

CHƯƠNG 1

Tổng quan về lập trình hướng đối tượng



Nội dung

- ❖ Lập trình thủ tục và lập trình hướng đối tượng
- ❖ Các nguyên tắc lập trình HĐT
- ❖ Các đặc điểm cơ bản của lập trình HĐT
- ❖ Các ngôn ngữ hỗ trợ lập trình HĐT

Lập trình hàm/thủ tục

- Chương trình là một dãy các lệnh. *Vị trí mỗi, chuỗi...*
- Chương trình là một hệ thống các thủ tục/ hàm.
Mỗi thủ tục/ hàm là một dãy các lệnh.
- Lập trình là xác định xem chương trình gồm bao nhiêu thủ tục/ hàm.
- Kết quả là hệ thống cấu trúc và mối quan hệ giữa các hàm/ thủ tục.

Lập trình hàm/thủ tục

☐ Phương pháp tiếp cận của lập thủ tục

- ✓ Chương trình là một hệ thống các thủ tục và hàm
- ✓ Mỗi thủ tục (hàm) sẽ thực hiện những công việc nào đó
- ✓ Chương trình sẽ gọi các thủ tục (hàm) để giải quyết vấn đề

☐ Ưu điểm

- ✓ Chương trình rõ ràng, dễ hiểu, dễ theo dõi.
- ✓ Tư duy giải thuật rõ ràng.

☐ Nhược điểm

- ✓ Các hàm đều sử dụng biến toàn cục, dữ liệu (tham số) phải truyền qua rất nhiều hàm trung gian trước khi thực hiện công việc, vì vậy khó kiểm soát và phát triển khi chương trình quá lớn.
- Khi thay đổi cấu trúc dữ liệu, phải thay đổi giải thuật.

Lập trình hàm/thủ tục

❑ Các cấu trúc sử dụng trong lập trình thủ tục:

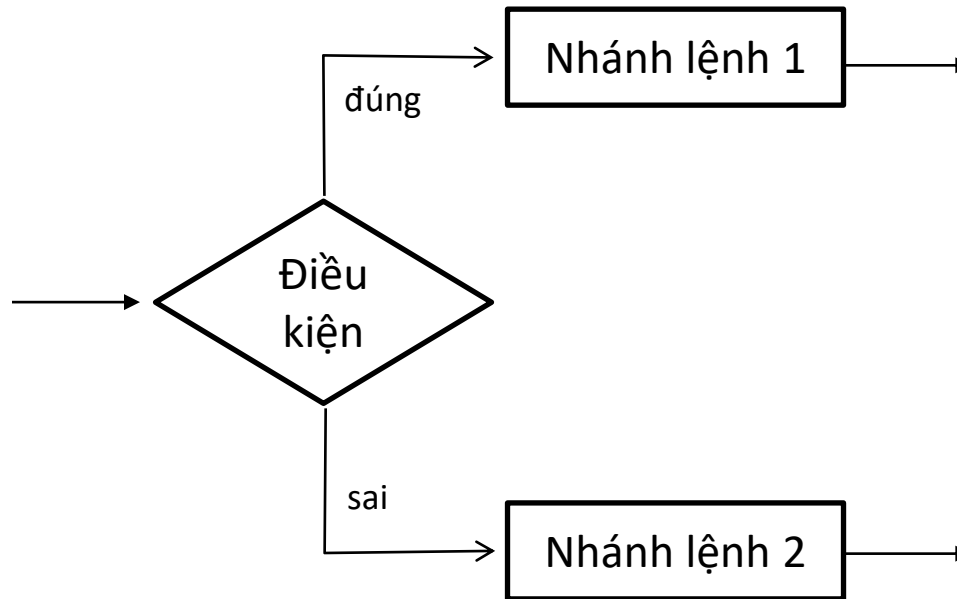
- Hiện nay có ba cấu trúc điều khiển thông dụng thông dụng để viết chương trình:

từ
trên
xác
định

Cấu trúc tuần tự (sequence): Các lệnh trong cấu trúc này lần lượt được thực hiện theo trình tự từng dòng lệnh, lệnh 1 → lệnh 2 → lệnh 3 → lệnh n...

