



Chương 1

TỔNG QUAN VỀ CÁCH TIẾP CẬN HƯỚNG ĐỐI TƯỢNG



Mục tiêu

- ◆ Nhận biết sự khác biệt giữa lập trình truyền thống và lập trình hướng đối tượng
- ◆ Nhận biết lớp và đối tượng
- ◆ Nhận biết các đặc trưng cơ bản của lập trình hướng đối tượng

Nội dung

- 1.1. Sơ lược các kỹ thuật lập trình
- 1.2. Đối tượng (Object)
- 1.3. Lớp (Class)
- 1.4. So sánh các cách tiếp cận lập trình
- 1.5. Đặc trưng của lập trình hướng đối tượng

1.1. Sơ lược các kỹ thuật lập trình

Lập trình tuyến tính

- ♦ Chương trình sẽ được thực hiện tuần tự từ đầu đến cuối, lệnh này kế tiếp lệnh kia cho đến khi kết thúc chương trình
- ♦ Chương trình đơn giản, dễ hiểu, nhưng không thể dùng phương pháp này cho những ứng dụng phức tạp
- ♦ Ví dụ: Basic, Fortran

1.1. Sơ lược các kỹ thuật lập trình

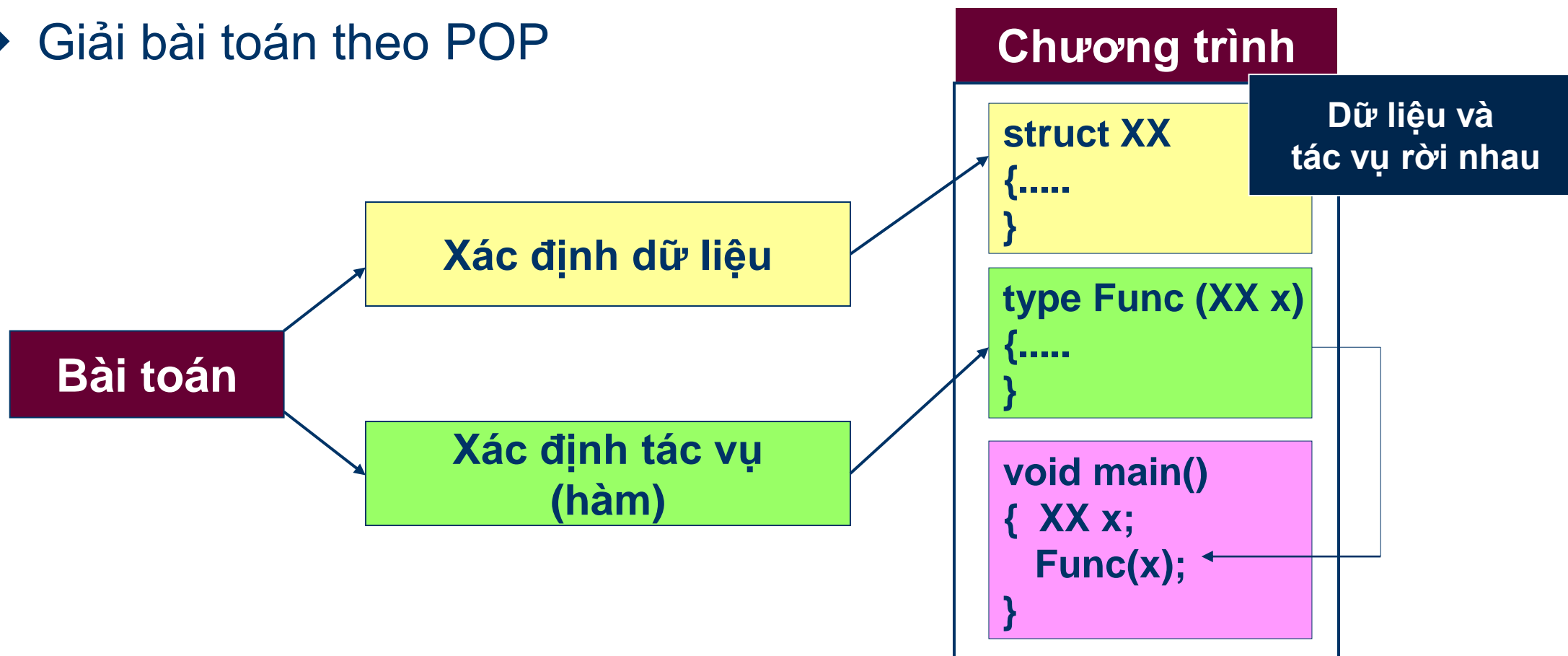
Lập trình hướng thủ tục (POP)

- ♦ Lập trình hướng thủ tục là cách lập trình lấy hành động làm trung tâm
- ♦ Chương trình được phân chia thành các công việc nhỏ hơn để dễ dàng quản lý, gọi là chương trình con (hàm)
- ♦ Dữ liệu và xử lý bị tách biệt
- ♦ Ví dụ: C, Pascal, COBOL, Fortran, Perl, HTML, VBScript

1.1. Sơ lược các kỹ thuật lập trình

Lập trình hướng thủ tục (tt)

- ♦ Giải bài toán theo POP



1.1. Sơ lược các kỹ thuật lập trình

Lập trình hướng thủ tục (tt)

- ♦ Vd1. Viết chương trình thực hiện các phép toán $+$, $-$, $*$, $/$ hai số thực.
- ♦ Vd2. Viết chương trình tính diện tích và chu vi của hình chữ nhật.

