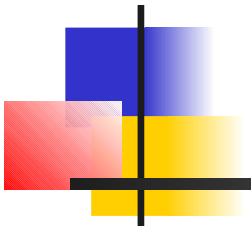
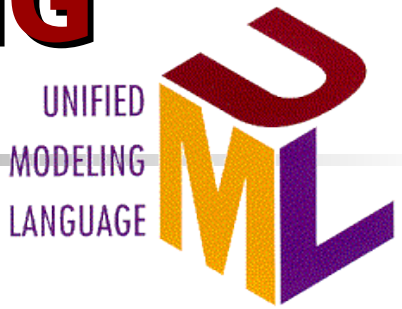
VIỆN KHOA HỌC VÀ CÔNG NGHỆ VIỆT NAM

**VIỆN CÔNG NGHỆ THÔNG TIN**

# PHÂN TÍCH THIẾT KẾ HƯỚNG ĐỐI TƯỢNG






**PGS.TS. Đặng Văn Đức**

# Nội dung

1. Tiến trình phát triển phần mềm theo hướng đối tượng
2. Giới thiệu Ngôn ngữ mô hình hóa thống nhất UML
3. Mô hình hóa nghiệp vụ
4. Mô hình hóa trường hợp sử dụng
5. Mô hình hóa tương tác đối tượng
6. Biểu đồ lớp và gói
7. Biểu đồ chuyển trạng thái và biểu đồ hoạt động
- ✓ **Biểu đồ kiến trúc vật lý và phát sinh mã trình**
9. Mô hình hóa dữ liệu
10. Bài học thực nghiệm

# Biểu đồ kiến trúc vật lý và phát sinh mã trình

1.Kiến trúc phần mềm?

2.Các thành phần

3.Biểu đồ thành phần

4.Bổ sung chi tiết cho thành phần

5.Biểu đồ triển khai

6.Các phần tử của biểu đồ triển khai

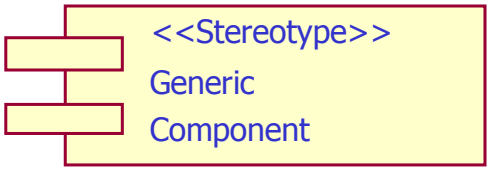
7.Phát sinh mã trình

# Kiến trúc phần mềm?

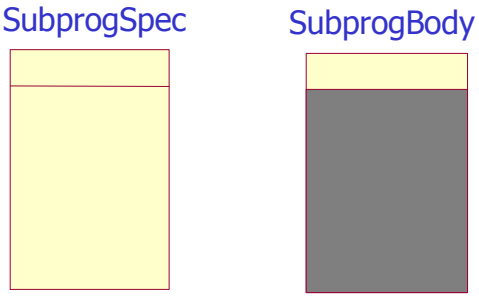
- Kiến trúc hệ thống là kế hoạch chi tiết của các bộ phận hình thành hệ thống
- UML định nghĩa:
  - Kiến trúc là cấu trúc tổ chức của hệ thống
  - Kiến trúc bao gồm các bộ phận tương tác thông qua giao diện
- Theo Buschman:
  - Kiến trúc phần mềm là mô tả các phân hệ, các thành phần của hệ thống phần mềm và các quan hệ giữa chúng
- Hai loại kiến trúc hệ thống
  - Kiến trúc logíc
    - Chỉ ra các lớp đối tượng và các quan hệ giữa chúng để hình thành chức năng hệ thống
    - Nó được thể hiện bằng các biểu đồ UC, biểu đồ lớp, trạng thái, hoạt động...
  - Kiến trúc vật lý
    - Là mô tả từ khía cạnh phần cứng và các mođun phần mềm trên đó
    - Nó được mô tả bằng các biểu đồ cài đặt: biểu đồ thành phần và biểu đồ triển khai

# Các thành phần

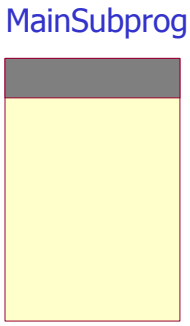
- Thành phần?
  - Là mô đun vật lý mã trình: thư viện mã nguồn, mã khả thực.
- Các loại thành phần




