

Chương 10

KẾ THỪA – INHERITANCE

- Nguyễn Hữu Lợi
- Đoàn Chánh Thống
- ThS. Nguyễn Thành Hiệp
- ThS. Trương Quốc Dũng
- ThS. Võ Duy Nguyên
- ThS. Nguyễn Văn Toàn
- TS. Nguyễn Duy Khánh
- TS. Nguyễn Tấn Trần Minh Khang

1. MỤC TIÊU



1. Mục tiêu

- Hiểu được các loại quan hệ?
- Hiểu được kế thừa trong lập trình hướng đối tượng (object-oriented programming) là gì?
- Hiểu được khái niệm cây kế thừa.
- Hiểu được khái niệm sơ đồ lớp.

2. QUAN HỆ

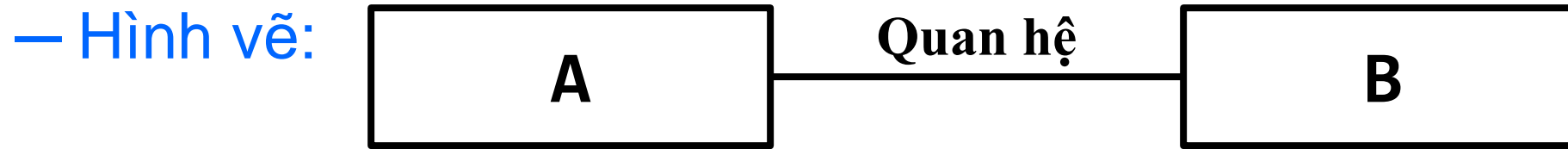
2. Quan hệ

Người ta chia các quan hệ thành những loại như sau:

- Quan hệ một một (1–1).
- Quan hệ một nhiều (1–n).
- Quan hệ nhiều nhiều (m–n).
- Quan hệ đặc biệt hóa, tổng quát hóa.

Quan hệ một một (1-1)

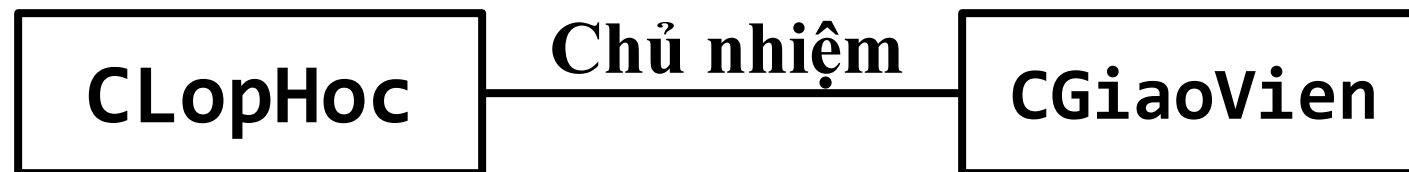
— Khái niệm: Hai lớp đối tượng được gọi là quan hệ một-một với nhau khi một đối tượng thuộc lớp này quan hệ với một đối tượng thuộc lớp kia và một đối tượng thuộc lớp kia quan hệ duy nhất với một đối tượng thuộc lớp này.



— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp A quan hệ với một đối tượng thuộc lớp B và một đối tượng lớp B quan hệ duy nhất với một đối tượng thuộc lớp A.

Quan hệ một một (1-1)

- Ví dụ 01: Xét ngữ cảnh trường PTTH Bạch Thái Bưởi thuộc quận Hồ Hoàn Kiếm thành phố Hà Nội trong niên khóa 2018 – 2019:



- Trong hình vẽ trên ta nói: một đối tượng thuộc lớp CLopHoc có quan hệ Chủ nhiệm với một đối tượng thuộc lớp CGiaoVien và ngược lại một đối tượng thuộc lớp CGiaoVien có quan hệ Chủ nhiệm với duy nhất một đối tượng thuộc lớp CLopHoc.

Quan hệ một một (1-1)

- Ví dụ 02: Xét ngữ cảnh nước Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam, theo hiến pháp và pháp luật tại thời điểm 2019:



- Trong hình vẽ trên ta nói: một đối tượng thuộc lớp CVo có quan hệ Hôn nhân với một đối tượng thuộc lớp CChong và ngược lại một đối tượng thuộc lớp CChong có quan hệ Hôn nhân với duy nhất một đối tượng thuộc lớp CVo.

Quan hệ một nhiều (1-n)

— Khái niệm: Hai lớp đối tượng được gọi là quan hệ một nhiều với nhau khi một đối tượng thuộc lớp này quan hệ với nhiều đối tượng thuộc lớp kia và một đối tượng lớp kia quan hệ duy nhất với một đối tượng thuộc lớp này.

— Hình vẽ:



— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp A quan hệ với nhiều đối tượng thuộc lớp B và một đối tượng lớp B có quan hệ duy nhất với một đối tượng thuộc lớp A.

Quan hệ một nhiều (1-n)

- Ví dụ 01: Xét ngữ cảnh trường PTTH Bạch Thái Bưởi trong niên khóa 2018 – 2019.



- Trong hình vẽ trên ta nói: một đối tượng thuộc lớp CLopHoc có quan hệ theo học với nhiều đối tượng thuộc lớp CHocSinh và ngược lại một đối tượng thuộc lớp CHocSinh có quan hệ theo học với duy nhất một đối tượng thuộc lớp CLopHoc.


Quan hệ một nhiều (1-n)

— Ví dụ 02: Xét ngữ cảnh quả đất.



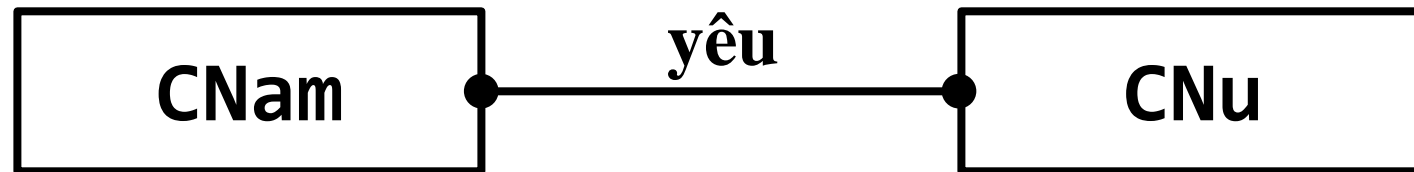
— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp CCha có quan hệ huyết thống với nhiều đối tượng thuộc lớp CCon và ngược lại một đối tượng thuộc lớp CCon có quan hệ huyết thống với duy nhất một đối tượng thuộc lớp CCha.

Quan hệ nhiều nhiều (m-n)

- Khái niệm: Hai lớp đối tượng được gọi là quan hệ nhiều-nhiều với nhau khi một đối tượng thuộc lớp này quan hệ với nhiều đối tượng thuộc lớp kia và một đối tượng lớp kia cũng có quan hệ với nhiều đối tượng thuộc lớp này.
- Hình vẽ:
- Trong hình vẽ trên ta nói: một đối tượng thuộc lớp A quan hệ với nhiều đối tượng thuộc lớp B và một đối tượng lớp B cũng có quan hệ với nhiều đối tượng thuộc lớp A.

Quan hệ nhiều nhiều (m-n)

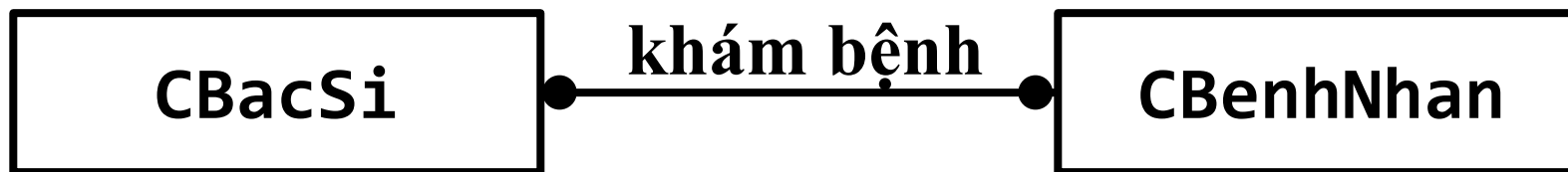
— Ví dụ 01: Xét ngữ cảnh quả đất từ hồi nằm đến bi chừ.



— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp CNaM quan hệ yêu với nhiều đối tượng thuộc lớp CNu và một đối tượng lớp CNu cũng có quan hệ yêu với nhiều đối tượng thuộc lớp CNaM.

Quan hệ nhiều nhiều (m-n)

— Ví dụ 02: Xét ngữ cảnh quả đất từ hồi ấy đến hồi này.

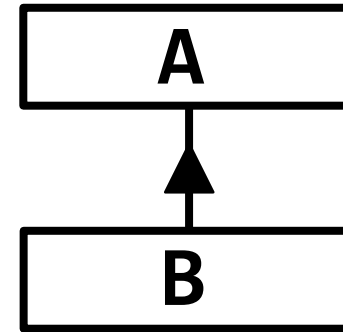


— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp **CBacSi** quan hệ **khám bệnh** với nhiều đối tượng thuộc lớp **CBenhNhan** và một đối tượng lớp **CBenhNhan** cũng có quan hệ **khám bệnh** với nhiều đối tượng thuộc lớp **CBacSi**.

Quan hệ đặc biệt hóa – tổng quát hóa

- **Khái niệm:** Hai lớp đối tượng được gọi là quan hệ đặc biệt hóa – tổng quát hóa với nhau khi, lớp đối tượng này là trường hợp đặc biệt của lớp đối tượng kia và lớp đối tượng kia là trường hợp tổng quát của lớp đối tượng này.

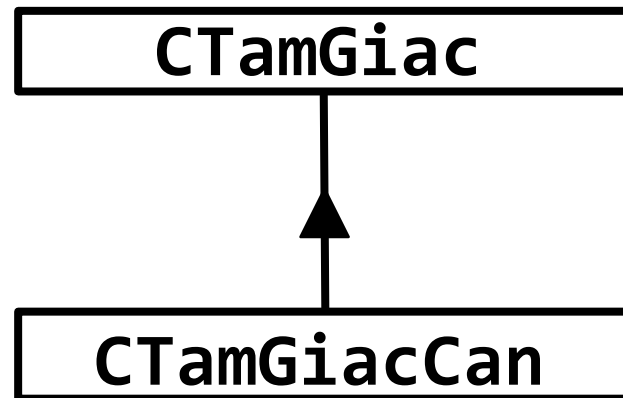
- Hình vẽ



- Trong hình vẽ trên ta nói: lớp đối tượng **B** là trường hợp đặc biệt của lớp đối tượng **A** và lớp đối tượng **A** là trường hợp tổng quát của lớp đối tượng **B**.

