

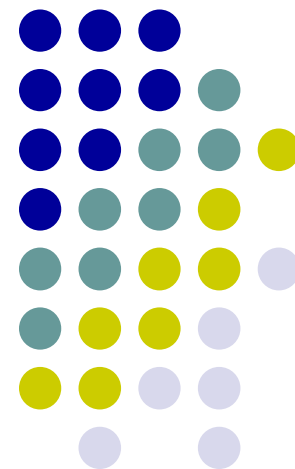
Bài 9. Thiết kế CSDL và Thiết kế chương trình

Nguyễn Hoài Anh

Khoa công nghệ thông tin

Học viện kỹ thuật quân sự

nguyenhoaianh@yahoo.com



NỘI DUNG

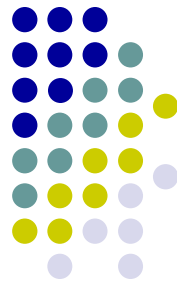


- Thiết kế CSDL
 - Mô hình dữ liệu
 - Truy xuất dữ liệu
- Thiết kế kiến trúc chương trình
 - Thiết kế kiến trúc
 - Thiết kế xử lý

THIẾT KẾ CƠ SỞ DỮ LIỆU



- Gồm 2 giai đoạn
 - Giai đoạn 1: Thiết kế CSDL logic
 - Thiết kế bảng quan hệ, trường dữ liệu phục vụ bảo mật
 - Thiết kế mô hình dữ liệu logic của hệ thống
 - Đặc tả cấu trúc CSDL logic
 - Giai đoạn 2: Thiết kế CSDL vật lý
 - Chọn công nghệ lưu trữ và quản lý dữ liệu
 - Thiết kế tệp, tổ chức tệp và cách thức truy xuất



THIẾT KẾ CSDL LOGIC

- **Mục đích:** xây dựng CSDL thỏa mãn
 - Hợp lý: đủ dùng và không dư thừa
 - Truy nhập thuận lợi
 - Tìm kiếm, cập nhật
 - Bổ sung và loại bỏ sao cho nhanh chóng và tiện dùng

THIẾT KẾ CSDL LOGIC



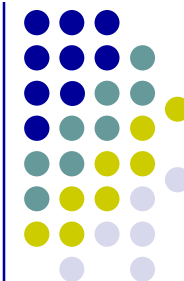
- Bảng quan hệ, trường dữ liệu cho bảo mật
 - Bảng quan hệ:
 - user, group,
 - employee,
 - permission
 - Trường dữ liệu:
 - Thêm iduser vào DONHANG để quản lý người tạo
 - Thêm maktra vào DONHANG để kiểm tra việc truy cập trái phép vào CSDL.

THIẾT KẾ CSDL LOGIC



- Mô hình dữ liệu logic của hệ thống
 - Nghiên cứu tình huống
 - DONHANG – CT_DONHANG: truy xuất cùng nhau
→ nghiên cứu gộp thành một bảng
 - Trường thanh tien = Soluong * Dongia: nếu thường xuyên đơn hàng có nhiều mặt hàng
→ nghiên cứu thêm trường thành tiền
 - Bảng dữ liệu tương ứng kho lưu tay → loại bỏ
 - Thêm bảng dữ liệu bảo mật và xác định quan hệ
 - Ước lượng khối lượng lưu trữ

THIẾT KẾ CSDL LOGIC



• Đặc tả cấu trúc

1. Số hiệu:	10	2. Tên bảng:	NSD	3. Bí danh:	NSD
4. Mô tả: Lưu trữ thông tin người sử dụng.					
5. Mô tả chi tiết các cột					
Số	Tên cột	Mô tả	Kiểu dữ liệu	Khuôn dạng	N
#1	ID	Số hiệu người sử dụng	N (6)	Số nguyên	✓
2	ID_NHOM	Số hiệu nhóm	N (6)	Số nguyên	✓
3	TEN	Tên người sử dụng	C (100)	Chữ cái	
4	TENLOGIN	Tên đăng nhập	C (10)	Chữ cái + chữ số	✓
5	MATKHAU	Mật khẩu	C (10)	Trù ký tự đặc biệt	✓
6	NGAY_CN	Ngày cập nhật	D (8)	MM/DD/YYYY	✓
7	XOA	Kiểm tra trạng thái xóa	N (1)	0 hoặc 1	✓
8	NGAY_BDSD	Ngày bắt đầu sử dụng	D (8)	MM/DD/YYYY	
9	NGAY_KTSD	Ngày kết thúc sử dụng	D (8)	MM/DD/YYYY	
6. Khoá ngoài					
Số	Tên	Cột khoá ngoài	Quan hệ với bảng		
1	ID_NHOM	ID_NHOM	NHOM_NSD		

THIẾT KẾ CSDL VẬT LÝ



- Mục đích

- Chuyển mô hình dữ liệu logic thành các đặc tả dữ liệu vật lý phù hợp điều kiện thiết bị và môi trường cụ thể trong không gian và nhu cầu sử dụng.

- Hai nội dung thiết kế

- Chọn công nghệ lưu trữ và quản lý dữ liệu
- Chuyển mô hình logic thành thiết kế vật lý và xác định phương án cài đặt trên hệ thống thiết bị.

