

# TIẾN TRÌNH PHẦN MỀM

## PHẦN II.2 – THIẾT KẾ



1

### Nội dung

- Định nghĩa về thiết kế
- Các nội dung thiết kế
- Một số vấn đề trong thiết kế
- Đặc trưng của thiết kế hoàn thiện
- Tài liệu thiết kế



2

## Định nghĩa về thiết kế



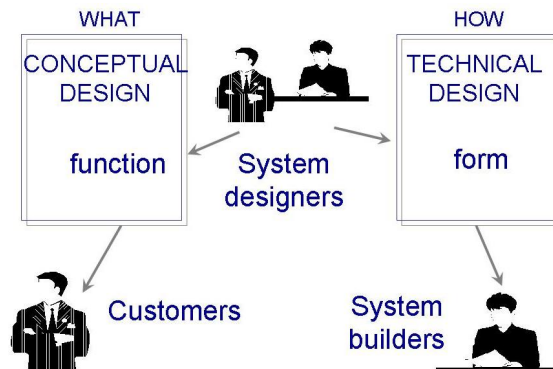
- Thiết kế là quá trình sáng tạo thực hiện việc chuyển đổi vấn đề thành giải pháp.
- Sự mô tả của một giải pháp còn được biết như là thiết kế.
  - Đặc tả các yêu cầu **định nghĩa vấn đề**.
  - Tài liệu thiết kế **xác định một giải pháp** cụ thể cho vấn đề.

3

## Định nghĩa về thiết kế



- **Thiết kế** là một quá trình tương tác gồm hai phần:
  - Thiết kế mức quan niệm (thiết kế hệ thống)
  - Thiết kế kỹ thuật



4

## Định nghĩa về thiết kế



- **Thiết kế mức quan niệm** nói với khách hàng **những cái** mà hệ thống phải làm:
  - Dữ liệu đến từ đâu?
  - Điều gì sẽ xảy ra với dữ liệu trong hệ thống?
  - (Đối với người sử dụng) Hệ thống trông giống cái gì?
  - Những lựa chọn nào sẽ được cung cấp cho người dùng?
  - Các báo cáo và màn hình giống cái gì?
  - Định thời gian cho các sự kiện?

5

## Định nghĩa về thiết kế



- **Thiết kế mức quan niệm**  
Các đặc trưng của một thiết kế mức quan niệm hoàn thiện:
  - Theo ngôn ngữ mà khách hàng có thể hiểu
  - Không có các từ kỹ thuật
  - Mô tả các chức năng của hệ thống
  - Độc lập với sự cài đặt
  - Được liên kết với các yêu cầu

6

## Định nghĩa về thiết kế



- **Thiết kế kỹ thuật** nói với lập trình viên những cái mà hệ thống phải làm như:
  - Các thành phần phần cứng chính và chức năng của chúng.
  - Sự phân cấp và các chức năng của các thành phần phần mềm.
  - Các cấu trúc dữ liệu và dòng dữ liệu.

7

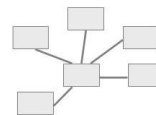
## Định nghĩa về thiết kế



- Sự khác nhau trong tài liệu thiết kế

"The user will be able to route messages to any other user on any other network computer."

CONCEPTUAL  
DESIGN



Network topology  
Protocol used  
Prescribed bps rate  
...

TECHNICAL  
DESIGN

8

## Định nghĩa về thiết kế



- Thiết kế một hệ thống là xác định một tập các thành phần và các giao diện giữa các thành phần để đáp ứng tập các yêu cầu được đặc tả.
- Những phương pháp tạo ra thiết kế
  - Phân rã theo mô đun
  - Phân rã hướng dữ liệu
  - Phân rã hướng sự kiện
  - Thiết kế trong ngoài
  - Thiết kế hướng đối tượng

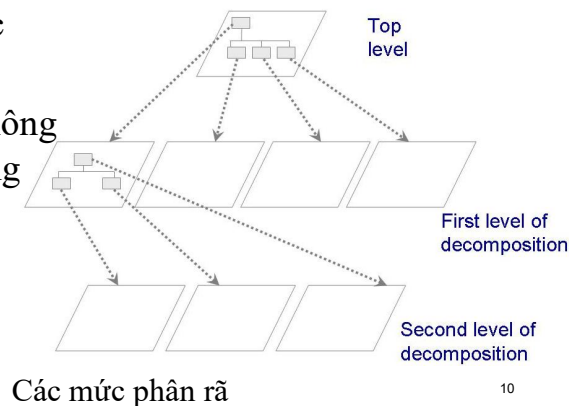
9

## Định nghĩa về thiết kế



Sự phân rã

- Mô tả dữ liệu hệ thống
- Mô tả chức năng mức cao
- Tạo ra sự phân cấp thông tin với các chi tiết tăng dần