Quan hệ đặc biệt hóa – tổng quát hóa

— Ví dụ 01:



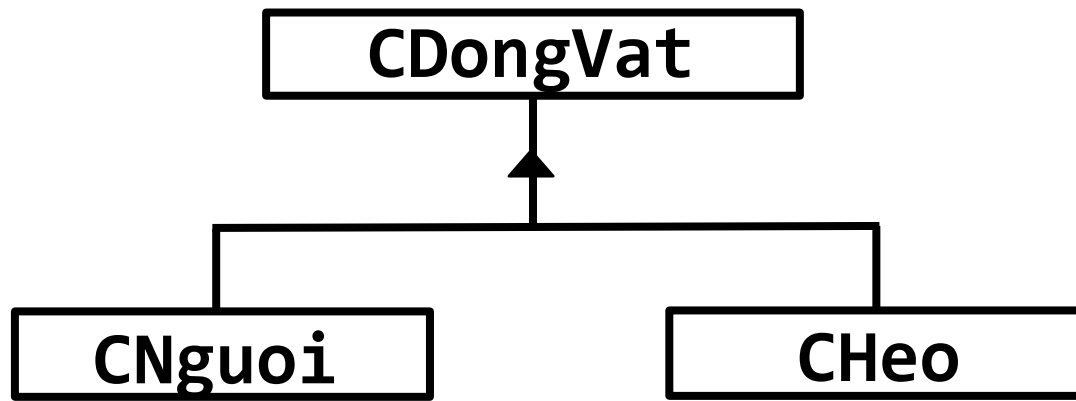
— Trong hình vẽ trên ta nói:

+ lớp đối tượng **CTamGiacCan** là trường hợp đặc biệt của lớp đối tượng **CTamGiac** và

+ lớp đối tượng **CTamGiac** là trường hợp tổng quát của lớp đối tượng **CTamGiacCan**.

Quan hệ đặc biệt hóa – tổng quát hóa

— Ví dụ 02:



— Trong hình vẽ trên ta nói:

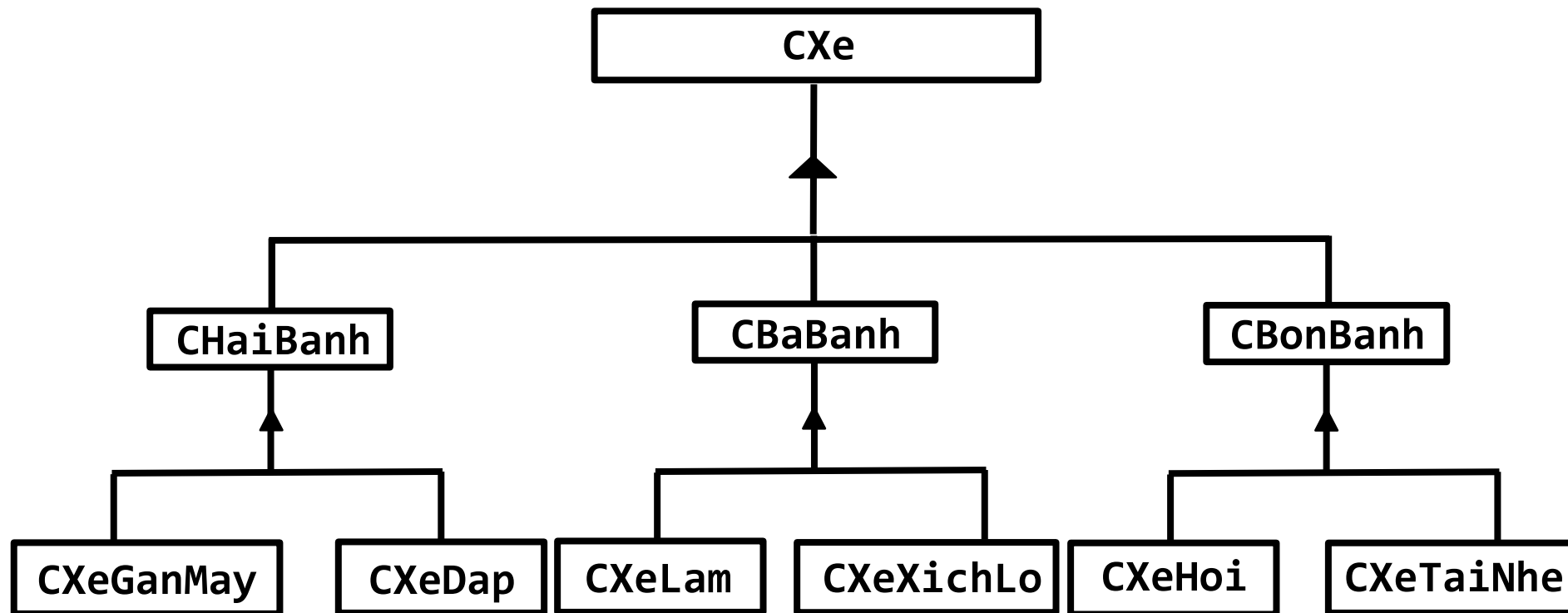
- + lớp đối tượng **CNgươi** và lớp đối tượng **CHeo** là trường hợp đặc biệt của lớp đối tượng **CDongVat** và
- + lớp đối tượng **CDongVat** là trường hợp tổng quát của lớp đối tượng **CNgươi** và lớp đối tượng **CHeo**.

3. CÂY KẾ THỪA

3. Cây kế thừa

- **Khái niệm:** Cây kế thừa là một cây đa nhánh thể hiện mối quan hệ đặc biệt hóa-tổng quát hóa giữa các lớp trong hệ thống, trong chương trình.
- Ví dụ: Hãy vẽ cây kế thừa cho các lớp đối tượng sau:
 - + Lớp CXdap
 - + Lớp CXeGanMay
 - + Lớp CXeHoi
 - + Lớp CXeHaiBanh
 - + Lớp CXeTaiNhe
 - + Lớp CXeLam
 - + Lớp CXe
 - + Lớp CXeBaBanh
 - + Lớp CXeBonBanh
 - + Lớp CXeXichLo

3. Cây kế thừa

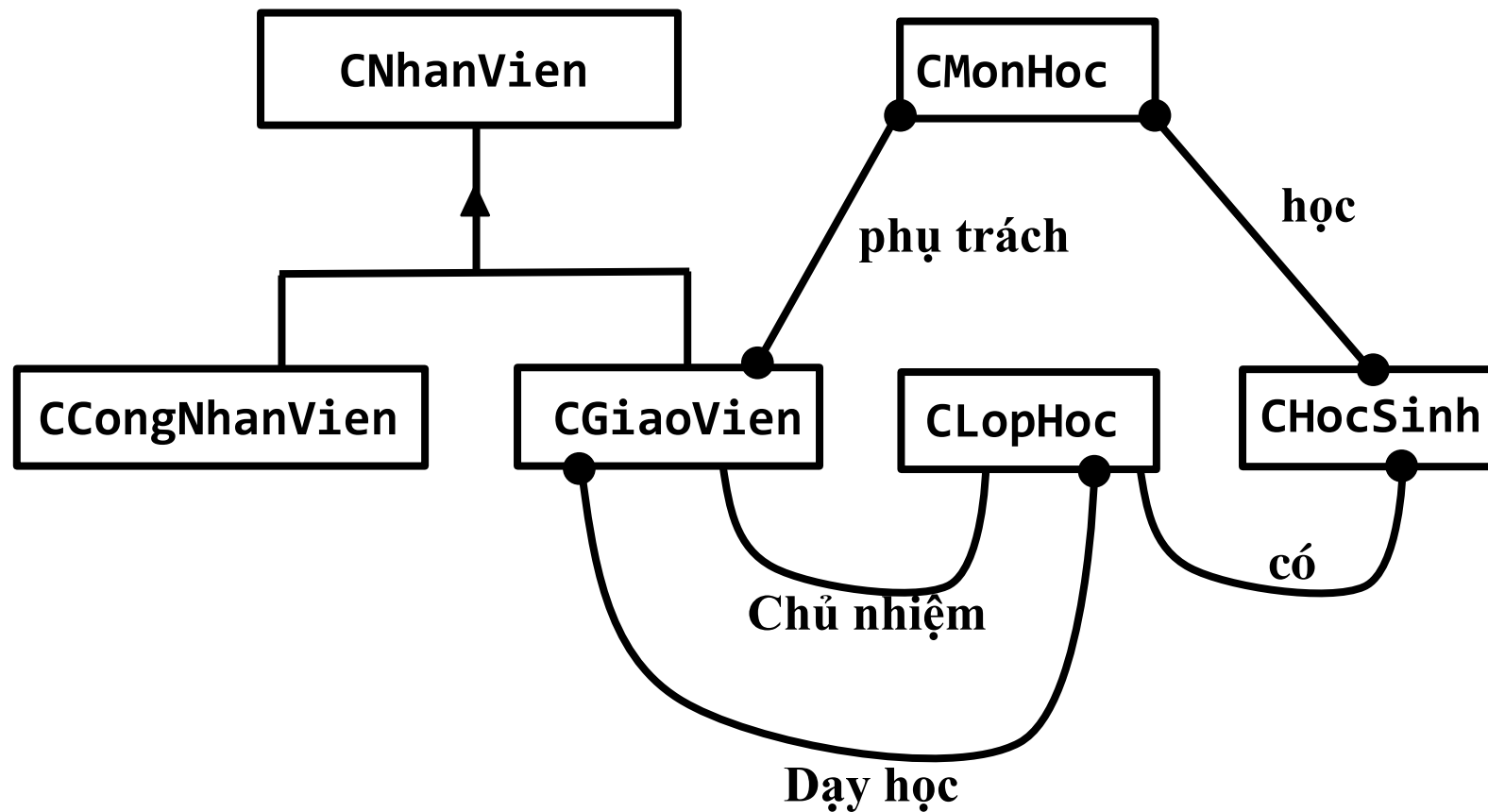


4. SƠ ĐỒ LỚP

Sơ đồ lớp

- **Khái niệm:** Sơ đồ lớp là sơ đồ thể hiện tất cả các mối quan hệ giữa các lớp trong hệ thống, trong chương trình.
- Xét ngữ cảnh trường PTTH Bạch Đằng trong niên học 2019-2020. Hãy vẽ sơ đồ lớp cho các lớp đối tượng sau:
 - + Lớp CGiaoVien
 - + Lớp CHocSinh
 - + Lớp CLopHoc
 - + Lớp CMonHoc
 - + Lớp CNhanVien
 - + Lớp CCongNhanVien
 - + Lớp CNhanVien: tất cả những nhân viên làm việc trong trường.
 - + Lớp CCongNhanVien: là các nhân viên làm việc trong nhà trường nhưng ko trực tiếp đứng lớp. Ví dụ: Bảo vệ, lao công, bảo mẫu.

