

Kế thừa

Nguyễn Khắc Huy

cuu duong than cong . com

BMCNPM – ĐHKHTN TPHCM
10/2015



Nội dung

- **Khái niệm kế thừa.**
- Tầm vực trong kế thừa.
- Định nghĩa lại phương thức.
- Quan hệ IS-A và HAS-A.
- Bài tập.

cuu duong than cong . com

cuu duong than cong . com



Khái niệm kế thừa

□ Vấn đề trùng lặp thông tin:

✓ Nhiều lớp có thông tin giống nhau.

✓ Có 2 dạng:

– Dạng chia sẻ: $A \cap B \neq \emptyset$.

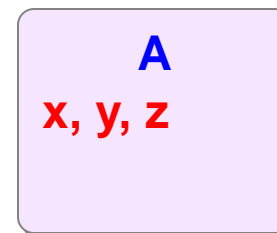
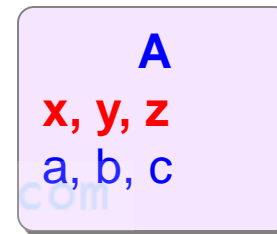
– Dạng mở rộng: $B = A + \epsilon$.

✓ Nhược điểm:

– Xây dựng tốn kém.

– Dung lượng lưu trữ lớn.

– Thay đổi phần chung khó khăn.



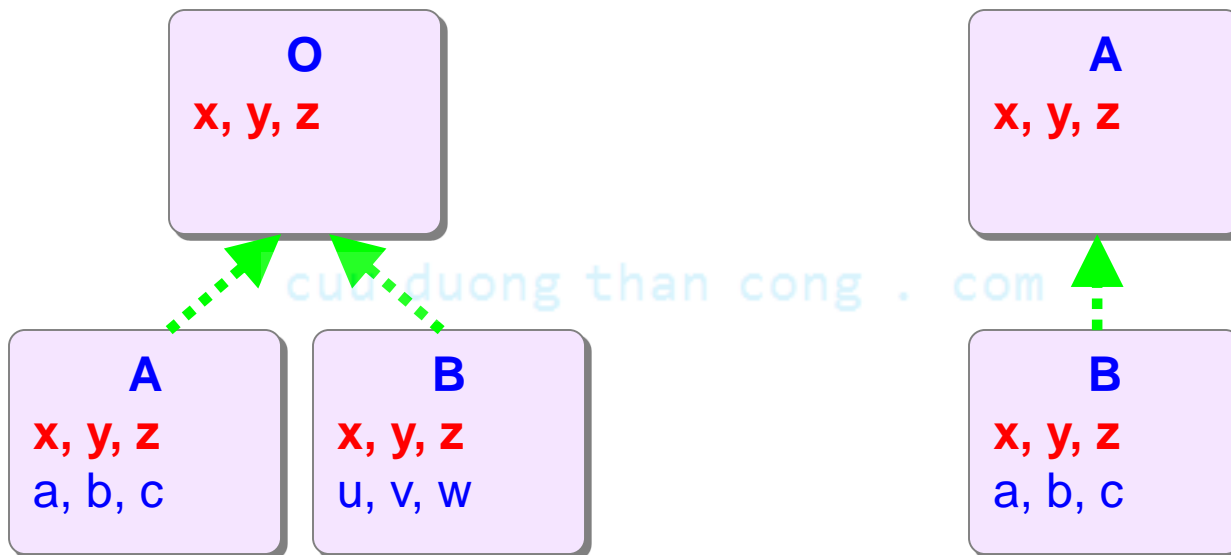
Giải quyết: tái sử dụng!!



Khái niệm kế thừa

□ Khái niệm kế thừa:

- ✓ Định nghĩa lớp mới dựa trên những lớp đã có.
- ✓ **Lớp cơ sở**: lớp dùng để định nghĩa lớp mới.
- ✓ **Lớp kế thừa**: lớp được định nghĩa từ lớp đã có.
- ✓ Lớp kế thừa thừa hưởng **TẤT CẢ** từ lớp cơ sở.



