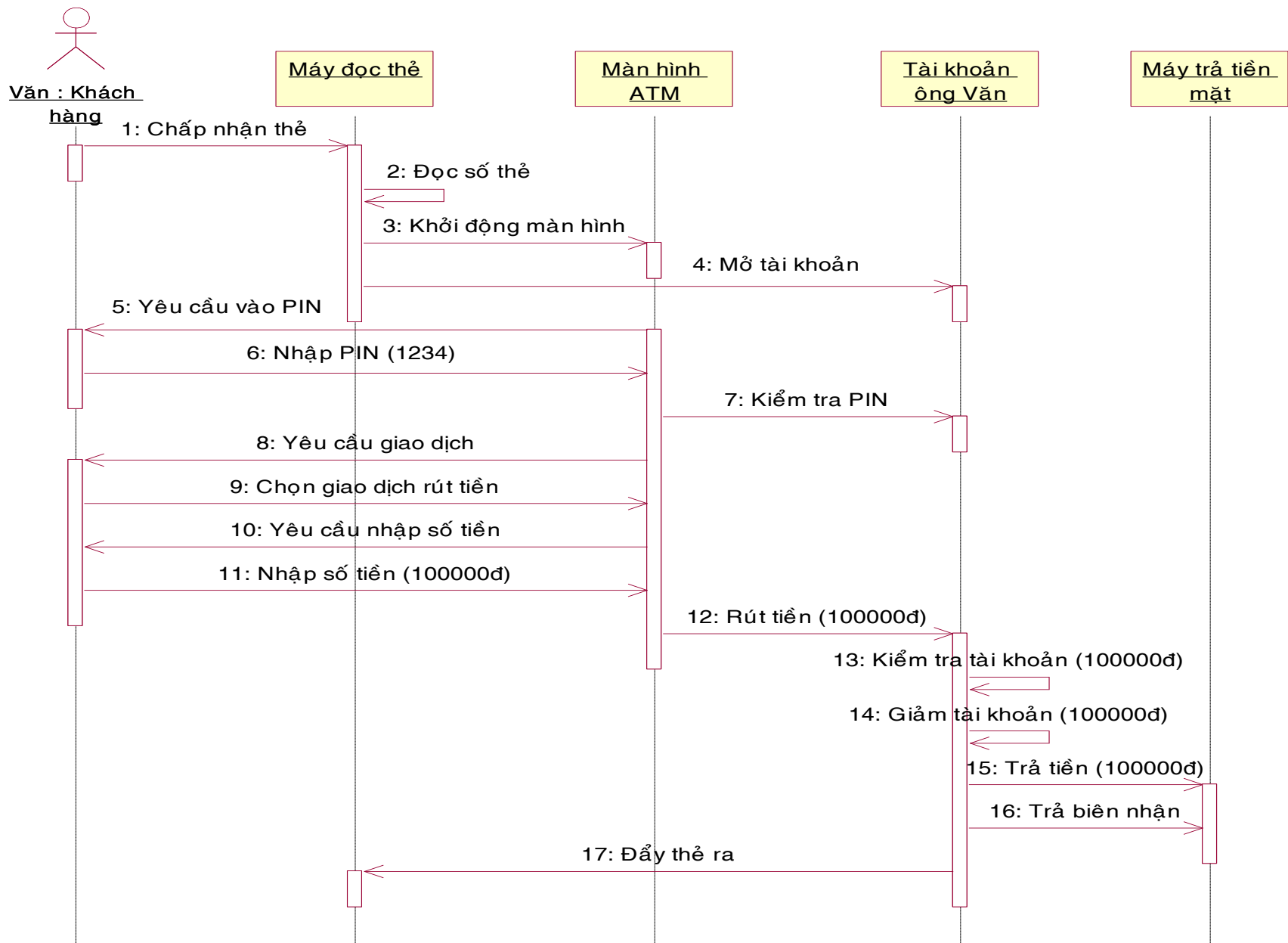
- **Bước 2:** Xác định **đối tượng** tham gia vào hệ thống:



## Biểu đồ tuần tự - ví dụ (tt)

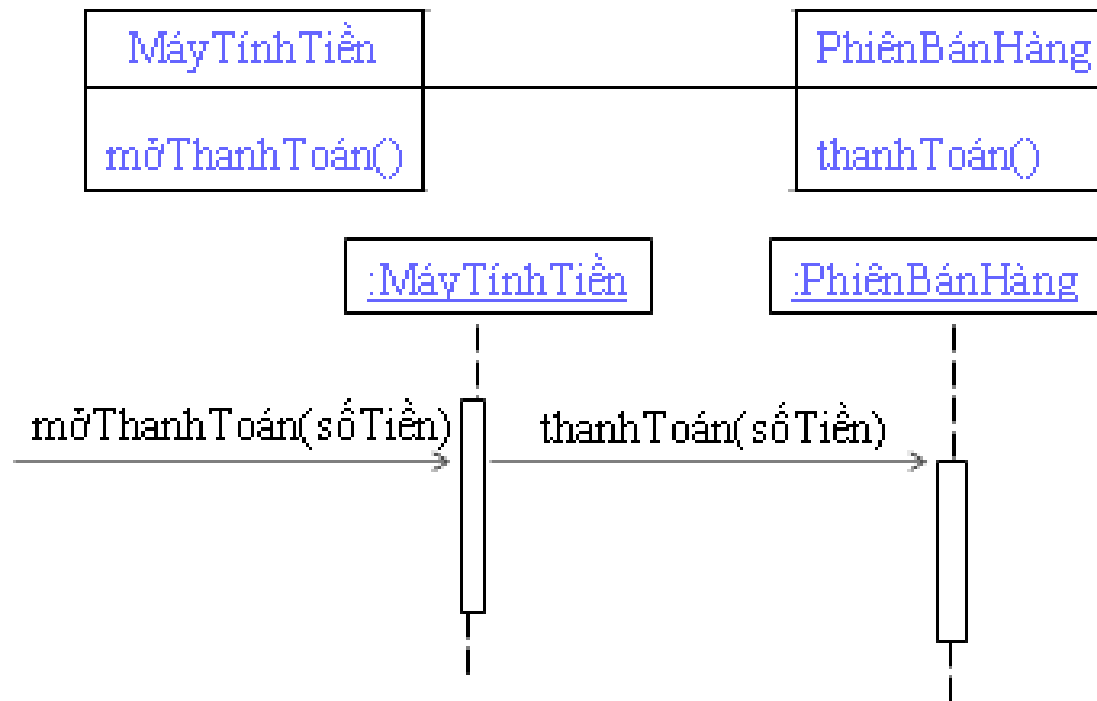
- **Bước 3:** Xác định các thông điệp chính:
  - dienTTDN(): điền thông tin đăng nhập
  - thuHienDN(): thực hiện danh nhập
  - chonMon(): chọn môn tạo đề
  - chonCH(): chọn câu hỏi
  - thucHienTronDe(): thực hiện trộn đề
  - themMoiCH(): thêm mới câu hỏi
- **Bước 4:** Xây dựng biểu đồ **Tuần tự**