Sơ đồ lớp



Access control

ĐIỀU KHIỂN TRUY XUẤT

Điều khiển truy xuất

— Một thuộc tính hay một phương thức khi được khai báo trong một lớp ta có thể khai báo trong 3 phạm vi khác nhau: **private**, **public** hoặc **protected**.

private
public
protected

— Ví dụ:

```
11.class A
12.{
13.    private:
14.        int a;
15.        void f();
16.    protected:
17.        int b;
18.        void g();
19.    public:
20.        int c;
21.        void h();
22.};
```

Điều khiển truy xuất

- Một thuộc tính hay một phương thức khi được khai báo trong một lớp ta có thể khai báo trong 3 phạm vi khác nhau: **private**, **public** hoặc **protected**.

private
public
protected

- Ví dụ:

```
11.class A
12.{
13.    private:
14.        int a;
15.        void f();
16.    protected:
17.        int b;
18.        void g();
19.    public:
20.        int c;
21.        void h();
22.};
```

Thuộc tính **a** và
phương thức **f**
được khai báo
trong phạm vi
private

Điều khiển truy xuất

- Một thuộc tính hay một phương thức khi được khai báo trong một lớp ta có thể khai báo trong 3 phạm vi khác nhau: **private**, **public** hoặc **protected**.

private
public
protected

- Ví dụ:

```
11.class A
12.{
13.    private:
14.        int a;
15.        void f();
16.    protected:
17.        int b;
18.        void g();
19.    public:
20.        int c;
21.        void h();
22.};
```

Thuộc tính **b** và
phương thức **g**
được khai báo
trong phạm vi
protected

Điều khiển truy xuất

- Một thuộc tính hay một phương thức khi được khai báo trong một lớp ta có thể khai báo trong 3 phạm vi khác nhau: **private**, **public** hoặc **protected**.

private
public
protected

- Ví dụ:

```
11.class A
12.{
13.    private:
14.        int a;
15.        void f();
16.    protected:
17.        int b;
18.        void g();
19.    public:
20.        int c;
21.        void h();
22.};
```

Thuộc tính **c** và
phương thức **h**
được khai báo
trong phạm vi
public

Quy tắc truy xuất

PRIVATE

Các thuộc tính và phương thức được khai báo trong phạm vi **private** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PRIVATE

— Ví dụ:

```
11. class A
12. {
13.     private:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15;
26. }
```


Quy tắc truy xuất

PRIVATE

Các thuộc tính và phương thức được khai báo trong phạm vi **private** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PRIVATE

— Ví dụ:

```
11. class A
12. {
13.     private:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15;
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **private**

Quy tắc truy xuất

PRIVATE

Các thuộc tính và phương thức được khai báo trong phạm vi **private** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PRIVATE

— Ví dụ:

```
11. class A
12. {
13.     private:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15;
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **private**

Quy tắc truy xuất

PRIVATE

Các thuộc tính và phương thức được khai báo trong phạm vi **private** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PRIVATE

— Ví dụ:

```
11. class A
12. {
13.     private:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **private**

Quy tắc truy xuất

PRIVATE

Các thuộc tính và phương thức được khai báo trong phạm vi **private** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PRIVATE

— Ví dụ:

```
11. class A
12. {
13.     private:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **private**

Quy tắc truy xuất

PRIVATE

Các thuộc tính và phương thức được khai báo trong phạm vi **private** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PRIVATE

— Ví dụ:

```
11. class A
12. {
13.     private:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15; SAI
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **private**

Quy tắc truy xuất

PRIVATE

Các thuộc tính và phương thức được khai báo trong phạm vi **private** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PRIVATE

— Ví dụ:

```
11. class A
12. {
13.     private:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15; SAI
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **private**

Quy tắc truy xuất

PRIVATE

Các thuộc tính và phương thức được khai báo trong phạm vi **private** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PRIVATE

— Ví dụ:

```
11. class A
12. {
13.     private:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15; SAI
21.     x.f(); SAI
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **private**

Quy tắc truy xuất

PROTECTED

Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PROTECTED

— Ví dụ:

```
11. class A
12. {
13.     protected:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15;
26. }
```

Quy tắc truy xuất

PROTECTED

Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PROTECTED

— Ví dụ:

```
11. class A
12. {
13.     protected:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15;
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **protected**

Quy tắc truy xuất

PROTECTED

Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PROTECTED

— Ví dụ:

```
11. class A
12. {
13.     protected:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15;
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **protected**

Quy tắc truy xuất

PROTECTED

Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PROTECTED

— Ví dụ:

```
11. class A
12. {
13.     protected:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **protected**

Quy tắc truy xuất

PROTECTED

Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PROTECTED

— Ví dụ:

```
11. class A
12. {
13.     protected:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **protected**

Quy tắc truy xuất

PROTECTED

Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PROTECTED

— Ví dụ:

```
11. class A
12. {
13.     protected:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15; SAI
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **protected**

Quy tắc truy xuất

PROTECTED

Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PROTECTED

— Ví dụ:

```
11. class A
12. {
13.     protected:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15; SAI
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **protected**

Quy tắc truy xuất

PROTECTED

Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của một lớp thì chỉ được phép truy xuất bên trong lớp (**accessible only inside the class**) và không được quyền truy xuất bên ngoài lớp.

PROTECTED

— Ví dụ:

```
11. class A
12. {
13.     protected:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15; SAI
21.     x.f(); SAI
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **protected**

Quy tắc truy xuất

PUBLIC

Các thuộc tính và phương thức được khai báo trong phạm vi **public** của một lớp thì không chỉ được phép truy xuất bên trong lớp và cả bên ngoài lớp.

PUBLIC

— Ví dụ:

```
11. class A
12. {
13.     public:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15;
26. }
```

Quy tắc truy xuất

PUBLIC

Các thuộc tính và phương thức được khai báo trong phạm vi **public** của một lớp thì không chỉ được phép truy xuất bên trong lớp và cả bên ngoài lớp.

— Ví dụ:

```
11. class A
12. {
13.     public:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15;
26. }
```

PUBLIC

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **public**

Quy tắc truy xuất

PUBLIC

Các thuộc tính và phương thức được khai báo trong phạm vi **public** của một lớp thì không chỉ được phép truy xuất bên trong lớp và cả bên ngoài lớp.

PUBLIC

— Ví dụ:

```
11. class A
12. {
13.     public:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15;
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **public**

Quy tắc truy xuất

PUBLIC

Các thuộc tính và phương thức được khai báo trong phạm vi **public** của một lớp thì không chỉ được phép truy xuất bên trong lớp và cả bên ngoài lớp.

PUBLIC

— Ví dụ:

```
11. class A
12. {
13.     public:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **public**

Quy tắc truy xuất

PUBLIC

Các thuộc tính và phương thức được khai báo trong phạm vi **public** của một lớp thì không chỉ được phép truy xuất bên trong lớp và cả bên ngoài lớp.

PUBLIC

— Ví dụ:

```
11. class A
12. {
13.     public:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15;
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **public**

Quy tắc truy xuất

PUBLIC

Các thuộc tính và phương thức được khai báo trong phạm vi **public** của một lớp thì không chỉ được phép truy xuất bên trong lớp và cả bên ngoài lớp.

PUBLIC

— Ví dụ:

```
11. class A
12. {
13.     public:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15; ĐÚNG
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **public**

Quy tắc truy xuất

PUBLIC

Các thuộc tính và phương thức được khai báo trong phạm vi **public** của một lớp thì không chỉ được phép truy xuất bên trong lớp và cả bên ngoài lớp.

PUBLIC

— Ví dụ:

```
11. class A
12. {
13.     public:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15; ĐÚNG
21.     x.f();
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **public**

Quy tắc truy xuất

PUBLIC

Các thuộc tính và phương thức được khai báo trong phạm vi **public** của một lớp thì không chỉ được phép truy xuất bên trong lớp và cả bên ngoài lớp.

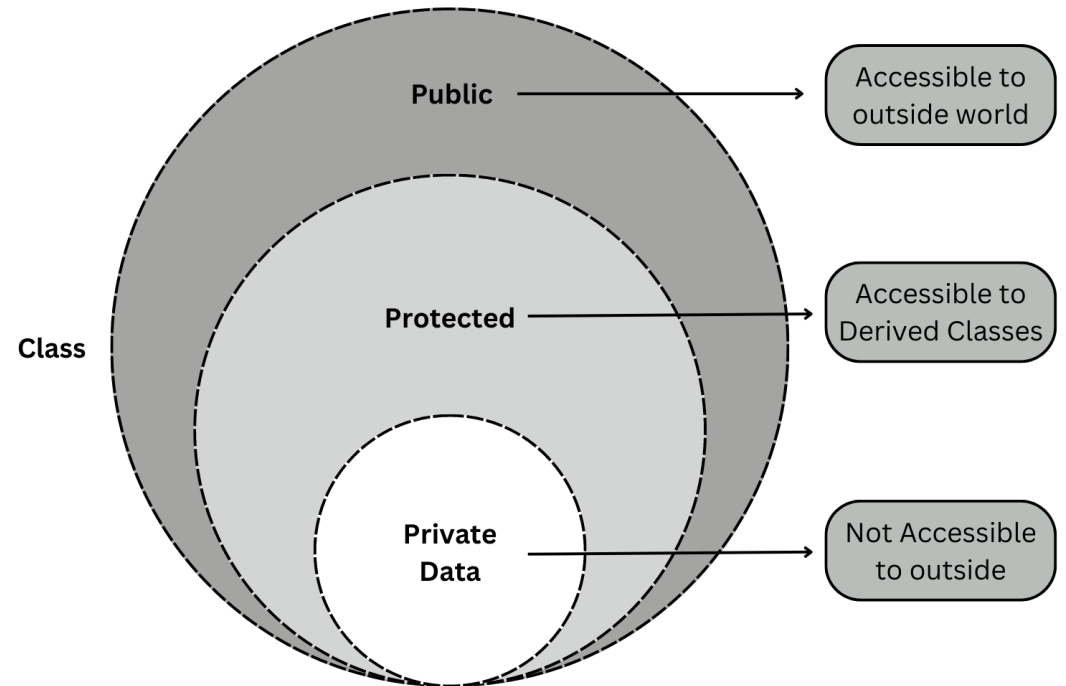
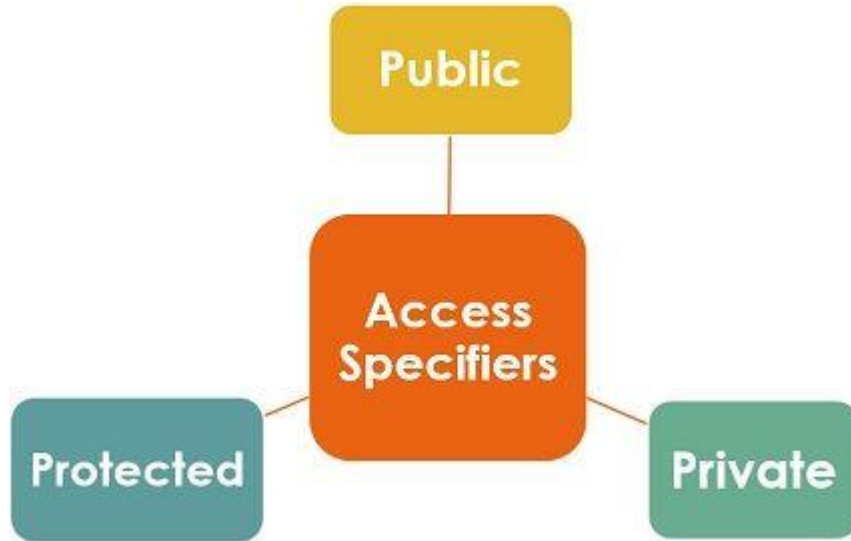
PUBLIC

— Ví dụ:

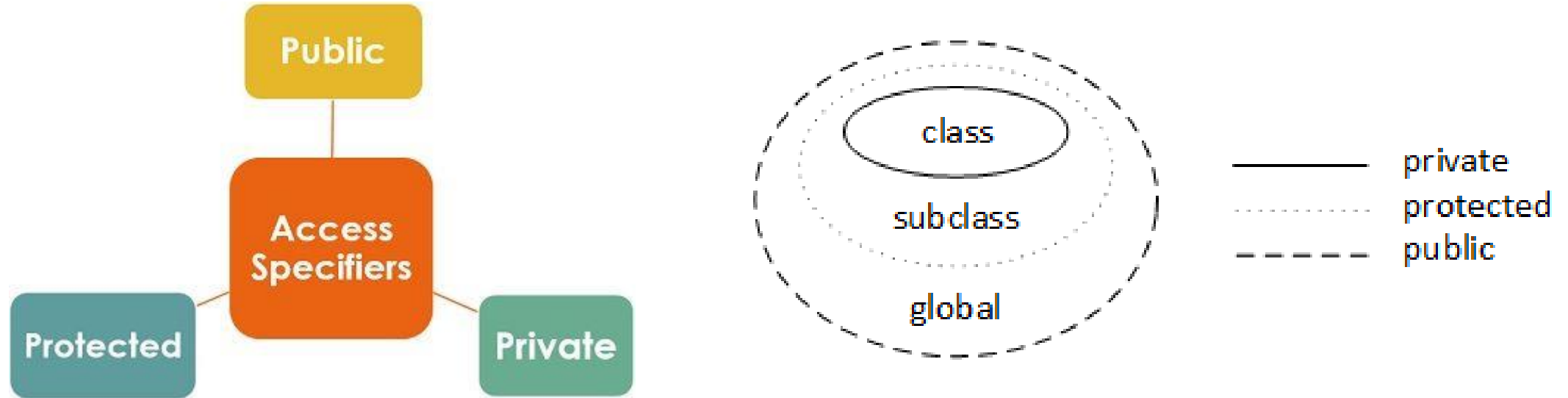
```
11. class A
12. {
13.     public:
14.         int a;
15.         void f();
16. };
17. void main()
18. {
19.     A x;
20.     x.a = 15; ĐÚNG
21.     x.f(); ĐÚNG
22. }
23. void A::f()
24. {
25.     a = 15; ĐÚNG
26. }
```

Thuộc tính **a** và phương thức **f** được khai báo trong phạm vi **public**

Quy tắc truy xuất



Quy tắc truy xuất

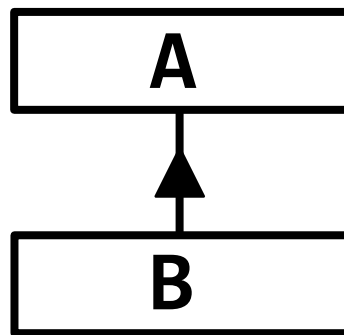


Inheritance

6. KẾ THỪA – INHERITANCE

Thế giới thực

— Hình vẽ

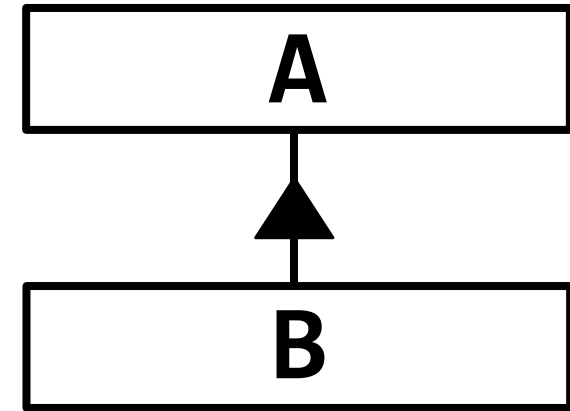


- Trong hình vẽ trên ta nói **A** và **B** có quan hệ đặc biệt hoá, tổng quát hoá với nhau.
- Trong đó **B** là trường hợp đặt biệt của **A** và **A** là trường hợp tổng quát của **B**.

Lập trình hướng đối tượng

— Xét khai báo.