THIẾT KẾ CSDL VẬT LÝ



- Thiết kế vật lý và phương án cài đặt
 - Thiết kế trường
 - Thiết kế bản ghi
 - Thiết kế tệp
 - Tổ chức tệp, bố trí tệp trong không gian

THIẾT KẾ CSDL VẬT LÝ



- Thiết kế vật lý và phương án cài đặt
 - Thiết kế trường
 - Đặc tả trường theo hệ quản trị CSDL đã chọn
 - Thiết kế bản ghi
 - Thiết kế tệp
 - Tổ chức tệp, bố trí tệp trong không gian

THIẾT KẾ CSDL VẬT LÝ



- Thiết kế vật lý và phương án cài đặt
 - Thiết kế trường
 - Thiết kế bản ghi
 - Nhóm các trường dữ liệu thành bản ghi vật lý
 - BENHNNHAN(idbn,hoten,diachi,ngayvao,giuong,khoa,tinhtrang,ngayra,thanhtoan)
BN1(idbn,hoten,diachi,khoa)
BN2(idbn,ngayvao,ngayra,giuong,tinhtrang,thanhtoan)
 - Thiết kế tệp
 - Tổ chức tệp, bố trí tệp trong không gian

THIẾT KẾ CSDL VẬT LÝ



- Thiết kế vật lý và phương án cài đặt
 - Thiết kế trường
 - Thiết kế bản ghi
 - Thiết kế tệp
 - Là phần nhỏ của bộ nhớ thứ cấp lưu các bản ghi một cách độc lập.
 - Các loại tệp: dữ liệu, lấy dữ liệu, giao dịch, làm việc, bảo vệ, sao lưu, lịch sử.
 - Tổ chức tệp, bố trí tệp trong không gian

THIẾT KẾ CSDL VẬT LÝ



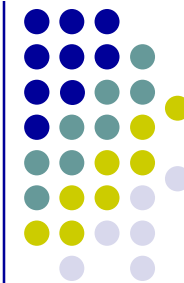
- Thiết kế vật lý và phương án cài đặt
 - Thiết kế trường
 - Thiết kế bản ghi
 - Thiết kế tệp
 - Tổ chức tệp, bố trí tệp trong không gian
 - Tổ chức: tuần tự, chỉ số
 - Phương thức truy cập: trực tiếp, gián tiếp

THIẾT KẾ CSDL VẬT LÝ



- Yêu tố cần quan tâm khi thiết kế
 - Lấy dữ liệu nhanh
 - Thông lượng các giao dịch xử lý lớn
 - Sử dụng hiệu quả không gian nhớ
 - Tránh được sai sót và mất dữ liệu
 - Tối ưu hóa nhu cầu tổ chức tệp
 - Đáp ứng được nhu cầu tăng trưởng dữ liệu
 - An toàn

THIẾT KẾ KIẾN TRÚC CTRÌNH

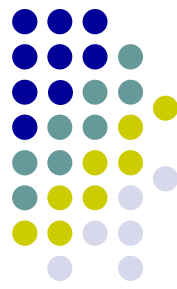


- **Khái niệm**

- là tập tất cả các modul (đơn thể) được sắp xếp theo một trật tự quy tắc xác định.
- được biểu diễn bởi lược đồ cấu trúc chương trình

- **Mục đích**

- Xây dựng một kiến trúc chương trình đúng đắn, hiệu quả mà với nội dung đó người lập trình có thể viết chương trình mà không cần hiểu cả hệ thống.



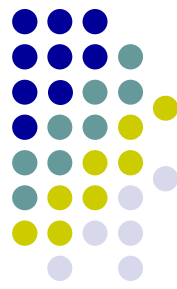
THIẾT KẾ KIẾN TRÚC CTRÌNH

- **Đầu vào**

- DFD hệ thống và đặc tả modul xử lý
- Thiết kế kiểm soát
- Thiết kế CSDL

- **Đầu ra**

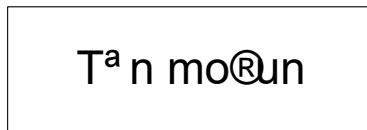
- Lược đồ cấu trúc: kiến trúc tổng thể của hệ thống máy tính dưới dạng modul chương trình.
- Đặc tả các modul chương trình.



THIẾT KẾ KIẾN TRÚC CTRÌNH

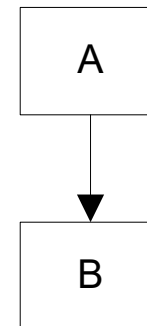
- Cách biểu diễn modul trong lược đồ cấu trúc

- Modul chương trình
- Modul có sẵn

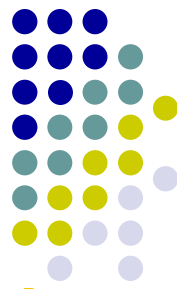


- Lời gọi modul

- Modul A gọi Modul B
B thực hiện xong quay
về A tại vị trí liền sau lời gọi



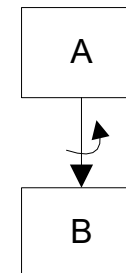
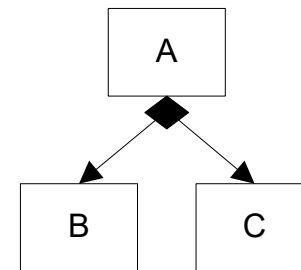
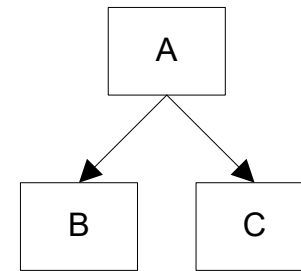
THIẾT KẾ KIẾN TRÚC CTRÌNH



