Viện Đại học Mở Hà Nội
Khoa Công nghệ Điện tử - Thông tin

Lập trình ứng dụng

Thiết kế hệ thống

Phần 1: Giới thiệu chung

Nội dung chính

- Mục đích của thiết kế
- Các nguyên tắc thiết kế
- Các phần cần thiết kế
  - Thiết kế CSDL ➔ CSDL ít nhất ở dạng chuẩn 3
  - Thiết kế kiến trúc ➔ Lược đồ cấu trúc chương trình
  - Thiết kế giao diện ➔ các menu, form nhập, mẫu báo cáo, thông báo

Mục đích của giai đoạn Thiết kế

- Là quá trình chuyển các y/c của phần mềm sang dạng biểu diễn của phần mềm mà nó có thể được đánh giá về chất lượng trước khi cài đặt.
- Thiếu thiết kế, việc cài đặt có thể gặp các vấn đề:
  - **Thiếu kế hoạch cài đặt**: không biết rõ thứ tự cài đặt các thành phần, do đó gây ra sự lộn xộn và khó khăn trong việc ước lượng và phân công công việc
  - **Không rõ ràng**: chưa hiểu rõ các y/c sẽ được cài đặt thế nào
  - **Khó nâng cấp và bảo trì**: khi có lỗi, rất khó xác định nó nằm ở phần nào. Khi muốn nâng cấp cũng không biết cần nâng cấp ở đâu, ảnh hưởng của nó đến hệ thống hiện tại thế nào

➔ **Ảnh hưởng xấu đến chất lượng và tiến độ làm phần mềm**

Các nguyên tắc thiết kế

- Sự trừu tượng hóa (abstraction)
- Làm mịn (tinh chỉnh từng bước - refinement)
- Modul hóa (modularity)

Các nguyên tắc thiết kế

- Sự trừu tượng:
  - Là sự tập trung vào một vấn đề ở một mức khái quát nào đó, và bỏ qua các chi tiết không liên quan
  - Quá trình thiết kế hệ thống đòi hỏi nhiều mức trừu tượng khác nhau
  - Với phần mềm thì có 3 loại trừu tượng
    - Trừu tượng thủ tục
    - Trừu tượng dữ liệu
    - Trừu tượng điều khiển