1.1. Sơ lược các kỹ thuật lập trình

Lập trình hướng thủ tục (tt)

♦ Vd1.

```
float Cong( float a, float b ){  
    return a+b;  
}  
float Tru( float a, float b ){  
    return a-b;  
}  
float Nhan( float a, float b ){  
    return a*b;  
}  
float Chia( float a, float b ){  
    return a/b;  
}
```

```
void main() {  
    float x= 9, y= 3, kq;  
    kq = Cong(x,y);  
    printf("Ket qua Cong=%f\n",kq);  
    kq = Tru(x,y);  
    printf("Ket qua Tru=%f\n"+kq);  
    kq = Nhan(x,y);  
    printf("Ket qua Nhan=%f\n"+kq);  
    kq = Chia(x,y);  
    printf("Ket qua Chia=%f\n"+kq);  
}
```


1.1. Sơ lược các kỹ thuật lập trình

Lập trình hướng thủ tục (tt)

♦ Vd2.

```
struct HìnhChuNhat {  
    int cdai;  
    int crong;  
};  
int TinhDienTich( HìnhChuNhat a ){  
    return a.cdai * a.crong;  
}  
int TinhChuVi( HìnhChuNhat a ){  
    return (a.cdai + a.crong) * 2;  
}
```

```
void main() {  
    HìnhChuNhat x;  
    x.cdai = 10;  
    x.crong = 5;  
    int dt = TinhDienTich(x);  
    int cv = TinhChuVi(x);  
    printf("Dien tich=%d\n",dt);  
    printf("Chu vi=%d\n",cv);  
}
```