- Cách biểu diễn modul trong lược đồ cấu trúc

- Lời gọi modul

- Modul A gọi Modul B rồi gọi modul C (thứ tự từ trái sang phải)
- Modul A gọi B hoặc C tùy thuộc kết quả phép chọn
- Modul A gọi B nhiều lần





THIẾT KẾ KIẾN TRÚC CTRÌNH

- Cách biểu diễn modul trong lược đồ cấu trúc
 - Thông tin chuyển giao giữa các modul
 - Truyền dữ liệu



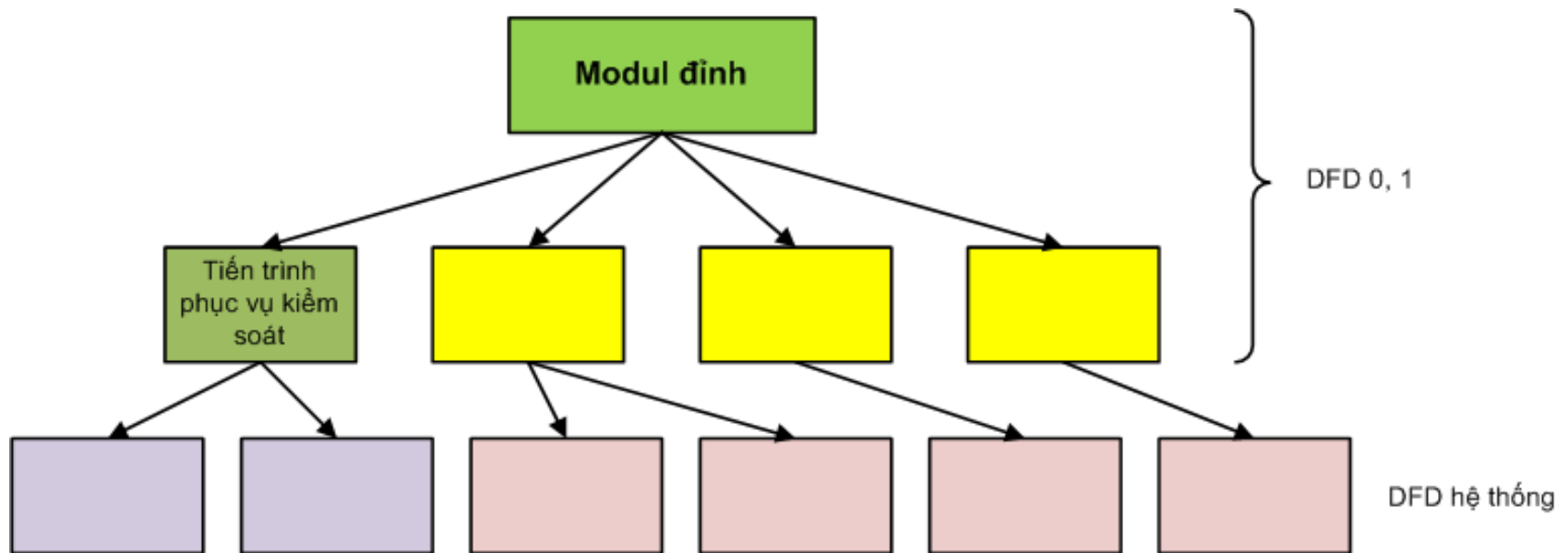
- Truyền thông tin điều khiển





THIẾT KẾ KIẾN TRÚC CTRÌNH

- Thiết kế kiến trúc hệ thống mức cao
 - Modul điều khiển, làm nhiệm vụ kết nối các modul phía dưới.





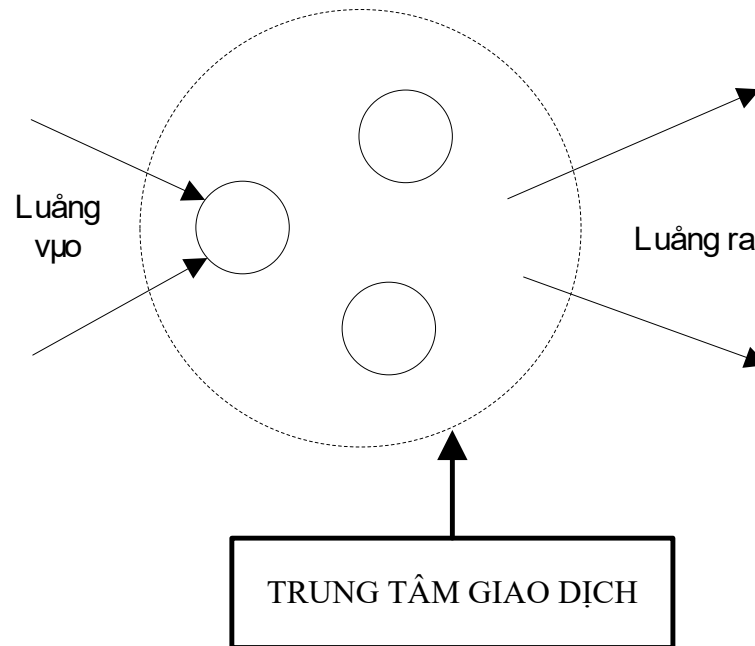
THIẾT KẾ KIẾN TRÚC CTRÌNH

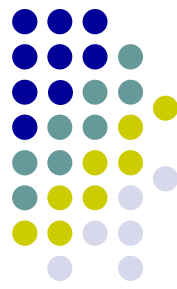
- Thiết kế kiến trúc mức thấp (modul xử lý)
 - Cấu trúc DFD hệ thống thành cấu trúc cây
 - Có hai dạng cấu trúc
 - Tập trung hướng giao dịch (transform centered design)
 - Tập trung hướng tương tác (transition centered design)