## Biểu đồ tuần tự (tt)

- Giữa biểu đồ lớp và biểu đồ tương tác có mối quan hệ chặt chẽ với nhau
- Ví dụ:



# Biểu đồ cộng tác

- Biểu đồ cộng tác (collaboration diagram)
  - mô tả sự tương tác giữa các đối tượng bằng việc nhấn mạnh cấu trúc kết hợp giữa các đối tượng và những thông điệp trao đổi giữa chúng
- Biểu đồ cộng tác là sự mở rộng của biểu đồ đối tượng
- Biểu đồ cộng tác chỉ ra
  - thứ tự gửi các thông điệp: mỗi thông điệp được gán một số tuần tự
  - điều kiện gửi các thông điệp



# Biểu đồ cộng tác (tt)

- Cấu trúc thông điệp được mô tả tổng quát như sau:  
**precondition / condition sequence \* \*|| iteration : result := message(parameters)**
  - “precondition /”:
    - ds số tuần tự của các thông điệp trước thông điệp cần gửi.
  - “condition”:
    - thông điệp chỉ được gửi đi khi đk được thỏa mãn.
  - “sequence”:
    - số tuần tự của thông điệp cần gửi.

# Biểu đồ cộng tác (tt)

- “\*”:
  - thông điệp được gửi đi nhiều lần một cách tuần tự.
- “\*||”:
  - thông điệp được gửi đi nhiều lần một cách đồng thời.
- “iteration”:
  - số lần gửi thông điệp một cách tuần tự hoặc đồng thời
- “result”:
  - giá trị trả về của thông điệp.
- “message”:
  - tên thông điệp
- “parameters”:
  - ds các tham số của thông điệp.

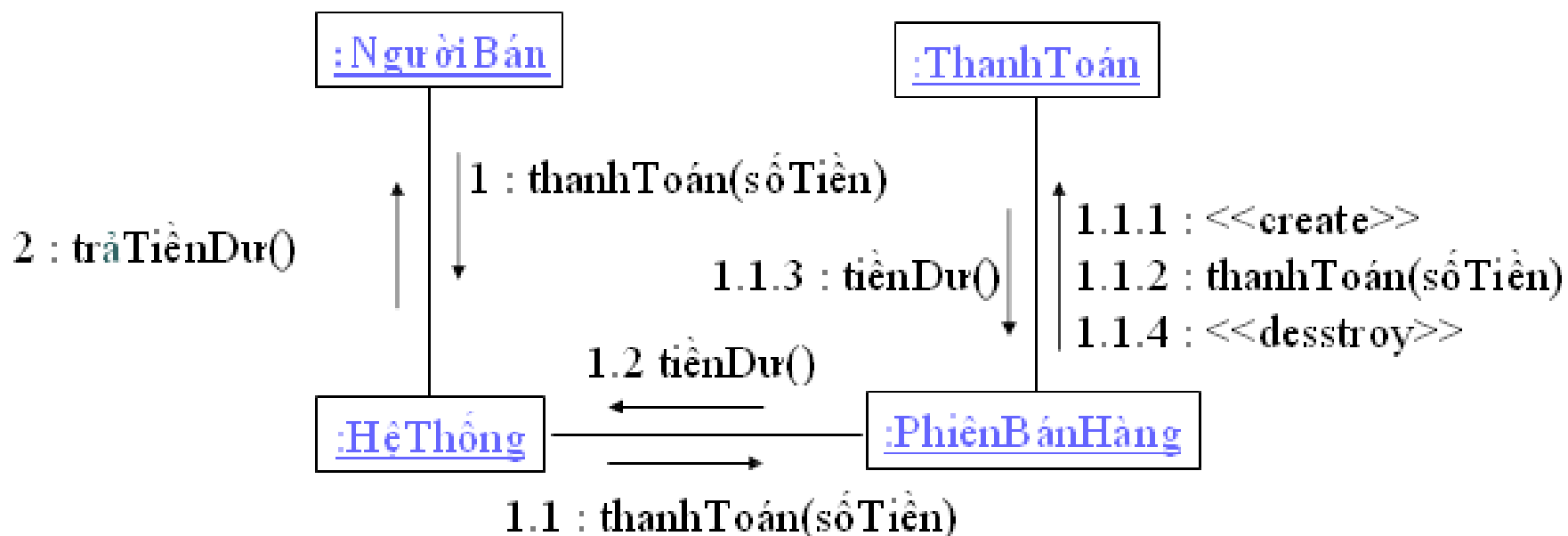
# Biểu đồ cộng tác (tt)

- Ví dụ:
  - **4 : hello()** :
    - thông điệp có số tuần tự là 4.
  - **[time = 12h] 1 : lunch()**:
    - thông điệp này chỉ được gửi đi lúc 12h.
  - **1.3.5 \* call()**:
    - thông điệp này được gửi đi nhiều lần.