1.1. Sơ lược các kỹ thuật lập trình

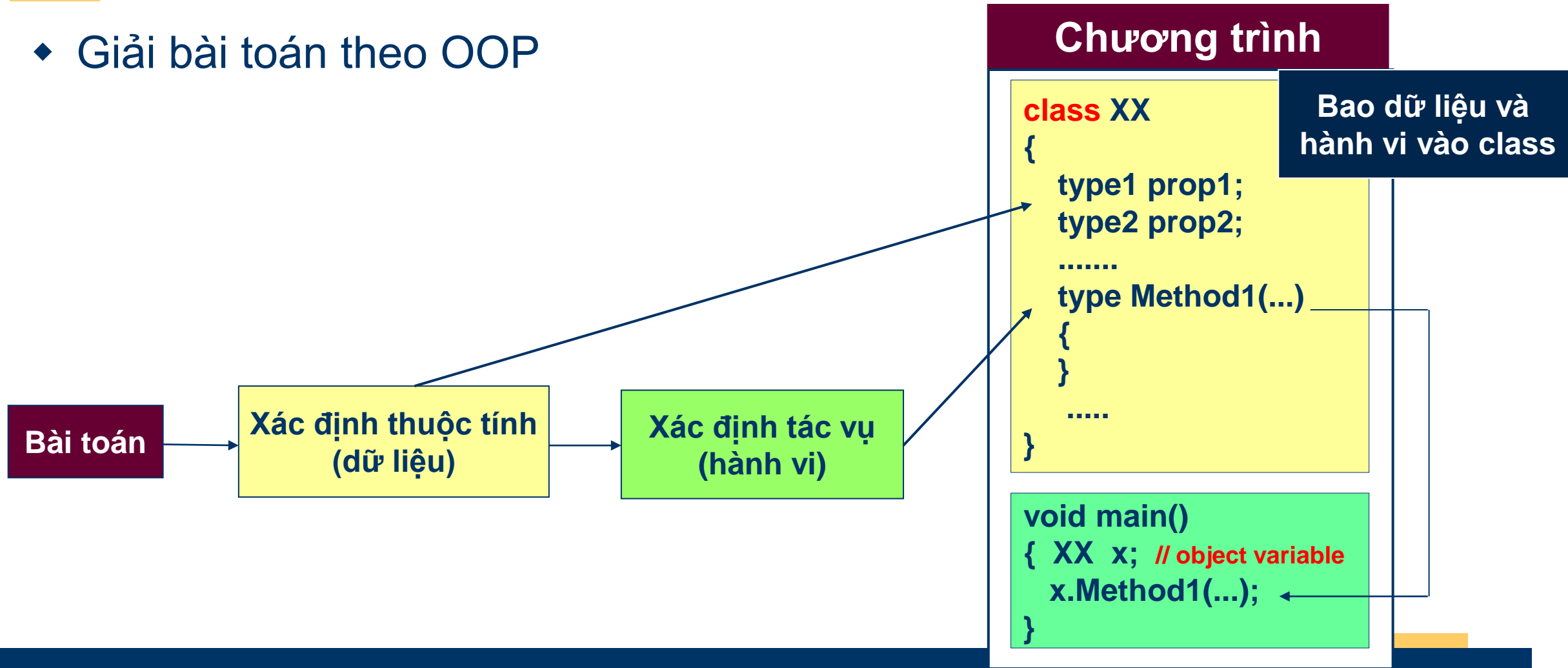
Lập trình hướng đối tượng

- ♦ Lập trình hướng đối tượng là cách lập trình lấy đối tượng (object) làm nền tảng để xây dựng chương trình.
- ♦ Một **ĐỐI TƯỢNG** sẽ bao GỒM các **THUỘC TÍNH** và **HÀNH VI** của nó.
- ♦ Một chương trình được viết theo hướng đối tượng sẽ bao gồm các đối tượng tương tác với nhau.
- ♦ Ví dụ: C++, C#, Visual Basic.NET, Java.

1.1. Sơ lược các kỹ thuật lập trình

Lập trình hướng đối tượng (tt)

- ♦ Giải bài toán theo OOP



1.1. Sơ lược các kỹ thuật lập trình

Lập trình hướng đối tượng (tt)

- ♦ Vd1. Viết chương trình thực hiện các phép toán $+$, $-$, $*$, $/$ hai số thực bằng kỹ thuật hướng đối tượng.
- ♦ Vd2. Viết chương trình tính diện tích và chu vi của hình chữ nhật bằng kỹ thuật hướng đối tượng.

1.1. Sơ lược các kỹ thuật lập trình

Lập trình hướng đối tượng (tt)

♦ Ví dụ 1.

```
class PhepTinh {  
    // dữ liệu:  
    float a, b;  
    // các hành vi:  
    float cong() {  
        return a + b;  
    }  
    float tru() {  
        return a - b;  
    }  
    float nhan() {  
        return a * b;  
    }  
    float chia() {  
        return a / b;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        PhepTinh pt = new PhepTinh();  
        pt.a = 9;  
        pt.b = 3;  
        float kq;  
        kq = pt.cong();  
        System.out.println("Ket qua Cong=" + kq);  
        kq = pt.tru();  
        System.out.println("Ket qua Tru=" + kq);  
        kq = pt.nhan();  
        System.out.println("Ket qua Nhan=" + kq);  
        kq = pt.chia();  
        System.out.println("Ket qua Chia=" + kq);  
    }  
}
```

1.1. Sơ lược các kỹ thuật lập trình

Lập trình hướng đối tượng (tt)

♦ Ví dụ 2.

```
class HìnhChuNhat {  
    // dữ liệu:  
    int cdai;  
    int crong;  
    // các hành vi:  
    int tinhDienTich() {  
        return cdai * crong;  
    }  
    int tinhChuVi() {  
        return (cdai + crong) * 2;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        HìnhChuNhat h = new HìnhChuNhat();  
        h.cdai = 10;  
        h.crong = 5;  
        int dt = h.TinhDienTich();  
        int cv = h.TinhChuVi();  
        System.out.println("Dien tich=" + dt);  
        System.out.println("Chu vi=" + cv);  
    }  
}
```

1.2. Đối tượng (Object)

- ♦ Trong thế giới thực, đối tượng là một **thực thể** (entity) cụ thể mà ta có thể sờ, nhìn thấy hay cảm nhận được.
- ♦ Dùng để mô tả người, sự vật hay khái niệm.

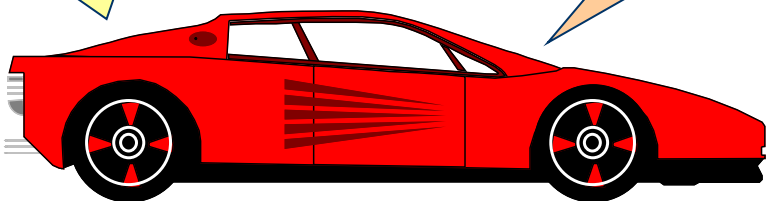
MỖI ĐỐI TƯỢNG CÓ THUỘC TÍNH VÀ HÀNH ĐỘNG CỦA NÓ

1.2. Đối tượng (Object)

Đối tượng trong thế giới thực

Tôi là 1 **chiếc xe hơi**.