THIẾT KẾ KIẾN TRÚC CTRÌNH



- Tập trung hướng giao dịch
 - Phát hiện trung tâm giao dịch thông tin chủ yếu

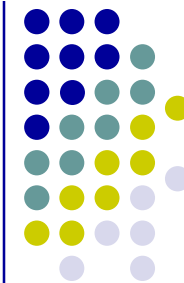




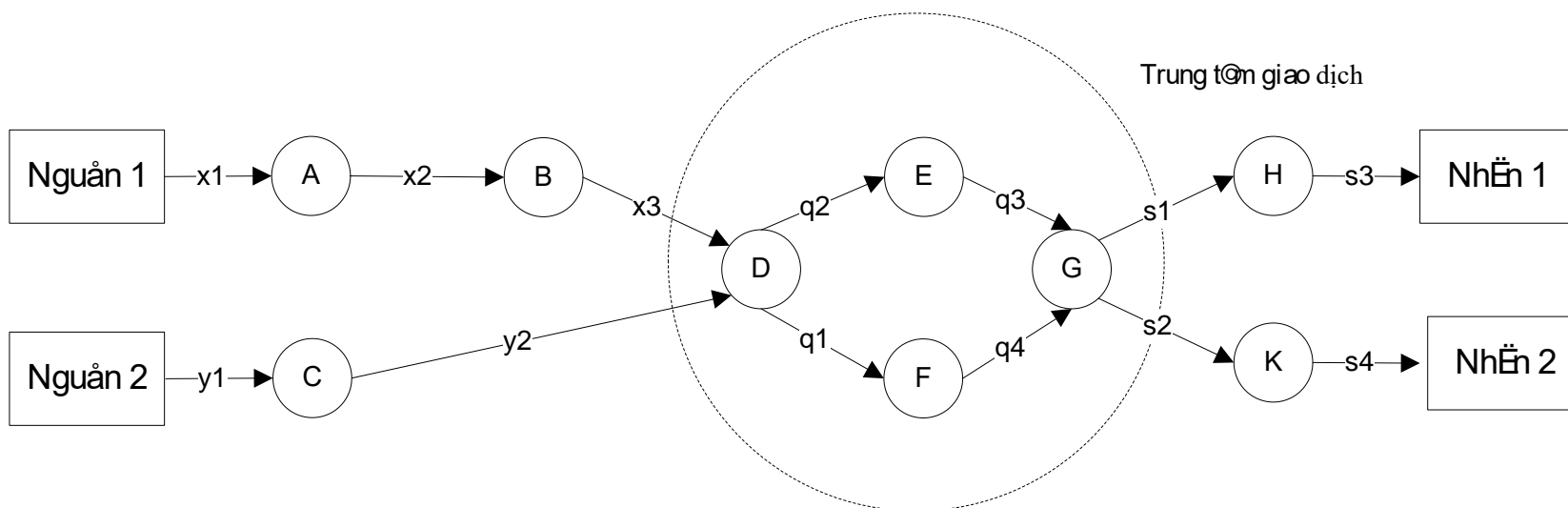
THIẾT KẾ KIẾN TRÚC CTRÌNH

- Tập trung hướng giao dịch
- Bước 1: Với luồng vào
 - Dõi theo các luồng dữ liệu vào và vượt qua các chức năng biến đổi thông tin sơ bộ cho đến khi
 - Hoặc các dữ liệu đó trở thành dữ liệu vào ở dạng trừu tượng nhất
 - Hoặc không còn xem chúng là dữ liệu vào nữa
 - Thì đánh dấu ngắt luồng vào
- Bước 2: Với luồng ra
 - Đi ngược dòng vượt qua các chức năng biến đổi thông tin cho đến khi
 - không xem đó là dữ liệu ra được nữa
 - thì đánh dấu ngắt luồng ra

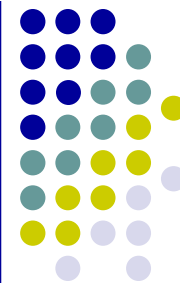
THIẾT KẾ KIẾN TRÚC CTRÌNH



- Tập trung hướng giao dịch
- Bước 3: căn cứ vào các điểm đánh dấu khoanh vùng các xử lý còn lại đây là trung tâm biến đổi.
 - Ví dụ