# Biểu đồ cộng tác (tt)

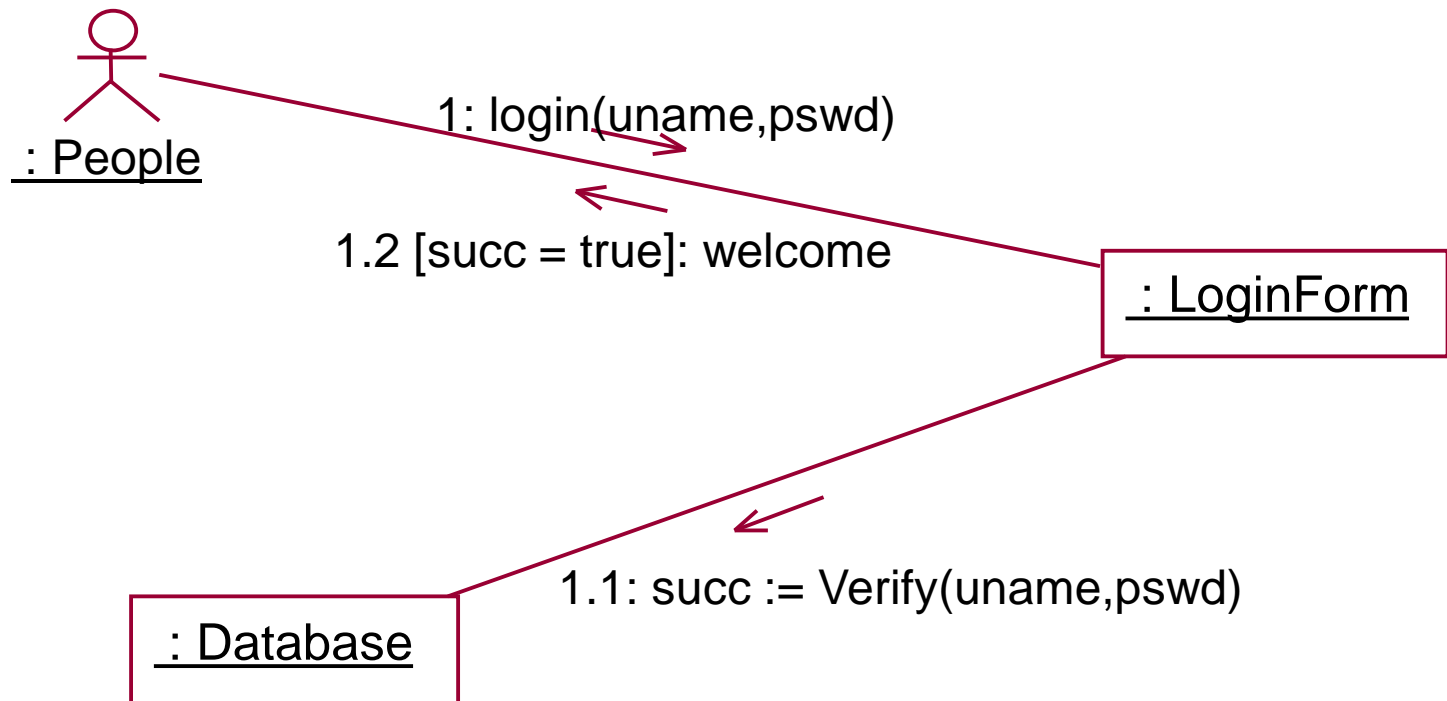
- Ví dụ: (tt)
  - **3 / \*|| [i:= 1..5] 1.2 : close():**
    - thông điệp này được gửi đi năm lần một cách đồng thời và sau thông điệp số 3.
  - **1.2, 2.3 / [t < 10] 3.1 name = getName():**
    - thông điệp này được gửi đi sau các thông điệp 1.2, 2.3 và với điều kiện  $t < 10$ .

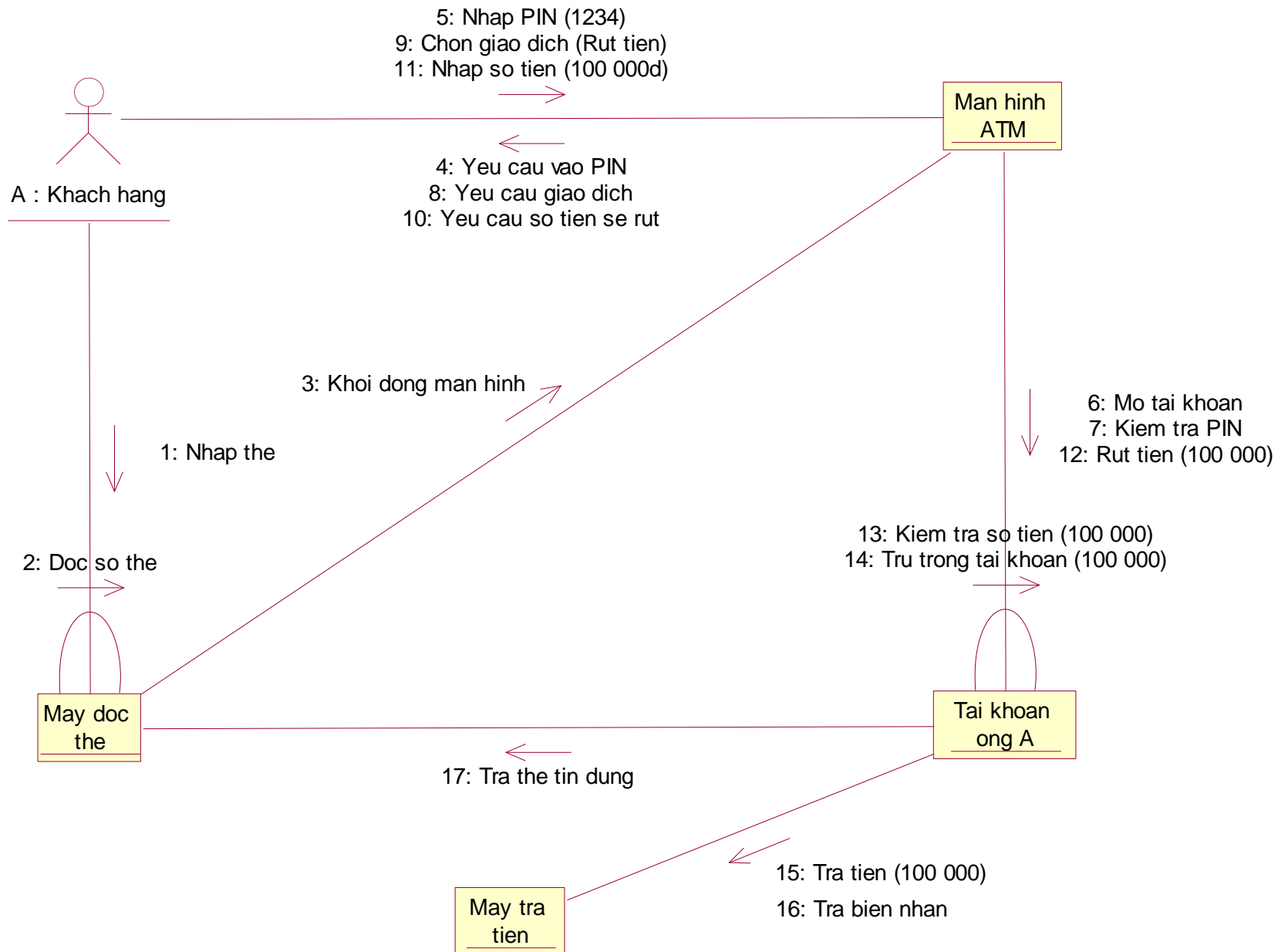
## Biểu đồ cộng tác (tt)