Thông tin của tôi gồm hãng sản xuất BMW, model A100, màu đỏ, giá 2 tỷ, chủ xe ông Ng Văn An



Tôi có khả năng:

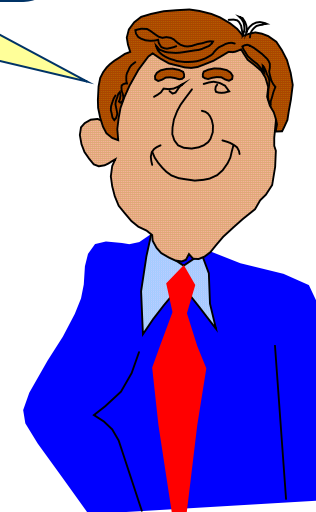
- Chạy
- Dừng
- Tăng tốc ...

Tôi là 1 **nhân viên**.

Thông tin của tôi gồm họ tên Ng Văn An, số CMND 123456789, nghề nghiệp nvhc, mức lương 3.99

Tôi có khả năng:

- Làm việc
- Lãnh lương
- Đi nhậu...



Tôi có khả năng:

- Bơi
- Được kho
- Được sốt cà ...



Tôi là 1 **con cá**.

Thông tin của tôi gồm loại cá thu, cân nặng 2 ký, đơn giá 330/kg, bị lưới ngày 1/1

1.2. Đối tượng (Object)

♦ Đối tượng phần mềm cũng có:

- Thuộc tính, đặc tính, dữ liệu (*attribute, property, data*): Mô tả các **đặc điểm** của đối tượng
- Phương thức, hành vi, hàm (*method, behavior, function*):
 - Liên quan đến những thứ đối tượng làm
 - Tác động lên dữ liệu của đối tượng

Thuộc tính:

Tiêu đề Abc, nội dung Xyz, kích thước 20x30, màu xám chủ đạo

Phương thức:

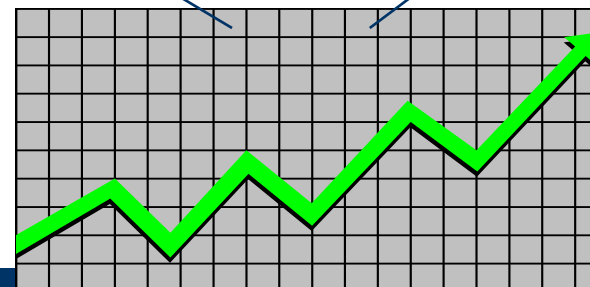
Vẽ, di chuyển, đổi kích thước, đóng, ...

Thuộc tính:

Số liệu Abc, mô tả Xyz, loại biểu đồ Line...

Phương thức:

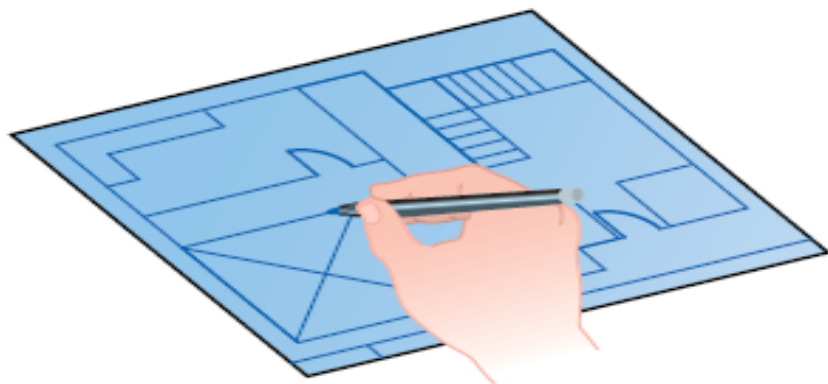
Vẽ, đổi số liệu, đổi dạng biểu đồ, ...



1.3. Lớp (Class)

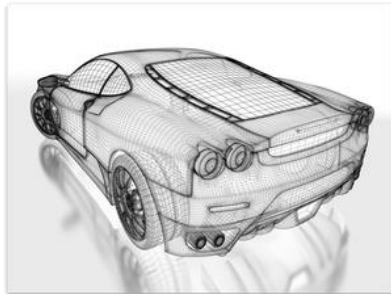
- ♦ Lớp là một khuôn mẫu để tạo ra đối tượng. Một lớp gồm:
 - Dữ liệu: mô tả các đặc trưng, tính chất của đối tượng
 - Các thao tác: là các hành động mà đối tượng đó có thể thực hiện được
- ♦ Lớp tạo ra đối tượng bằng cách gán giá trị cụ thể cho các thuộc tính

→ **Đối tượng là một thể hiện (instance) của một lớp**



1.3. Lớp (Class)

Ví dụ: Class và Object



NhanVien

- Mã nhân viên
- Họ và tên
- Năm sinh
- Nghề nghiệp
- Địa chỉ
- Hệ số lương
- Chức vụ

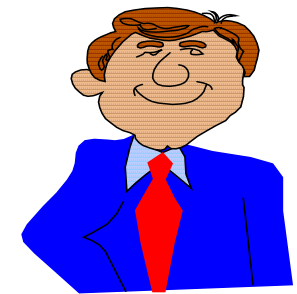
- + Làm việc()
- + Nghỉ phép()
- + Lãnh lương()



