

# Phân tích thiết kế hướng đối tượng

## Bài 9: Các sơ đồ tương tác

**TS. Nguyễn Hiếu Cường**

Bộ môn CNPM, Khoa CNTT, Trường ĐH GTVT

Email: [cuonggt@gmail.com](mailto:cuonggt@gmail.com)

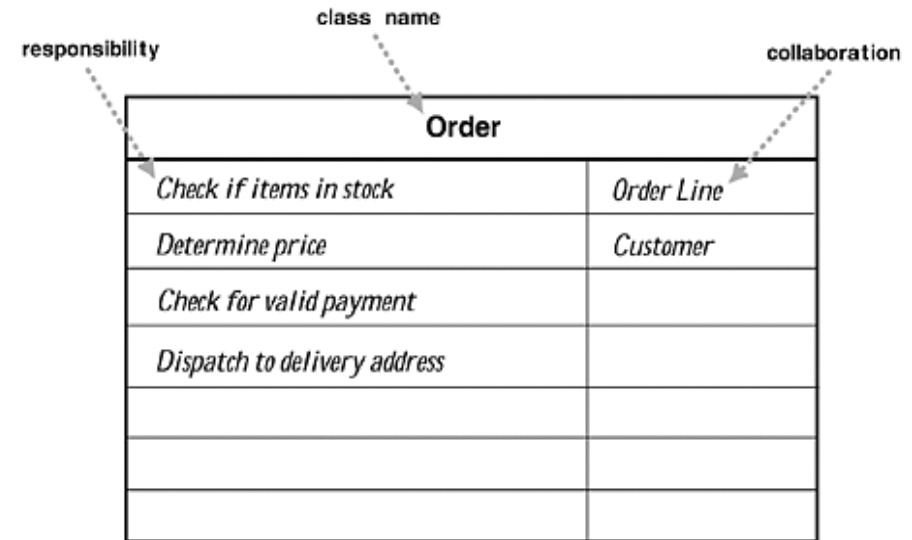
# Nội dung

---

- Hiểu vai trò và các ký hiệu của các sơ đồ tương tác trong OOAD
- Biết cách lựa chọn sơ đồ phù hợp
  - Sơ đồ Trình tự (Sequence diagram)
  - Sơ đồ Giao tiếp (Communication diagram)
- Biết cách chuyển từ use case/kịch bản sang sơ đồ trình tự
- Có thể thực hành với các case study

# Thẻ CRC

- Thẻ CRC có thể cho biết:
  - Trách nhiệm của các lớp
  - Quan hệ giữa các lớp
- Không cho biết những gì?



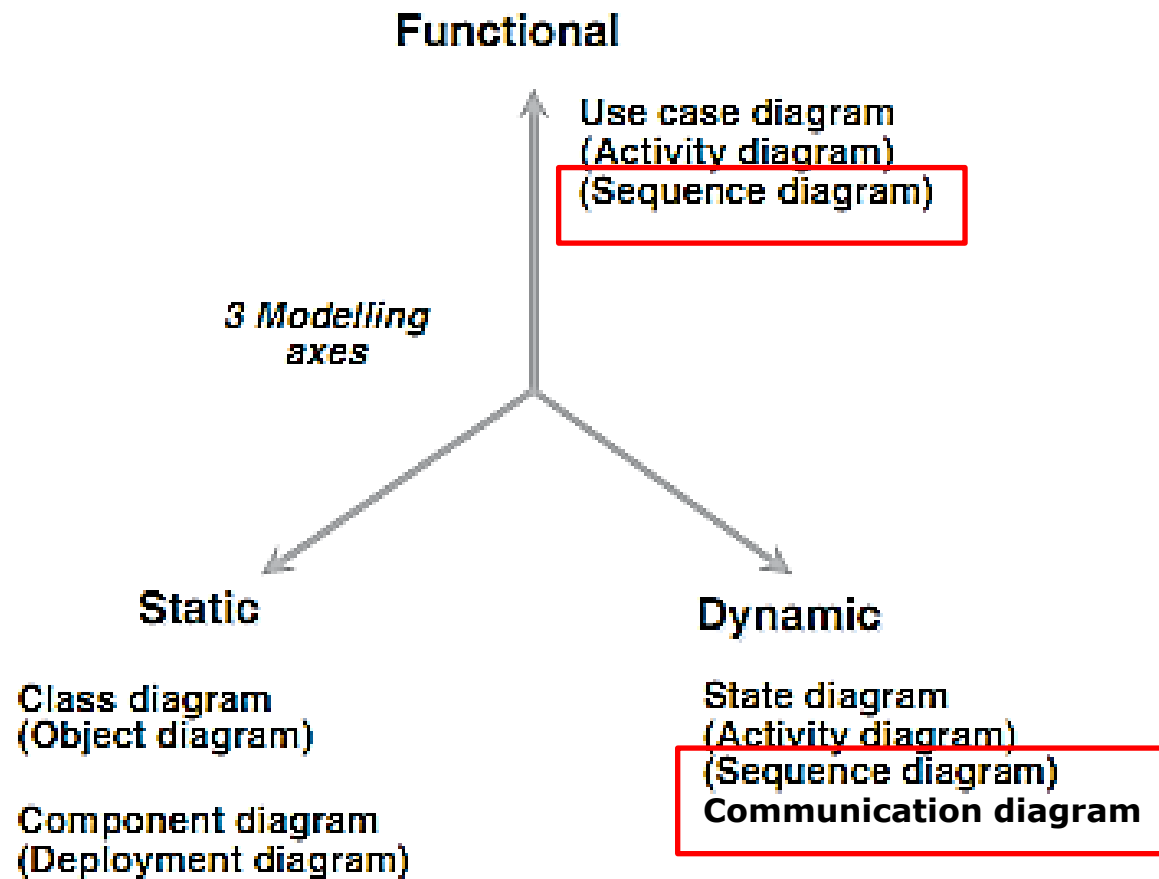
*Trình tự tương tác giữa các đối tượng*

# Tương tác trong OOAD là gì?


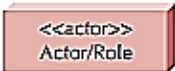
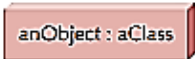


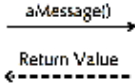

---

- Tương tác:
  - Các đối tượng hợp tác để hoàn thành một use case
  - Thể hiện bằng trao đổi thông điệp (messages)  
→ Giúp kiểm tra lại phân công trách nhiệm lớp
- Đặc điểm các sơ đồ tương tác
  - Sơ đồ trình tự: nhấn mạnh *trình tự thời gian*
  - Sơ đồ giao tiếp [hoặc Collaboration in UML 1.x]: nhấn mạnh cấu trúc *cộng tác giữa các đối tượng*