Đặc tả thành phần
bằng Stereotype



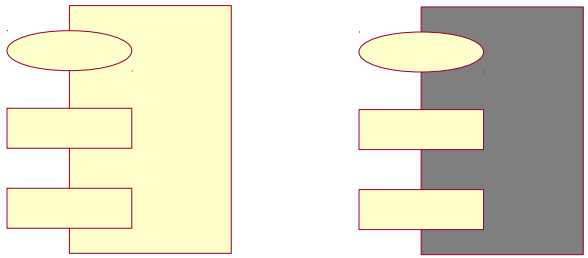

Đặc tả và thân chương trình con
Tập hợp các hàm
Không chứa định nghĩa lớp



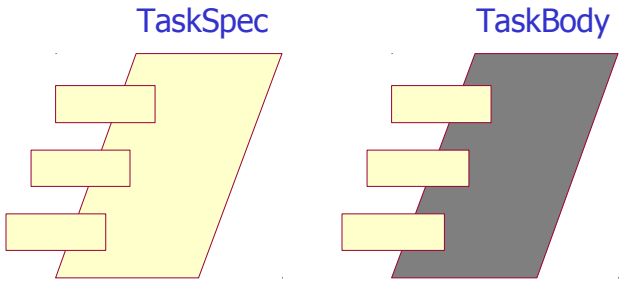

Chương trình chính
Chứa đầu vào chương trình

# Các thành phần

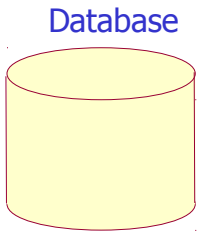
- Các loại thành phần




Đặc tả và thân gói
Gói là cài đặt lớp
Đặc tả gói là tệp header



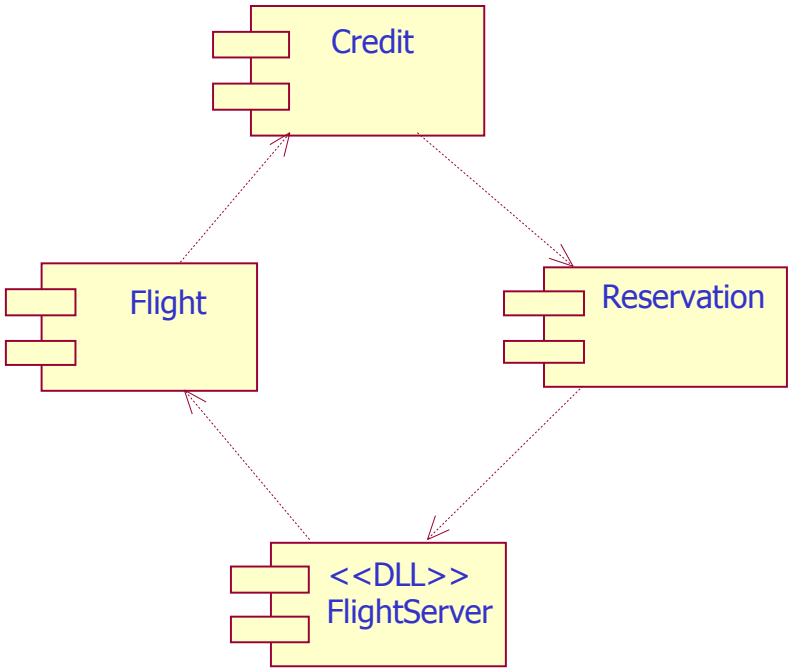

Đặc tả và thân nhiệm vụ
Là các thành phần Run-time
Biểu diễn các gói có thread độc lập