```
1. class A
2. {
3. | ...
4. };
5. class B:<từ khóa dẫn xuất> A
6. {
7. | ...
8. };
```



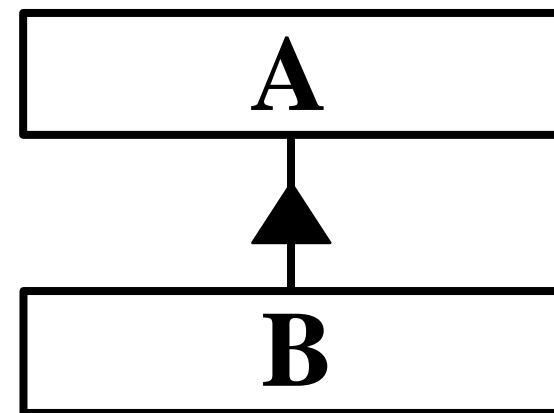
- Khai báo trên ta nói lớp đối tượng **B** kế thừa từ lớp đối tượng **A**.
- Lớp đối tượng **A** được gọi là lớp cơ sở.
- Lớp đối tượng **B** được gọi là lớp dẫn xuất từ lớp đối tượng **A**.

Từ khóa dẫn xuất

— Xét khai báo.

```

1. class A
2. {
3. | ...
4. };
5. class B:<từ khóa dẫn xuất> A
6. {
7. | ...
8. };
  
```



— Trong ngôn ngữ C++ có ba loại **từ khóa dẫn xuất** đó là: **private**, **protected** và **public**. Thông thường trong thực tế người ta hay sử dụng từ khóa dẫn xuất **public** là nhiều nhất.

Ví dụ kế thừa

— Ví dụ 01: Khai báo lớp tam giác (CTamGiac) và lớp tam giác cân (CTamGiacCan).

```
1. class CTamGiac
```

```
2. {
```

```
3. |    ...
```

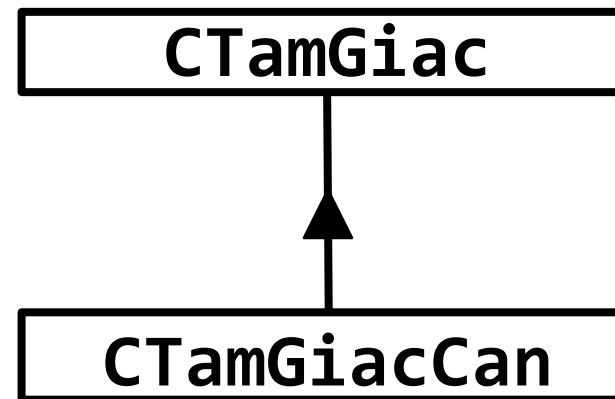
```
4. };
```

```
5. class CTamGiacCan:public CTamGiac
```

```
6. {
```

```
7. |    ...
```

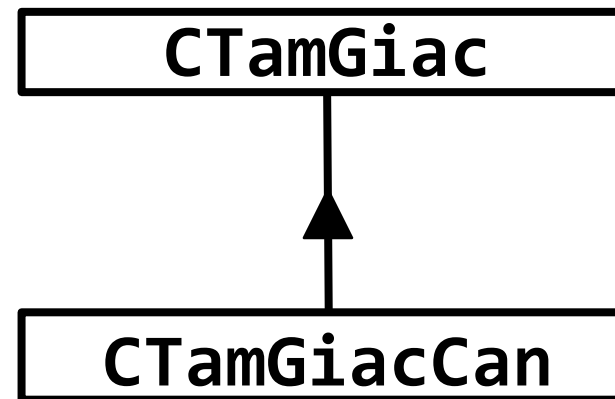
```
8. };
```



Ví dụ kế thừa

— Ví dụ 01: Khai báo lớp tam giác (CTamGiac) và lớp tam giác cân (CTamGiacCan).

```
1. class CTamGiac
2. {
3. |    ...
4. };
5. class CTamGiacCan:public CTamGiac
6. {
7. |    ...
8. };
```

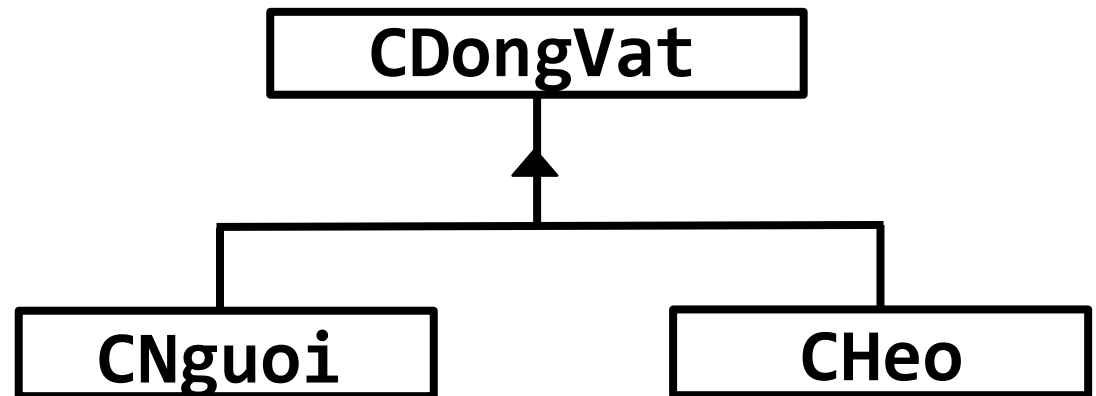


Dòng số 5 ta nói lớp đối tượng **CTamGiacCan** kế thừa từ lớp đối tượng **CTamGiac** bằng từ khóa dẫn xuất **public**

Ví dụ kế thừa

— Ví dụ 02: Khai báo lớp động vật, lớp heo và lớp người.

```
11. class CDongVat
12. {
13. |    ...
14. };
15. class CHeo:private CDongVat
16. {
17. |    ...
18. };
19. class CNgnoi:public CDongVat
20. {
21. |    ...
22. };
```



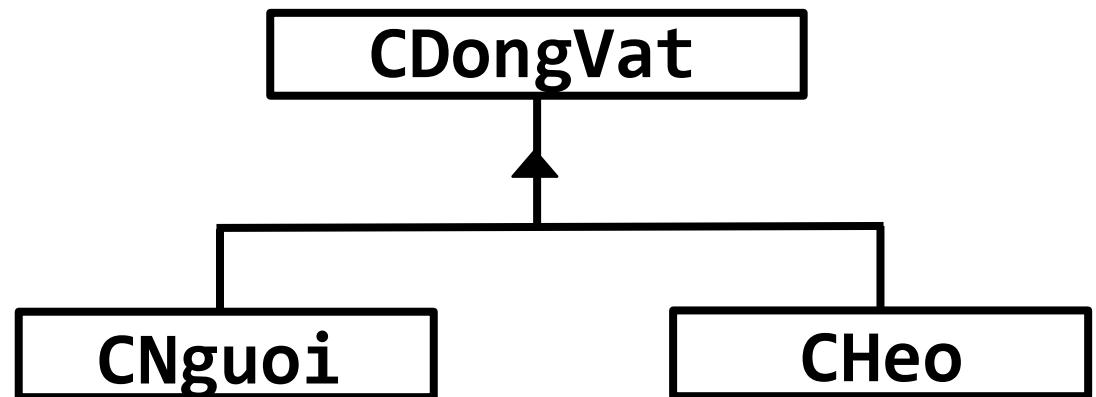
Ví dụ kế thừa

— Ví dụ 02: Khai báo lớp động vật, lớp heo và lớp người.

```

11. class CDongVat
12. {
13. |    ...
14. };
15. class CHeo:private CDongVat
16. {
17. |    ...
18. };
19. class CNgnoi:public CDongVat
20. {
21. |    ...
22. };

```

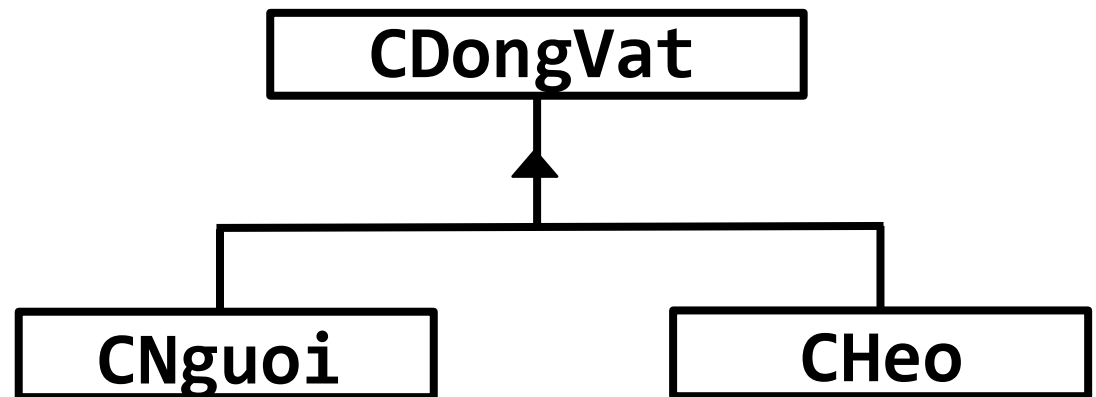


Dòng số 15 ta nói lớp đối tượng CHeo kế thừa từ lớp đối tượng CDongVat bằng từ khóa dẫn xuất private

Ví dụ kế thừa

— Ví dụ 02: Khai báo lớp động vật, lớp heo và lớp người.