# Các loại sơ đồ



# Các ký hiệu

<b>An Actor:</b> <ul style="list-style-type: none"> <li>Is a person or system that derives benefit from and is external to the system</li> <li>Participates in a sequence by sending and/or receiving messages</li> <li>Are placed across the top of the diagram</li> <li>Is depicted as either a stick figure (default) or if a non-human actor is involved, as a rectangle with &lt;&lt;actor&gt;&gt; in it (alternative)</li> </ul>	 
<b>An Object:</b> <ul style="list-style-type: none"> <li>Participates in a sequence by sending and/or receiving messages</li> <li>Are placed across the top of the diagram</li> </ul>	
<b>A Lifeline:</b> <ul style="list-style-type: none"> <li>Denotes the life of an object during a sequence</li> <li>Contains an "X" at the point at which the class no longer interacts</li> </ul>	
<b>An Execution Occurrence:</b> <ul style="list-style-type: none"> <li>Is a long narrow rectangle placed atop a lifeline</li> <li>Denotes when an object is sending or receiving messages</li> </ul>	
<b>A Message:</b> <ul style="list-style-type: none"> <li>Conveys information from one object to another one</li> <li>An operation call is labeled with the message being sent and a solid arrow, while a return is labeled with the value being returned and shown as a dashed arrow</li> </ul>	
<b>Object Destruction:</b> <ul style="list-style-type: none"> <li>An X is placed at the end of an object's lifeline to show that it is going out of existence</li> </ul>	

**Actor**

**Object**

**Lifeline:** đường chấm dọc

**Activation bar:** hình chữ nhật mỏng

**Message:** mũi tên, có nhãn là lời gọi phương thức

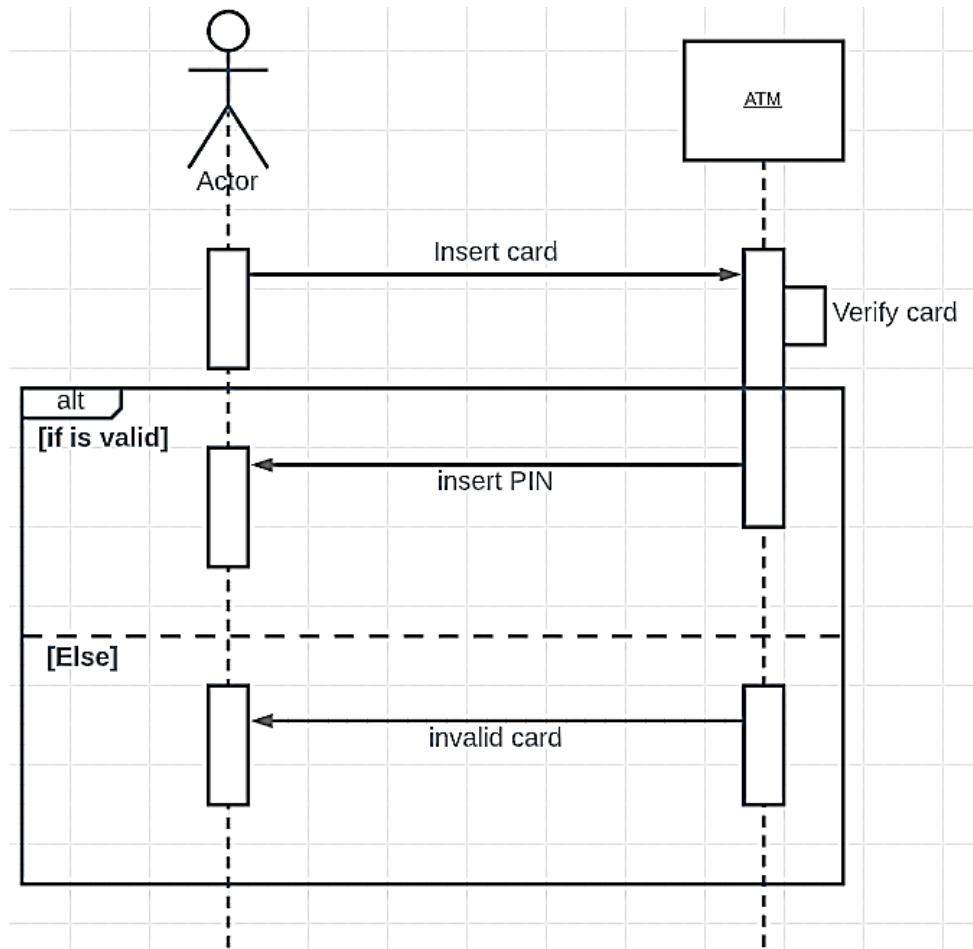
**Return message:** mũi tên nét đứt

# Xây dựng sơ đồ trình tự

---

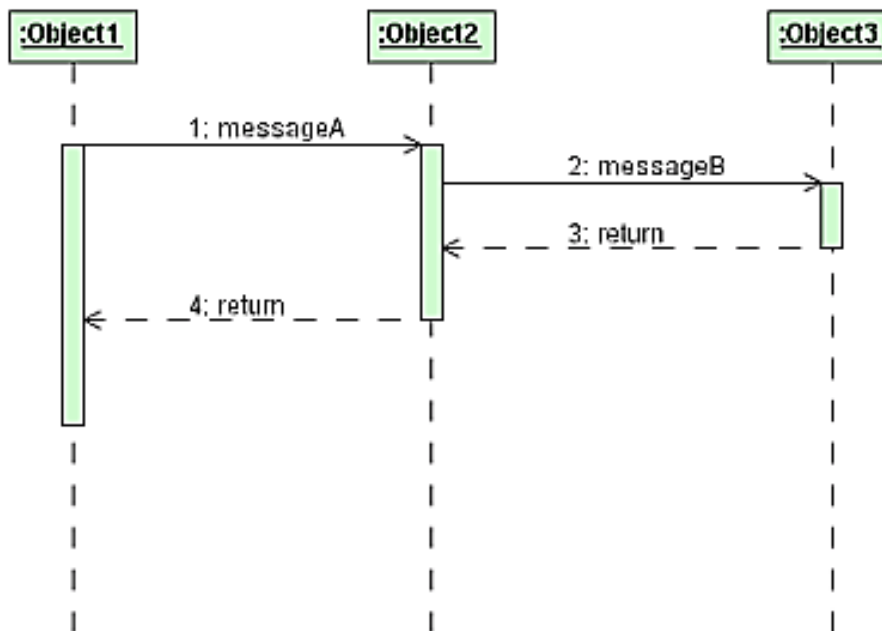
1. Xác định ngữ cảnh (là một **kịch bản** trong một **ca sử dụng**)
2. Nhận diện các tác nhân và đối tượng tham gia
3. Thiết lập đường sống (lifeline) cho từng đối tượng
4. Dựa vào kịch bản, viết các thông điệp (message) tương tác giữa các đối tượng và có thể có kết quả trả về (return line)
5. Có thể thêm các kích hoạt (activation bar) cho các đối tượng

# Ví dụ (sơ đồ trình tự hệ thống)





# Ví dụ (sơ đồ trình tự)



**Use case:** Issue bike  
**Actors:** Receptionist  
**Goal:** To hire out a bike

**Overview:**

When a customer comes into the shop they choose a bike to hire. The Receptionist looks up the bike on the system and tells the customer how much it will cost to hire the bike for a specified period. The customer pays, is issued with a receipt, then leaves with the bike.