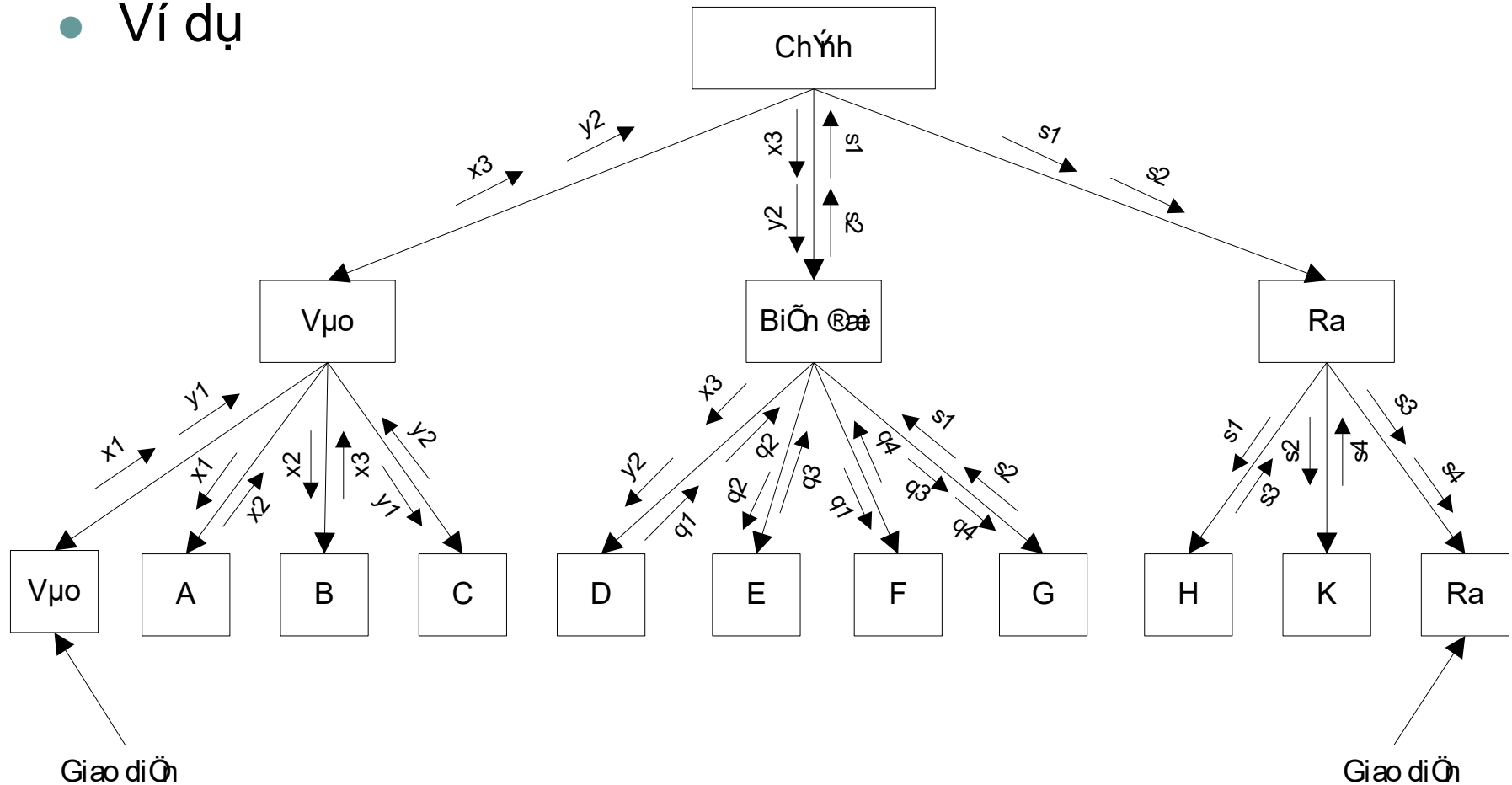
THIẾT KẾ KIẾN TRÚC CTRÌNH



- Tập trung hướng giao dịch
- Bước 4: Vẽ lược đồ chương trình ở hai mức cao nhất
 - Mức 1: một modul chính
 - Mức 2: gồm 3 modul
 - Một modul **vào** cho các luồng dữ liệu vào
 - Một modul **ra** cho các luồng dữ liệu ra
 - Một modul cho trung tâm biến đổi
- Bước 5: Triển khai mỗi modul ở mức 2 xuống mức thấp hơn và làm xuất hiện dần các modul tương ứng với các chức năng xử lý trong DFD