```
11. class CDongVat
12. {
13. |    ...
14. };
15. class CHeo:private CDongVat
16. {
17. |    ...
18. };
19. class CNgnoi:public CDongVat
20. {
21. |    ...
22. };
```



Dòng số 19 ta nói lớp đối tượng **CNgnoi** kế thừa từ lớp **CDongVat** bằng từ khóa dẫn xuất **public**

Access Control and Inheritance

7. BẢNG QUI TẮC KẾ THỪA

Bảng quy tắc kế thừa

Từ khóa dẫn xuất Phạm vi lớp cơ sở	Private	Public
Private		
Protected	private	protected
Public	private	public

Bảng quy tắc kế thừa

Từ khóa dẫn xuất \ Phạm vi lớp cơ sở	Private	Public
Private		
Protected	private	protected
Public	private	public

- Các thuộc tính và phương thức được khai báo trong phạm vi **private** của lớp cơ sở thì sẽ không được hiểu ở lớp dẫn xuất.

Bảng quy tắc kế thừa

Từ khóa dẫn xuất \ Phạm vi lớp cơ sở	Private	Public
Private		
Protected	private	protected
Public	private	public

- Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của lớp cơ sở nếu được dẫn xuất bằng từ khóa **private** thì các thuộc tính và phương thức đó sẽ được hiểu ở lớp dẫn xuất như là thành phần **private** của lớp dẫn xuất.

Bảng quy tắc kế thừa

Từ khóa dẫn xuất \ Phạm vi lớp cơ sở	Private	Public
Private		
Protected	private	protected
Public	private	public

- Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của lớp cơ sở nếu được dẫn xuất bằng từ khóa **public** thì các thuộc tính và phương thức đó sẽ được hiểu ở lớp dẫn xuất như là thành phần **protected** của lớp dẫn xuất.

Bảng quy tắc kế thừa

Từ khóa dẫn xuất \ Phạm vi lớp cơ sở	Private	Public
Private		
Protected	private	protected
Public	private	public

- Các thuộc tính và phương thức được khai báo trong phạm vi **public** của lớp cơ sở nếu được dẫn xuất bằng từ khóa **private** thì các thuộc tính và phương thức đó sẽ được hiểu ở lớp dẫn xuất như là thành phần **private** của lớp dẫn xuất.

Bảng quy tắc kế thừa

Từ khóa dẫn xuất \ Phạm vi lớp cơ sở	Private	Public
Private		
Protected	private	protected
Public	private	public

- Các thuộc tính và phương thức được khai báo trong phạm vi **public** của lớp cơ sở nếu được dẫn xuất bằng từ khóa **public** thì các thuộc tính và phương thức đó sẽ được hiểu ở lớp dẫn xuất như là thành phần **public** của lớp dẫn xuất.

Bảng quy tắc kế thừa

<div>Từ khóa dẫn xuất</div> <div>Phạm vi lớp cơ sở</div>	Private	Public
Private	(1)	(2)
Protected	(3)	(4)
Public	(5)	(6)

Bảng quy tắc kế thừa

<div>Từ khóa dẫn xuất</div> <div>Phạm vi lớp cơ sở</div>	(10)	(11)
	(7)	(2)
(8)	(3)	(4)
(9)	(5)	(6)

Bảng quy tắc kế thừa

12 \ 13	(10)	(11)
	(1)	(2)
	(3)	(4)
	(5)	(6)

Bảng quy tắc kế thừa

8. TOÁN TỬ GÁN – KẾ THỪA

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 11 được đọc
như sau: Khai báo
lớp đối tượng A.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 15 được đọc
như sau: Lớp đối
tượng B kế thừa từ
lớp đối tượng A
bằng từ khóa dẫn
xuất public.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 21 đọc là: a nhỏ
là đối tượng thuộc lớp
A lớn.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 22 đọc là: b nhỏ
là đối tượng thuộc lớp
B lớn.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 23 đọc là: đối tượng b nhỏ được gán cho đối tượng a nhỏ.

Dòng 23 đọc là: đối tượng a nhỏ được gán bằng đối tượng b nhỏ.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 24 đọc là: đối
tượng a nhỏ được gán
cho đối tượng b nhỏ.

Dòng 24 đọc là: đối
tượng b nhỏ được gán
bằng đối tượng a nhỏ.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 11 được đọc
như sau: Khai báo
lớp đối tượng A
lớn.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 15 được đọc
như sau: Lớp đối
tượng B kế thừa từ
lớp đối tượng A lớn
bằng từ khóa dẫn
xuất public.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 21 đọc là: a nhỏ
là con trỏ đối tượng
thuộc lớp A lớn. Miền
giá trị của con trỏ đối
tượng a nhỏ là địa chỉ
ô nhớ.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 22 đọc là: **b nhỏ**
là con trỏ đối tượng
thuộc lớp B lớn. **Miền**
giá trị của con trỏ đối
tượng b nhỏ là địa chỉ
ô nhớ.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 23 đọc là: **x là một đối tượng thuộc lớp đối tượng A lớn.**

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 24 đọc là: **y** là
một đối tượng thuộc
lớp đối tượng B lớn.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 25 đọc là (cách 01): con trỏ đối tượng a nhỏ giữ địa chỉ của đối tượng x.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 25 đọc là (cách
02): địa chỉ của đối
tượng x được gán cho
con trỏ đối tượng a
nhỏ.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 26 đọc là: địa chỉ
của đối tượng y được
gán cho con trỏ đối
tượng b nhỏ.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 27 đọc là: địa chỉ
của đối tượng y được
gán cho con trỏ đối
tượng a nhỏ.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Dòng 28 đọc là: địa chỉ
của đối tượng x được
gán cho con trỏ đối
tượng b nhỏ.

Toán tử gán trong kế thừa

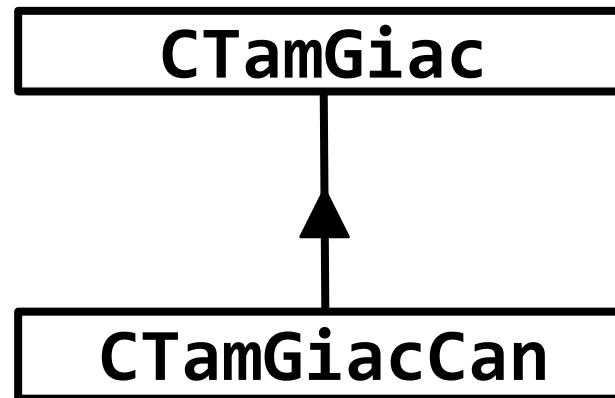
Ví dụ dẫn nhập 02:
Hãy cho biết đoạn
chương trình dưới
đây câu lệnh nào
đúng, câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Toán tử gán trong kế thừa

- Toán tử gán trong kế thừa được thực hiện theo nguyên tắc: trường hợp đặc biệt có thể được gán cho trường hợp tổng quát, và trường hợp tổng quát thì không thể gán cho trường hợp đặc biệt được.



Toán tử gán trong kế thừa

- Quy tắc trên áp dụng cho tất cả các ngôn ngữ hỗ trợ lập trình hướng đối tượng như:
 - + C++,
 - + Java,
 - + VB.NET,
 - + C#,
 - + Python,...

Toán tử gán trong kế thừa

- Áp dụng quy tắc trên cho ngôn ngữ lập trình hướng đối tượng C++ ta có thể nói như sau: một đối tượng thuộc lớp dẫn xuất (trường hợp đặc biệt) có thể được gán cho một đối tượng thuộc lớp cơ sở (trường hợp tổng quát). Điều ngược lại là sai, nghĩa là một đối tượng thuộc lớp cơ sở (trường hợp tổng quát) không được quyền gán cho một đối tượng thuộc lớp dẫn xuất (trường hợp đặc biệt).

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```


Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 11 được đọc
như sau: Khai báo
lớp đối tượng A
lớn.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 15 được đọc
như sau: Lớp đối
tượng B kế thừa từ
lớp đối tượng A
bằng từ khóa dẫn
xuất public.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 21 đọc là: a nhỏ
là đối tượng thuộc lớp
A lớn.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Dòng 22 đọc là: b nhỏ
là đối tượng thuộc lớp
B lớn.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Một đối tượng
thuộc lớp dẫn xuất
có thể được gán
cho một đối tượng
thuộc lớp cơ sở.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Một đối tượng
thuộc lớp dẫn xuất
có thể được gán
cho một đối tượng
thuộc lớp cơ sở.

ĐÚNG

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Một đối tượng
thuộc lớp cơ sở
không được quyền
gán cho một đối
tượng thuộc lớp
dẫn xuất.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 01:
Hãy cho biết trong
chương trình dưới
đây câu lệnh nào
đúng câu lệnh nào
sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A a;
22. |    B b;
23. |    a = b;
24. |    b = a;
25. }
```

Một đối tượng
thuộc lớp cơ sở
không được quyền
gán cho một đối
tượng thuộc lớp
dẫn xuất.

Toán tử gán trong kế thừa

- Mở rộng quy tắc trên cho con trỏ đối tượng ta có thể nói như sau: một con trỏ đối tượng thuộc lớp cơ sở có thể giữ địa chỉ của một đối tượng thuộc lớp dẫn xuất. Ngược lại, một con trỏ đối tượng thuộc lớp dẫn xuất không thể giữ địa chỉ của một đối tượng thuộc lớp cơ sở.
- Hiển nhiên, con trỏ đối tượng của một lớp được quyền giữ địa chỉ của đối tượng cùng thuộc về lớp đó.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12.{
13.|    ...
14.};
15. class B:public A
16.{
17.|    ...
18.};
19. void main()
20.{
21.|    A* a;
22.|    B* b;
23.|    A x;
24.|    B y;
25.|    a = &x;
26.|    b = &y;
27.|    a = &y;
28.|    b = &x;
29.}
```

Dòng 11 được đọc như sau: Khai báo lớp đối tượng A lớn.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12.{
13.|    ...
14.};
15. class B:public A
16.{
17.|    ...
18.};
19. void main()
20.{
21.|    A* a;
22.|    B* b;
23.|    A x;
24.|    B y;
25.|    a = &x;
26.|    b = &y;
27.|    a = &y;
28.|    b = &x;
29.}
```

Dòng 15 được đọc như sau: Lớp đối tượng B kế thừa từ lớp đối tượng A bằng từ khóa dẫn xuất public.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12.{
13.|    ...
14.};
15. class B:public A
16.{
17.|    ...
18.};
19. void main()
20.{
21.|    A* a;
22.|    B* b;
23.|    A x;
24.|    B y;
25.|    a = &x;
26.|    b = &y;
27.|    a = &y;
28.|    b = &x;
29.}
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a; ĐÚNG
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b; ĐÚNG
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x; ĐÚNG
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```


Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y; ĐÚNG
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Con trỏ đối tượng của một lớp được quyền giữ địa chỉ của đối tượng cùng thuộc về lớp đó.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x; ĐÚNG
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Con trỏ đối tượng của một lớp được quyền giữ địa chỉ của đối tượng cùng thuộc về lớp đó.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Con trỏ đối tượng của một lớp được quyền giữ địa chỉ của đối tượng cùng thuộc về lớp đó.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y; ĐÚNG
27. |    a = &y;
28. |    b = &x;
29. }
```

Con trỏ đối tượng của một lớp được quyền giữ địa chỉ của đối tượng cùng thuộc về lớp đó.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12.{
13.|    ...
14.};
15. class B:public A
16.{
17.|    ...
18.};
19. void main()
20.{
21.|    A* a;
22.|    B* b;
23.|    A x;
24.|    B y;
25.|    a = &x;
26.|    b = &y;
27.|    a = &y;
28.|    b = &x;
29.}
```

Một con trỏ đối tượng thuộc lớp cơ sở có thể giữ địa chỉ của một đối tượng thuộc lớp dẫn xuất.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y; ĐÚNG
28. |    b = &x;
29. }
```

Một con trỏ đối tượng thuộc lớp cơ sở có thể giữ địa chỉ của một đối tượng thuộc lớp dẫn xuất.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x;
29. }
```

Một con trỏ đối tượng thuộc lớp dẫn xuất không thể giữ địa chỉ của một đối tượng thuộc lớp cơ sở.