**Cross-reference:**

R3, R4, R5, R6, R7, R8, R9, R10

**Typical course of events:**

Actor action	System response
1 The customer chooses a bike	
2 The Receptionist keys in the bike number	3 Displays the bike details including the daily hire rate and deposit
4 Customer specifies length of hire	
5 Receptionist keys this in	6 Displays total hire cost
7 Customer agrees the price	
8 Receptionist keys in the customer details	9 Displays customer details
10 Customer pays the total cost	
11 Receptionist records amount paid	12 Prints a receipt

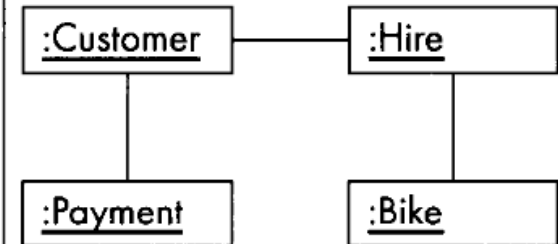
Kịch bản thuê xe  
thành công

# Ví dụ cụ thể của kịch bản

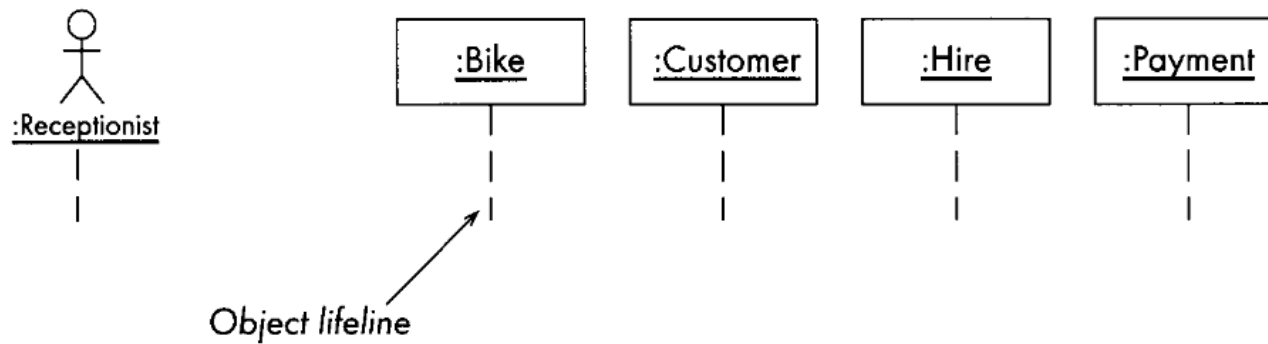
- Stephanie arrives at the shop at 9.00am one Saturday and chooses a mountain bike
- Annie sees that its number is 468
- Annie enters this number into the system
- The system confirms that this is a woman's mountain bike and displays the daily rate (£2) and the deposit (£60)
- Stephanie says she wants to hire the bike for a week
- Annie enters this and the system displays the total cost  $\text{£}14 + \text{£}60 = \text{£}74$
- Stephanie agrees this
- Annie enters Stephanie's name, address and telephone number into the system
- Stephanie pays the £74
- Annie records this on the system and the system prints out a receipt
- Stephanie agrees to bring the bike back by 5.00pm on the following Saturday.

# Xác định các lớp

- Stephanie arrives at the shop at 9.00am one Saturday and chooses a mountain bike
- Annie sees that its number is 468
- Annie enters this number into the system
- The system confirms that this is a woman's mountain bike and displays the daily rate (£2) and the deposit (£60) (:Bike)
- Stephanie says she wants to hire the bike for a week
- Annie enters this and the system displays the total cost £14 + £60 = £74 (:Bike)
- Stephanie agrees this
- Annie enters Stephanie's name, address and telephone number into the system (:Customer)
- Stephanie pays the £74
- Annie records this on the system and the system prints out a receipt (:Payment collaborating with :Customer)
- Stephanie agrees to bring the bike back by 5.00pm on the following Saturday.