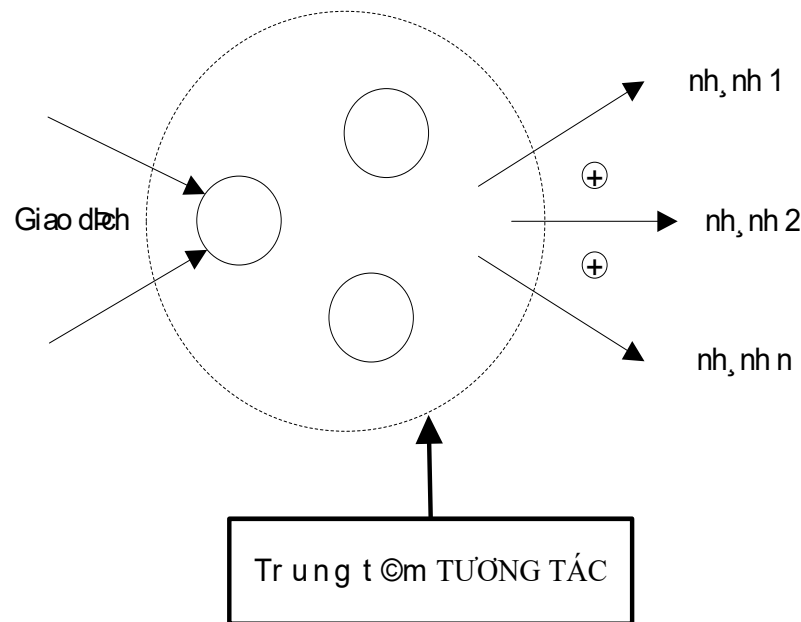
- Ví dụ



THIẾT KẾ KIẾN TRÚC CTRÌNH



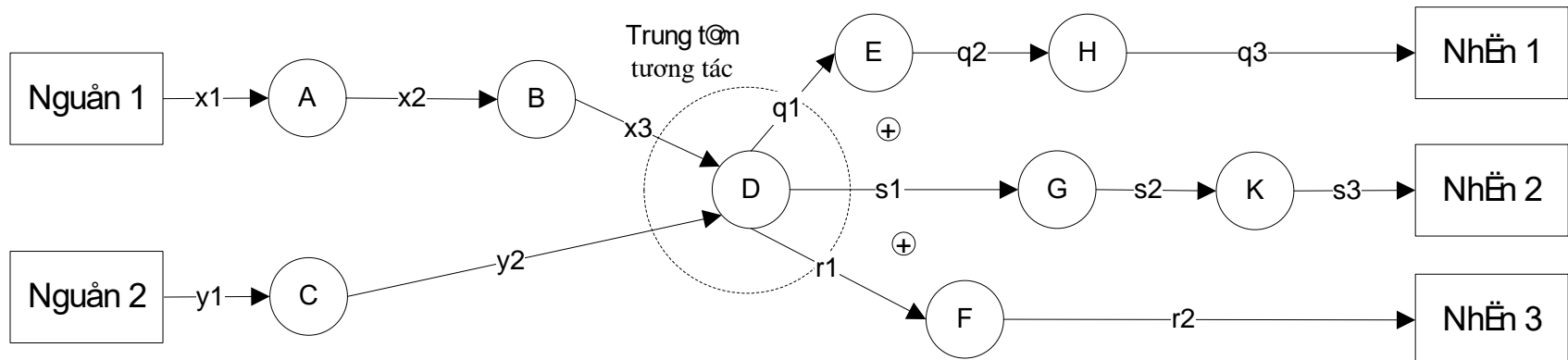
- Tập trung hướng tương tác
 - Phát hiện trung tâm tương tác thông tin chủ yếu



THIẾT KẾ KIẾN TRÚC CTRÌNH



- Tập trung hướng tương tác
- Bước 1: Xác định và khoanh vùng các trung tâm tương tác trong DFD
- Bước 2: Xác định các loại tương tác khác nhau
 - tương ứng với các luồng ra của trung tâm tương tác
 - đồng thời xác định các chức năng được khởi động từ trung tâm tương tác đó.