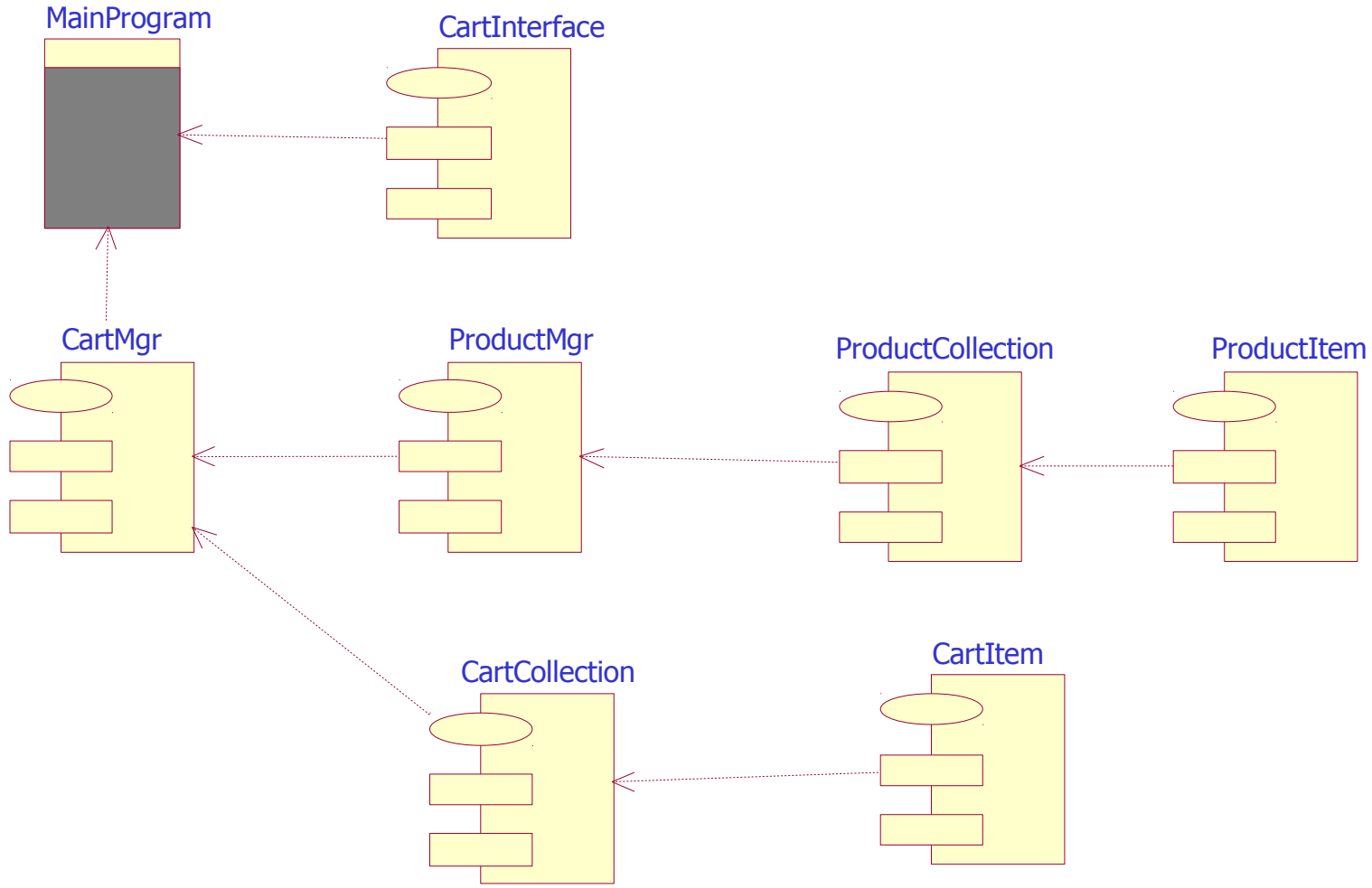

Biểu diễn CSDL
Chứa một hay nhiều lược đồ

# Biểu đồ thành phần

- Biểu đồ thành phần là biểu đồ hiển thị các thành phần trong hệ thống và phụ thuộc giữa chúng
  - Thành phần A phụ thuộc vào thành phần B khi vài lớp trong A phụ thuộc vào vài lớp trong B
- Biểu đồ cho biết
  - Thư viện nào được sử dụng, tệp khả thực (.exe) nào được tạo ra khi dịch chương trình
  - Các quan hệ giữa các thư viện mã trình
- Có khả năng tổ chức các thành phần vào các gói



