

# **Chương 2. Tổng quan về lập trình hướng đối tượng**

Nội dung

#2

1. Giới thiệu các phương pháp lập trình cổ điển
2. Giới thiệu phương pháp lập trình hướng đối tượng

# Phương pháp lập trình (PPLT)? #3

- Xây dựng phần mềm bao gồm rất nhiều công đoạn: phân tích & thiết kế, cài đặt, kiểm tra/thử nghiệm và bảo trì.

- Cài đặt (programming/coding) chỉ là 1 phần trong quá trình trên.

## Phương pháp lập trình?

#4

- C++/C#/Java/v.v... là NNLT để viết chương trình.

- PPLT là hệ thống hướng dẫn các giai đoạn cần thiết, cấu trúc của một chương trình.
- PPLT là các cách tiếp cận giúp cho quá trình cài đặt hiệu quả hơn.

Các yêu cầu chính của phần

mềm  #5

- Tính tái sử dụng (reusability)
- Tính mở rộng (extensibility)
- Tính mềm dẻo (flexibility)

Các PPLT cổ điển

#6

## Lập trình tuyến tính

- Chương trình là một dãy các lệnh.
- Lập trình là xây dựng các lệnh trong dãy lệnh.
- Không mang tính thiết kế.

# Các PPLT cổ điển

#7

## Lập trình thủ tục / hàm

- 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.

## Các PPLT cổ điển

#8

### Lập trình đơn thể

- Chương trình là một hệ thống những đơn thể.

Đơn thể là một hệ thống các thủ tục/ hàm.

- Phân tích và tìm ra các đơn thể.
- Gom nhóm các thành phần tương tự nhau về ý nghĩa, phạm vi...

Các PPLT cổ điển

#9

VD: Xét chương trình nhập vào họ tên, điểm

văn, điểm toán của một học sinh và xuất điểm trung bình tương ứng. Hãy viết chương trình trên bằng các phương pháp.

Cài đặt với pp lập trình tuyến tính (chỉ dùng 1 hàm main & biến toàn cục)

static string hoten;

```
static int van, toan;  
static float dtb;  
static void Main(string[] args)  
{  
    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());  
    dtb = (float)(toan + van) / 2;  
    Console.WriteLine("Diem trung binh: {0: 0.00}", dtb);  
}
```

Cài đặt với pp lập trình tuyến tính (chỉ dùng 1 hàm

## main và biến cục bộ

#11

```
static void Main(string[] args)
{
    string hoten;
    int van, toan;
    float dtb;
    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());
    dtb = (float)(toan + van) / 2;
```

```
Console.WriteLine("Diem trung binh: {0: 0.00}", dtb);
```

Cài đặt với pp lập trình tuyến tính (chỉ dùng 1 hàm main và cấu trúc toàn cục)

#12

```
struct HOCSINH {  
    public string hoten; public int van,  
    toan; public float dtb;  
}  
static HOCSINH hs;
```

```
static void Main(string[] args)  
{  
    Console.Write("Nhap ho ten: ");  
    hs.hoten = Console.ReadLine();  
    Console.Write("Nhap diem toan:  
"); hs.toan =  
    int.Parse(Console.ReadLine());  
    Console.Write("Nhap diem van: ");  
    hs.van =  
    int.Parse(Console.ReadLine());
```

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

Cài đặt với pp lập trình tuyến tính (chỉ dùng 1 hàm main và cấu trúc cục bộ)

#13

```
static void Main(string[] args)  
{  
    HOCSINH hs;  
        public float dtb;  
}
```

```
struct HOCSINH  
{  
    public string hoten;  
    public int van, toan;
```

```
hs.van) / 2;  
Console.WriteLine("Diem trung  
binh: {0: 0.00}", hs.dtb);
```

```
Console.Write("Nhap ho ten: ");  
hs.hoten = Console.ReadLine();  
Console.Write("Nhap diem toan:  
");  
hs.toan=int.Parse(Console.ReadLi  
ne()); Console.Write("Nhap diem  
van: ");  
hs.van=int.Parse(Console.ReadLin  
e()); hs.dtb = (float)(hs.toan +
```



Cài đặt với pp lập trình thủ tục (dùng biến toàn

cục) #14

```
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());    van =  
Console.Write("Nhap diem van: "); int.Parse(Console.ReadLine()); }
```

Cài đặt với pp lập trình thủ tục (dùng biến toàn

cục) #15

```
static void Xuat()
```

```
{
```

```
    Console.WriteLine("Diem trung binh: {0: 0.00}",
```

```
dtb); }  
static void TinhTrungBinh()  
{  
    dtb = (float)(toan + van) / 2;  
}
```

Cài đặt với pp lập trình thủ tục (dùng biến cục bộ)

#16

```
static void Nhap(out string ht, out int v, out int
```

```
t) {  
    Console.Write("Nhap ho ten: ");  
    ht = Console.ReadLine();  
    Console.Write("Nhap diem toan: ");  
    t = int.Parse(Console.ReadLine());  
    Console.Write("Nhap diem van: ");  
    v = int.Parse(Console.ReadLine());  
}
```

Cài đặt với pp lập trình thủ tục (dùng biến cục bộ)

#17

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

Cài đặt với pp lập trình thủ tục (dùng biến cục bộ)

#18

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

Phải quan tâm đến tham số: Trị, chiều và giá trị trả về của mỗi phương thức.