Khái niệm kế thừa

□ Khai báo trong C++:

class <Lớp kế thừa> : <Loại kế thừa> <Lớp cơ sở>

□ Loại kế thừa:

✓ public, private, protected.

□ Ví dụ:

class A : public O

{

private:

// Khai báo thuộc tính mới của A.

public:

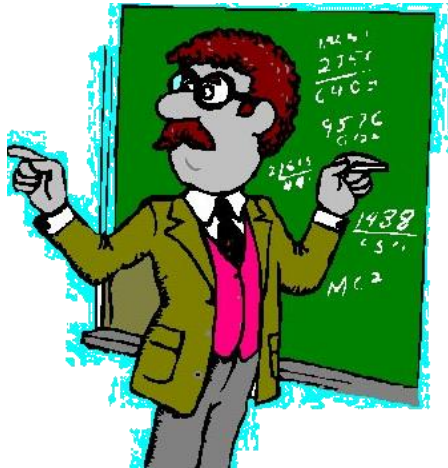
// Khai báo phương thức mới của A.

};



Khái niệm kế thừa

□ Ví dụ



Giáo viên

Thông tin:

Họ tên.
Mức lương.
Số ngày nghỉ.

Công việc:

Giảng dạy.
Tính lương.

Thông tin:

Họ tên.
Mức lương.
Số ngày nghỉ.

Lớp chủ nhiệm.

Công việc:

Giảng dạy.
Tính lương.
Sinh hoạt chủ nhiệm.



GVCN



Khái niệm kế thừa

□ Ví dụ:

```
class GiaoVien
{
private:
    char    *m_sHoTen;
    float   m_fMucLuong;
    int     m_iSoNgayNghỉ;
public:
    GiaoVien(char *sHoTen,
              float fMucLuong,
              int iSoNgayNghỉ);
    void giangDay();
    float tinhLuong();
};
```

Lớp kế thừa

Lớp cơ sở

```
class GVCN : public GiaoVien
{
private:
    char*m_sLopCN;
public:
    GVCN(char *sHoTen,
          float fMucLuong,
          int iSoNgayNghỉ,
          char *sLopCN);
    void sinhHoatCN();
};
```

GVCN thừa hưởng **TẤT CẢ** thuộc tính và phương thức của **GiaoVien**

Khái niệm kế thừa

□ Ví dụ:

```
void main()
```

```
{
```

```
    GiaoVien gv1("Minh", 500000, 5);
```

```
    gv1.giangDay();
```

```
    float fLuong1 = gv1.tinhLuong();
```

```
    GVCN gv2("Hanh", 700000, 3);
```

```
    gv2.giangDay();
```

```
    gv2.sinhHoatCN();
```

```
    float fLuong2 = gv2.tinhLuong();
```

```
}
```



Nội dung

- Khái niệm kế thừa.
- **Tầm vực trong kế thừa.**
- Định nghĩa lại phương thức.
- Quan hệ IS-A và HAS-A.
- Bài tập.