# Thí dụ Biểu đồ thành phần



# Bổ sung chi tiết cho thành phần

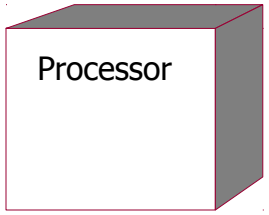
- Stereotype
  - Lựa chọn biểu tượng để biểu diễn thành phần
  - Có thể là:
    - <none>, ActiveX, Applet, Subroutine Spec, dll... tự định nghĩa
- Language
  - Trong Rose có thể gán ngôn ngữ cho thành phần
  - Cho khả năng phat sinh các ngôn ngữ khác nhau cho mỗi thành phần
- Declaration
  - Gán các khai báo vào mã trình của từng thành phần
- Class
  - Gán lớp vào thành phần trước khi phát sinh mã trình
  - Có thể ánh xạ một hay nhiều lớp vào một thành phần
- Dependency
  - Thành phần chỉ có một loại quan hệ: quan hệ phụ thuộc
  - Tránh hình thành quan hệ vòng

# Biểu đồ triển khai

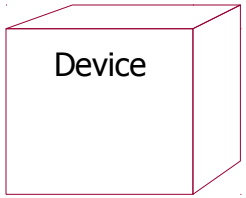
- Biểu đồ triển khai mô tả kiến trúc phần cứng (các nút) có phần mềm chạy trên chúng, bao gồm các bộ xử lý, các tiến trình, các thiết bị và các kết nối giữa chúng
  - Mô tả tôpô của hệ thống
  - Chỉ ra toàn bộ các nút trên mạng, kết nối giữa chúng và các phần mềm chạy trên chúng
- Nút là đối tượng vật lý có tài nguyên tính toán
  - Máy tính, máy in, thiết bị đọc thẻ từ và truyền tin
- Giữa các nút là kết nối giao tiếp, kiểu kết nối được thể hiện bằng stereotype

# Các phần tử của biểu đồ triển khai

- Bộ xử lý
  - Là máy xử lý: máy chủ, máy trạm
  - Bổ sung thuộc tính:
    - Stereotype
    - Mô tả vật lý của bộ xử lý: tốc độ, dung lượng nhớ
    - Lập lịch xử lý: Preemptive, Non-preemptive, Cyclic, Executive, Manual