Ông NVA



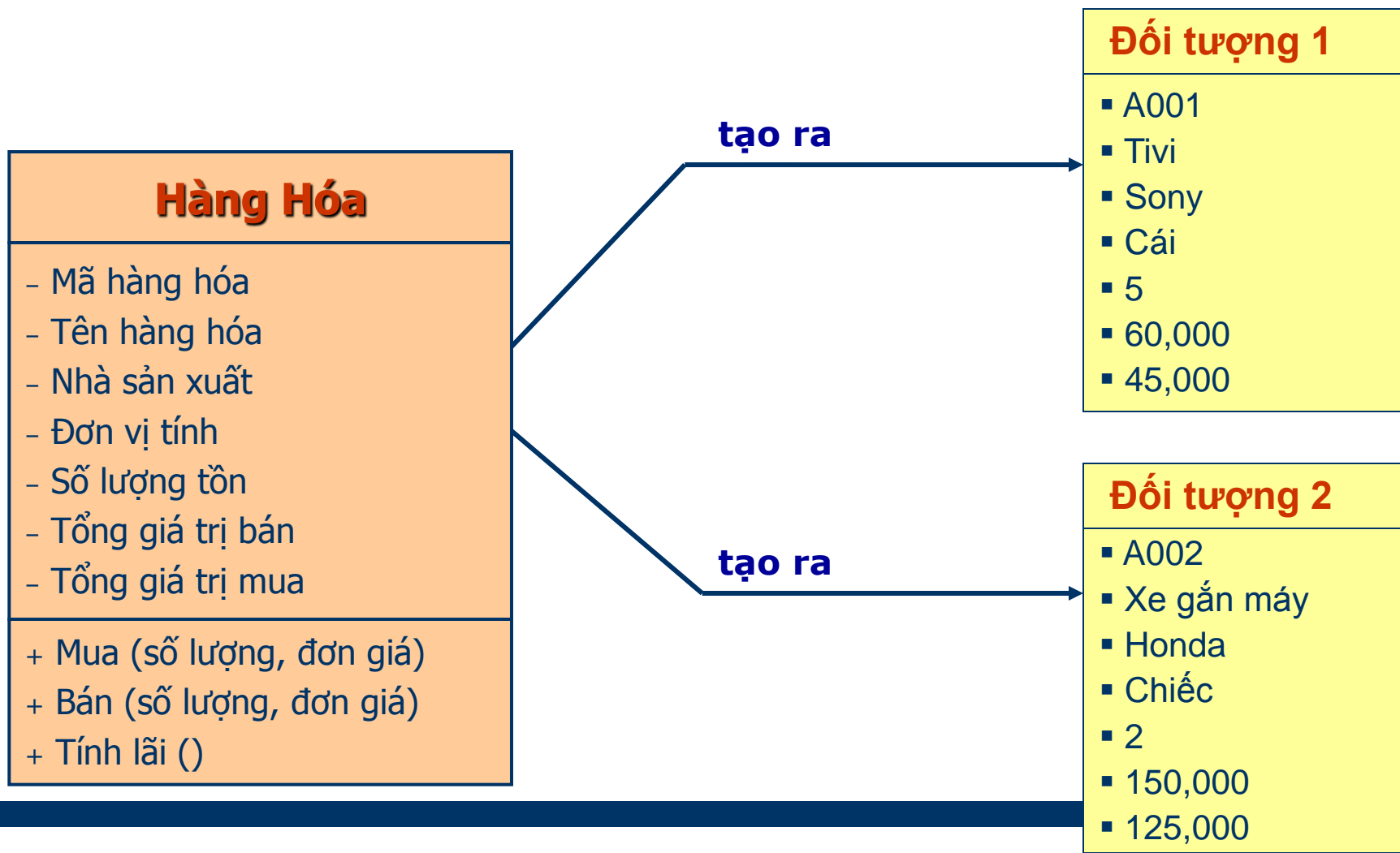
Bà TTB



Ông LVC

1.3. Lớp (Class)

Ví dụ: Class và Object



1.3. Lớp (Class)

Xác định lớp đối tượng

- ♦ Ví dụ 1: Xác định các lớp đối tượng có thể trong mô tả sau:

The user must be allowed to specify each product by its primary characteristics, including its name and product number. If the bar code does not match the product, then an error should be generated to the message window and entered into the error log. The summary report of all transactions must be structured as specified in section 7.A.