10

## Định nghĩa về thiết kế



Tính mô đun hóa

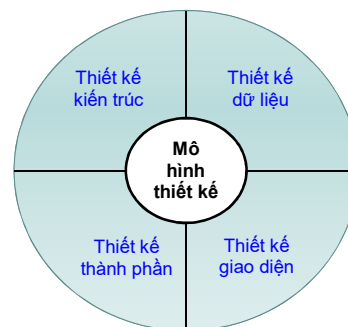
- Mô đun hay thành phần (component): các bộ phận hợp lại của thiết kế.
- Một hệ thống là có tính mô đun khi
  - Mỗi hoạt động của hệ thống được thực hiện bởi chỉ một thành.
  - Đầu vào và đầu ra của mỗi thành phần là rãnh mạch
    - Tất cả các đầu vào là cần thiết cho chức năng của nó.
    - Tất cả các kết xuất được tạo ra bởi một trong các hoạt động của nó.

11

## Các nội dung thiết kế



- Nhà thiết kế thực hiện các công việc:
  - Thiết kế kiến trúc
  - Thiết kế dữ liệu
  - Thiết kế giao diện
  - Thiết kế thủ tục (thuật toán)



12

# Thiết kế kiến trúc



- Thiết kế kiến trúc

- Cung cấp một khung nhìn toàn diện về hệ thống cần xây dựng.
- Một loại kiến trúc phần mềm liên quan tới các thành phần, các liên kết và các ràng buộc trên các thành phần kết hợp.
- Liên kết các thành phần của hệ thống với các khả năng đã được xác định trong đặc tả yêu cầu.