## Biểu đồ cộng tác – Ví dụ

- Ví dụ: Biểu đồ cộng tác cho *use-case* Login của hệ thống đăng ký môn học tín chỉ qua WEB





# Biểu đồ trạng thái

---

- Chuẩn UML đưa ra Biểu đồ trạng thái để biểu diễn hành vi của một phần tử bất kỳ bằng cách chỉ ra đáp ứng của nó đối với các sự kiện bên ngoài
- Thông thường Biểu đồ trạng thái được áp dụng cho đối tượng/lớp → biểu diễn hành vi của lớp

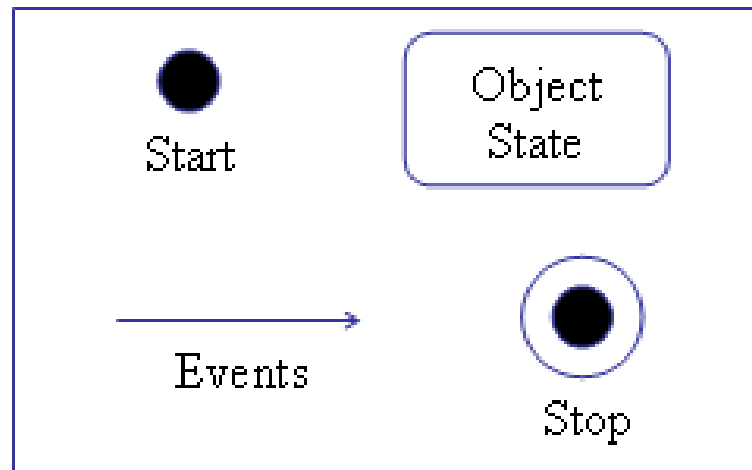


# Biểu đồ trạng thái (tt)

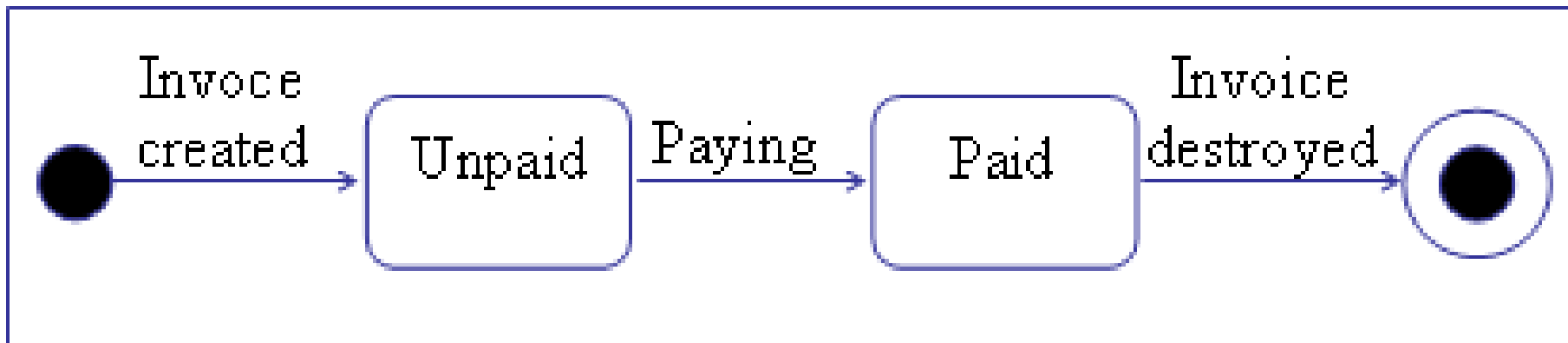
- Trạng thái và sự biến đổi trạng thái:
  - Trạng thái là một kết quả của các hoạt động trước đó đã được đối tượng thực hiện
  - Được xác định qua giá trị các thuộc tính cũng như các nối kết của đối tượng với các đối tượng khác.
    - Hóa đơn (đối tượng) đã được trả tiền (trạng thái).
    - Chiếc xe ô tô (đối tượng) đang đứng yên (trạng thái).
  - Đối tượng thay đổi trạng thái khi có sự kiện xảy ra:
    - VD: có ai đó trả tiền cho hóa đơn, bật động cơ xe ô tô hay là lấy chồng lấy vợ.