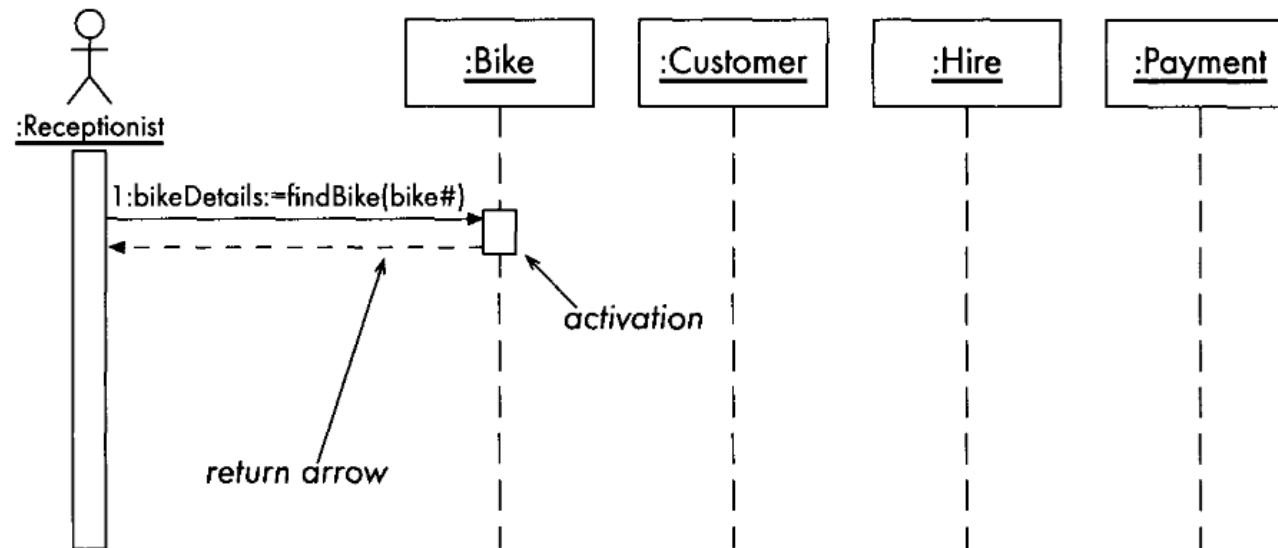


# Xác định các tác nhân và đối tượng



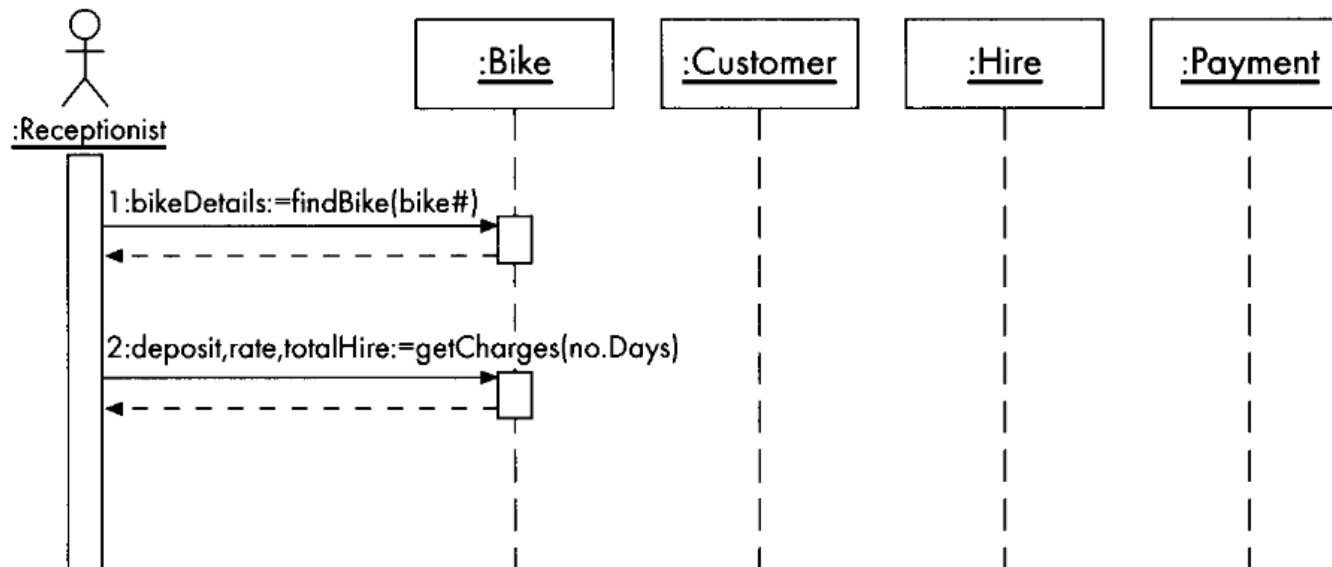
# Thêm các thông điệp

- 1 Stephanie arrives at the shop at 9.00am one Saturday and chooses a mountain bike
- 2 Annie sees that its number is 468
- 3 Annie enters this number into the system
- 4 The system confirms that this is a woman's mountain bike and displays the daily rate (£2) and the deposit (£60).



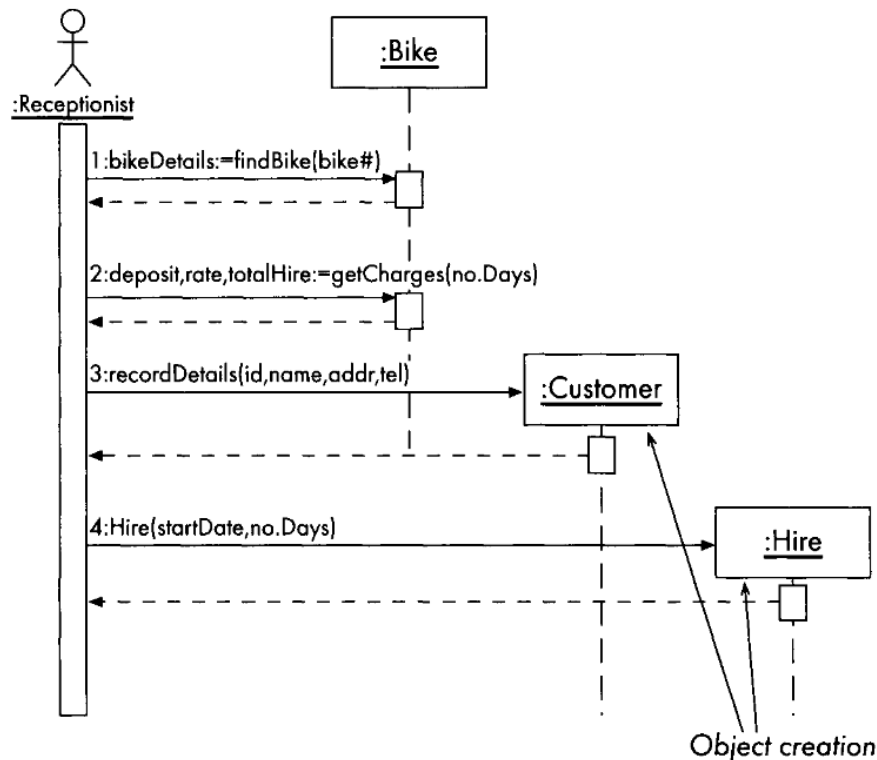
# Thêm các thông điệp

- 5 Stephanie says she wants to hire the bike for a week
- 6 Annie enters this and the system displays the total cost £14 + £60 = £74.



# Thêm các thông điệp

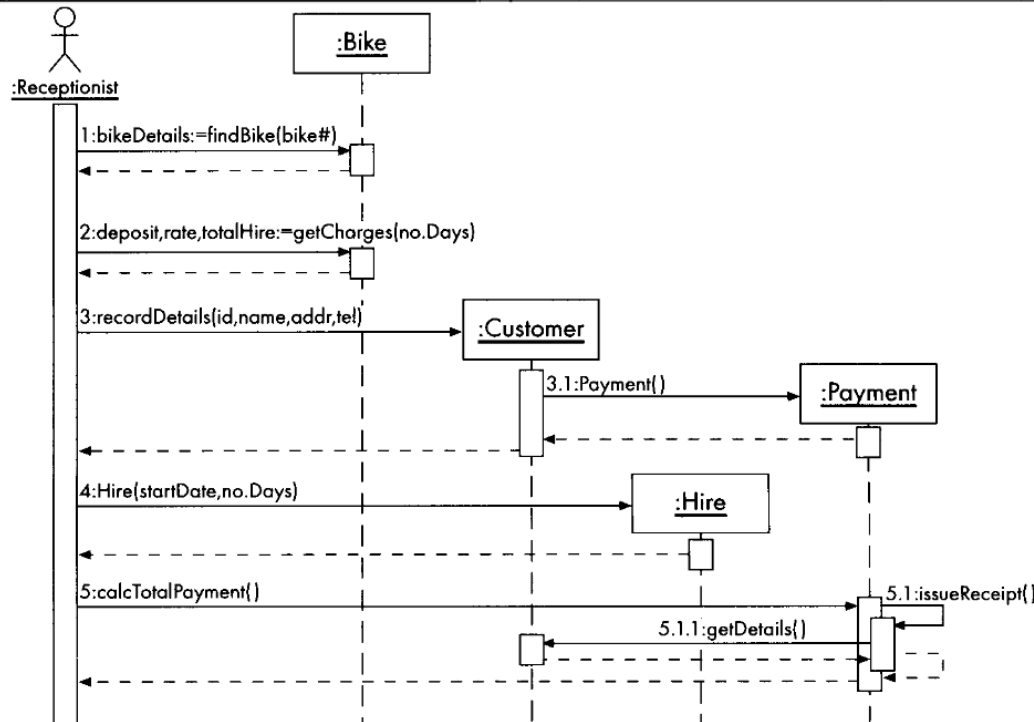
- 7 Stephanie agrees this
- 8 Annie enters Stephanie's name, address and telephone number into the system.