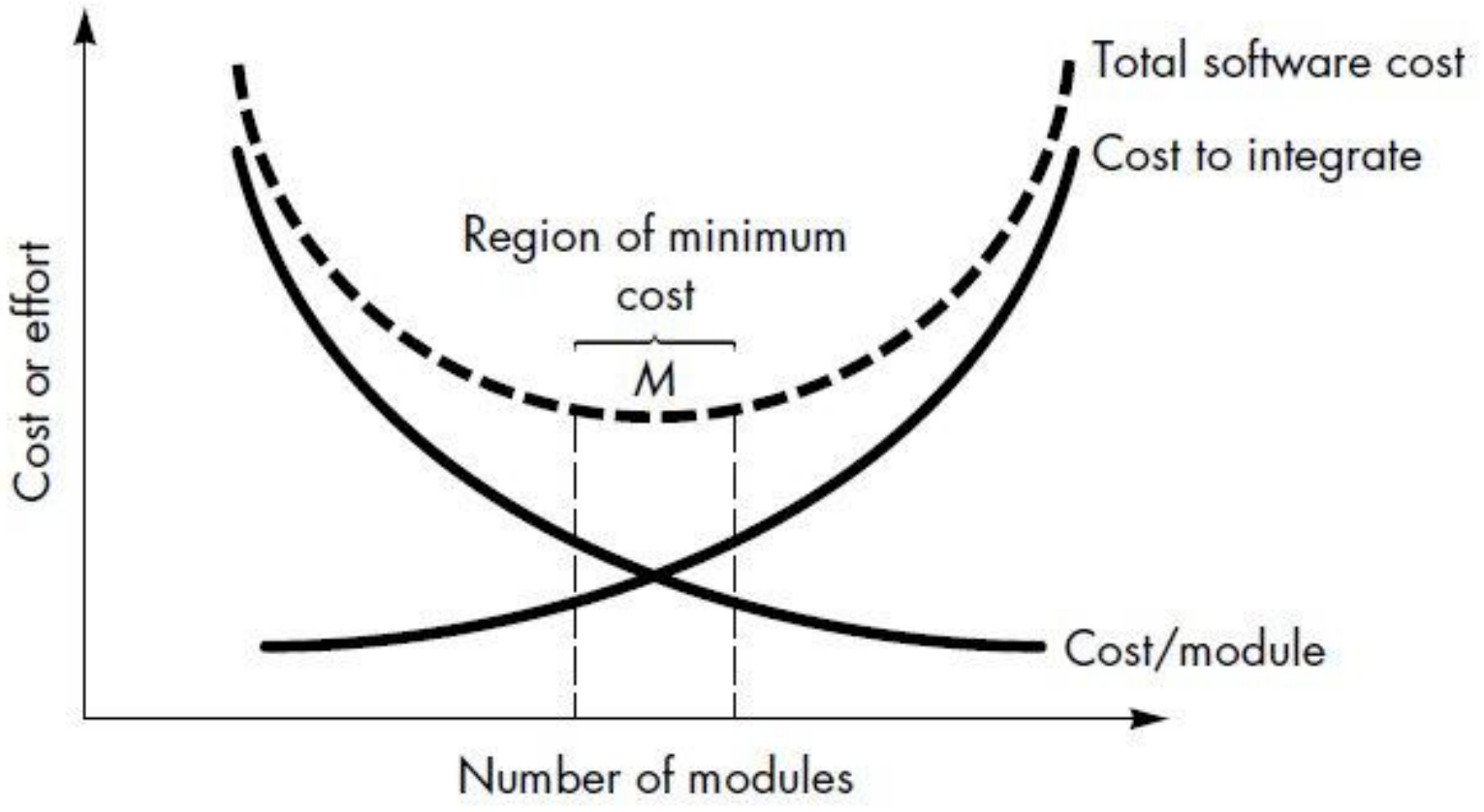
Các nguyên tắc thiết kế

- Làm mịn (tinh chỉnh từng bước - refinement)
  - Là quá trình làm chi tiết hóa từng thành phần của một biểu diễn nào đó, để dần đưa nó sang biểu diễn ở dạng chi tiết hơn (giảm mức độ trừu tượng)
  - Việc làm mịn giúp cho việc chuyển đổi này diễn ra một cách không đột ngột và dễ dàng quản lý.

Các nguyên tắc thiết kế

- Modul hóa (modularity):
  - Là quá trình phân chia hệ thống/phần mềm thành các thành phần riêng rẽ có tên và tương đối độc lập
  - Là một kỹ thuật cơ bản nhất để quản lý một cách hiệu quả độ phức tạp của hệ thống
  - Modul hóa tốt có thể giúp giảm thiểu thời gian và chi phí phát triển hệ thống

Modul hóa



Quan hệ giữa modul hóa và chi phí phần mềm

Module hóa hiệu quả

- Che dấu thông tin
  - Là cách thiết kế làm sao để thông tin trong một modul (cả chức năng và dữ liệu) là không nhìn thấy và không truy nhập được từ các thành phần bên ngoài mà không có nhu cầu về thông tin đó
- Độc lập chức năng (functional independence)
  - Là tính chất phản ánh mức độ đơn nhất về chức năng và đơn giản về giao diện của một modul. Nó được đo lường theo 2 tiêu chuẩn:
    - *Mức độ cố kết* (cohesion): càng cao càng tốt.
    - *Mức độ tương liên* (coupling): càng thấp càng tốt.

Mức độ cố kết

- Khái niệm:

  Mức độ cố kết của một modul là một đơn vị đo về sức mạnh chức năng của modul đó. Mức độ này càng cao thì tính độc lập chức năng cũng càng cao.