# Biểu đồ trạng thái (tt)

- Biểu đồ trạng thái thể hiện những khía cạnh quan tâm khi xem xét trạng thái của một đối tượng:
  - Trạng thái ban đầu
  - Một số trạng thái ở giữa
  - Một hoặc nhiều trạng thái kết thúc
  - Sự biến đổi giữa các trạng thái



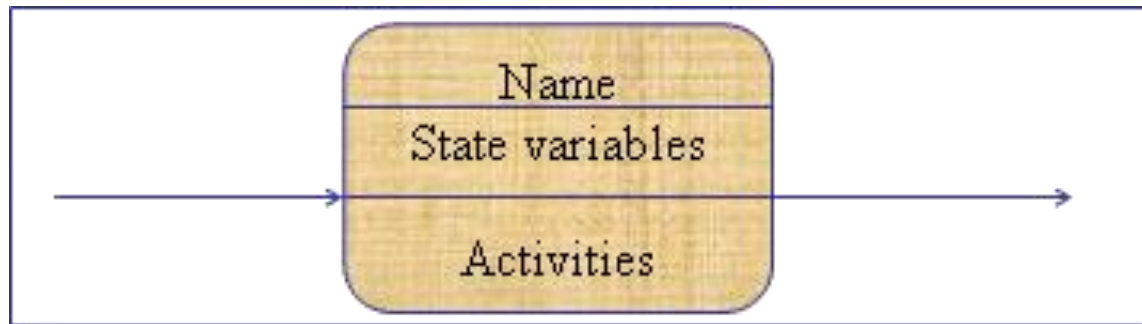
## Biểu đồ trạng thái (tt)



**Biểu đồ trạng thái thực hiện hóa đơn**

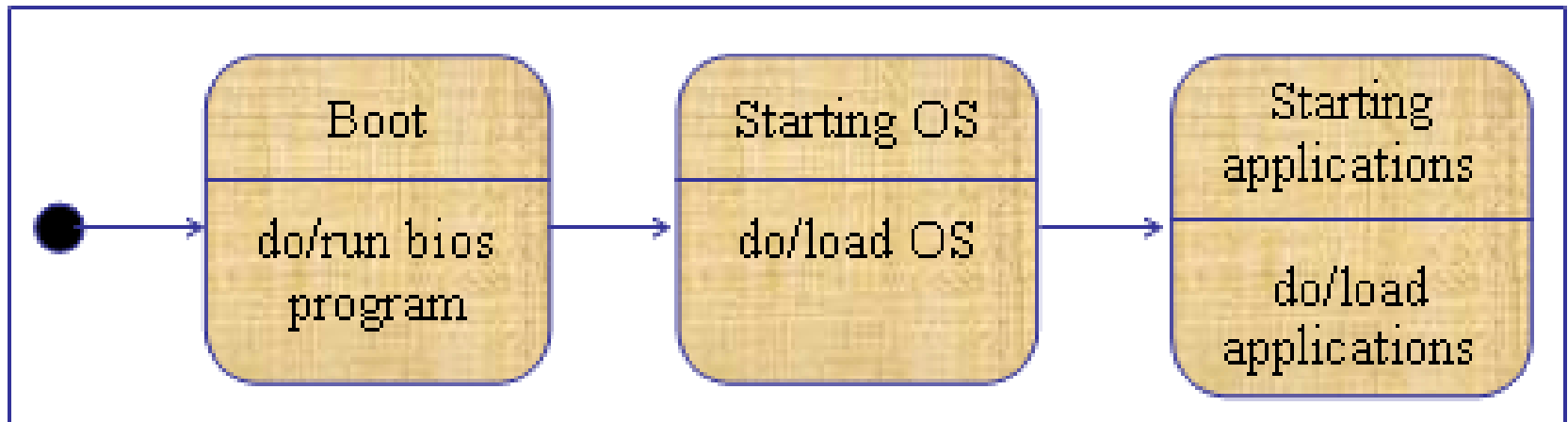
# Biểu đồ trạng thái (tt)

- Một trạng thái có thể có ba thành phần, như hình sau:



- Name:**
  - chỉ ra tên của trạng thái
- State variables (không bắt buộc):**
  - dành cho các biến trạng thái.
- Activities (không bắt buộc):**
  - phần dành cho hoạt động

## Biểu đồ trạng thái (tt)



- Biến đổi trạng thái không có sự kiện từ ngoài. Sự thay đổi trạng thái xảy ra khi các hoạt động trong mỗi trạng thái được thực hiện xong.

## Biểu đồ trạng thái (tt)

- **Nhận biết trạng thái và sự kiện:** Hỏi một số các câu hỏi thích hợp:
  - Một đối tượng có thể có những trạng thái nào?
  - Những sự kiện nào có thể xảy ra?
  - Trạng thái mới sẽ là gì?
  - Có những thủ tục nào sẽ được thực thi?:
  - Chuỗi tương tác giữa các đối tượng là gì?

## Biểu đồ trạng thái (tt)

---

- Qui định nào sẽ được áp dụng cho các phản ứng của các đối tượng với nhau?
- Những sự kiện và sự chuyển tải nào là không thể xảy ra?
- Cái gì khiến cho một đối tượng được tạo ra?
- Cái gì khiến cho một đối tượng bị hủy?
- Cái gì khiến cho đối tượng cần phải được tái phân loại (reclassified)?