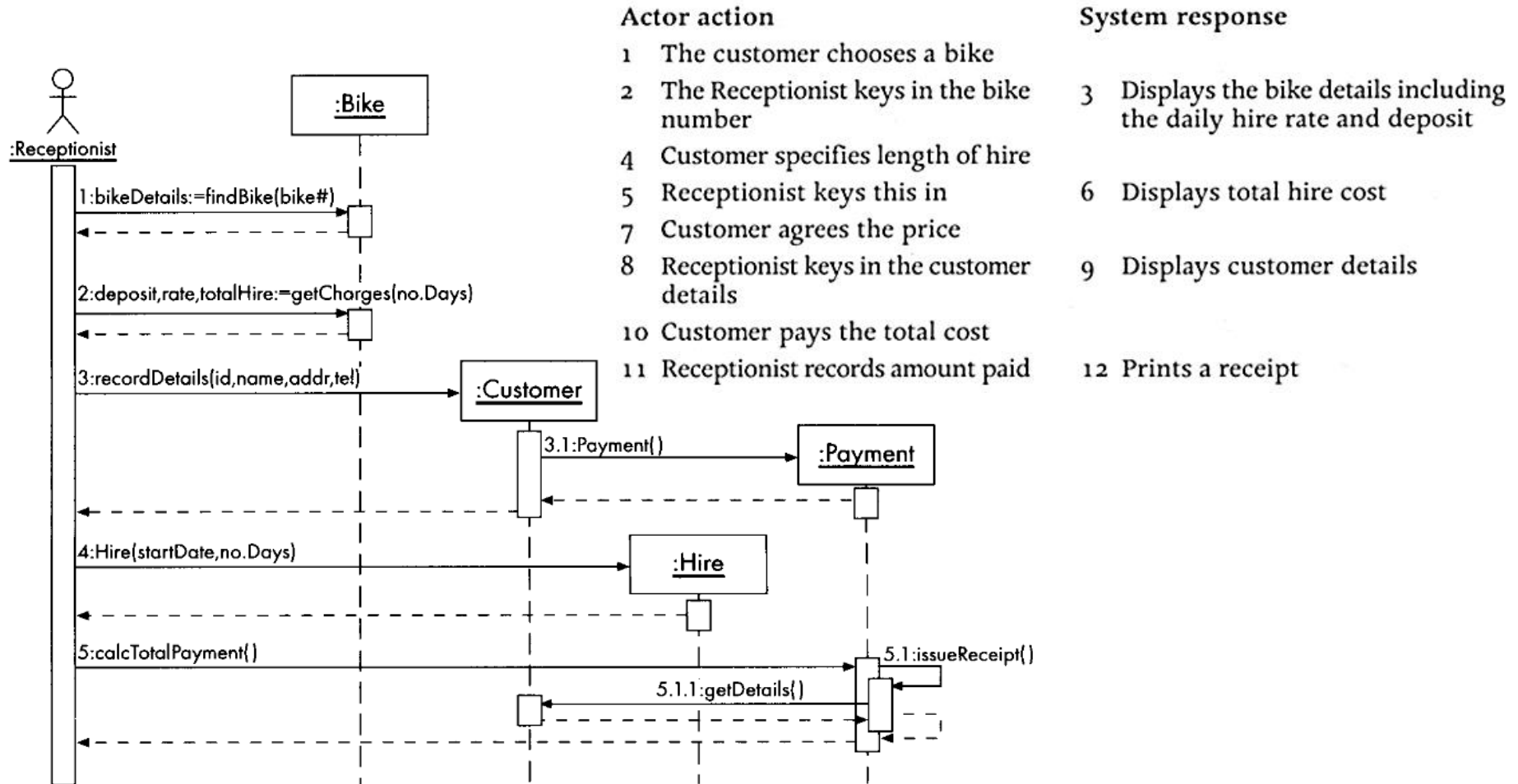


# Thêm các thông điệp

- 9 Stephanie pays the £74
- 10 Annie records this on the system and the system prints out a receipt
- 11 Stephanie agrees to bring the bike back by 5.00pm on the following Saturday



# Kịch bản và sơ đồ



# Các loại lớp

---

- Lớp thực thể (Entity class): mô tả bài toán (domain)
  - Lớp biên (Boundary class)
  - Lớp điều khiển (Control class)
  - BCE dùng để phân tách:
    - Ai giao tiếp với người dùng (Boundary)
    - Ai điều phối logic (Control)
    - Ai giữ dữ liệu (Entity)
- BCE giúp tăng tính rõ ràng, giảm coupling, tăng cohesion

# Đặc điểm của BCE

---

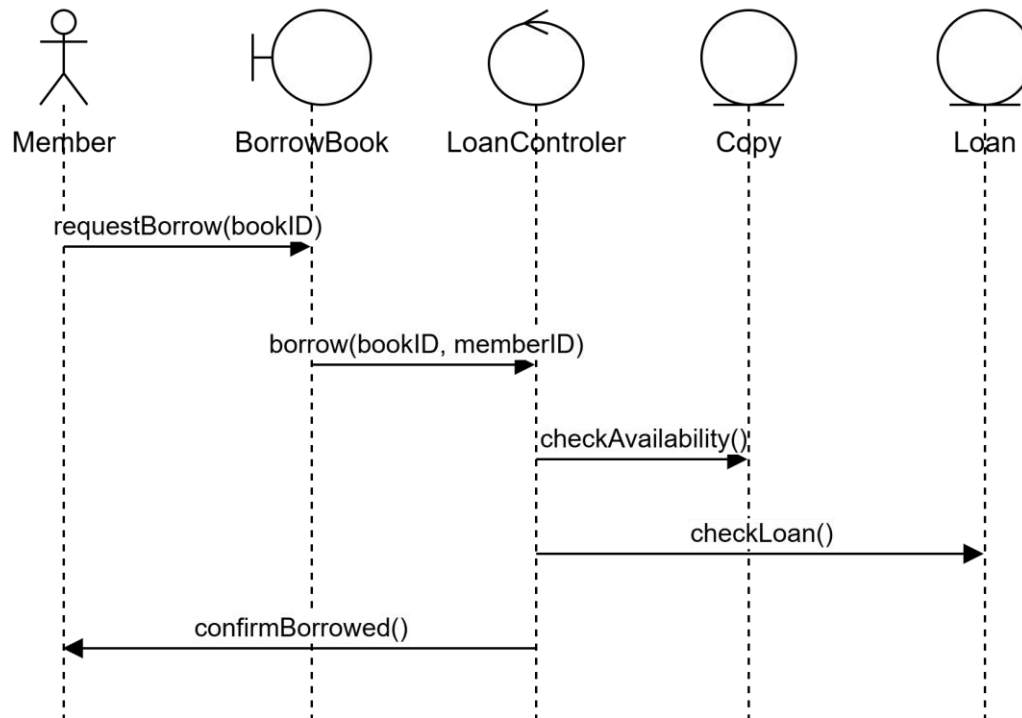
- Boundary class
  - Thường gắn với UI, form, API endpoint
- Control class
  - Điều phối logic, điều khiển trong một use case cụ thể
- Entity class
  - Tồn tại lâu dài, lưu dữ liệu domain
  - Thường ánh xạ sang bảng CSDL

# Ví dụ: Borrow Book

---

- Vẽ sơ đồ trình tự cho use case Borrow Book
  - Actor: Member
  - Boundary: BorrowForm, Control: LoanController
  - Entity: Copy, Loan
  - Luồng chính:
    1. Member → BorrowForm: requestBorrow(bookID)
    2. BorrowForm → LoanController: borrow(bookID, memberID)
    3. LoanController → Copy: checkAvailability()
    4. LoanController → Loan: createLoan()
    5. LoanController → BorrowForm: confirmBorrowed()

