

# Lập trình hướng đối tượng

## BÀI 1

### XÂY DỰNG VÀ SỬ DỤNG LỚP

# **Nội dung:** XÂY DỰNG VÀ SỬ DỤNG LỚP

1. Xác định các lớp đối tượng, các thuộc tính và các phương thức, thiết kế sơ đồ lớp.
2. Đọc sơ đồ lớp, cài đặt lớp.
3. Phạm vi truy xuất
4. Sử dụng lớp.
5. Cài đặt chương trình ứng dụng hướng đối tượng

## 2.1. Xác định các lớp đối tượng

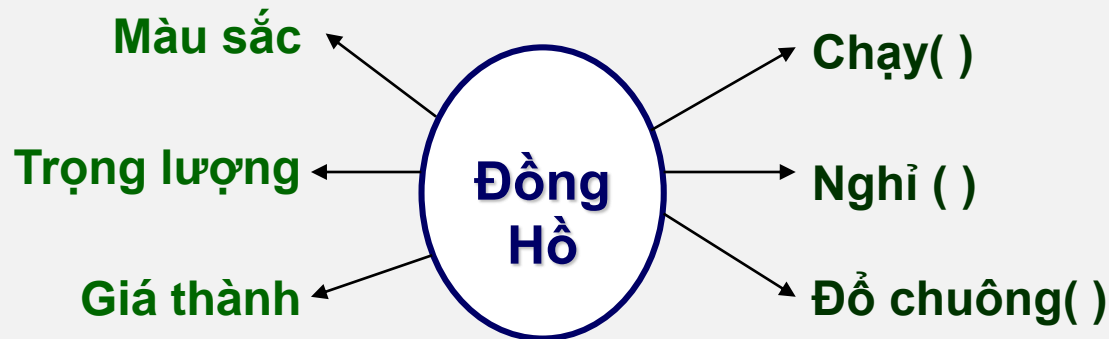
- Tìm các danh từ và động từ
  - Danh từ chung -> các lớp đối tượng
  - Danh từ riêng -> các đối tượng
- Xác định các lớp cần thiết
- Xác định các phương thức <- động từ
- Xác định các thuộc tính <- danh từ
- Xác định quan hệ kết tập
- Xác định quan hệ kế thừa

# Xác định các lớp đối tượng (tt)

- Ví dụ: Mô phỏng quần thể đồng hồ

Động từ	Danh từ	
Chạy	Đồng hồ	Màu sắc
Nghỉ		Trọng lượng
Đổ chuông		Giá thành

## Sơ đồ lớp



Đồng Hồ
Màu sắc
Trọng lượng
Giá thành
Chạy()
Nghỉ()
Đổ chuông()

## 2.2. Định nghĩa lớp

- **Mẫu 1:**
  - Các thuộc tính và phương thức được định nghĩa trong thân lớp.

```
class <Tên_Lớp>
{
    // Định nghĩa các thuộc tính
    // Định nghĩa các phương thức
};
```

# Định nghĩa lớp (tt)

- **Mẫu 2:**

- Các thuộc tính và nguyên mẫu phương thức được định nghĩa trong thân lớp.
- Nội dung phương thức được định nghĩa bên ngoài lớp.

```
class <Tên_Lớp>
{
    // Định nghĩa các thuộc tính
    // Khai báo nguyên mẫu các phương thức
};
//Định nghĩa các phương thức
```

# Định nghĩa lớp (tt)

- Giải thích

- <Tên\_Lớp>: do người dùng đặt tùy ý, viết theo đúng quy tắc đặt tên của ngôn ngữ lập trình C++.
- Định nghĩa các thuộc tính:

```
<kiểu_dữ_liệu> <tên_thuộc_tính>;
```

- Khai báo và định nghĩa phương thức trong thân lớp

```
<kiểu_trả_về> <tên_PT>([danh sách đối])  
{  
    // Thân phương thức;  
}
```

# Định nghĩa lớp (tt)

- Định nghĩa phương thức bên ngoài thân lớp

```
<kiểu_trả_về> <Tên_Lớp>::<tên_PT>([ds đối])  
{  
    //Nội dung phương thức  
}
```

:: gọi là toán tử phạm vi



# Định nghĩa lớp (tt) – Ví dụ mẫu 1

```
class Nguoi
{
    private:
        char hoTen[30]    ;
        int tuoi;
        char que[60];
    public:
void nhap()
{cout<<"Nhap ho ten "; flush(stdin);gets(hoTen)
cout<<"Nhap tuoi"; cin>>tuoi;
cout<<"Nhap que quan « ;fflush(stdin);gets(que);
}
void xuat()
{cout<<"Ho va ten:"<<hoTen<<endl;
cout<<"Tuoi " <<tuoi<<endl;
cout<<"Que quan: " <<que<<endl;
}
};
```

# Định nghĩa lớp (tt) – Ví dụ mẫu 2

```
class Nguoi{
    private:
        char hoTen[30];
        int tuoi;
        char que[45];
    public:
        void nhap();
        void xuat();
};

void Nguoi::nhap() {
    cout<<"Nhap ho ten "; flush(stdin);gets(hoTen);
    cout<<"Nhap tuoi"; cin>>tuoi;
    cout<<"Nhap que quan ";fflush(stdin);gets(que);
}

void Nguoi::xuat() {
    cout<<"Ho va ten: "<<hoTen<<endl;
    cout<<"Tuoi: "<<tuoi<<endl;
    cout<<"Que quan: "<<que<<endl;
}
```