- **Cấu trúc lựa chọn (selection):** Một điều kiện sẽ được kiểm tra trước khi chuyển hướng xử lý theo nhánh lệnh 1 hay nhánh lệnh 2

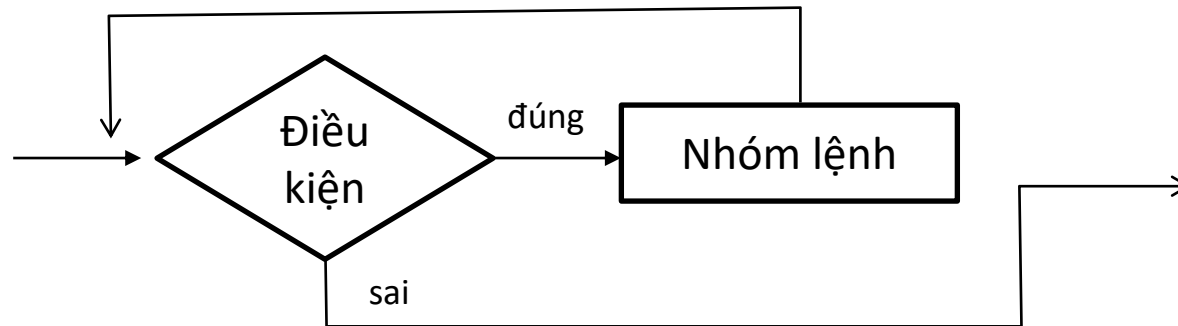
Lập trình hàm/thủ tục



if ()
switch
else

Lập trình hàm/thủ tục

- **Cấu trúc lặp (iteration):** Một nhóm lệnh sẽ được thực hiện lặp đi lặp lại cho tới khi điều kiện được kiểm tra không còn đúng nữa.



for
while
do while (i > n)

Lập trình hàm/thủ tục

❑ Ví dụ về lập trình thủ tục (dùng biến toàn cục)

```
static string hoten;  
static int van, toan;  
static float dtb;  
static void Main(string[] args)  
{  
    Nhap();  
    TinhTrungBinh();  
    Xuat();  
}
```

```
static void Nhap()  
{  
    Console.Write("Nhap ho ten: ");  
    hoten = Console.ReadLine();  
    Console.Write("Nhap diem toan: ");  
    toan = int.Parse(Console.ReadLine());  
    Console.Write("Nhap diem van: ");  
    van = int.Parse(Console.ReadLine());  
}
```

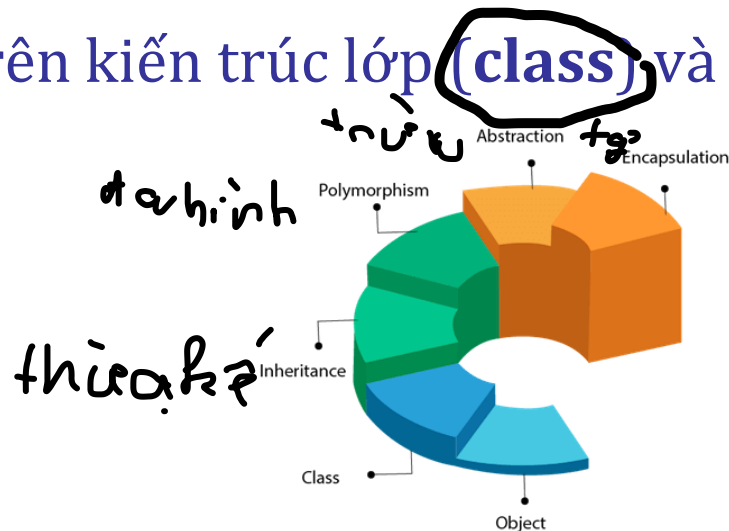



Lập trình hàm/thủ tục

```
static void Xuat()
{
    Console.WriteLine("Diem trung binh: {0: 0.00}", dtb);
}
static void TinhTrungBinh()
{
    dtb = (float)(toan + van) / 2;
}
```

Lập trình hướng đối tượng

- Lập trình hướng đối tượng là xây dựng những **lớp đối tượng** làm nền tảng để xây dựng chương trình. Đối tượng là sự liên kết giữa dữ liệu và phương thức thực hiện trên dữ liệu này.
- Lập trình hướng đối tượng là phương pháp lập trình dựa trên kiến trúc lớp **(class)** và đối tượng **(object)**



Nguồn: Internet

Lập trình hướng đối tượng

❑ Đặc điểm cơ bản

- ✓ Tập trung vào dữ liệu thay cho các hàm.
- ✓ Chương trình được chia thành các đối tượng độc lập.
- ✓ Cấu trúc dữ liệu được thiết kế sao cho đặc tả được các đối tượng.
- ✓ Dữ liệu bên trong mỗi đối tượng được bảo mật.
- ✓ Các đối tượng trao đổi với nhau thông qua các hàm.
- ✓ Chương trình được thiết kế theo hướng tiếp cận từ dưới lên.