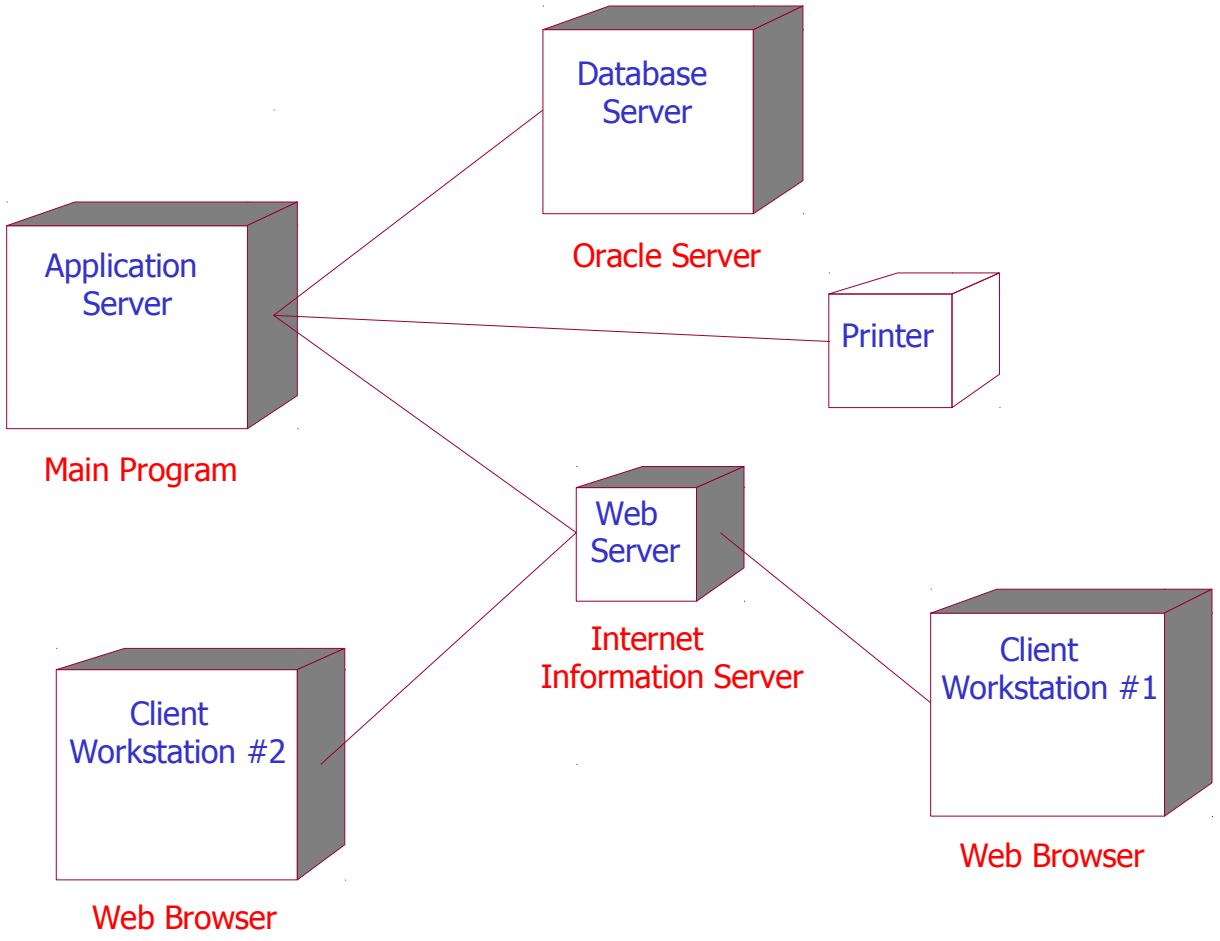



- Thiết bị
  - Là phần cứng chỉ có một mục đích: máy in, scanner...
  - Bổ sung thuộc tính:
    - Stereotype
    - Mô tả vật lý của thiết bị




- Kết nối
  - Là liên kết vật lý giữa các thiết bị và bộ xử lý
  - Bổ sung stereotype và đặc tính vật lý cho kết nối: T1
- Bổ sung tiến trình cho bộ xử lý

# Thí dụ biểu đồ triển khai



# Phát sinh mã trình

- Sáu bước cơ bản để phát sinh mã trình
  - Kiểm tra mô hình
  - Tạo lập thành phần
  - Ánh xạ lớp vào thành phần
  - Gán thuộc tính phát sinh mã trình
  - Chọn lớp, thành phần hay gói để phát sinh mã
  - Phát sinh mã trình

# Phát sinh mã trình

- Bước 1:  Kiểm tra mô hình
  - Rose có chức năng kiểm tra mô hình độc lập ngôn ngữ để đảm bảo tính nhất quán trong mô hình
  - Khi kiểm tra có thể phát hiện các lỗi sau
    - Ánh xạ không đầy đủ: Các đối tượng hay thông điệp trong biểu đồ trình tự chưa ánh xạ vào thao tác hay lớp trong biểu đồ lớp
    - Vi phạm xâm nhập: Thí dụ, hai lớp trong hai gói có quan hệ nhưng vẽ thiếu quan hệ giữa hai gói
    - Kiểm tra phụ thuộc ngôn ngữ: Sẽ phát hiện, thí dụ, nhiều lớp cùng tên khai báo public trong một modun chương trình
- Bước 2: Tạo lập thành phần
  - Tạo lập thành phần để chứa lớp
  - Trước khi phát sinh mã trình phải ánh xạ các lớp vào thành phần tương ứng
  - Bổ sung quan hệ thành phần trên Biểu đồ thành phần

# Phát sinh mã trình

- Bước 3:  Ánh xạ lớp vào thành phần
  - Mỗi thành phần mã nguồn biểu diễn tệp mã nguồn cho một hoặc vài lớp
    - Thí dụ C++: Mỗi lớp ánh xạ đến hai thành phần – Các tệp Header và Body
  - Bước này yêu cầu ánh xạ lớp vào thành phần tương ứng
- Bước 4: Đặt đặc tính cho phát sinh mã trình
  - Nhiều đặc tính có thể gán cho lớp, thuộc tính, thành phần của mô hình để điều khiển mã được phát sinh như thế nào.
    - Thí dụ C++: Đặc tính GenerateGetOperation điều khiển việc có phát sinh hàm Get() hay không.
    - Thí dụ khác: GenerateDefaultConstructor
  - Đặt tập đặc tính tạm thời
    - Thay vì thay đổi trực tiếp tập đặc tính ta có thể tạo ra tập đặc tính tạm thời để sử dụng, không ảnh hưởng đến tập đặc tính mặc định
  - Hủy bỏ tập đặc tính tạm thời