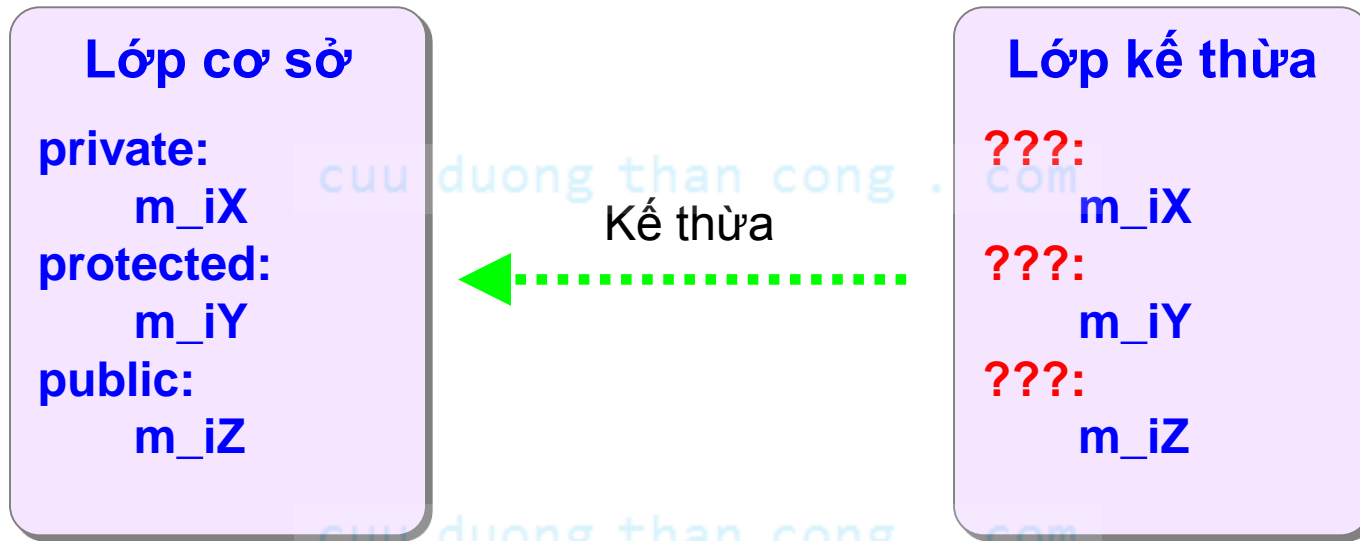
cuu duong than cong . com

cuu duong than cong . com



Tầm vực trong kế thừa

□ Tầm vực thay đổi thế nào khi kế thừa?



➔ **Loại kế thừa quyết định!!**



Tầm vực trong kế thừa

□ Bảng tầm vực trong kế thừa

Tầm vực	Kế thừa public	Kế thừa protected	Kế thừa private
public	public	protected	private
protected	protected	protected	private
private	<i>Không thể truy xuất</i>	<i>Không thể truy xuất</i>	<i>Không thể truy xuất</i>



Nội dung

- Khái niệm kế thừa.
- Tầm vực trong kế thừa.
- **Định nghĩa lại phương thức.**
- Quan hệ IS-A và HAS-A.
- Bài tập.

cuu duong than cong . com



Định nghĩa lại phương thức

□ Kế thừa một phần:

✓ Không kế thừa “máy móc” tất cả.

✓ **Lớp kế thừa có thể thay đổi những gì đã kế thừa!!**

➔ **Định nghĩa lại phương thức đã kế thừa.**

Lớp kế thừa thừa hưởng **TẤT CẢ** thuộc tính và phương thức của lớp cơ sở **TRỪ** những phương thức được định nghĩa lại!!



Định nghĩa lại phương thức

□ Ví dụ:

- ✓ GVCN kế thừa từ GiaoVien.
- ✓ GVCN tính lương khác GiaoVien.
 - Lương GV = Mức lương – Số ngày nghỉ * 10000.
 - Lương GVCN = Lương GV + Phụ cấp 50000.

→ **Viết lại phương thức tính Lương() cho lớp GVCN.**

cuu duong than cong . com



Định nghĩa lại phương thức

- Ví dụ:

```
class GiaoVien
```

```
{
```

```
private:
```

```
    char    *m_sHoTen;
```

```
    float   m_fMucLuong;
```

```
    int     m_iSoNgayNghỉ;
```

```
public:
```

```
    GiaoVien(char *sHoTen, float fMucLuong, int iSoNgayNghỉ);
```

```
    void giangDay();
```

```
    float tinhLuong()
```

```
    {
```

```
        return m_fMucLuong – m_iSoNgayNghỉ * 10000;
```

```
    }
```

```
};
```