Lập trình hướng đối tượng

❑ Một số ưu điểm nổi bật

- ✓ Không có nguy cơ dữ liệu bị thay đổi tự do trong chương trình.
- ✓ Khi thay đổi cấu trúc dữ liệu của một đối tượng, không cần thay đổi mã nguồn của các đối tượng khác.
- ✓ Có thể sử dụng lại mã nguồn, tiết kiệm tài nguyên.
- ✓ Phù hợp với các dự án phần mềm lớn, phức tạp.

SV A

SV. B

Lập trình hướng đối tượng

Đối tượng là gì?

- Đối tượng trong thế giới thực: là một thực thể cụ thể mà ta có thể sờ, nhìn thấy hay cảm nhận được.
- Đối tượng phần mềm: dùng để biểu diễn các đối tượng trong thế giới thực.
- Mỗi đối tượng bao gồm 2 thành phần: *thuộc tính* và *hành động*.

Lập trình hướng đối tượng

Đối tượng là gì?

VD: một người A

- Một người có các thuộc tính: tên, tuổi, địa chỉ, màu mắt, ...
- Các hành động: đi, nói, thở, ...

Một đối tượng là 1 thực thể bao gồm
thuộc tính & hành động

Lập trình hướng đối tượng

Lớp đối tượng là gì?

- Lớp đối tượng thể hiện cho một nhóm các đối tượng giống nhau (cùng thuộc tính & hành động)
- VD: học sinh A, học sinh B, học sinh C...

Học sinh

Lập trình hướng đối tượng

Ví dụ chương trình hướng đối tượng dùng ngôn ngữ lập trình C#

```
public class SinhVien
{
    public string Ma;
    public string Ten;
    public void XuatThongTin()
    {
        Console.WriteLine(Ma+"\t"+Ten);
    }
}
```

```
static void Main(string[] args)
{
    Console.OutputEncoding = Encoding.UTF8;
    SinhVien svA = new SinhVien();
    svA.Ma = "K194115";
    svA.Ten = "Kiều Hồng Lam";
    SinhVien svB = new SinhVien();
    svB.Ma = "K194061";
    svB.Ten = "Trần Văn Chuyên";
    svA.XuatThongTin();
    svB.XuatThongTin();
    Console.ReadLine();
}
```


Lập trình hướng đối tượng

Các tính chất của lập trình HĐT:

- **Tính đóng gói (Encapsulation):** Khả năng bảo vệ dữ liệu riêng biệt và phương thức tác động lên dữ liệu đó. Do vậy chúng ta không phải quan tâm tới *“phải làm như thế nào”* mà chỉ điều khiển bằng *“làm việc gì”*. Đóng gói giúp đồng nhất giữa dữ liệu và các thao tác tác động lên dữ liệu đó.

Lập trình hướng đối tượng

- **Tính thừa kế (inheritance):** Giúp tạo đối tượng mới từ đối tượng có sẵn, bổ sung những đặc tính cần thiết trong đối tượng mới.
- ✓ Lớp đối tượng đã có được sử dụng lại gọi là lớp cơ sở.
- ✓ Lớp thừa kế lớp cơ sở gọi là *lớp dẫn xuất*.

Lập trình hướng đối tượng

- **Tính đa hình (polymorphism):** Đa hình nghĩa là có nhiều hình thái khác nhau. Đa hình xuất hiện khi có một cấu trúc cấp bậc của các lớp và chúng liên quan với nhau bởi tính thừa kế.
- **Tính trừu tượng (abstraction):** Thể hiện tính tổng quát hóa lên một vấn đề mà không cần quan tâm đến các chi tiết bên trong. Là phương pháp trừu tượng hóa định nghĩa lên những hành động, tính chất của loại đối tượng nào đó cần phải có.

Lập trình hướng đối tượng

- Các ngôn ngữ hỗ trợ LTHĐT
 - ✓ C++
 - ✓ C#, VB.Net, J#, VC++
 - ✓ Java
 - ✓ JavaScript
 - ✓ PHP
 - ✓ ...

FAQs