1.3. Lớp (Class)

Xác định lớp đối tượng

- ♦ Ví dụ 2: Xác định các lớp đối tượng có thể trong mô tả sau:

ĐƠN VI: _____ HÓA ĐƠN BÁN LẺ Số: _____

Họ và tên người mua hàng: _____

Địa chỉ: _____

Số TT	Tên Hàng và quy cách phẩm chất	D. vị tính	Số lượng	Giá đơn vị	THÀNH TIỀN
1	Kem trị sỗ mụn bé	tube	01	150	150.000
2	DHC vitamin C	gói	01	150	150.000
3	Cc Melano serum	tube	01	270	270.000
4	Soft Stone	cái	01	215	215.000
5	Bb cream b'son	tube	01	260	260.000
6	Kit CN Biore	chai	01	180	180.000
7	Kit Biore kid	chai	01	190	190.000
8	Mũi lăn mũi + ship (kèm nước)		01	180	180.000
9					
10					
11					
12					
13					
14					
CỘNG:					1.595.000

Cộng thành tiền (viết bằng chữ) _____

Ngày.....tháng.....năm 20....

NGƯỜI NHẬN HÀNG ĐÃ NHẬN ĐỦ TIỀN NGƯỜI VIẾT HÓA ĐƠN

1.4. So sánh các cách tiếp cận lập trình

PP truyền thống	PP hướng đối tượng
Đi từ tổng quan rồi chia nhỏ thành các bài toán con, cụ thể hơn	Đi từ chi tiết đến trừu tượng hóa ở mức cao
Phải tạo ràng buộc giữa dữ liệu và hành động	Dữ liệu tự liên kết với hành động
Khó sử dụng lại mã nguồn	Dễ dàng sử dụng lại mã nguồn

1.5. Đặc trưng của lập trình hướng đối tượng

- ♦ Trừu tượng (Abstraction)
- ♦ Đóng gói (Encapsulation)
- ♦ Thừa kế (Inheritance)
- ♦ Đa hình (Polymorphism)

1.5. Đặc trưng của LT HĐT

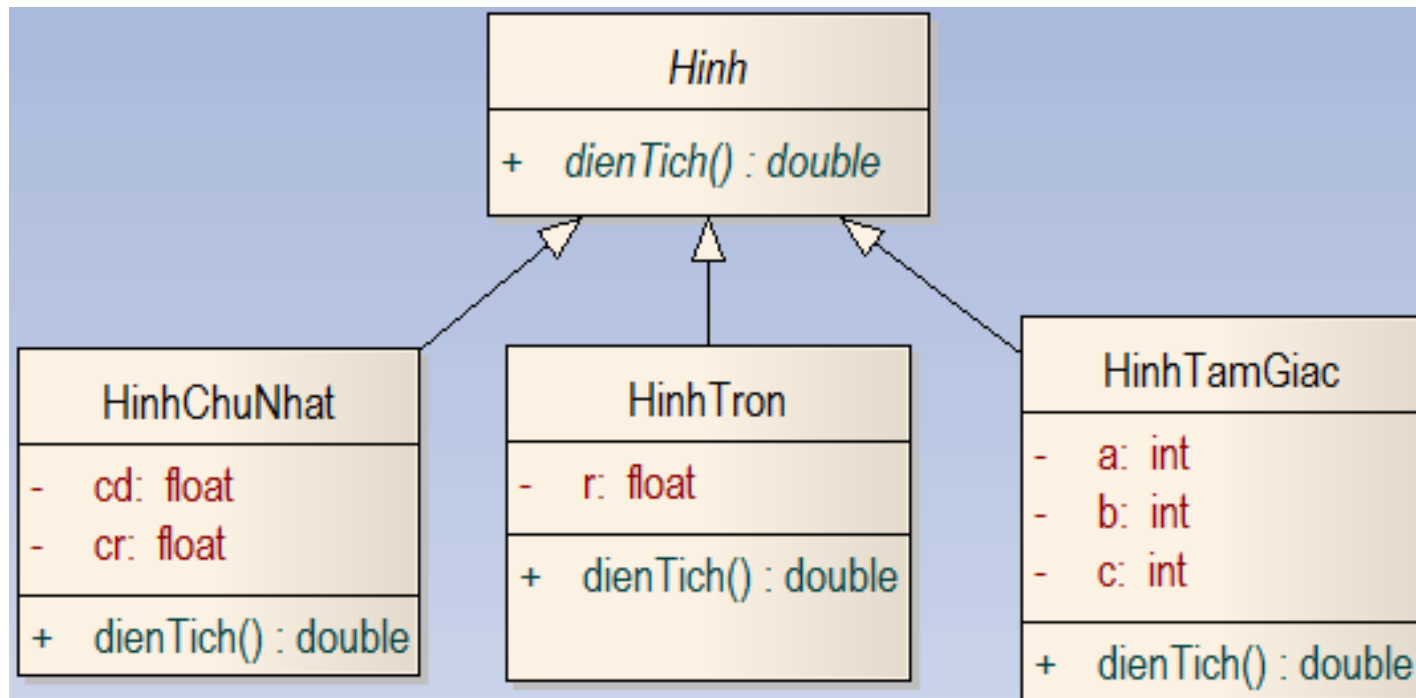
Tính trừu tượng [1]