Toán tử gán trong kế thừa

Ví dụ dẫn nhập 02: Hãy cho biết đoạn chương trình dưới đây câu lệnh nào đúng, câu lệnh nào sai.

— Chương trình

```
11. class A
12. {
13. |    ...
14. };
15. class B:public A
16. {
17. |    ...
18. };
19. void main()
20. {
21. |    A* a;
22. |    B* b;
23. |    A x;
24. |    B y;
25. |    a = &x;
26. |    b = &y;
27. |    a = &y;
28. |    b = &x; SAI
29. }
```

Một con trỏ đối tượng thuộc lớp dẫn xuất không thể giữ địa chỉ của một đối tượng thuộc lớp cơ sở.

CÂU HỎI ÔN TẬP

Câu hỏi ôn tập

— Trong bài học ta học bao nhiêu loại quan hệ?

Câu hỏi ôn tập

- Trong bài học ta học bao nhiêu loại quan hệ?
- Trong bài học ta học 4 loại quan hệ.

Câu hỏi ôn tập

— Kể tên các loại quan hệ đã học trong bài học?

Câu hỏi ôn tập

- Kể tên các loại quan hệ đã học trong bài học?
- Quan hệ một một (1–1).
- Quan hệ một nhiều (1–n).
- Quan hệ nhiều nhiều (m–n).
- Quan hệ đặc biệt hóa, tổng quát hóa.

Câu hỏi ôn tập

— Quan hệ một – một giữa hai lớp đối tượng là gì?

Câu hỏi ôn tập

- Quan hệ một – một giữa hai lớp đối tượng là gì?
- Hai lớp đối tượng được gọi là quan hệ một-một với nhau khi một đối tượng thuộc lớp này quan hệ với một đối tượng thuộc lớp kia và một đối tượng thuộc lớp kia quan hệ duy nhất với một đối tượng thuộc lớp này.

Câu hỏi ôn tập

— Quan hệ một – nhiều giữa hai lớp đối tượng là gì?

Câu hỏi ôn tập

- Quan hệ một – nhiều giữa hai lớp đối tượng là gì?
- Hai lớp đối tượng được gọi là quan hệ một nhiều với nhau khi một đối tượng thuộc lớp này quan hệ với nhiều đối tượng thuộc lớp kia và một đối tượng lớp kia quan hệ duy nhất với một đối tượng thuộc lớp này.

Câu hỏi ôn tập

— Quan hệ nhiều – nhiều giữa hai lớp đối tượng là gì?

Câu hỏi ôn tập

- Quan hệ nhiều – nhiều giữa hai lớp đối tượng là gì?
- Hai lớp đối tượng được gọi là quan hệ nhiều-nhiều với nhau khi một đối tượng thuộc lớp này quan hệ với nhiều đối tượng thuộc lớp kia và một đối tượng lớp kia cũng có quan hệ với nhiều đối tượng thuộc lớp này.

Câu hỏi ôn tập

- Quan hệ đặc biệt hóa – tổng quát hóa giữa hai lớp đối tượng là gì?

Câu hỏi ôn tập

- Quan hệ đặc biệt hóa – tổng quát hóa giữa hai lớp đối tượng là gì?
- Hai lớp đối tượng được gọi là quan hệ đặc biệt hóa – tổng quát hóa với nhau khi, lớp đối tượng này là trường hợp đặc biệt của lớp đối tượng kia và lớp đối tượng kia là trường hợp tổng quát của lớp đối tượng này.

Câu hỏi ôn tập

— Cây kế thừa là gì?

Câu hỏi ôn tập

- Cây kế thừa là gì?
- Cây kế thừa là một cây đa nhánh thể hiện mối quan hệ đặc biệt hóa-tổng quát hóa giữa các lớp trong hệ thống, trong chương trình.

Câu hỏi ôn tập

— Sơ đồ lớp là gì?

Câu hỏi ôn tập

- Sơ đồ lớp là gì?
- Sơ đồ lớp là sơ đồ thể hiện tất cả các mối quan hệ giữa các lớp trong hệ thống, trong chương trình.

Câu hỏi ôn tập

- Một thuộc tính hay một phương thức khi được khai báo trong một lớp ta có thể khai báo trong những phạm vi nào?

Câu hỏi ôn tập

- Một thuộc tính hay một phương thức khi được khai báo trong một lớp ta có thể khai báo trong những phạm vi nào?
- Một thuộc tính hay một phương thức khi được khai báo trong một lớp ta có thể khai báo trong 3 phạm vi khác nhau: `private`, `public` hoặc `protected`.

Câu hỏi ôn tập

- Một thuộc tính hay một phương thức khi được khai báo trong phạm vi **private** qui tắc truy xuất như thế nào?

Câu hỏi ôn tập

- Một thuộc tính hay một phương thức khi được khai báo trong phạm vi **private** qui tắc truy xuất như thế nào?
- Các thuộc tính và phương thức được khai báo trong phạm vi **private** của một lớp thì chỉ được phép truy xuất bên trong lớp (accessible only inside the class) và không được quyền truy xuất bên ngoài lớp.

Câu hỏi ôn tập

- Một thuộc tính hay một phương thức khi được khai báo trong phạm vi **protected** qui tắc truy xuất như thế nào?

Câu hỏi ôn tập

- Một thuộc tính hay một phương thức khi được khai báo trong phạm vi **protected** qui tắc truy xuất như thế nào?
- Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của một lớp thì chỉ được phép truy xuất bên trong lớp (accessible only inside the class), lớp dẫn xuất và không được quyền truy xuất bên ngoài lớp.

Câu hỏi ôn tập

- Một thuộc tính hay một phương thức khi được khai báo trong phạm vi **public** qui tắc truy xuất như thế nào?

Câu hỏi ôn tập

- Một thuộc tính hay một phương thức khi được khai báo trong phạm vi **public** qui tắc truy xuất như thế nào?
- Các thuộc tính và phương thức được khai báo trong phạm vi **public** của một lớp thì không chỉ được phép truy xuất bên trong lớp, lớp dẫn xuất và cả bên ngoài lớp.

Câu hỏi ôn tập

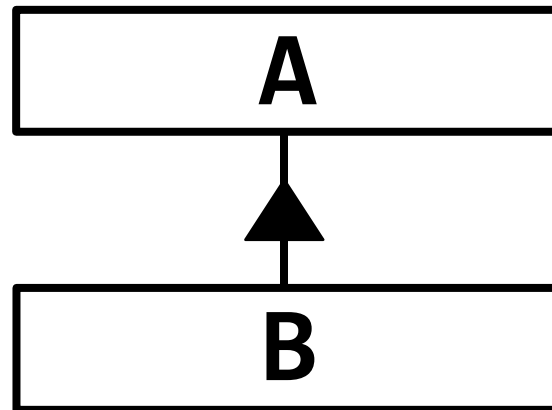
— Kế thừa tiếng anh là gì?

Câu hỏi ôn tập

- Kế thừa tiếng anh là gì?
- Kế thừa tiếng anh là Inheritance.

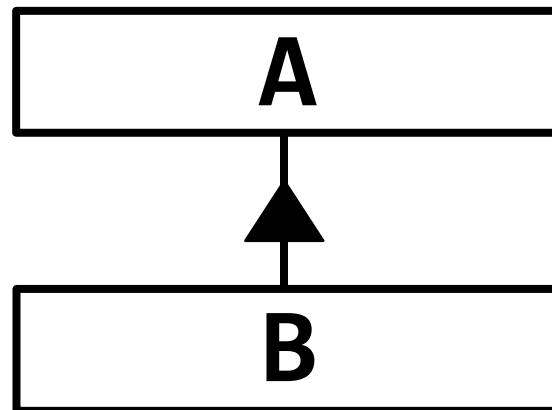
Câu hỏi ôn tập

— Trong hình vẽ dưới trường hợp tổng quát là gì?



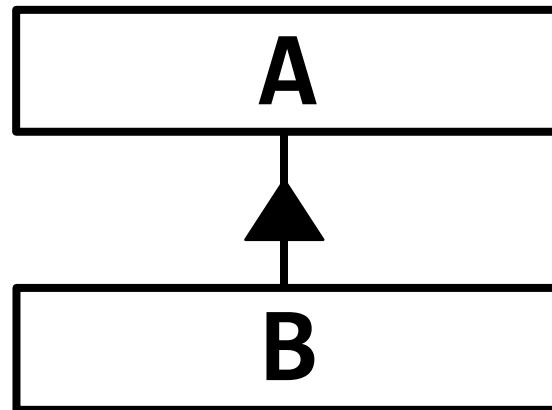
Câu hỏi ôn tập

- Trong hình vẽ dưới trường hợp tổng quát là gì?
- Trong hình vẽ dưới trường hợp tổng quát là A.



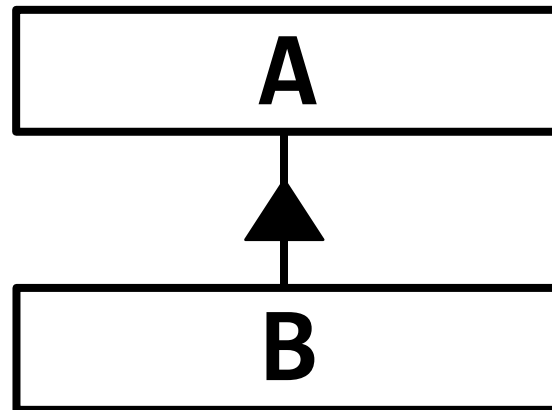
Câu hỏi ôn tập

— Trong hình vẽ dưới trường hợp đặc biệt là gì?



Câu hỏi ôn tập

- Trong hình vẽ dưới trường hợp đặc biệt là gì?
- Trong hình vẽ dưới trường hợp đặc biệt là B.



Câu hỏi ôn tập

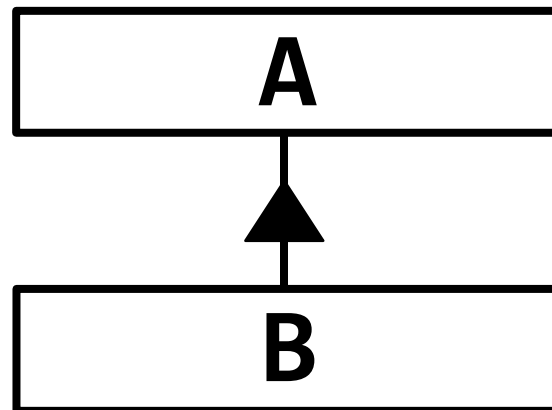
— Trong ngôn ngữ lập trình C++ có mấy từ khóa dẫn xuất?