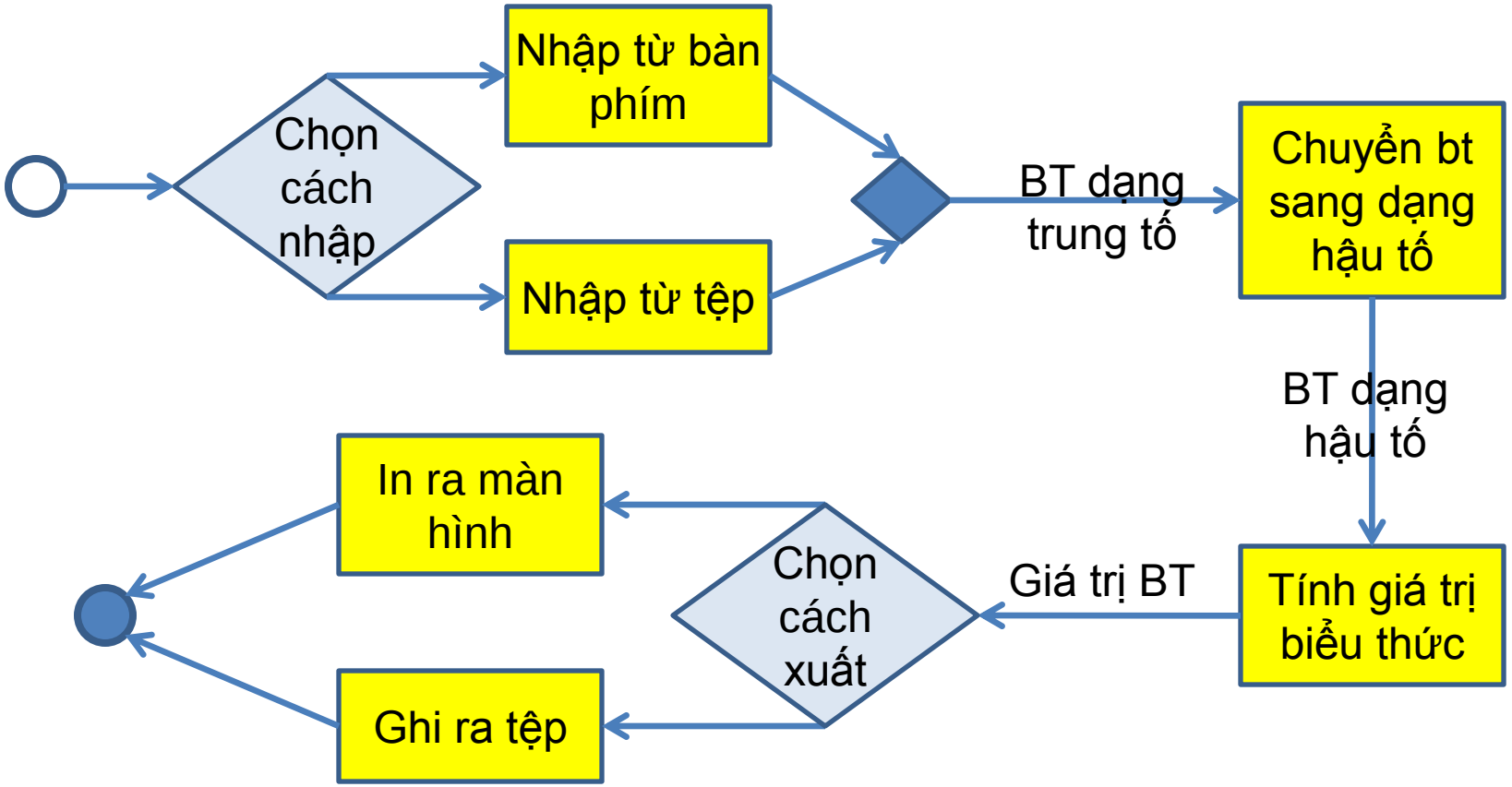
Các loại cố kết và mức độ của chúng

Mức độ	Loại cố kết	Ý nghĩa
Thấp	Cố kết trùng khớp	Modul bao gồm một dãy các công việc mà liên quan rất ít đến nhau
	Cố kết logic	Modul bao gồm một dãy các công việc mà có liên quan đến nhau một cách logic
	Cố kết thời gian	Modul bao gồm một dãy các công việc mà phải hoàn thành trong cùng một khoảng tg.
Vừa	Cố kết thủ tục	Các công việc trong modul đó liên quan đến nhau và phải được thực hiện theo một trật tự nhất định
	Cố kết truyền thông	Khi các công việc trong một modul cùng sử dụng một phần nào đó của một cấu trúc dữ liệu
Cao	Cố kết thủ tục rõ ràng	Khi modul đó chỉ thực hiện một công việc