- ♦ *Tính trừu tượng* cho phép bỏ qua các chi tiết không cần thiết và chỉ đưa ra các thuộc tính và phương thức cần thiết của đối tượng
- ♦ Ví dụ: Chọn những thông tin cần thiết từ các thông tin thu thập được về khách hàng để xây dựng ứng dụng cho ngân hàng

- ☒ Full Name
- ☒ Address
- ☒ Contact Number
- ☒ Tax Information
- ☒ Favorite Food
- ☒ Favorite Movie
- ☒ Favorite Actor
- ☒ Favorite Band

1.5. Đặc trưng của LT HĐT Tính trừu tượng [2]

- ♦ *Trừu tượng* tức là chung chung, không có thực. Đặc tính này được thể hiện trong khái niệm **lớp trừu tượng** (abstract class) và **giao diện** (interface)
- ♦ Ví dụ:



1.5. Đặc trưng của LT HĐT Tính đóng gói [1]

- ♦ *Che dấu cài đặt chi tiết bên trong: chỉ cần biết nó làm được gì mà không cần biết bên trong nó làm như thế nào*
- ♦ Ví dụ: Hãy nêu một món đồ dùng trong nhà và mô tả cách sử dụng nó (vd/ TV, máy giặt, laptop,...). Sau đó thử mô tả các thành phần bên trong của món đồ này cũng như mô tả chi tiết kỹ thuật làm nó hoạt động
 - Mô tả cách vận hành dễ hơn mô tả chi tiết chính xác làm sao nó hoạt động. Và hầu như mọi người thậm chí không biết có những thành phần gì bên trong UD

1.5. Đặc trưng của LT HĐT

Tính đóng gói [2]

- ♦ *Che dấu dữ liệu (data hiding)*: là sự che giấu dữ liệu riêng của mỗi đối tượng, không cho bên ngoài sử dụng dữ liệu trực tiếp mà chỉ được truy xuất thông qua hệ thống các phương thức có sẵn của lớp
- ♦ Để cài đặt tính đóng gói, cần thực hiện:
 - Khai báo các thuộc tính là *private* (để các lớp khác không thể truy cập trực tiếp được)
 - Tạo các phương thức *get/set* là *public* để lấy và sửa giá trị của thuộc tính

1.5. Đặc trưng của LT HĐT

Tính đóng gói: Ví dụ

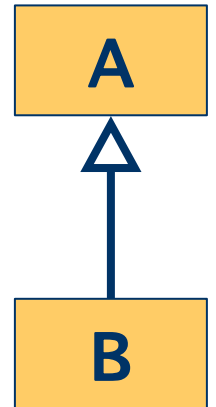
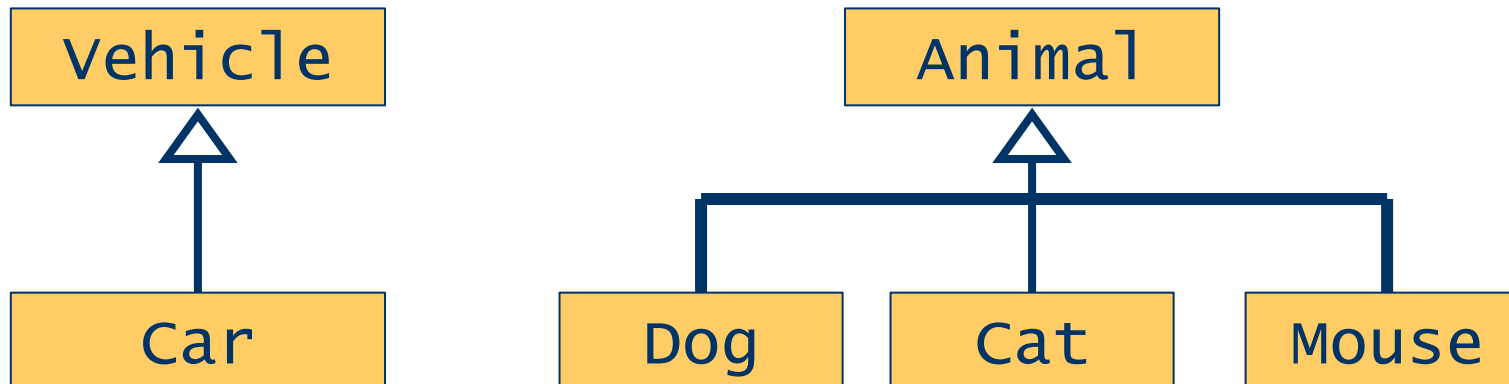
```
public class Person {  
    // khai báo các thuộc tính của là private  
    private String Cmnd;  
    private String HoTen;  
    // tạo các phương thức getter/setter  
    public String getCmnd() {  
        return Cmnd;  
    }  
    public void setCmnd(String cmd) {  
        Cmnd = cmd;  
    }  
    public String getHoTen() {  
        return HoTen;  
    }  
    public void setHoTen(String hoTen) {  
        HoTen = hoTen;  
    }  
}
```

```
public class Test {  
    public static void main(String[] args) {  
        Person person = new Person();  
        // gán giá trị họ tên cho đt person  
        // thông qua setHoTen(); và gán số  
        // chứng minh nhân dân thông qua  
        // setCmnd()  
        person.setHoTen("Trần Văn Bình");  
        person.setCmnd("212321678");  
        // truy cập đến tên của đt person thông  
        // qua phương thức getHoten() và số  
        // chứng minh nhân dân thông qua phương  
        // thức getCmnd()  
        System.out.println("Tên: " +  
            person.getHoTen() + ", số cmd: " +  
            person.getCmnd());  
    }  
}
```