## 2.3. Phạm vi truy xuất

- Phạm vi truy xuất được thiết lập cho các thành phần của lớp gồm các thuộc tính và các phương thức.
- Phạm vi truy xuất xác định phần mã lệnh nào của chương trình được phép truy xuất vào các thành phần của lớp.
- Các loại phạm vi truy xuất gồm:
  - **private** – riêng tư
  - **protected** – bảo vệ
  - **public** – công cộng

# Phạm vi truy xuất (tt)

- **private:** Các thành phần có phạm vi **private** chỉ được truy xuất bởi các phương thức trong nội bộ lớp.
- **protected:** Các thành phần có phạm vi **protected** có thể được truy xuất bởi các phương thức bên trong lớp và các phương thức của **lớp dẫn xuất trực tiếp từ lớp** có các thành phần này.
- **public:** Các thành phần có phạm vi **public** có thể được truy xuất bởi mọi mã lệnh trong chương trình.

# Phạm vi truy xuất (tt)

- Cách đặt từ khóa chỉ định phạm vi truy xuất:

**<từ\_khóa\_phạm\_vi:>**

- Vị trí đặt từ khóa chỉ định phạm vi truy xuất:
  - Trước các thành phần cần xác định phạm vi truy xuất.
- Lưu ý:
  - Nếu không đặt từ khóa chỉ định phạm vi truy xuất trước các thành phần của lớp thì mặc định các thành phần này có phạm vi truy xuất là **private**.

# Phạm vi truy xuất (tt) – Ví dụ

```
class Nguoi
{
    private:
        char hoTen[30];
        char ngaySinh[11];
        float chieuCao;
    public:
        void nhap();
        void xuat();
};
```

# Phạm vi truy xuất (tt) – Ví dụ

```
class Nguoi
{
    char hoTen[30];
    char ngaySinh[11];
    float chieuCao;

    public:
        void nhap();
        void xuat();
};
```

Các thành phần này có  
phạm vi truy xuất là  
**private**

## 2.4. Sử dụng lớp

- Chương trình thực thi do sự hoạt động của các đối tượng được tạo ra trong chương trình.
- Trong chương trình, các lớp được xem như các kiểu dữ liệu, các đối tượng là các biến.
- Việc sử dụng lớp trong chương trình để khai báo các đối tượng cũng giống như sử dụng các kiểu dữ liệu để khai báo các biến.
- Để tạo ra các đối tượng trong chương trình cần khai báo thông qua các lớp đã được định nghĩa.



# Sử dụng lớp (tt)

- Khai báo đối tượng

```
<Tên_lớp> <tên_đối_tượng>;
```

- Ví dụ

```
Ngnoi ngnoi_1, ngnoi_2, ngnoi[20];
```

- Trong khai báo trên:

- `ngnoi_1, ngnoi_2`: là các đối tượng thuộc lớp `Ngnoi`.
- `ngnoi[20]`: là mảng chứa 20 đối tượng thuộc lớp `Ngnoi`.

# Sử dụng lớp (tt)

- Truy xuất phương thức của đối tượng

`<tên_đối_tượng>.<tên_phương_thức>`

- Ví dụ

`nguo_i_1.nhap();`

`nguo_i_1.xuat();`

- Truy xuất vào các thuộc tính của đối tượng

`<tên_đối_tượng>.<thuộc_tính>`

- Ví dụ

`nguo_i_1.hoTen, nguo_i[2].ngaySinh`

## 2.5. Cài đặt chương trình ứng dụng

1. Viết chương trình hướng đối tượng giải quyết bài toán tính diện tích hình thang.
2. Viết chương trình hướng đối tượng giải quyết bài toán tính chu vi, diện tích hình tròn.
3. Viết chương trình hướng đối tượng giải quyết bài toán giải phương trình bậc nhất một ẩn.
4. Viết chương trình hướng đối tượng giải quyết bài toán giải phương trình bậc 2.
5. Viết chương trình hướng đối tượng giải quyết bài toán tính chu vi và diện tích của một tam giác.
6. Viết chương trình hướng đối tượng giải quyết bài toán, nhập và hiển thị thông tin của một học sinh gồm: Họ tên, ngày, tháng, năm sinh, giới tính, điểm trung bình, xếp loại đạo đức.

# Cài đặt chương trình ứng dụng (tt)

- Cài đặt chương trình hướng đối tượng thực hiện các yêu cầu:
  - Thiết kế, cài đặt lớp CanBo bao gồm các thuộc tính: Mã cán bộ, họ và tên, ngày sinh, số ngày làm việc trong tháng và các phương thức:
  - Phương thức nhap() để nhập thông tin của cán bộ.
  - Phương thức tinhLuong(): Trả về lương của cán bộ theo công thức  $Lương = Số\ ngày\ làm\ việc\ trong\ tháng * 250.000$ .
  - Phương thức xuat(): hiển thị thông tin của cán bộ ra màn hình.
  - Xây dựng chương trình chính nhập thông tin cho 1 cán bộ. Xuất thông tin của cán bộ đó ra màn hình và cho biết Lương của cán bộ đó là bao nhiêu.