# Borrow Book

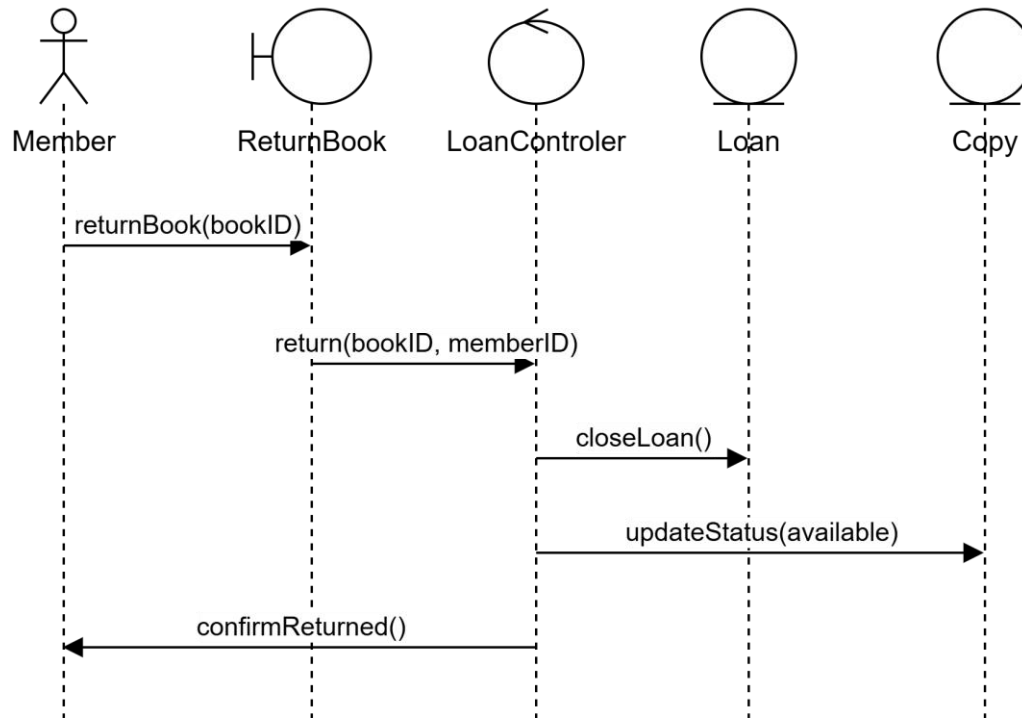


# Ví dụ: Return Book

---

- Vẽ sơ đồ trình tự cho use case Return Book
  - Actor: Member
  - Boundary: ReturnForm, Control: LoanController
  - Entity: Loan, Copy
  - Luồng chính:
    1. Member → ReturnForm: returnBook(copyID)
    2. ReturnForm → LoanController: return(copyID, memberID)
    3. LoanController → Loan: closeLoan()
    4. LoanController → Copy: updateStatus(Available)
    5. LoanController → ReturnForm: confirmReturned()

# Return Book





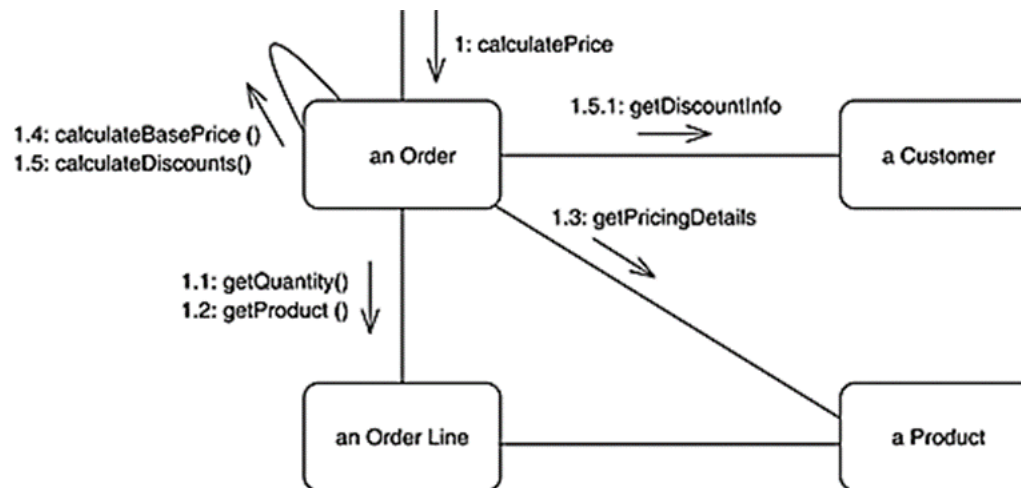
# Sơ đồ giao tiếp

---

- Thể hiện sự tương tác giữa các đối tượng
- Tác dụng tương đương với sơ đồ trình tự
- Khi nào nên dùng sơ đồ giao tiếp?
  - Khi muốn làm rõ mối quan hệ giữa các đối tượng
  - Phù hợp để phân tích kết nối phức tạp giữa nhiều đối tượng

# Sơ đồ giao tiếp

- Các ký hiệu
  - **Object** = hình chữ nhật
  - **Message** = mũi tên giữa các đối tượng
    - Số thứ tự tin nhắn: 1, 1.1, 2... → để thể hiện thứ tự
  - **Nhãn** = “số thứ tự + tên hành động”