13

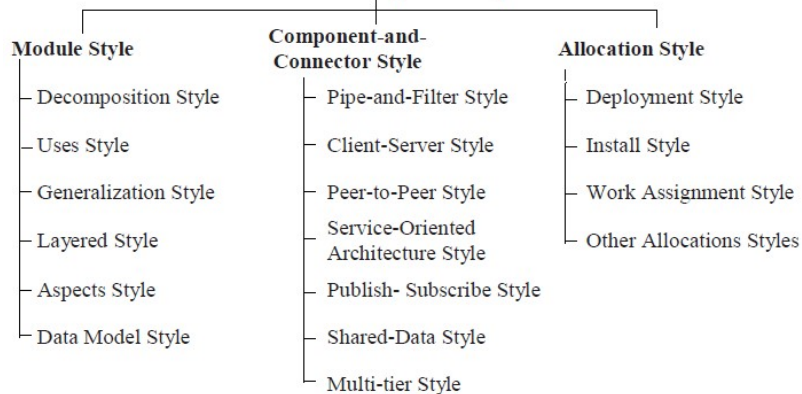
# Thiết kế kiến trúc

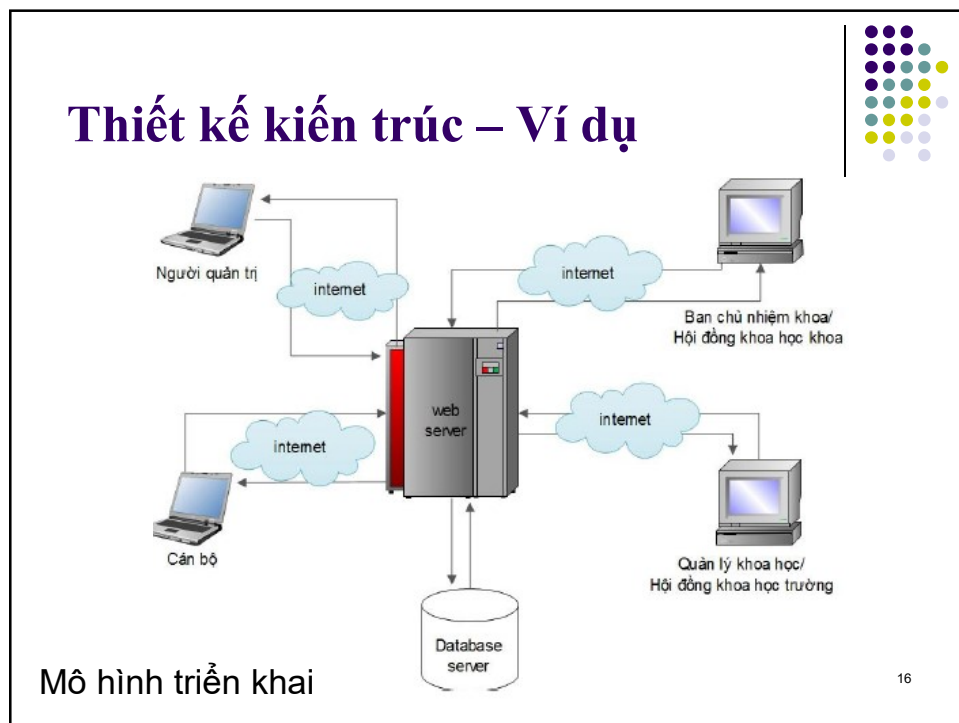
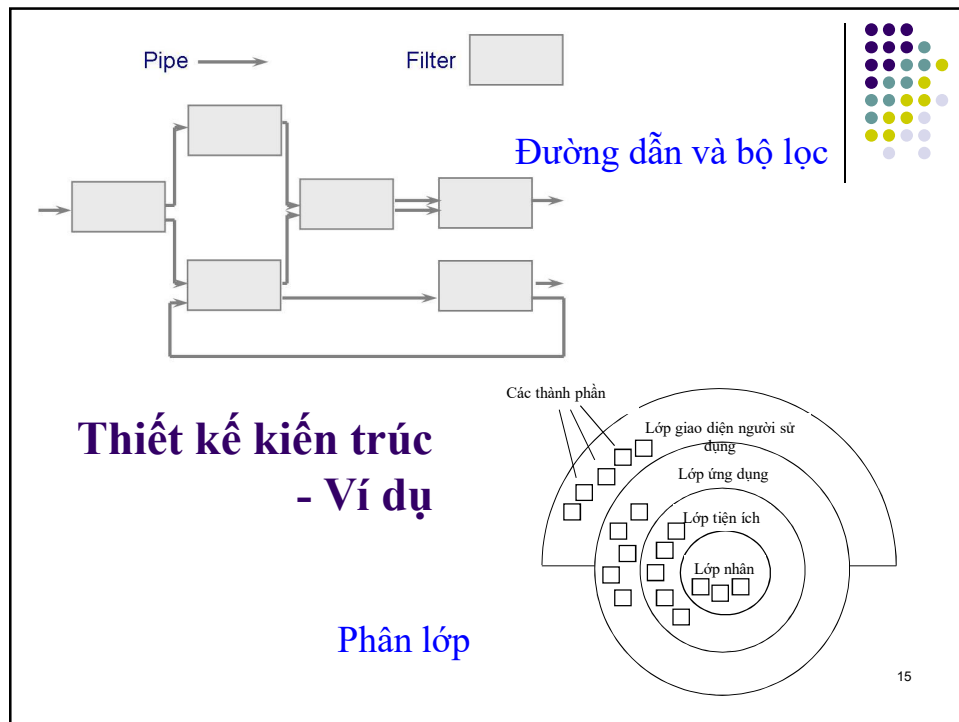