# Biểu đồ hoạt động

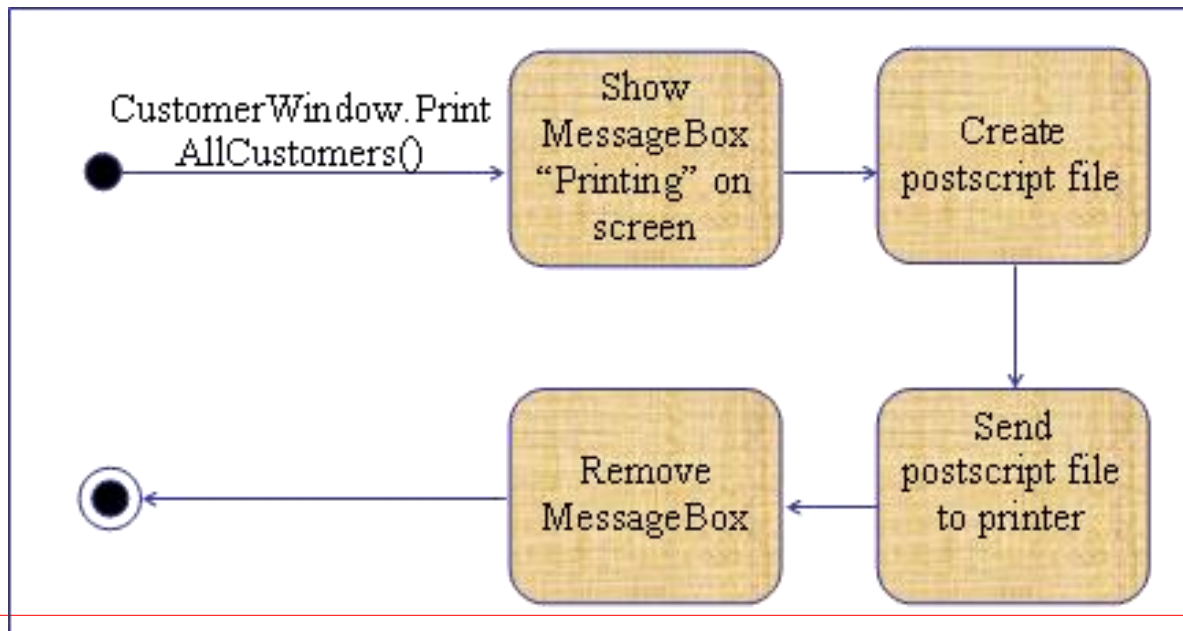
- Biểu đồ hoạt động (activity diagram) là một biến thể của biểu đồ trạng thái trong đó **trạng thái là sự thực thi một hành động và sự dịch chuyển** được kích hoạt khi hành động hoàn tất
- Được dùng để mô tả một thủ tục hay thuật giải → tập trung vào các hành động
- Mỗi hành động được ký hiệu bằng hình vẽ như sau





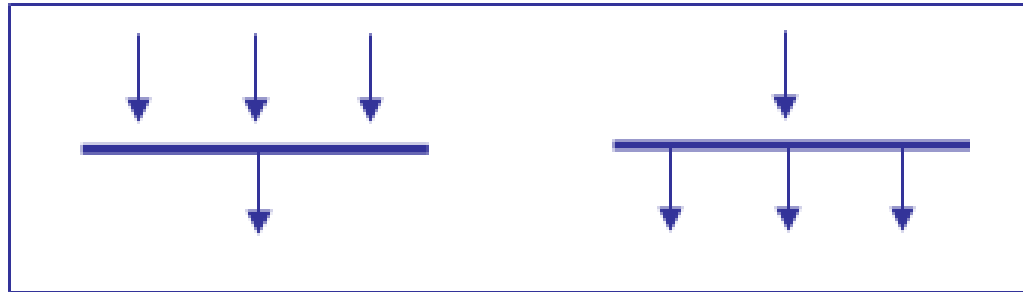
# Biểu đồ hoạt động (tt)

- Biểu đồ hoạt động
  - chỉ ra các hành động cùng mối quan hệ giữa chúng
  - có thể có một điểm bắt đầu và một điểm kết thúc.
  - sử dụng những ký hiệu như biểu đồ trạng thái bình thường.



# Biểu đồ hoạt động (tt)

- **Thanh đồng bộ hóa (Synchronisation bar):**
  - cho phép mở ra hoặc đóng lại các nhánh chạy song song nội bộ trong tiến trình.



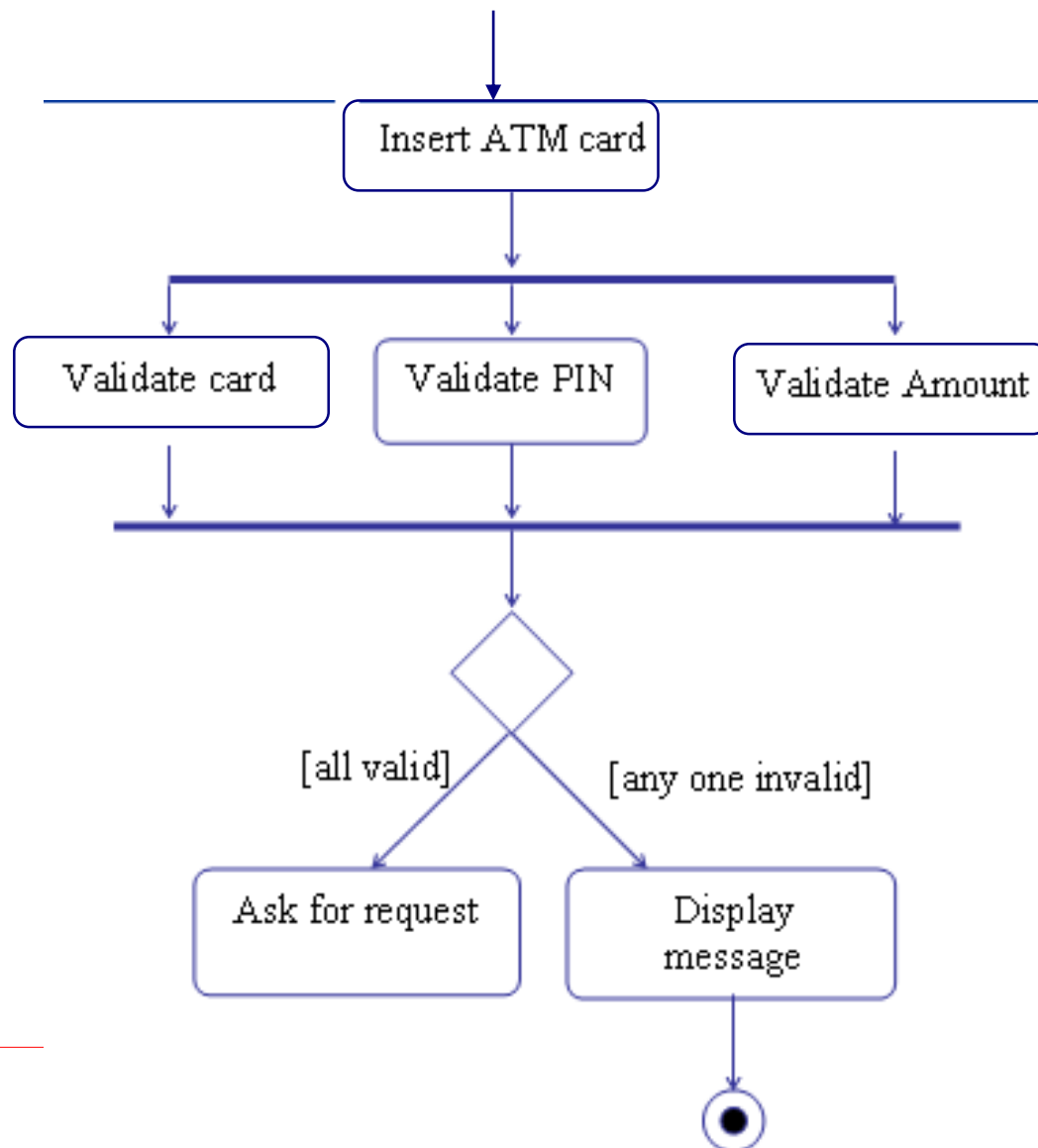
- **Điều kiện canh giữ (Guard Condition):**
  - các biểu thức logic có giá trị hoặc đúng hoặc sai. Điều kiện canh giữ được thể hiện trong ngoặc vuông