1.5. Đặc trưng của LT HĐT

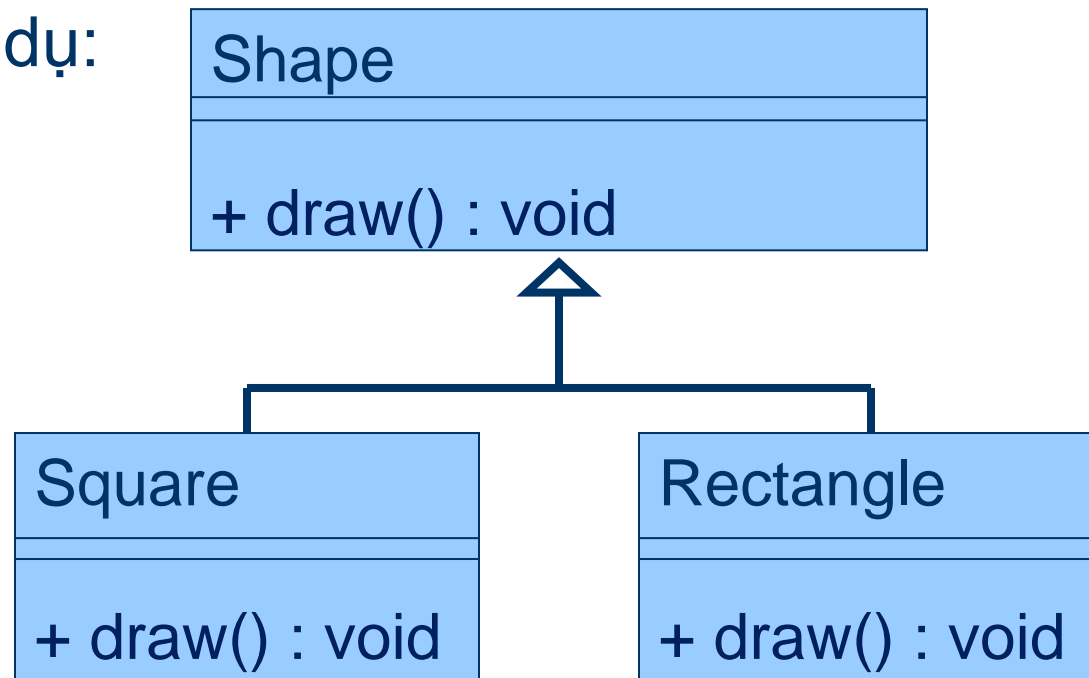
Tính thừa kế

- ♦ Là cơ chế cho phép một lớp B có được các thuộc tính và hành vi của lớp A, như thể các thuộc tính và hành vi đó đã được định nghĩa tại lớp B
- ♦ Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không cần phải định nghĩa lại
- ♦ Ví dụ:



1.5. Đặc trưng của LT HĐT Tính đa hình

- ♦ *Tính đa hình* là khả năng một đối tượng có thể tham chiếu đến nhiều loại đối tượng khác nhau tại những thời điểm khác nhau
- ♦ Thể hiện thông qua việc gửi các **thông điệp** (*message*)
- ♦ Ví dụ:



Shape shape;

shape = new Square();

shape.draw();

shape = new Rectangle();

shape.draw();



Câu hỏi kiểm tra

1. Ưu điểm của lập trình HĐT.
2. Nêu và giải thích 4 đặc trưng của lập trình HĐT.
3. Thể hiện tính đóng gói trong lập trình HĐT như thế nào.
4. Phân biệt lớp và đối tượng.
5. Khi xây dựng lớp đối tượng, cần xác định những gì.

Bài tập

1. Liệt kê các thuộc tính và hành vi của sinh viên, của hóa đơn
2. Giả sử viết một chương trình tính lương trả hàng tháng cho công nhân. Lương trả dựa trên số sản phẩm họ làm được. Ngoài ra chương trình còn tính phí bảo hiểm y tế và bảo hiểm xã hội cho công nhân.

Xác định đối tượng trong bài toán này, cùng với các thuộc tính và hành vi của đối tượng đó.