- Thiết kế kiến trúc

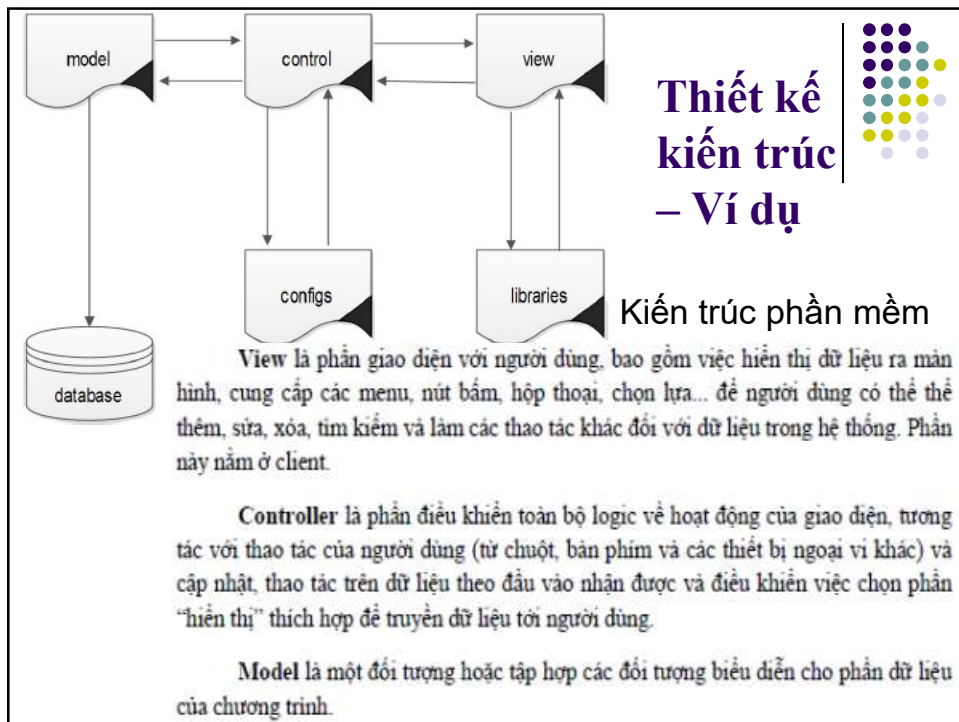
- Các loại kiến trúc phần mềm

## Các cấu trúc kiến trúc phần mềm phổ biến





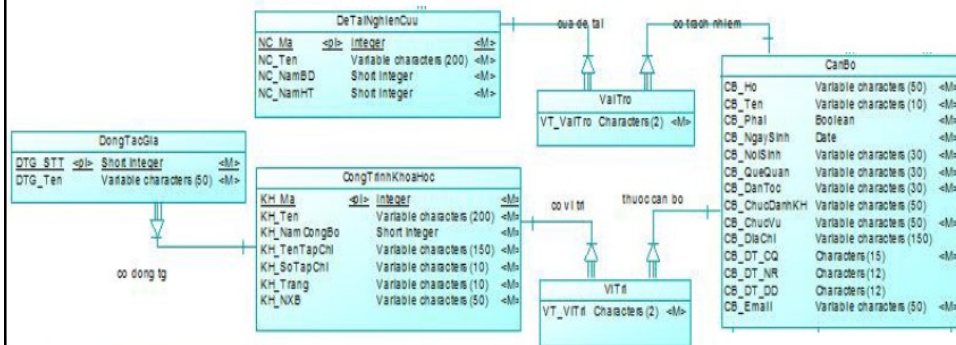




## Thiết kế dữ liệu

- **Thiết kế dữ liệu**
  - Các thành phần dữ liệu và bảng để tạo CSDL.
  - Các cấu trúc dữ liệu.
- **Các mức thiết kế dữ liệu**
  - Thiết kế cấu trúc logic: các quan hệ chuẩn, các khóa, các tham chiếu, các cấu trúc thao tác dữ liệu.
  - Thiết kế cấu trúc vật lý: các file, các kiểu, kích thước.

## Thiết kế dữ liệu – Ví dụ



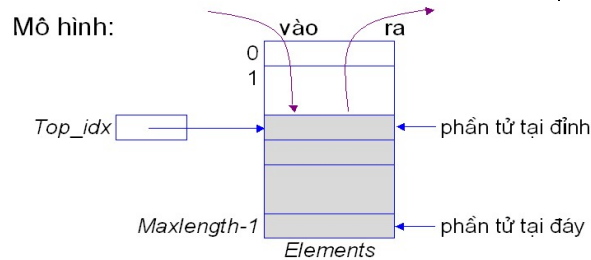
19

## Thiết kế dữ liệu – Ví dụ

STT	Tên trường	Kiểu	Kích thước	Khóa chính	Not null	RBTV khóa ngoài	Diễn giải
1.	KH_ma	int	5	x	x		Mã công trình khoa học
2.	KH_ten	varchar	200		x		Tên công trình khoa học
3.	KH_namcongbo	smallint			x		Năm công bố
4.	KH_tentapchi	varchar	150		x		Tên tạp chí
5.	KH_sotapchi	varchar	10		x		Số tạp chí
6.	KH_trang	varchar	10		x		Trang
7.	KH_nxb	varchar	50		x		Nhà xuất bản

20

## Thiết kế dữ liệu – Ví dụ



- Khai báo

```
#define Max_Length 100; //độ dài của mảng
typedef Element_Type float ;//kiểu phần tử của ngăn xếp
typedef struct {
    Element_Type Elements[Max_Length]; //Lưu nội dung của các phần tử
    int Top_Idex; //Chỉ số của đỉnh ngăn xếp
} Stack;
Stack Operator;
```

21

## Thiết kế giao diện

- Thiết kế giao diện
  - Các form nhập liệu.
  - Các reports và những kết xuất mà hệ thống phải sinh ra.

22






## Thiết kế giao diện – Ví dụ

**Hình 1: Giao diện Đăng Nhập**

STT	Loại Điều Khiển	Tên Điều Khiển	Nội dung thực hiện	Giá trị mặc định
1.	Form	frmDangNhap	Tạo giao diện đăng nhập hệ thống	
2.	DotNetBar.ButtonX	btHinhAnh	Hiện thị ảnh	
3.	DotNetBar.TextBoxX	txtTaiKhoan	Cho phép nhập	
4.	DotNetBar.TextBoxX	txtMatKhau	Cho phép nhập	
5.	DotNetBar.ButtonX	btDangNhap	Thực hiện kiểm tra sự hợp lệ của người dùng	
6.	DotNetBar.ButtonX	btHuyBo	Đóng frmDangNhap	
7.	StatusStrip	statusTrangThai	Hiện thị trạng thái phím Caps Lock	

## Thiết kế giao diện – Ví dụ



STT	Tên Bảng	Phương thức			
		Truy vấn	Thêm	Sửa	Xóa
1.	NGUOI_SU_DUNG_CHUONG_TRINH	✓			

STT	Tên Điều Khiển	Ràng buộc
1.	txtTaiKhoan	Chỉ cho phép nhập ký tự số, chiều dài là 3.
2.	txtMatKhau	Hiện thị với ký tự đặc biệt (*), chiều dài là 30.
3.	btDangNhap	Chỉ bật khi txtTaiKhoan và txtMatKhau có nội dung.