# Biểu đồ hoạt động (tt)

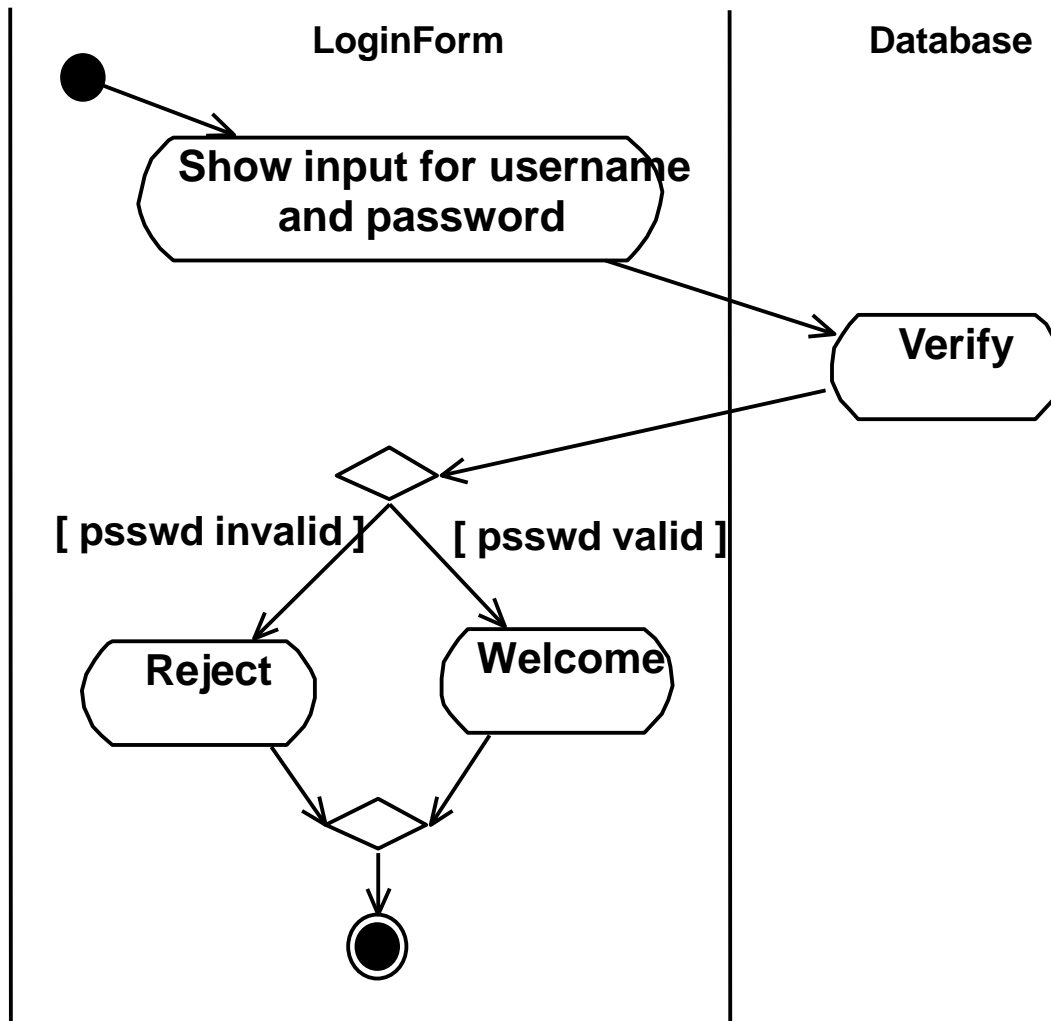
---

- **Quyết định rẽ nhánh:**
  - hình thoi có một đường vào và nhiều nhánh ra, mỗi nhánh được gán một guard-condition (Các nhánh ra được nhập lại bằng một hình thoi khác)
- **Mỗi “đường bơi” (swimlane):**
  - đại diện một lớp hoặc một actor
- **Điểm quyết định (Decision Point):**
  - dùng để chỉ ra các thay đổi khả thi. Kí hiệu hình thoi.