Cài đặt với pp lập trình thủ tục (dùng biến cấu trúc cục bộ)

#19

```
struct HOCSINH
{
    public string hoten;
    public int van, toan;
    public float dtb;
}
static void Main(string[] args)
```

```
{  
    HOCSINH hs;  
    Nhap(out hs);  
    Xuat(hs);  
}
```

Cài đặt với pp lập trình thủ tục (dùng biến cấu trúc cục bộ)

```
#20  
static void Nhap(out HOCSINH  
hs)  
{  
    Console.Write("Nhap ho ten: ");  
    hs.hoten = Console.ReadLine();  
    Console.Write("Nhap diem toan:  
");  
    hs.toan =  
    int.Parse(Console.ReadLine());  
    Console.Write("Nhap diem van:  
");  
    hs.van =  
    int.Parse(Console.ReadLine());
```



```

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

```

```

static void Xuat(HOCSINH hs)
    }

```

Cài đặt với pp lập trình thủ tục (dùng biến cấu trúc toàn cục)

#21

```

struct HOCSINH
{
public string hoten;
public int van, toan;
public float dtb;
}

```

```

static HOCSINH hs;
static void Main(string[] args) {
Nhap();
Xuat();
}
static void Nhap()

```

```

{
Console.Write("Nhap ho ten: ");
hs.hoten = Console.ReadLine();
Console.Write("Nhap diem toan: ");
hs.toan=int.Parse(Console.ReadLine());
Console.Write("Nhap diem van: ");
hs.van = int.Parse(Console.ReadLine());
}
hs.dtb = (float)(hs.toan + hs.van) / 2;
}
static void Xuat()
{
Console.WriteLine("Diem trung binh: {0:0.00}", hs.dtb);
}

```

## PP Lập trình hướng đối

tượng  #22

- Chương trình là một hệ thống những lớp đối

tượng. Mỗi một lớp đối tượng về mặt thực tế tương ứng với những đối tượng có xuất hiện trong thực tế.

PP Lập trình hướng đối

tượng  #23

- LT hướng đối tượng là xây dựng những lớp đối tượng và yêu cầu chúng thực hiện những trách nhiệm của mình.
- LT hướng đối tượng là phương pháp LT dựa trên kiến trúc lớp (class) và đối tượng (object)

Đối tượng là gì ?

#24

●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.

Đối tượng là gì ?

#25

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**

Lớp đối tượng là gì ?

#26

- 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...

Thiết kế phần mềm hướng đối tượng



- Trừu tượng hóa dữ liệu và các hàm/ thủ tục liên quan
- Chia hệ thống ra thành các lớp/ đối tượng ■  
Mỗi lớp/ đối tượng có các tính năng và hành động chuyên biệt
  - Các lớp có thể được sử dụng để tạo ra nhiều đối tượng cụ thể

# Sự trừu tượng

hoá  #28

**Thế giới thực Trừu tượng hóa Phần mềm**

Thực thể

Thuộc tính Hành động

Dữ liệu hàm

Một số khái niệm

Thế giới thực	PPLT	Ngôn ngữ lập trình
Đối tượng trong thế giới thực	Đối tượng	Biến có kiểu cấu trúc
Khái niệm chung về đối tượng	Lớp đối tượng	Kiểu dữ liệu cấu trúc
Các thông tin được quan tâm về 1 đối tượng	Thuộc tính	Thành phần thuộc tính của kiểu cấu trúc
Các khả năng của đối tượng	Hành động	Các phương thức
Phân công giữa các đối tượng	Yêu cầu	Gọi thực hiện phương thức

# Đặc điểm của pp lập trình HĐT

#30

- **Tính đóng gói (Encapsulation):** Khả năng cất giữ riêng biệt dữ liệu 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.

# Đặc điểm của pp lập trình HĐT

#31

- **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*.

## Đặc điểm của pp lập trình HĐT

#32

- **Tính đa hình (polymorphism):** Cho phép gửi cùng một thông điệp đến những đối tượng khác nhau cùng có chung một đặc điểm.

# Một số ngôn ngữ lập trình

HĐT  #33

- C++



- C#, VB.Net, J#, VC++

- Java

- JavaScript

- PHP

- ...

## Các bước thiết kế đối tượng

- Bước 1: Xây dựng sơ đồ đối tượng
  - Xác định các lớp đối tượng
  - Xác định các quan hệ giữa các lớp
- Bước 2: Thiết kế các lớp

Thiết kế thuộc tính, các hành động

- Bước 3: Cài đặt các lớp
- Bước 4: Sử dụng các lớp để tạo ra các đối tượng

# FAQs

#35