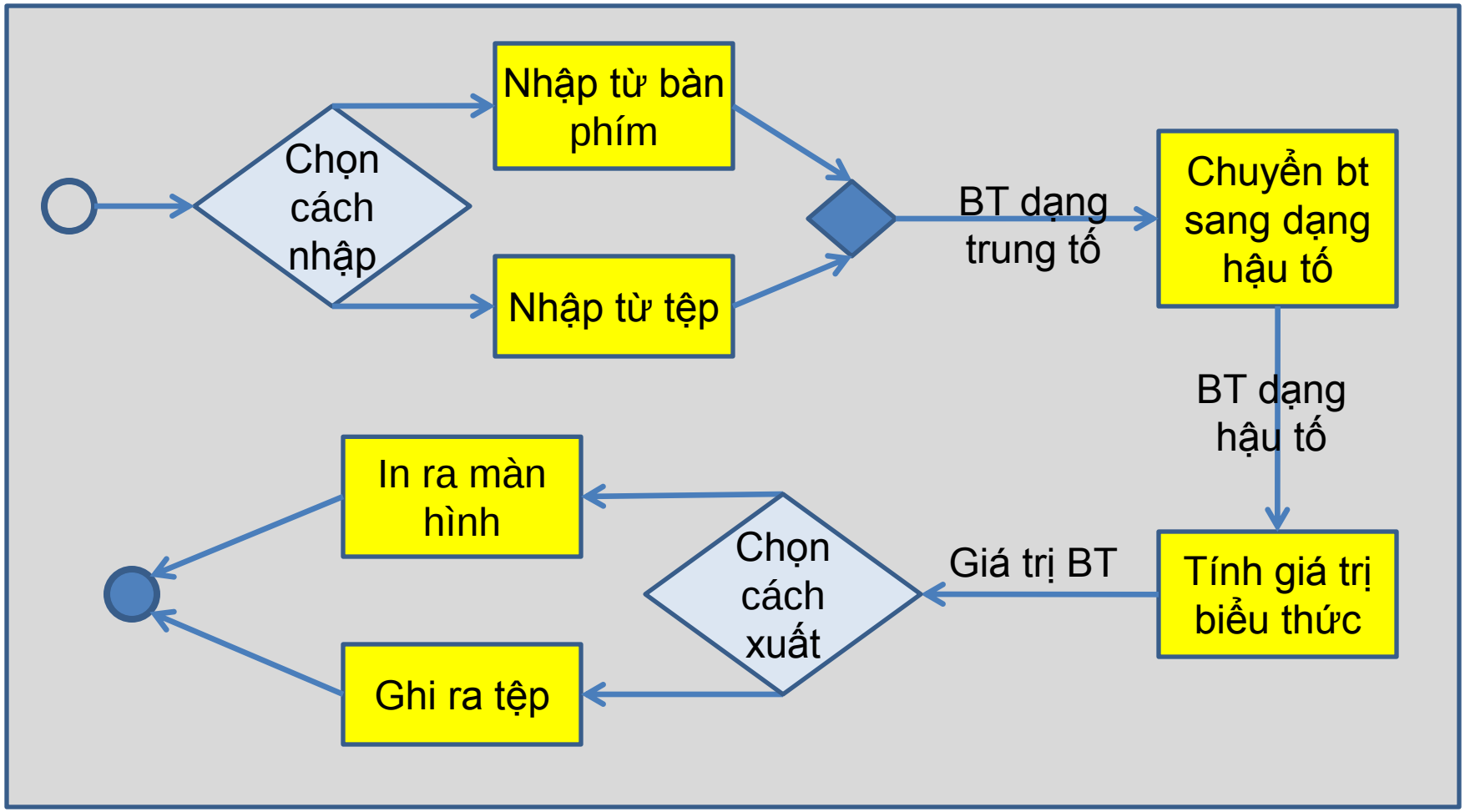
Ví dụ về mức độ cố kết

- Bài toán: viết một chương trình tính giá trị một biểu thức số học mà có thể được nhập từ bàn phím hay từ một tệp văn bản. Kết quả đưa ra cũng có thể đưa ra màn hình hoặc ghi vào tệp văn bản.
- Sơ đồ cho giải thuật của bài toán trên được cho ở hình sau:

Ví dụ về mức độ cố kết

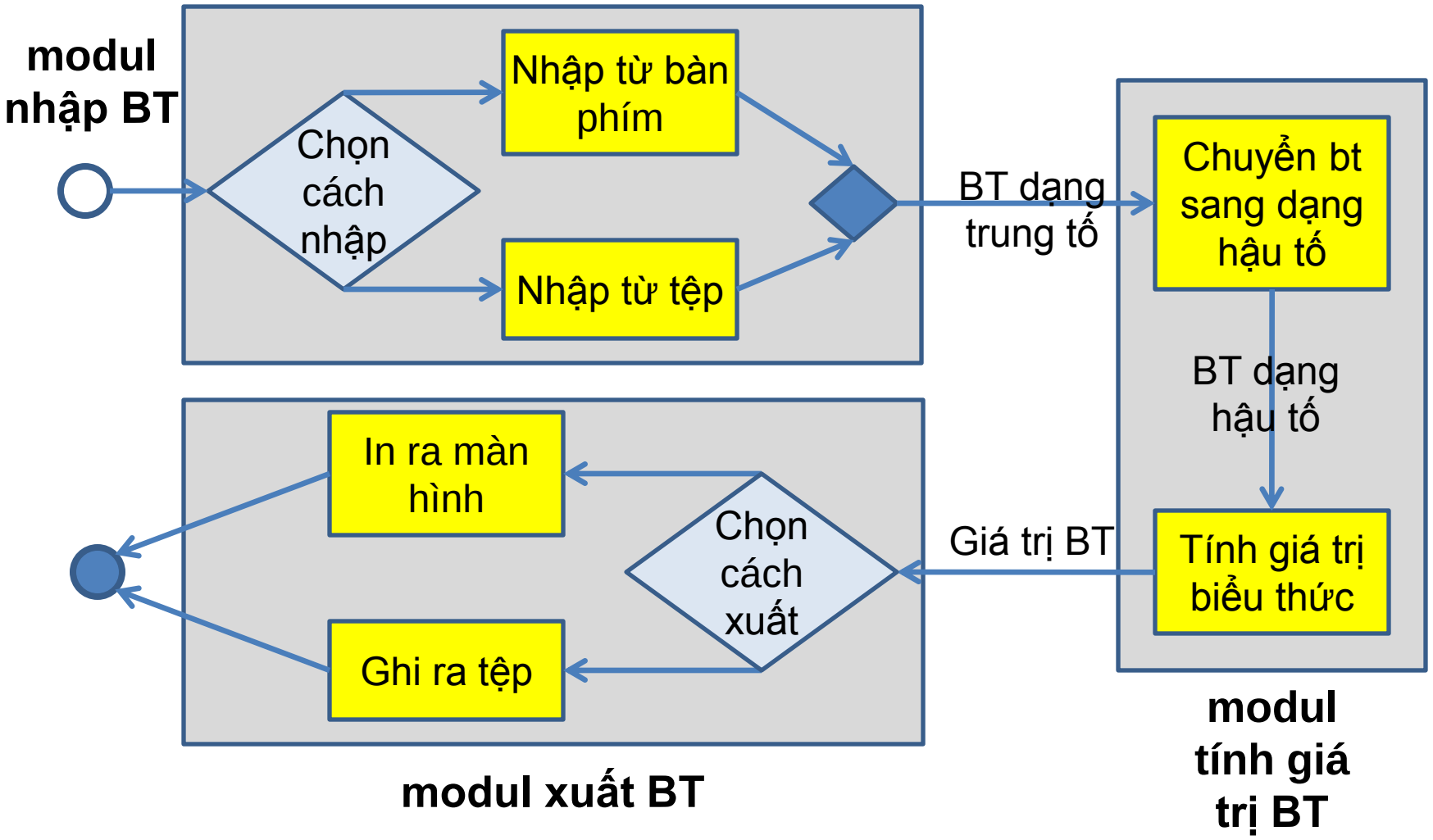


Mức độ cố kết thấp: trùng khớp

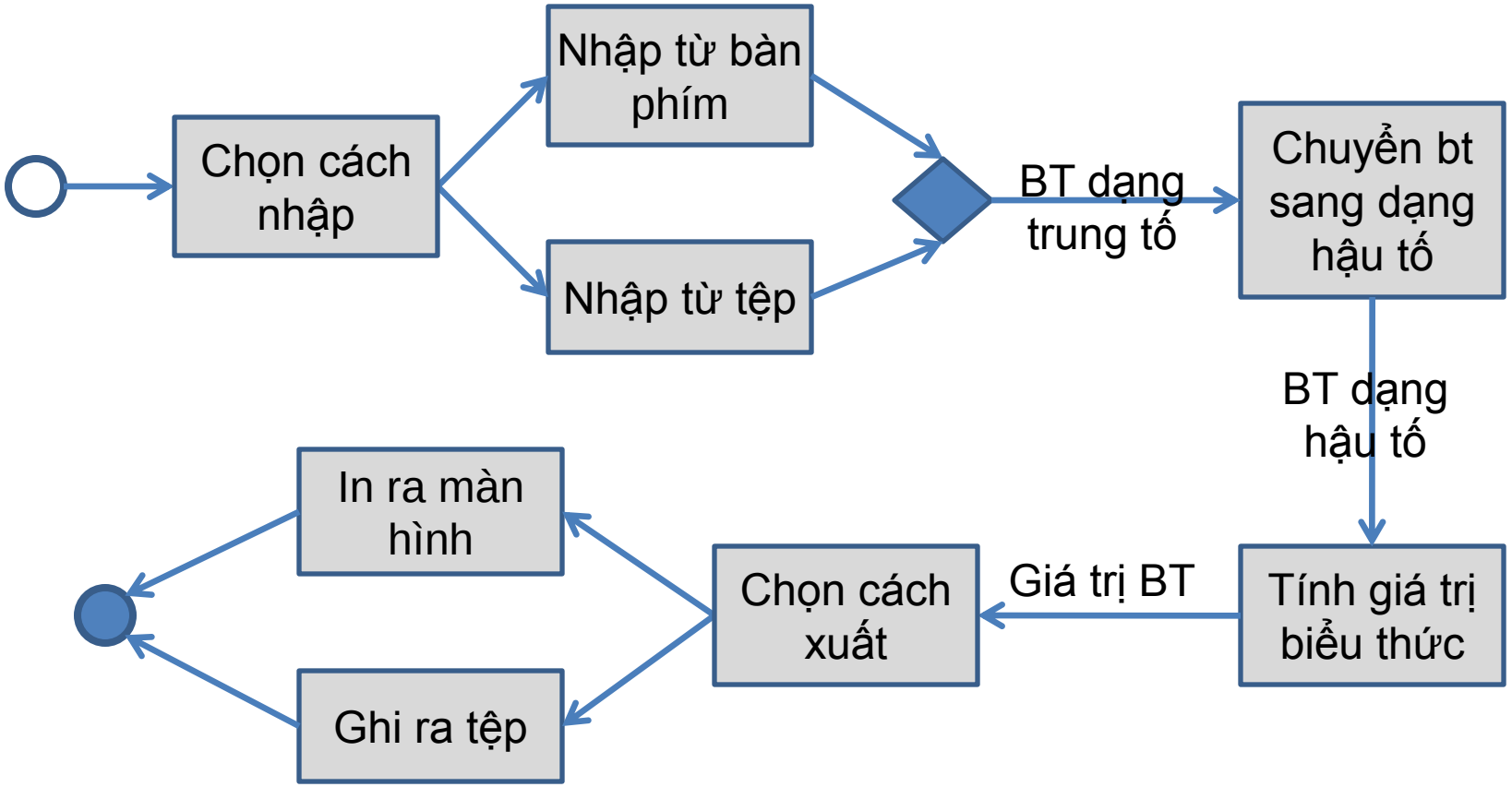


Một modul làm toàn bộ các công việc

Mức độ cố kết vừa: thủ tục



Mức độ cố kết cao: thủ tục rõ ràng



Mức độ tương liên

- Khái niệm:

  Mức độ tương liên của một modul đơn vị đo lường mức độ kết nối của modul đó với các modul khác. Điều này phụ thuộc vào độ phức tạp của giao diện, điểm truy nhập hay tham chiếu của modul.

Mức độ tương liên

Mức độ	Loại	Ý nghĩa
Thấp	Tương liên dữ liệu	Là khi một modul truyền tham số đến modul khác