# Biểu đồ hoạt động (tt)



# Biểu đồ hoạt động – ví dụ



Biểu đồ hoạt động cho tác vụ submit của LoginForm

# Khi nào thì sử dụng biểu đồ nào?

- **Biểu đồ tuần tự và biểu đồ cộng tác:**
  - được dùng khi muốn xem xét ứng xử động của nhiều đối tượng/ lớp trong nội bộ một cảnh kịch của một trường hợp sử dụng.
  - rất hữu dụng trong việc chỉ ra sự cộng tác giữa các đối tượng, nhưng chúng lại không hữu dụng khi muốn miêu tả ứng xử chính xác của một đối tượng.

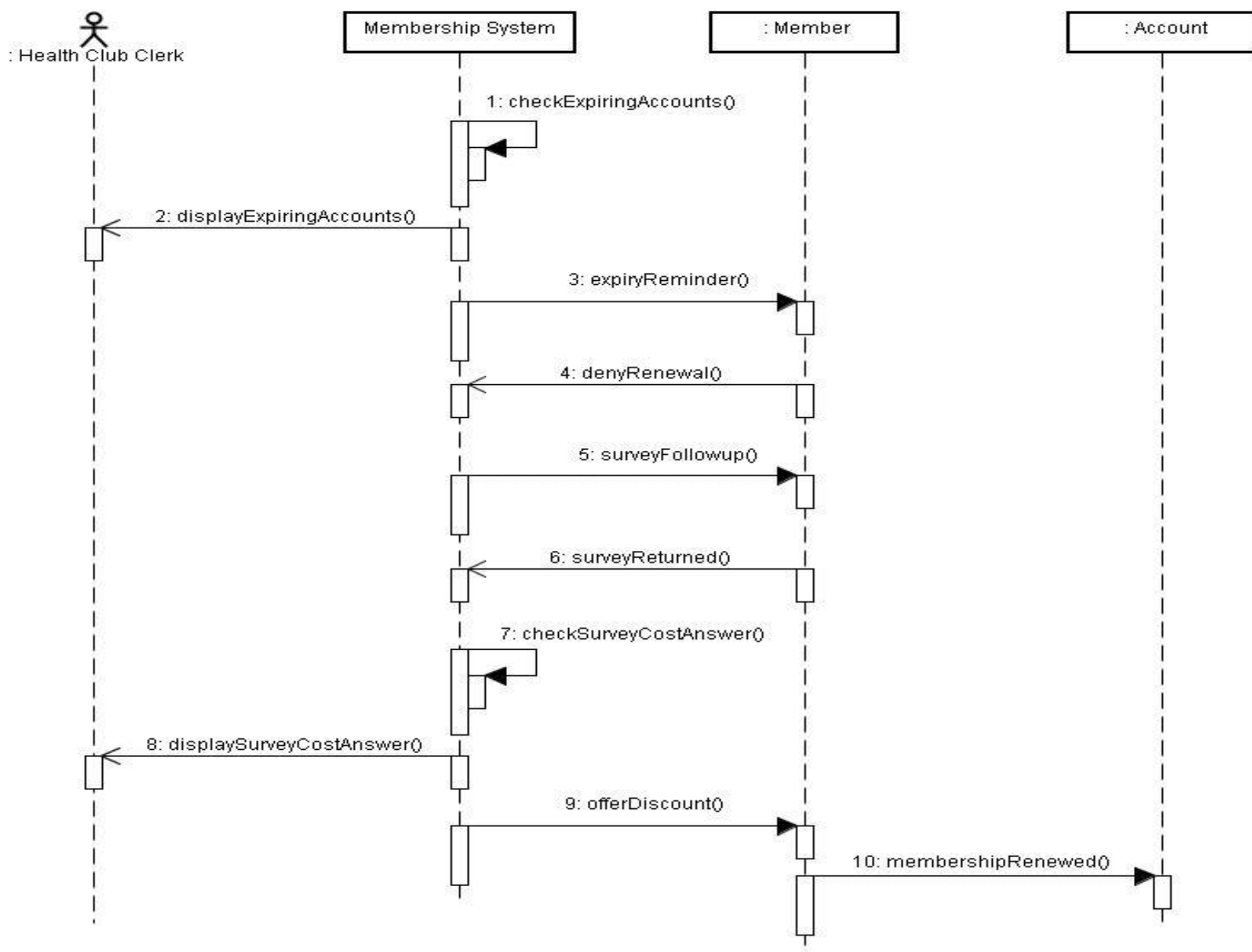
# Khi nào thì sử dụng biểu đồ nào? (tt)

- **Biểu đồ trạng thái:**
  - được dùng để thể hiện ứng xử chính xác của một đối tượng.
- **Biểu đồ hoạt động:**
  - được dùng để thể hiện lối ứng xử xuyên suốt nhiều trường hợp sử dụng hoặc các tiểu trình xảy ra song song của một lần thực thi.

Create a sequence diagram for the following scenario description for a health club membership system.

When members join the health club, they pay a fee for a certain length of time. The club wants to mail out reminder letters to members asking them to renew their memberships one month before they expire. About half of the members do not renew their memberships. These members are sent follow-up surveys to complete about why they decided not to renew so that the club can learn how to increase retention. If the member did not renew because of cost, a special discount is offered to that customer. Typically, 25 % of accounts are reactivated because of this offer.





Create a collaboration diagram for the following scenario description for a health club membership system.

When members join the health club, they pay a fee for a certain length of time. The club wants to mail out reminder letters to members asking them to renew their memberships one month before they expire. About half of the members do not renew their memberships. These members are sent follow-up surveys to complete about why they decided not to renew so that the club can learn how to increase retention. If the member did not renew because of cost, a special discount is offered to that customer. Typically, 25 % of accounts are reactivated because of this offer.

Every time a member enters the club, an attendant takes his or her card and scans it to make sure the person is an active member. If the member is not activate, the system presents the amount of money it costs to renew the membership. The customer is given the chance to pay the fee and use the club, and the system makes note of the reactivation of the account so that special attention can be given to this customer when the next round of renewal notices are dispensed.