24

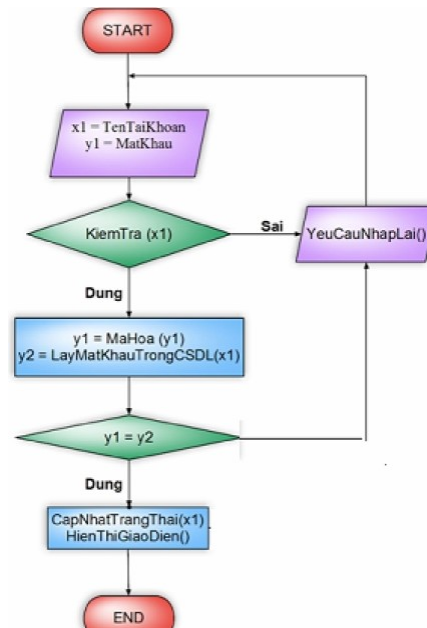
## Thiết kế thủ tục



- Thiết kế thủ tục (thuật toán)
  - Giải thích quá trình xử lý từ input đến output.
- Phương pháp biểu diễn
  - Lưu đồ giải thuật.
  - Ngôn ngữ giả.

25

## Thiết kế thủ tục – Ví dụ



Lưu đồ xử lý khi nhấn nút “Đăng nhập”

26

## Một số vấn đề trong thiết kế



- Thiết kế cộng tác
  - Hầu hết các dự án làm việc cộng tác.
  - Các vấn đề trong thiết kế cộng tác
    - Ai là người phù hợp nhất để thiết kế từng bộ phận của hệ thống.
    - Viết tài liệu thiết kế như thế nào.
    - Làm thế nào để kết hợp các thành phần thiết kế thành một thiết kế hợp nhất.
  - Các vấn đề trong việc thực hiện thiết kế cộng tác
    - Sự khác nhau về kinh nghiệm cá nhân, sự hiểu biết, sở thích.

27

## Một số vấn đề trong thiết kế



- Thiết kế giao diện
  - Các vấn đề then chốt được xem xét khi thiết kế giao diện:
    - Các vấn đề về văn hóa (quốc tịch, dân tộc, giới tính, nghề nghiệp, tuổi tác, vùng miền).
    - Sở thích của người dùng.
  - Một số lưu ý khi thiết kế giao diện:
    - Nên có sự đồng nhất giữa các giao diện (menu, lệnh, hiển thị...).
    - Đặt tên nhãn ngắn gọn, dễ nhớ.
    - Tối ưu trong trình bày hộp thoại và di chuyển chuột.

28

## Một số vấn đề trong thiết kế



- Một số lưu ý khi thiết kế giao diện
  - Hạn chế nhập dữ liệu trực tiếp, nếu có thể, nên cho người dùng chọn lựa từ một số dữ liệu có sẵn.
  - Yêu cầu xác nhận những tác vụ mang tính phá hủy (xoá dữ liệu).
  - Chấp nhận lỗi từ phía người sử dụng.
  - Nên cung cấp feedback cho người dùng.
  - Tạo ra thông báo lỗi có ý nghĩa.
  - Cung cấp trợ giúp.

29

## Một số vấn đề trong thiết kế



- Sự đồng thời
  - Các vấn đề
    - Tính nhất quán của dữ liệu được chia sẻ giữa các thành phần mà chúng thực thi tại cùng thời điểm.
    - Đảm bảo rằng một hoạt động không can thiệp vào các hoạt động khác.
  - Các giải pháp
    - *Sự đồng bộ*: phương pháp cho phép hai hoạt động thực hiện đồng thời mà không can thiệp vào nhau.
    - *Loại trừ lẫn nhau*: một quy trình truy xuất một phần tử dữ liệu, không có quy trình nào khác ảnh hưởng tới phần tử.
    - *Giám sát*: một đối tượng trừu tượng kiểm soát sự loại trừ lẫn nhau của một quy trình cụ thể.