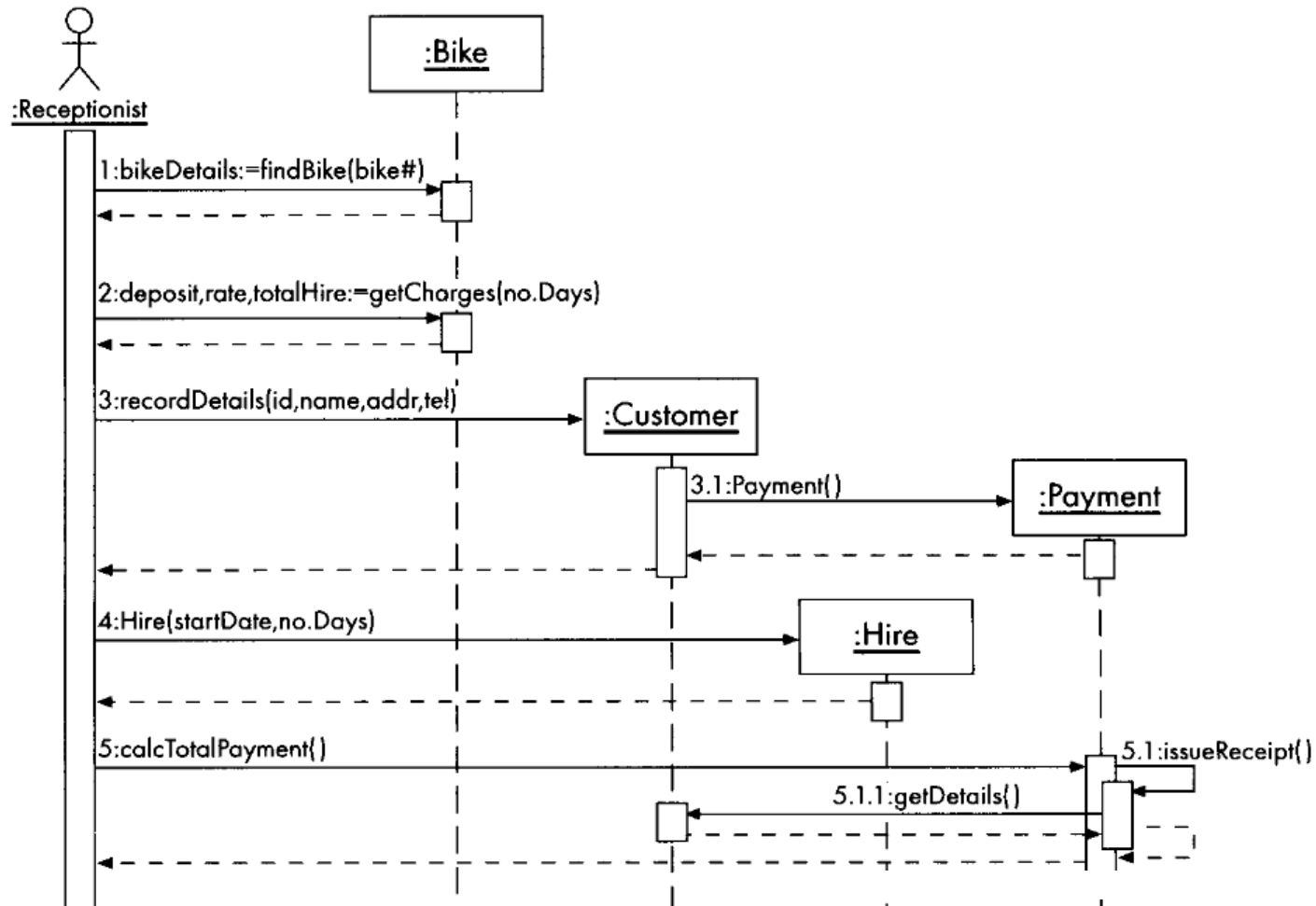


# Xây dựng sơ đồ giao tiếp

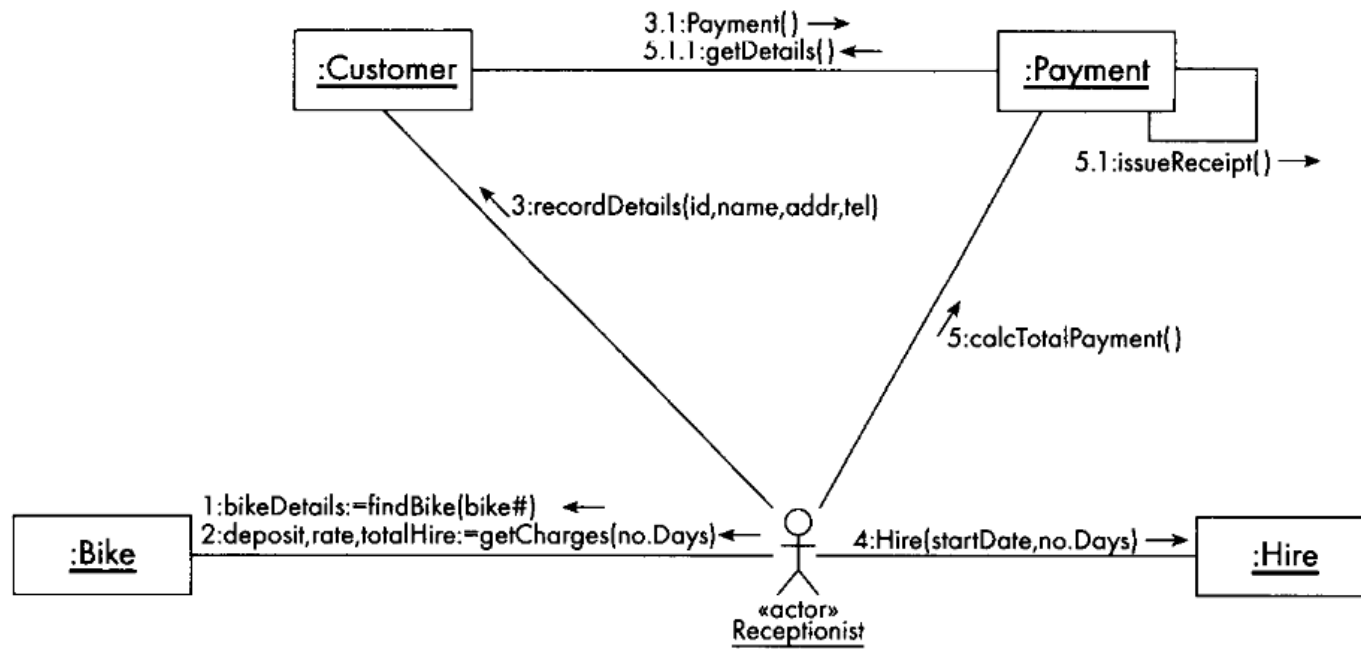
---

1. Xác định ngữ cảnh
2. Nhận diện các tác nhân và đối tượng tham gia
3. Thêm các thông điệp tương tác giữa các đối tượng
4. Các thông điệp nên đánh số để rõ thứ tự thực hiện

# Ví dụ

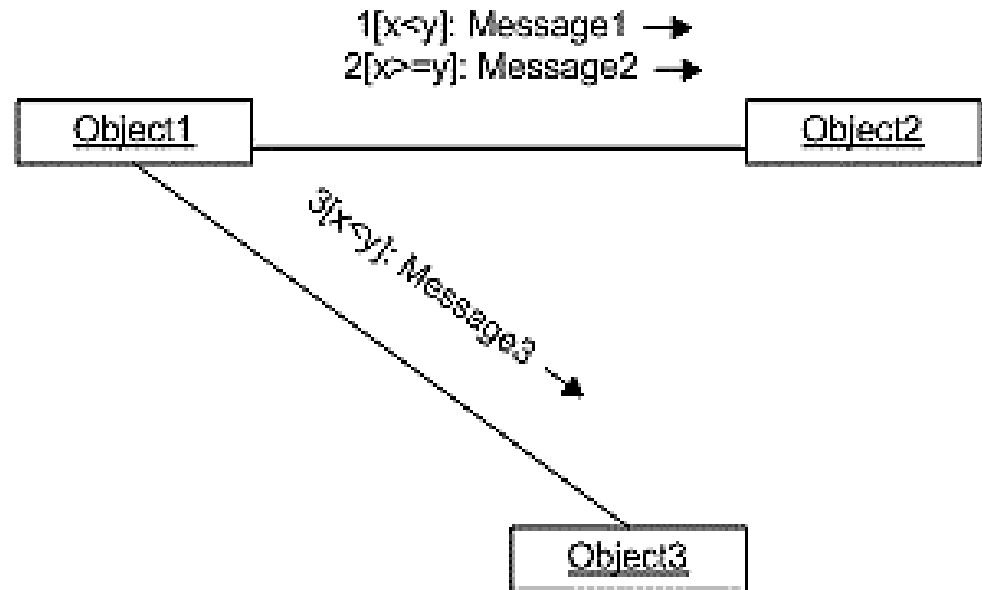


# Ví dụ



# Xử lý điều kiện

```
if (x < y)
{
    object2.Message1();
    object3.Message3();
}
else
{
    object2.Message2();
}
```