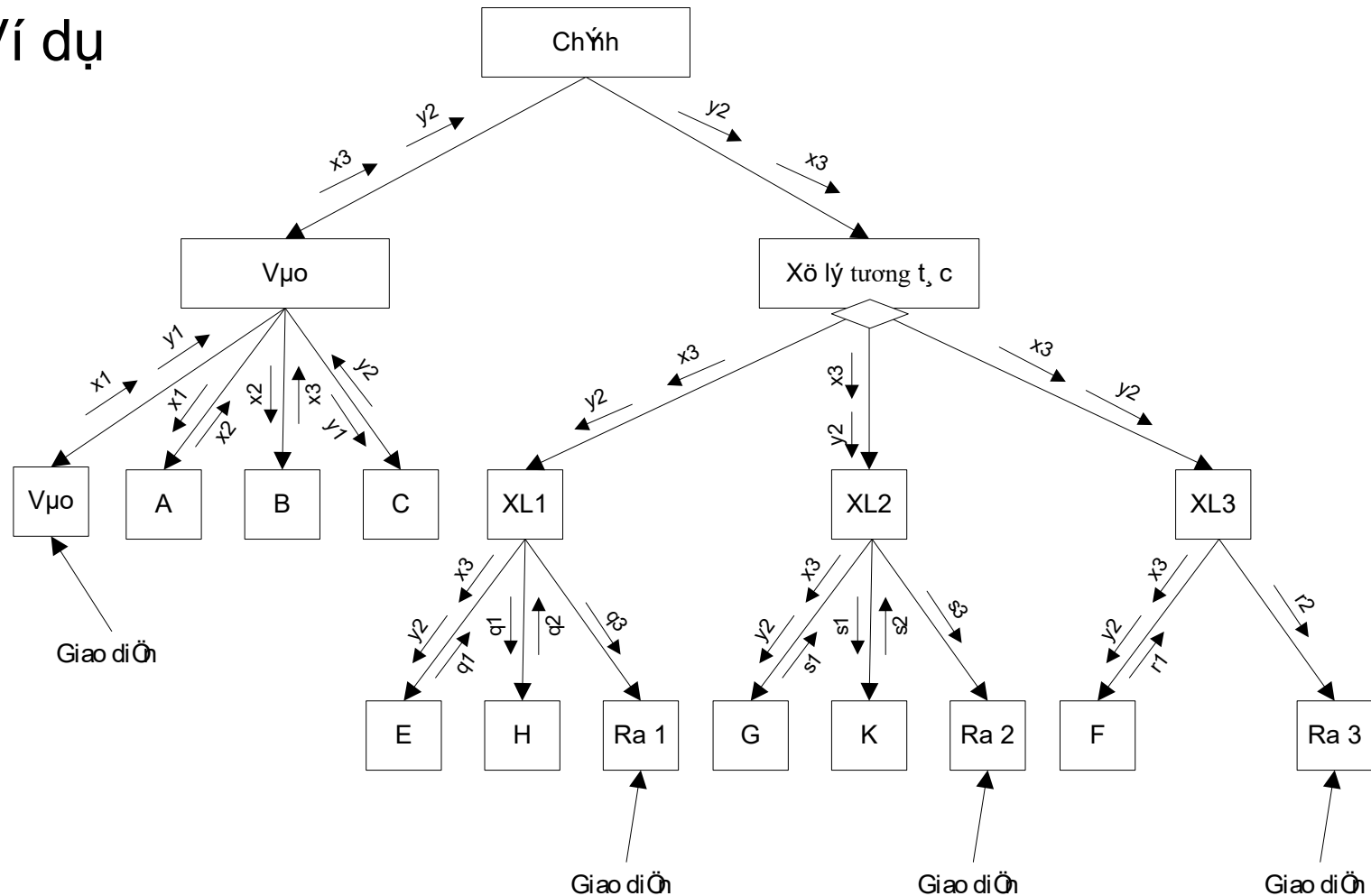
THIẾT KẾ KIẾN TRÚC CTRÌNH

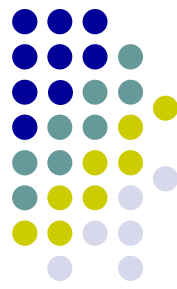
- Tập trung hướng tương tác
- Bước 3: Vẽ lược đồ chương trình ở hai mức cao nhất
 - Mức 1: một modul chính
 - Mức 2: gồm 2 modul
 - Một modul cho đầu vào tương tác
 - Một modul cho xử lý tương tác
- Bước 4: Triển khai mỗi modul ở mức 2 xuống mức thấp hơn,
 - mỗi xử lý một modul và được môđul xử lý tương tác gọi qua phép chọn.
 - Khi triển khai cần phát hiện các modul dùng chung.

THIẾT KẾ KIẾN TRÚC CTRÌNH

- Tập trung hướng tương tác

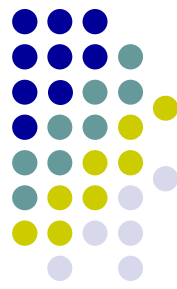
- Ví dụ





THIẾT KẾ KIẾN TRÚC CTRÌNH

- Lưu ý
- Không có tiêu chuẩn cụ thể về chất lượng của lược đồ chương trình, nên tuân theo các quy tắc
 - Giảm độ rộng bằng cách tăng độ sâu của lược đồ chương trình để tăng độ kết dính, giảm tính kết nối đơn thể.
 - Hạn chế tối đa phạm vi ảnh hưởng của đơn thể.
 - Hạn chế tối đa các dòng thông tin vào ra một đơn thể.
 - Tốt nhất là tạo các đơn thể một đầu vào, một đầu ra.
 - Quy tắc này cho phép giảm tính gắn kết, tăng tính độc lập chức năng của các đơn thể.



THIẾT KẾ KIẾN TRÚC CTRÌNH

- **Chất lượng thiết kế**

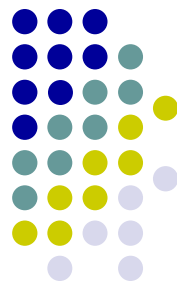
- Tiêu chuẩn một thiết kế tốt
 - Kết dính chặt chẽ
 - Ghép nối lỏng lẻo
- **Sự kết dính** là độ đo sự kết lại với nhau giữa các bộ phận trong một thành phần.
- Các loại kết dính có mức chặt chẽ tăng dần
 1. Kết dính gom nhóm
 2. Kết dính hội hợp logic
 3. Kết dính theo thời điểm
 4. Kết dính thủ tục
 5. Kết dính truyền thông
 6. Kết dính tuần tự
 7. Kết dính chức năng
 8. Kết dính đối tượng



THIẾT KẾ KIẾN TRÚC CTRÌNH

- **Chất lượng thiết kế**

- Sự ghép nối chỉ ra mức độ tương tác bên trong giữa các đơn vị thành phần của một chương trình
- Các loại kết dính có mức lỏng lẻo giảm dần
 1. Ghép nối dữ liệu
 2. Ghép nối nhãn
 3. Ghép nối điều khiển
 4. Ghép nối chung
 5. Ghép nối nội dung

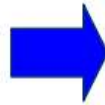


THIẾT KẾ KIẾN TRÚC CTRÌNH

● Nguyên tắc và hướng dẫn cho 1 thiết kế tốt

Các nguyên tắc thiết kế

- Môđun hoá
- Kiểm soát chức năng
- Độc lập tương đối
- Kích thước hợp lý
- Mỗi môđun chỉ thực hiện 1 và chỉ 1 chức năng
- Một môđun có thể sử dụng lại nhiều lần



Hướng dẫn thiết kế

- Chia thành các nhân tố
- Mở rộng kiểm tra
- Đảm bảo ghép nối lỏng
- Chọn kích cỡ hợp lý
- Đảm bảo kết dính chặt

- Tạo các môđun sử dụng chung, sử dụng lại



THIẾT KẾ KIẾN TRÚC CTRÌNH

- Đặc tả modul chương trình
 - Phương tiện sử dụng
 - Sơ đồ khối
 - Ngôn ngữ có cấu trúc
 - Ngôn ngữ có cấu trúc: ba cấu trúc cơ sở
 - **Tuần tự**: các câu lệnh mô tả cho hoạt động của chức năng được viết lần lượt
 - **Tuyển chọn**: chọn 1 trong nhiều hành động
 - If ... then ... else ... hoặc “Nếu ... thì ... ngược lại ...”
 - **Lặp**: các hành động được lặp lại theo một số lần
 - For each tương ứng “Với mỗi”

Đầu vào : đơn hàng

Đầu ra : tình trạng

KIỂM TRA KHẢ NĂNG ĐÁP ỨNG MẶT HÀNG (đơn hàng, tình trạng)

FOR EACH mặt hàng trong đơn hàng

BEGIN đọc tệp **IÒN KHO**, sử dụng SHMH làm khoá

Tình trạng = 'OK'