Câu hỏi ôn tập

- Trong ngôn ngữ lập trình C++ có mấy từ khóa dẫn xuất?
- Trong ngôn ngữ lập trình C++ có 3 từ khóa dẫn xuất đó là `private`, `protected`, `public`.

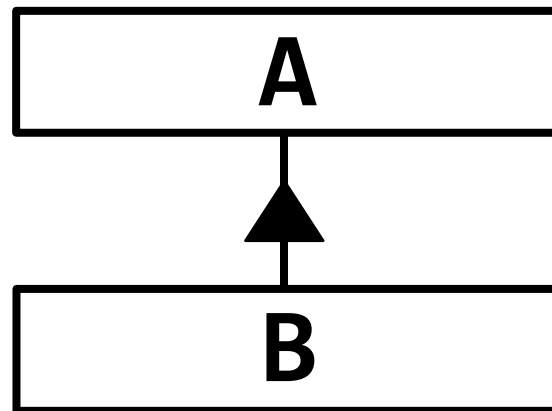
Câu hỏi ôn tập

— Trong hình vẽ dưới lớp đối tượng cơ sở là lớp đối tượng nào?



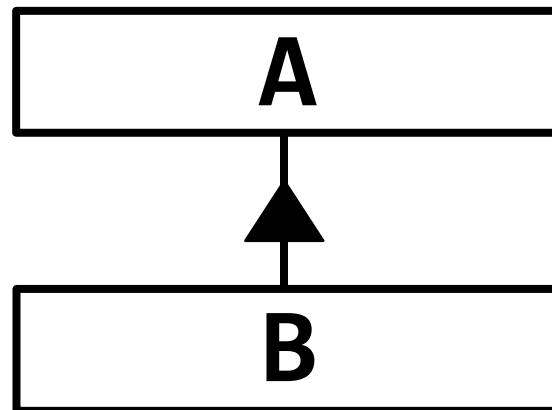
Câu hỏi ôn tập

- Trong hình vẽ dưới lớp đối tượng cơ sở là lớp đối tượng nào?
- Trong hình vẽ dưới lớp đối tượng cơ sở là lớp đối tượng A.



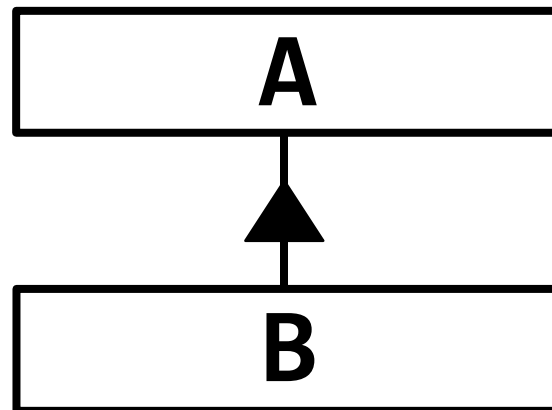
Câu hỏi ôn tập

— Trong hình vẽ dưới lớp đối tượng dẫn xuất là lớp đối tượng nào?



Câu hỏi ôn tập

- Trong hình vẽ dưới lớp đối tượng dẫn xuất là lớp đối tượng nào?
- Trong hình vẽ dưới lớp đối tượng dẫn xuất là lớp đối tượng B.



Câu hỏi ôn tập

— Dòng lệnh số 15 được đọc như thế nào?

```
11. class CDongVat
12. {
13. |    ...
14. };
15. class CHeo:private CDongVat
16. {
17. |    ...
18. };
```

Câu hỏi ôn tập

— Dòng lệnh số 15 được đọc như thế nào?

```
11. class CDongVat
12. {
13. |    ...
14. };
15. class CHeo:private CDongVat
16. {
17. |    ...
18. };
```

— Dòng số 15 được đọc như sau: lớp đối tượng **CHeo** kế thừa từ lớp **CDongVat** bằng từ khóa dẫn xuất **private**.

Câu hỏi ôn tập

— Lớp đối tượng nào là lớp đối tượng cơ sở?

```
11. class CDongVat
12. {
13. |    ...
14. };
15. class Cheo:private CDongVat
16. {
17. |    ...
18. };
```


Câu hỏi ôn tập

— Lớp đối tượng nào là lớp đối tượng cơ sở?

```
11. class CDongVat
12. {
13. |    ...
14. };
15. class Cheo: private CDongVat
16. {
17. |    ...
18. };
```

— Lớp đối tượng cơ sở là lớp đối tượng **CDongVat**.

Câu hỏi ôn tập

— Lớp đối tượng nào là lớp đối tượng dẫn xuất?

```
11. class CDongVat
12. {
13. |    ...
14. };
15. class Cheo:private CDongVat
16. {
17. |    ...
18. };
```

Câu hỏi ôn tập

— Lớp đối tượng nào là lớp đối tượng dẫn xuất?

```
11. class CDongVat
12. {
13. | ...
14. };
15. class Cheo: private CDongVat
16. {
17. | ...
18. };
```

— Lớp đối tượng dẫn xuất là lớp đối tượng **CHeo**.

Câu hỏi ôn tập

— Lớp cơ sở tiếng anh là gì?

Câu hỏi ôn tập

- Lớp cơ sở tiếng anh là gì?
- Lớp cơ sở tiếng anh là .

Câu hỏi ôn tập

— Lớp dẫn xuất tiếng anh là gì?

Câu hỏi ôn tập

- Lớp dẫn xuất tiếng anh là gì?
- Lớp dẫn xuất tiếng anh là .

Câu hỏi ôn tập

- Các thuộc tính và phương thức được khai báo trong phạm vi **private** của lớp cơ sở thì có được hiểu ở lớp dẫn xuất không?

Câu hỏi ôn tập

- Các thuộc tính và phương thức được khai báo trong phạm vi **private** của lớp cơ sở thì có được hiểu ở lớp dẫn xuất không?
- Các thuộc tính và phương thức được khai báo trong phạm vi **private** của lớp cơ sở thì sẽ không được hiểu ở lớp dẫn xuất.

Câu hỏi ôn tập

- Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của lớp cơ sở thì có được hiểu ở lớp dẫn xuất không?

Câu hỏi ôn tập

- Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của lớp cơ sở thì có được hiểu ở lớp dẫn xuất không?
- Các thuộc tính và phương thức được khai báo trong phạm vi **protected** của lớp cơ sở thì sẽ được hiểu ở lớp dẫn xuất.

Câu hỏi ôn tập

- Các thuộc tính và phương thức được khai báo trong phạm vi **public** của lớp cơ sở thì có được hiểu ở lớp dẫn xuất không?

Câu hỏi ôn tập

- Các thuộc tính và phương thức được khai báo trong phạm vi **public** của lớp cơ sở thì có được hiểu ở lớp dẫn xuất không?
- Các thuộc tính và phương thức được khai báo trong phạm vi **public** của lớp cơ sở thì sẽ được hiểu ở lớp dẫn xuất.

Câu hỏi ôn tập

- Toán tử gán trong kế thừa của oop thực hiện theo nguyên tắc nào?

Câu hỏi ôn tập

- Toán tử gán trong kế thừa của oop thực hiện theo nguyên tắc nào?
- Toán tử gán trong kế thừa của oop được thực hiện theo nguyên tắc: trường hợp đặc biệt có thể được gán cho trường hợp tổng quát, và trường hợp tổng quát thì không thể gán cho trường hợp đặc biệt được.

Câu hỏi ôn tập

- Một đối tượng thuộc lớp dẫn xuất (trường hợp đặc biệt) có thể được gán cho một đối tượng thuộc lớp cơ sở (trường hợp tổng quát) không?

Câu hỏi ôn tập

- Một đối tượng thuộc lớp dẫn xuất (trường hợp đặc biệt) có thể được gán cho một đối tượng thuộc lớp cơ sở (trường hợp tổng quát) không?
- Một đối tượng thuộc lớp dẫn xuất (trường hợp đặc biệt) có thể được gán cho một đối tượng thuộc lớp cơ sở (trường hợp tổng quát).

Câu hỏi ôn tập

- Một đối tượng thuộc lớp cơ sở (trường hợp tổng quát) có thể được gán cho một đối tượng thuộc lớp dẫn xuất (trường hợp đặc biệt) không?

Câu hỏi ôn tập

- Một đối tượng thuộc lớp cơ sở (trường hợp tổng quát) có thể được gán cho một đối tượng thuộc lớp dẫn xuất (trường hợp đặc biệt) không?
- Một đối tượng thuộc lớp cơ sở (trường hợp tổng quát) không thể gán cho một đối tượng thuộc lớp dẫn xuất (trường hợp đặc biệt).

Câu hỏi ôn tập

- Con trỏ đối tượng của một lớp được quyền giữ địa chỉ của một đối tượng cùng thuộc về lớp đó không?

Câu hỏi ôn tập

- Con trỏ đối tượng của một lớp được quyền giữ địa chỉ của một đối tượng cùng thuộc về lớp đó không?
- Con trỏ đối tượng của một lớp được quyền giữ địa chỉ của một đối tượng cùng thuộc lớp.

Câu hỏi ôn tập

- Con trỏ đối tượng của lớp cơ sở được quyền giữ địa chỉ của một đối tượng cùng thuộc về lớp dẫn xuất hay không?

Câu hỏi ôn tập

- Con trỏ đối tượng của lớp cơ sở được quyền giữ địa chỉ của một đối tượng cùng thuộc về lớp dẫn xuất hay không?
- Con trỏ đối tượng của lớp cơ sở được quyền giữ địa chỉ của một đối tượng cùng thuộc về lớp dẫn xuất.

Câu hỏi ôn tập

- Con trỏ đối tượng của lớp dẫn xuất được quyền giữ địa chỉ của một đối tượng cùng thuộc về lớp cơ sở hay không?

Câu hỏi ôn tập

- Con trỏ đối tượng của lớp dẫn xuất được quyền giữ địa chỉ của một đối tượng cùng thuộc về lớp cơ sở hay không?
- Con trỏ đối tượng của lớp dẫn xuất không được quyền giữ địa chỉ của một đối tượng cùng thuộc về lớp cơ sở.

Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

TS. Nguyễn Tấn Trần Minh Khang

Chương 10

KẾ THỪA – INHERITANCE – PYTHON

- Nguyễn Hữu Lợi
- Đoàn Chánh Thống
- ThS. Nguyễn Thành Hiệp
- ThS. Trương Quốc Dũng
- ThS. Võ Duy Nguyên
- ThS. Nguyễn Văn Toàn
- TS. Nguyễn Duy Khánh
- TS. Nguyễn Tấn Trần Minh Khang

MỤC TIÊU

1. Mục tiêu

- Hiểu được các loại quan hệ?
- Hiểu được kế thừa trong lập trình hướng đối tượng (object-oriented programming) là gì?
- Hiểu được khái niệm cây kế thừa.
- Hiểu được khái niệm sơ đồ lớp.