Định nghĩa lại phương thức

□ Ví dụ:

```
class GVCN : public GiaoVien
```

```
{
```

```
private:
```

```
    char    *m_sLopCN;
```

```
public:
```

```
    GVCN(char *sHoTen,  
          float fMucLuong,  
          int iSoNgayNghỉ,  
          char *sLopCN);
```

```
    void sinhHoatCN();
```

```
    float tinhLuong()
```

```
{
```

```
        return GiaoVien::TinhLuong() + 50000;
```

```
}
```

```
};
```

```
void main()
```

```
{
```

```
    GiaoVien gv1("Minh", 500000, 5);  
    gv1.giangDay();
```

```
    float fLuong1 = gv1.tinhLuong();
```

```
    GVCN gv2("Hanh", 700000, 3);
```

```
    gv2.giangDay();
```

```
    float fLuong2 = gv2.tinhLuong();
```

```
}
```



Nội dung

- Khái niệm kế thừa.
- Tầm vực trong kế thừa.
- Định nghĩa lại phương thức.
- **Quan hệ IS-A và HAS-A.**
- Bài tập.

cuu duong than cong . com



Quan hệ IS-A và HAS-A

□ Quan hệ IS-A:

- ✓ Lớp A quan hệ IS-A với lớp B
 - A là một trường hợp đặc biệt của B.
 - A cùng loại với B.

□ Ví dụ:

- ✓ GVCN là một GiaoVien đặc biệt.
- ✓ HìnhVuong là một HìnhChuNhat đặc biệt.
- ✓ ConMeo là một ConVat đặc biệt.



Quan hệ IS-A và HAS-A

□ Quan hệ HAS-A:

- ✓ Lớp A quan hệ HAS-A với lớp B
 - A bao hàm B.
 - A chứa B.
 - B là một bộ phận của A.

□ Ví dụ:

- ✓ ChiecXe chứa BanhXe.
- ✓ QuyenSach chứa TrangSach.



Quan hệ IS-A và HAS-A

□ Dr. Guru khuyên: **luật xây dựng lớp.**

✓ A có quan hệ IS-A với B.

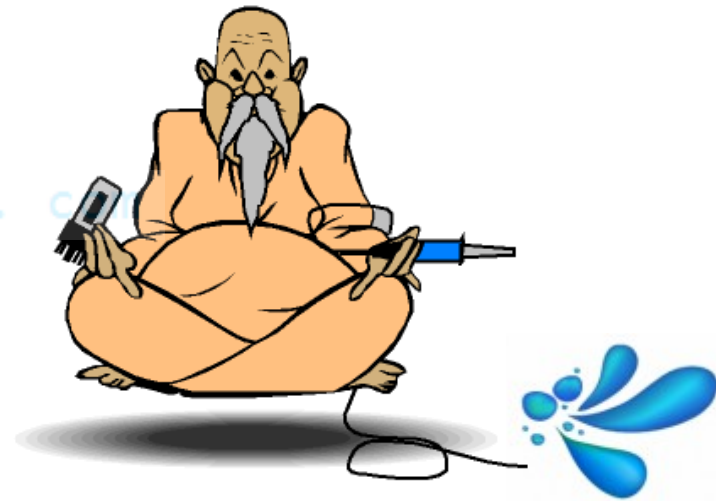
➔ **Cho A kế thừa B.**

✓ A có quan hệ HAS-A với B.

➔ **Cho B là một thuộc tính của A.**

□ Ví dụ:

```
class ConMeo : public ConVat { };  
class ChiecXe  
{  
private:  
    BanhXe *m_pBanhXe;  
};
```



Tóm tắt

□ Khái niệm kế thừa:

- ✓ Định nghĩa lớp mới dựa trên những lớp đã có.
- ✓ Lớp kế thừa thừa hưởng tất cả từ lớp cơ sở.