30

## Các đặc trưng của một thiết kế hoàn thiện



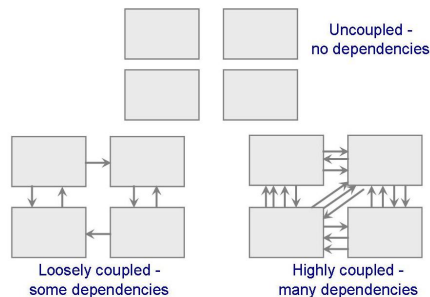
- Sự độc lập của thành phần
  - Sự nối kết (coupling): mức độ tương tác giữa các mô đun.
  - Sự gắn kết (cohesion): mức độ tương tác bên trong một mô đun.
- Nhận dạng và xử lý các ngoại lệ (Exception).
- Ngăn chặn và chấp nhận các lỗi trong giới hạn cho phép
  - Chủ động
  - Bị động

31

## Sự nối kết (coupling)



- Các thành phần *được nối kết cao* khi có một lượng lớn các phụ thuộc.
- Các thành phần *được nối kết lỏng lẻo* khi có một sự phụ thuộc nào đó nhưng sự kết nối lẫn nhau giữa các thành phần yếu.
- Các thành phần *không được nối kết* khi không có các quan hệ nào cả.

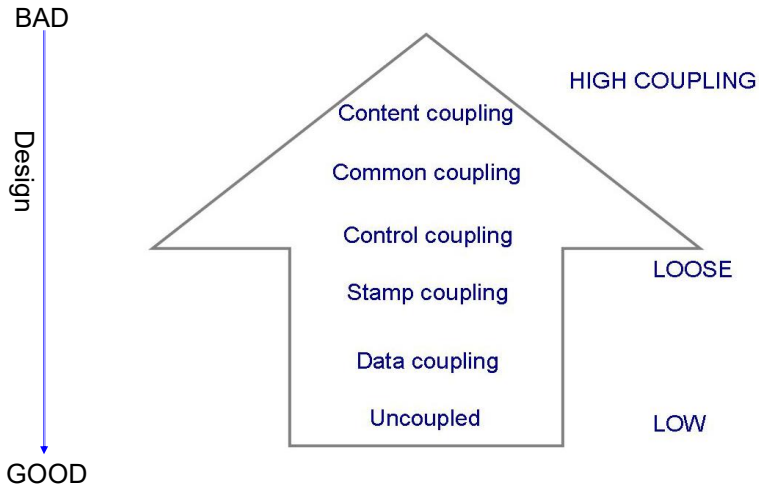


32



## Sự nối kết (coupling)

- Ta có thể đo sự nối kết theo mức độ phụ thuộc



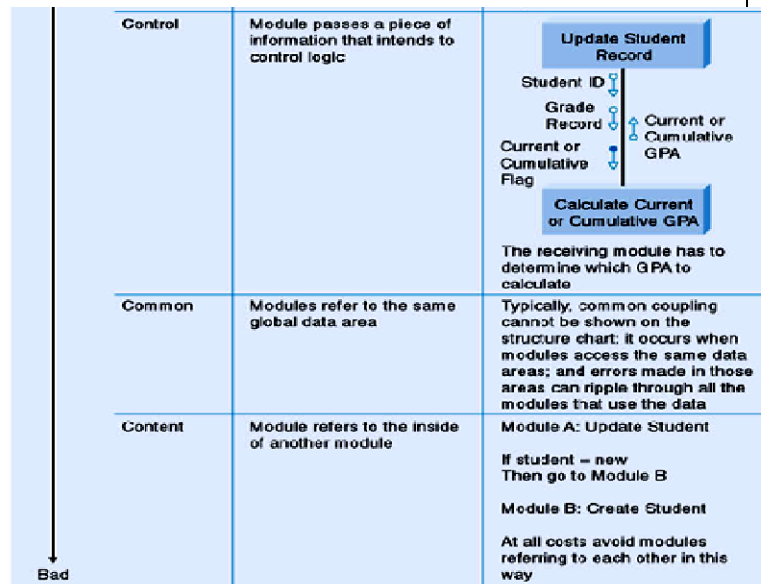
33

## Sự nối kết (coupling)

TYPE		DEFINITION	EXAMPLE
Good	Data	Modules pass fields of data or messages	<p>All couples that are passed are used by the receiving module</p>
	Stamp	Modules pass record structures	<p>The entire student record is not used by the receiving module, only the <i>student ID</i> field</p>

5 - 34

## Sự nối kết (coupling)



5 - 35

## Sự nối kết

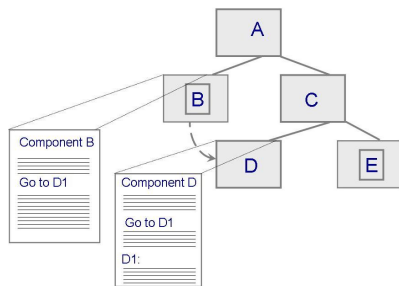
### • Các loại nối kết

- Nối kết nội dung (content): một thành phần sửa dữ liệu nội bộ của một thành phần khác hay một thành phần rẽ nhánh sang một thành phần khác.
- Nối kết chung (common): các thành phần truy xuất và làm thay đổi dữ liệu chung.
- Nối kết điều khiển (control): một thành phần truyền các tham số để điều khiển hoạt động của một thành phần khác.
- Nối kết cấu trúc dữ liệu (Stamp): cấu trúc dữ liệu được sử dụng để truyền thông tin từ một thành phần này sang một thành phần khác và bản thân cấu trúc dữ liệu được truyền đi.
- Nối kết dữ liệu (data): chỉ có dữ liệu được truyền từ một thành phần này sang một thành phần khác.