# Phát sinh mã trình

- Bước 5: Chọn lớp, thành phần hay gói
  - Có thể chọn lớp, thành phần hay gói để phát sinh mã trình vào các thời điểm khác nhau
  - Phát sinh mã từ biểu đồ hay Browser
  - Có thể phát sinh mã trình cho một vài lớp, thành phần hay gói đồng thời
- Bước 6: Phát sinh mã trình
  - Lựa chọn ngôn ngữ theo yêu cầu để phát sinh mã từ mô hình

# Phát sinh mã trình

- Cái gì đã được phát sinh từ mô hình?
  - Thực tế
    - Không có công cụ mô hình hóa nào phát sinh mã trình đầy đủ
    - Rose cũng chỉ phát sinh khung chương trình
  - Các phần tử được phát sinh
    - Lớp: Mọi lớp trong mô hình được sinh mã
    - Thuộc tính: Mã trình sẽ chứa các thuộc tính lớp bao gồm phạm vi, kiểu dữ liệu và giá trị mặc định, các hàm Get(), Set().
    - Signature: Các thao tác được khai báo trong mã trình cùng với danh sách tham số, kiểu dữ liệu của tham số và kiểu giá trị cho lại của thao tác
    - Quan hệ: Một số quan hệ trong mô hình được chuyển sang thuộc tính
    - Thành phần: Mỗi thành phần được hiện thực trong tệp tương ứng
    - Tài liệu: Tài liệu trong mô hình được chèn vào nơi thích ứng trong mã trình

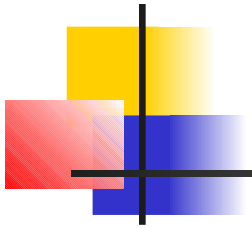
# Phát sinh mã trình

- Nhiệm vụ của người phát triển sau khi Rose sinh mã trình
  - Thu thập các tệp mã trình, viết mã trình cho các thao tác lớp
  - Thiết kế giao diện đồ họa
- Thí dụ đoạn mã trình do Rose phát sinh

```
#include "stdafx.h"
#include "Order.h"
//##ModelId=3A77E3CD0280
Boolean Order::Create()
{
  // TODO: Add your specialized code here.
  // NOTE: Requires a correct return value to compile.
}
//##ModelId=3A77E3E60316
Boolean Order::SetInfo(Integer OrderNum, String Customer, Date OrderDate, Date FillDate)
{
  // TODO: Add your specialized code here.
  // NOTE: Requires a correct return value to compile.
}
//##ModelId=3A77E40E0230
String Order::GetInfo()
{
  // TODO: Add your specialized code here.
  // NOTE: Requires a correct return value to compile.
```

# Phát sinh mã trình

```
class Order
{
public:
  //##ModelId=3A7F695F019A
  OrderItem* theOrderItem;
  //##ModelId=3A77E3CD0280
  Boolean Create();
  //##ModelId=3A77E3E60316
Boolean SetInfo(Integer OrderNum, String Customer, Date OrderDate, Date
FillDate);
  //##ModelId=3A77E40E0230
  String GetInfo();
private:
  //##ModelId=3A7E13F9038E
  Integer OrderNumber;
  //##ModelId=3A7E14260122
  String CustomerName;
  //##ModelId=3A7E14470208
  Date OrderDate;
  //##ModelId=3A7E145303D4
  Date OrderFillDate;
};
```



# Tóm tắt

- Bài này đã xem xét các vấn đề sau
  - Kiến trúc vật lý của hệ thống
  - Xây dựng biểu đồ thành phần
    - Các thành phần phần mềm và quan hệ giữa chúng
    - Các phần tử đồ họa vẽ biểu đồ thành phần
  - Xây dựng biểu đồ triển khai
    - Các phần tử đồ họa vẽ biểu đồ triển khai
  - Các bước chuyển đổi mô hình thành phần mềm