IF lượng đáp ứng < Lượng yêu cầu

THEN BEGIN Gửi 'thông báo số 1' cho người dùng

CASE trả lời

+ Huỷ đơn hàng

Thực hiện huỷ đơn hàng

Tình trạng = 'Huỷ'

+ Rút bớt số lượng

Lượng yêu cầu = Lượng yêu cầu - lượng cắt giảm

Tình trạng = 'Giảm'

+ Bỏ mặt hàng

Loại mặt hàng khỏi đơn hàng

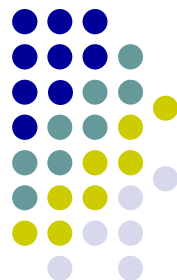
Tình trạng = 'cắt mặt hàng'

END

END

END

Lưu đơn hàng vào tệp **ĐƠN HÀNG**





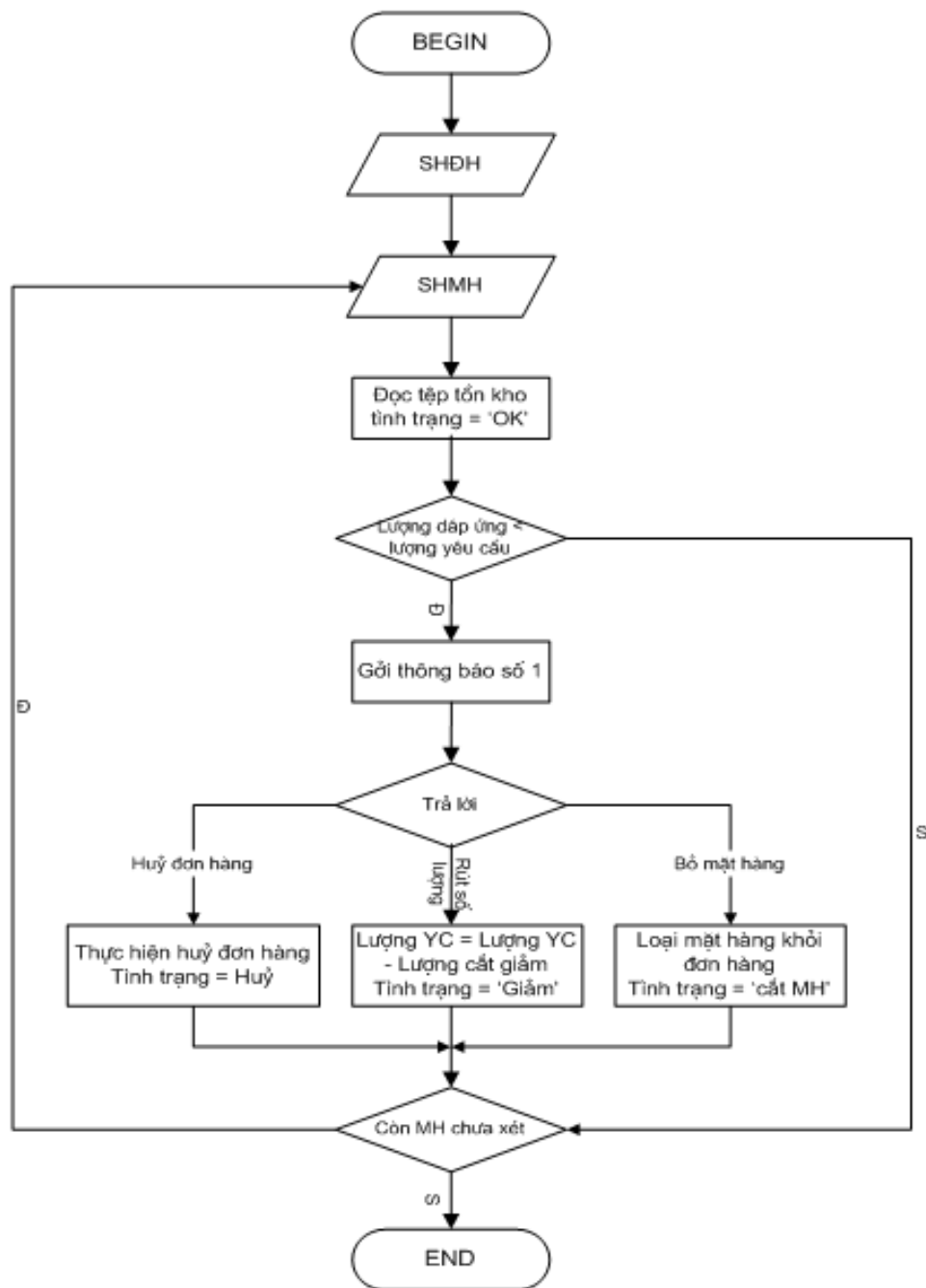
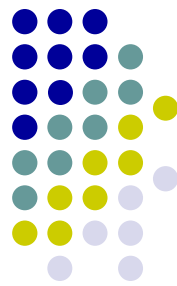
THIẾT KẾ KIẾN TRÚC CTRÌNH

- Đặc tả modul chương trình
 - Sơ đồ khối

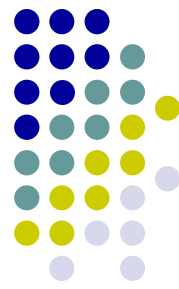
Một số khối cơ bản

- Khối bắt đầu/kết thúc
- Khối vào/ra dữ liệu
- Khối rẽ nhánh
- Khối thực hiện
- Chỉ dẫn





THẢO LUẬN



- Quy tắc thiết kế CSDL
- Quy trình thiết kế kiến trúc chương trình