# Tóm tắt

---

- Sequence = trình tự thời gian
- Communication = cấu trúc cộng tác
- Cả hai bổ sung cho nhau
- Deliverable (Tài liệu cần có):

## **Sơ đồ tương tác** cho các **use case** chính

- Đầu vào quan trọng để bước tiếp theo: Design Class Diagram

# Câu hỏi

---

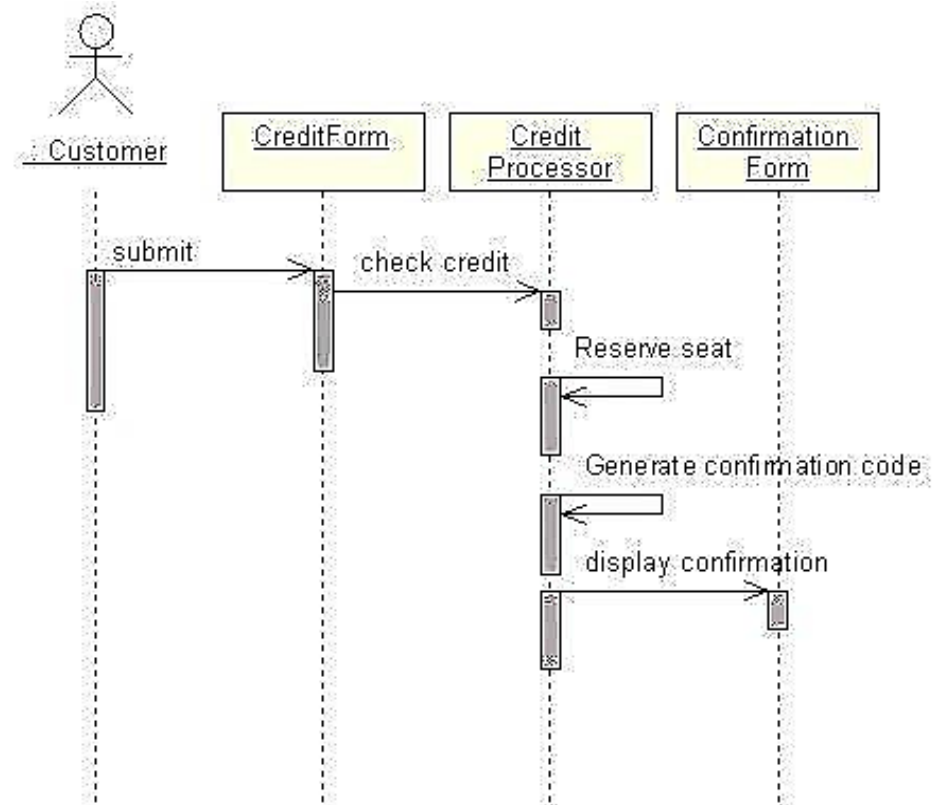
1. Trình bày về hai loại sơ đồ tương tác: tên gọi, tác dụng, cách xây dựng và đặc điểm chính?
2. Tác dụng của sơ đồ trình tự là gì?
3. Hãy so sánh sơ đồ trình tự và sơ đồ trình tự hệ thống?
4. Hãy so sánh hai loại sơ đồ trình tự và sơ đồ giao tiếp? Ưu điểm của mỗi loại sơ đồ này là gì?



# Bài tập

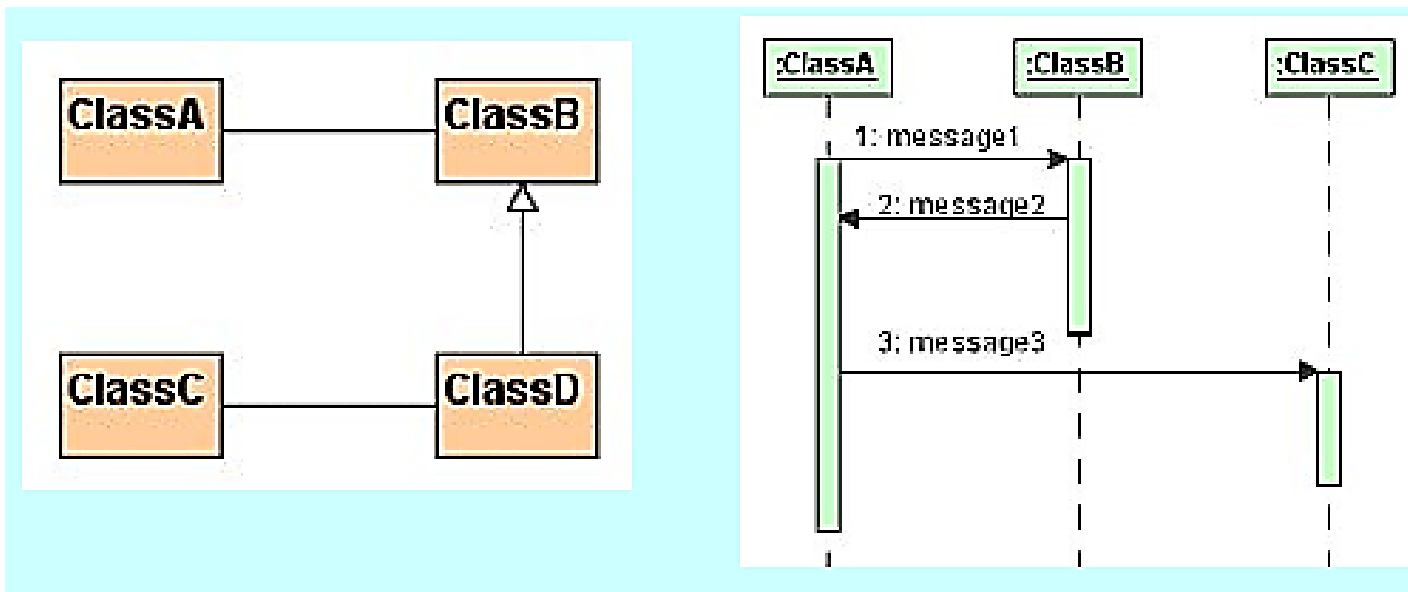
1. Xem sơ đồ trình tự hình bên thì phương thức nào sẽ cần phải cài đặt cho lớp CreditProcessor?

- a. checkCredit, generateConfirmationCode, displayCofirmation
- b. checkCredit, generateConfirmationCode
- c. checkCredit, generateConfirmationCode, reserveSeat
- d. checkCredit, reserveSeat, displayCofirmation



# Bài tập

2. Cho biết sơ đồ trình tự sau có vấn đề gì?



*Sơ đồ lớp cho thấy: đối tượng của A và C không thể có tương tác với nhau*