36

## Sự nối kết

- Ví dụ



Nối kết nội dung

Nối kết chung

Global: A1  
A2  
A3  
Variables: V1  
V2

Common data area  
and variable names

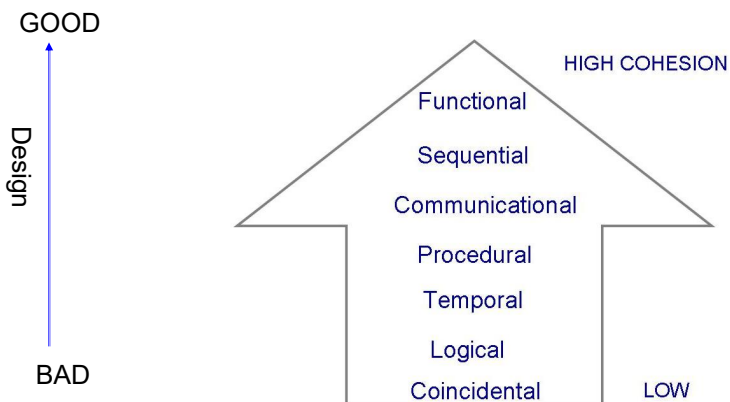
Component X  
Change  $V1$  to zero

Component Y  
Increment  $V1$

Component Z  
 $V1 = V2 + A1$

## Sự gắn kết (cohesion)

- Một thành phần là gắn kết nếu tất cả các thành viên của thành phần được điều khiển với mục đích và cần thiết để thực hiện cùng một công việc
- Một số dạng gắn kết



38

## Sự gắn kết (cohesion)

note: GPA =  
grade  
point  
average

Type	Definition	Example
Good Functional	Module performs one problem-related task	<div>Calculate Current GPA</div> <p>The module calculates current GPA only</p>
Sequential	Output from one task is used by the next	<div>Format and Validate Current GPA</div> <p>Two tasks are performed, and the formatted GPA from the first task is the input for the second task</p>
Communicational	Elements contribute to activities that use the same inputs or outputs	<div>Calculate Current and Cumulative GPA</div> <p>Two tasks are performed because they both use the student grade record as input</p>
Procedural	Elements are performed in sequence but do not share data	<div>Print Grade Listing</div> <p>The module includes the following: housekeeping, produce report</p>

5 - 39

## Sự gắn kết (cohesion)

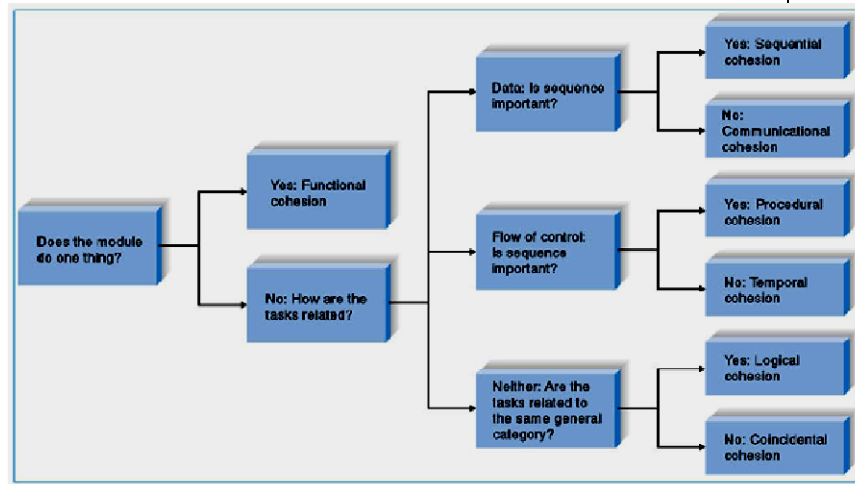
note: GPA =  
grade  
point  
average

Temporal	Activities are related in time	<div>Initialize Program Variables</div> <p>Although the tasks occur at the same time, each task is unrelated</p>
Logical	List of activities; which one to perform is chosen outside of module	<div>Perform Customer Transaction</div> <p>This module will open a checking account, open a savings account, or calculate a loan, depending on the message that is sent by its control module</p>
Coincidental	No apparent relationship	<div>Perform Activities</div> <p>This module performs different functions that have nothing to do with each other: update customer record, calculate loan payment, print exception report, analyze competitor pricing structure</p>