Vừa	Tương liên điều khiển	Là khi một modul truyền một thông tin điều khiển (cờ điều khiển) đến modul khác
	Tương liên ngoài	Là khi một modul phụ thuộc vào một thiết bị bên ngoài (như các t/b nhập/xuất)
Cao	Tương liên chung dữ liệu	Là khi một số modul tham chiếu/chia sẻ đến cùng một đối tượng dữ liệu toàn cục
	Tương liên nội dung	Là khi một modul sử dụng dữ liệu hay điều khiển thông tin trong phạm vi của một modul khác; Nó cũng xuất hiện khi có tồn tại lệnh rẽ nhánh trong modul

Ví dụ về mức độ tương liên

- Xét chương trình sắp xếp một dãy số L. Nó gồm các modul sau:
  - Nhập-dữ-liệu(L): nhập giá trị cho dãy số L cần sắp xếp.
  - Sắp-xếp(L, L_s): sắp xếp dãy L thành dãy Ls mà được sắp xếp.
  - Hiển-thị-kết-quả(L_s): in dãy kết quả đã sắp xếp Ls.

Ví dụ về mức độ tương liên: vừa

```
int L[N],Ls[N];  //Khai báo biến toàn cục
void main(){
   NhapDL();
   SapXep();
   HienThi();
}
void NhapDL(){
   //Nhap DL cho L
}
void SapXep(){
   //Sap xep L thanh Ls
}
void HienThi(){
   //Hien thi Ls
}
```

Ví dụ về mức độ tương liên: thấp

```
int L[N],Ls[N];  //Khai báo biến toàn cục
void main(){
   NhapDL(L);
   SapXep(L,Ls);
   HienThi(Ls);
}
void NhapDL(L){
   //Nhap DL cho L
}
void SapXep(L, Ls){
   //Sap xep L thanh Ls
}
void HienThi(Ls){
   //Hien thi Ls
}
```

Cảm ơn!