□ Tầm vực trong kế thừa:

- ✓ Tầm vực thay đổi tùy theo loại kế thừa.

□ Định nghĩa lại phương thức:

- ✓ Thay đổi những phương thức kế thừa từ lớp cơ sở.

□ Quan hệ IS-A và HAS-A:

- ✓ IS-A: A là trường hợp đặc biệt của $B \Rightarrow A$ kế thừa B .

- ✓ HAS-A: A bao hàm $B \Rightarrow B$ là thuộc tính của A .



Nội dung

- Khái niệm kế thừa.
- Tầm vực trong kế thừa.
- Định nghĩa lại phương thức.
- Quan hệ IS-A và HAS-A.
- **Bài tập.**

cuu duong than cong . com

cuu duong than cong . com



Bài tập

- **Bài tập 6.1:** Những cặp đối tượng sau có quan hệ IS-A hay HAS-A?

Khai báo lớp cho từng cặp thể hiện quan hệ giữa chúng.

- Hình vuông / Hình chữ nhật.
- Đa giác / Cạnh.
- Giám đốc / Nhân viên.
- Hình tròn / Hình Ellipse.
- Máy bay / Động cơ.
- Câu / Từ.
- Mỹ phẩm / Hàng hóa.
- Cây lúa / Cây lương thực.
- Thư viện / Sách.
- Phim hoạt hình / Phim ảnh.



Bài tập

- **Bài tập 6.2:** Cho lớp TaiKhoan như bên dưới

```
class TaiKhoan
{
private:
    float    m_fSoDu = 0;
public:
    float baoSoDu() { return m_fSoDu; }
    void napTien(float fSoTien) { m_fSoDu += fSoTien; }
    void rutTien(float fSoTien)
    {
        if (fSoTien <= m_fSoDu)
            m_fSoDu -= fSoTien;
    }
};
```



Bài tập

- **Bài tập 6.2 (tt.):** Dựa trên lớp TaiKhoan, xây dựng lớp TaiKhoanTietKiem như sau:
 - Có thêm thông tin:
 - Kỳ hạn gửi.
 - Lãi suất.
 - Số tháng đã gửi.
 - Khi nạp tiền, số tháng đã gửi được tính lại từ đầu.
 - Chỉ được rút tiền khi đến kỳ hạn.
 - Cho phép tăng số tháng đã gửi.
 - Tính số dư tại thời điểm hiện tại.

cuu duong than cong . com



Bài tập

- **Bài tập 6.3:** Một chiếc xe máy chạy 100km tốn 2lit xăng, cứ chở thêm 10kg hàng xe tốn thêm 0.1lit xăng.

Một chiếc xe tải chạy 100km tốn 20lit xăng, cứ chở thêm 1000kg hàng xe tốn thêm 1lit xăng.

Dùng kế thừa xây dựng lớp XeMay và XeTai cho phép:

- Chất một lượng hàng lên xe.
- Bỏ bớt một lượng hàng xuống xe.
- Đổ một lượng xăng vào xe.
- Cho xe chạy một đoạn đường.
- Kiểm tra xem xe đã hết xăng chưa.
- Cho biết lượng xăng còn trong xe.



Bài tập

- **Bài tập 6.4:** Xây dựng lớp MyString cho phép thực hiện các công việc sau
 - Phương thức khởi tạo từ 1 char*
 - Phương thức khởi tạo mặc định
 - Toán tử +, []
 - Toán tử ép kiểu char*
 - Các toán tử so sánh: >, <, ==, !=
 - Các phương thức xử lý
 - ✓ append
 - ✓ indexOf, lastIndexOf
 - ✓ substr
 - ✓ insert



Lời cảm ơn

- Nội dung được xây dựng dựa trên slide trình bày của Thầy Đinh Bá Tiến, Thầy Nguyễn Minh Huy.

cuu duong than cong . com

cuu duong than cong . com