**FIGURE 11-14**  
Types of Cohesion (GPA = grade-point average)

5 - 40

## Cây quyết định sự gắn kết (cohesion) (Cohesion Decision Tree)



5 - 41

## Lưu ý thiết kế

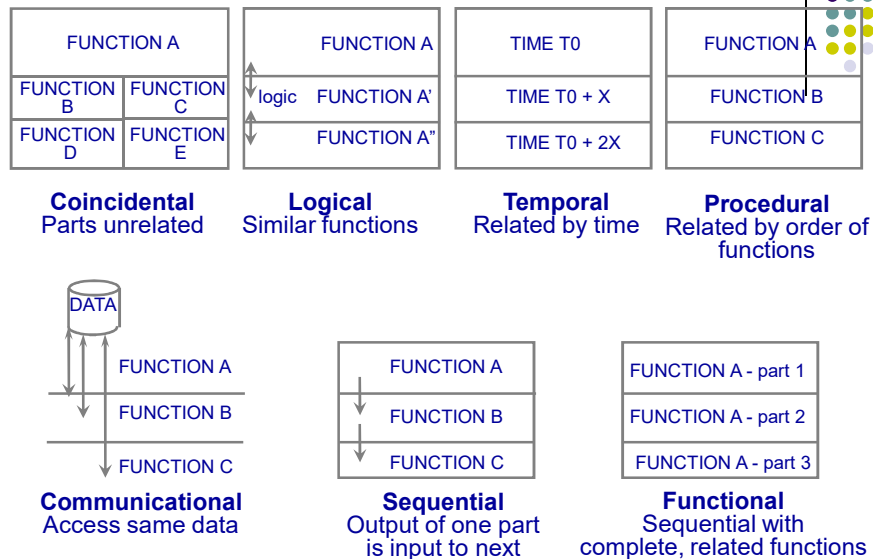


Quá trình xử lý với sự gắn kết "mức thấp"

- Phân chia các công việc thành các module khác nhau
- Hạn chế sử dụng cờ điều khiển

5 - 42

## Sự gắn kết (cohesion)



43

## Nhận dạng và xử lý ngoại lệ

- Các ngoại lệ: những tình huống ngược lại với cái mà ta thực sự muốn hệ thống làm. Các ngoại lệ điển hình gồm:
  - Không thực hiện được việc cung cấp một dịch vụ.
  - Cung cấp dữ liệu hay dịch vụ sai.
  - Làm hư dữ liệu.
- Các ngoại lệ có thể được xử lý theo một trong ba cách sau
  - Thử lại: phục hồi hệ thống về trạng thái trước đó và cố gắng thực hiện dịch vụ bằng một chiến lược khác.
  - Hiệu chỉnh: phục hồi hệ thống về trạng thái trước đó, hiệu chỉnh một mặt nào đó của hệ thống và cố gắng thực hiện lại bằng cùng một chiến lược.
  - Báo cáo: phục hồi hệ thống về trạng thái trước đó, báo cáo vấn đề với thành phần xử lý lỗi và không cung cấp dịch vụ.

44

## Ngăn chặn và chấp nhận các lỗi



- Phát hiện lỗi chủ động: định kỳ kiểm tra các dấu hiệu về lỗi hoặc cố gắng giải quyết trước khi lỗi xuất hiện.
- Phát hiện lỗi bị động: chờ cho đến khi lỗi xuất hiện trong suốt sự thực thi.
- Hiệu chỉnh lỗi: sự đền bù của hệ thống cho sự hiện diện của lỗi.
- Chấp nhận lỗi: cô lập những thiệt hại bị gây ra bởi lỗi.

45

## Viết tài liệu thiết kế



- Tài liệu thiết kế gồm các mục:
  - Nêu lý do cơ bản của thiết kế.
    - Phác thảo những vấn đề then chốt và các thỏa hiệp
  - Mô tả về các thành phần của hệ thống.
  - Xác định cách mà người dùng tương tác với hệ thống.
  - Tập các biểu đồ và ký pháp hình thức mô tả toàn bộ tổ chức và cấu trúc của hệ thống.

46

## Viết tài liệu thiết kế



- Xác định cách mà người sử dụng tương tác với hệ thống
  - Các menu và các định dạng màn hình hiển thị.
  - Giao diện người dùng: các phím chức năng, v.v.
  - Kết nhập: dữ liệu đến từ đâu, cách mà chúng được định dạng, chúng được lưu giữ trên phương tiện nào.
  - Kết xuất: dữ liệu đến từ đâu, cách mà chúng được định dạng, chúng được lưu giữ trên phương tiện nào.
  - Các đặc trưng chức năng chung.
  - Các ràng buộc về sự thực thi.
  - Các thủ tục lưu giữ.
  - Cách phương pháp xử lý lỗi.

47

## HẾT PHẦN II.2 THIẾT KẾ



48