QUAN HỆ

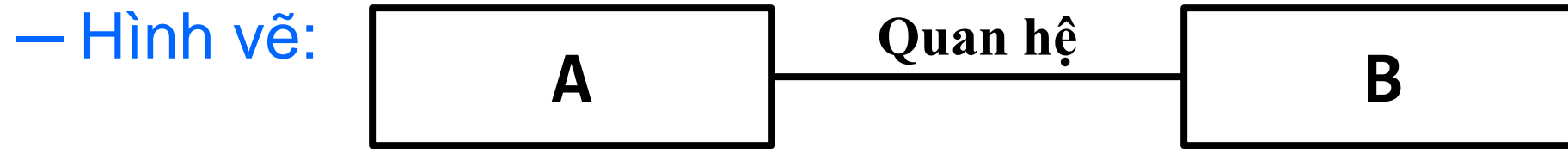
2. Quan hệ

Người ta chia các quan hệ thành những loại như sau:

- Quan hệ một một (1–1).
- Quan hệ một nhiều (1–n).
- Quan hệ nhiều nhiều (m–n).
- Quan hệ đặc biệt hóa, tổng quát hóa.

Quan hệ một một (1-1)

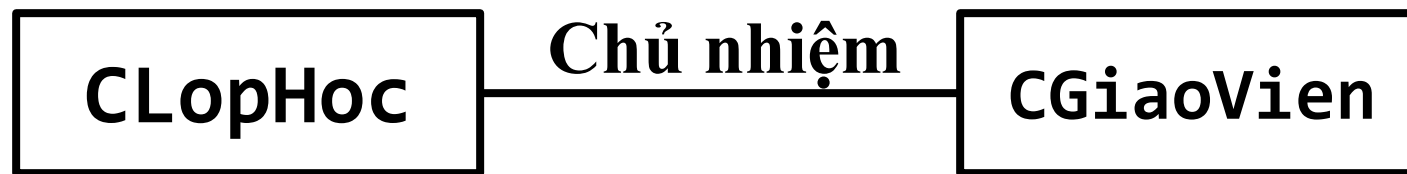
— Khái niệm: Hai lớp đối tượng được gọi là quan hệ một-một với nhau khi một đối tượng thuộc lớp này quan hệ với một đối tượng thuộc lớp kia và một đối tượng thuộc lớp kia quan hệ duy nhất với một đối tượng thuộc lớp này.



— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp A quan hệ với một đối tượng thuộc lớp B và một đối tượng lớp B quan hệ duy nhất với một đối tượng thuộc lớp A.

Quan hệ một một (1-1)

- Ví dụ 01: Xét ngữ cảnh trường PTTH Bạch Thái Bưởi thuộc quận Hồ Hoàn Kiếm thành phố Hà Nội trong niên khóa 2018 – 2019:



- Trong hình vẽ trên ta nói: một đối tượng thuộc lớp CLopHoc có quan hệ Chủ nhiệm với một đối tượng thuộc lớp CGiaoVien và ngược lại một đối tượng thuộc lớp CGiaoVien có quan hệ Chủ nhiệm với duy nhất một đối tượng thuộc lớp CLopHoc.

Quan hệ một một (1-1)

- Ví dụ 02: Xét ngữ cảnh nước Cộng Hòa Xã Hội Chủ Nghĩa Việt Nam, theo hiến pháp và pháp luật tại thời điểm 2019:



- Trong hình vẽ trên ta nói: một đối tượng thuộc lớp CVo có quan hệ Hôn nhân với một đối tượng thuộc lớp CChong và ngược lại một đối tượng thuộc lớp CChong có quan hệ Hôn nhân với duy nhất một đối tượng thuộc lớp CVo.

Quan hệ một nhiều (1-n)

— Khái niệm: Hai lớp đối tượng được gọi là quan hệ một nhiều với nhau khi một đối tượng thuộc lớp này quan hệ với nhiều đối tượng thuộc lớp kia và một đối tượng lớp kia quan hệ duy nhất với một đối tượng thuộc lớp này.

— Hình vẽ:



— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp A quan hệ với nhiều đối tượng thuộc lớp B và một đối tượng lớp B có quan hệ duy nhất với một đối tượng thuộc lớp A.

Quan hệ một nhiều (1-n)

- Ví dụ 01: Xét ngữ cảnh trường PTTH Bạch Thái Bưởi trong niên khóa 2018 – 2019.



- Trong hình vẽ trên ta nói: một đối tượng thuộc lớp **CLopHoc** có quan hệ theo học với nhiều đối tượng thuộc lớp **CHocSinh** và ngược lại một đối tượng thuộc lớp **CHocSinh** có quan hệ theo học với duy nhất một đối tượng thuộc lớp **CLopHoc**.


Quan hệ một nhiều (1-n)

— Ví dụ 02: Xét ngữ cảnh quả đất.



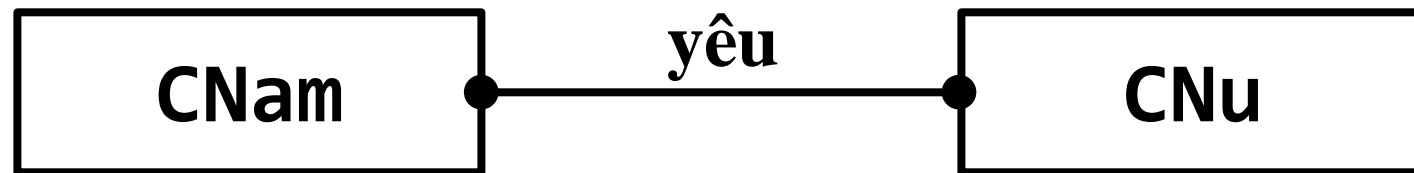
— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp CCha có quan hệ huyết thống với nhiều đối tượng thuộc lớp CCon và ngược lại một đối tượng thuộc lớp CCon có quan hệ huyết thống với duy nhất một đối tượng thuộc lớp CCha.

Quan hệ nhiều nhiều (m-n)

- Khái niệm: hai lớp đối tượng được gọi là quan hệ nhiều-nhiều với nhau khi một đối tượng thuộc lớp này quan hệ với nhiều đối tượng thuộc lớp kia và một đối tượng lớp kia cũng có quan hệ với nhiều đối tượng thuộc lớp này.
- Hình vẽ:
- Trong hình vẽ trên ta nói: một đối tượng thuộc lớp A quan hệ với nhiều đối tượng thuộc lớp B và một đối tượng lớp B cũng có quan hệ với nhiều đối tượng thuộc lớp A.

Quan hệ nhiều nhiều (m-n)

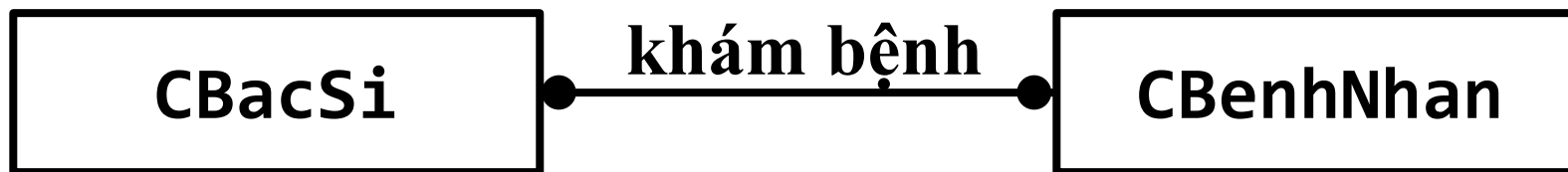
— Ví dụ 01: Xét ngữ cảnh quả đất từ hồi nằm đến bi chừ.



— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp CNaM quan hệ yêu với nhiều đối tượng thuộc lớp CNu và một đối tượng lớp CNu cũng có quan hệ yêu với nhiều đối tượng thuộc lớp CNaM.

Quan hệ nhiều nhiều (m-n)

— Ví dụ 02: Xét ngữ cảnh quả đất từ hồi ấy đến hồi này.

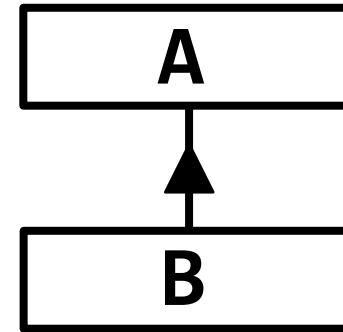


— Trong hình vẽ trên ta nói: một đối tượng thuộc lớp **CBacSi** quan hệ **khám bệnh** với nhiều đối tượng thuộc lớp **CBenhNhan** và một đối tượng lớp **CBenhNhan** cũng có quan hệ **khám bệnh** với nhiều đối tượng thuộc lớp **CBacSi**.

Quan hệ đặc biệt hóa – tổng quát hóa

- **Khái niệm:** Hai lớp đối tượng được gọi là quan hệ đặc biệt hóa – tổng quát hóa với nhau khi, lớp đối tượng này là trường hợp đặc biệt của lớp đối tượng kia và lớp đối tượng kia là trường hợp tổng quát của lớp đối tượng này.

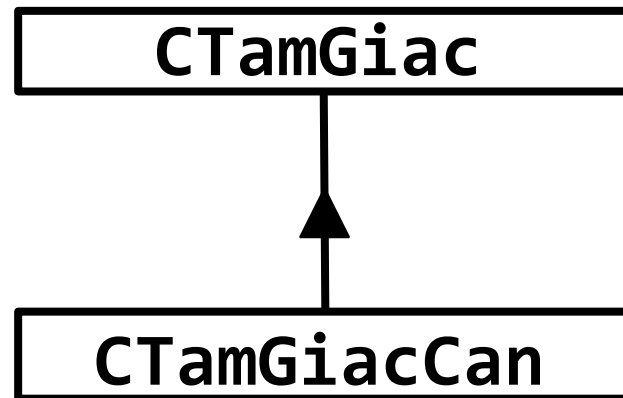
- Hình vẽ



- Trong hình vẽ trên ta nói: lớp đối tượng **B** là trường hợp đặc biệt của lớp đối tượng **A** và lớp đối tượng **A** là trường hợp tổng quát của lớp đối tượng **B**.

Quan hệ đặc biệt hóa – tổng quát hóa

— Ví dụ 01:



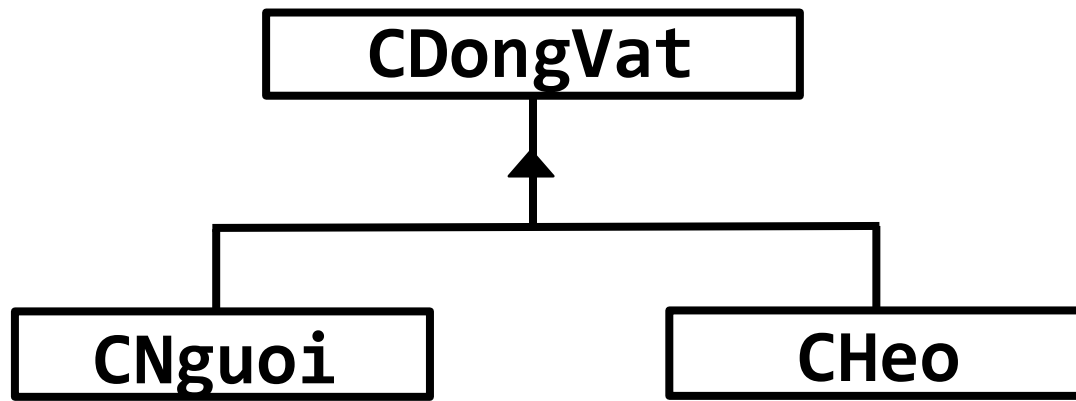
— Trong hình vẽ trên ta nói:

+ lớp đối tượng **CTamGiacCan** là trường hợp đặc biệt của lớp đối tượng **CTamGiac** và

+ lớp đối tượng **CTamGiac** là trường hợp tổng quát của lớp đối tượng **CTamGiacCan**.

Quan hệ đặc biệt hóa – tổng quát hóa

— Ví dụ 02:



— Trong hình vẽ trên ta nói:

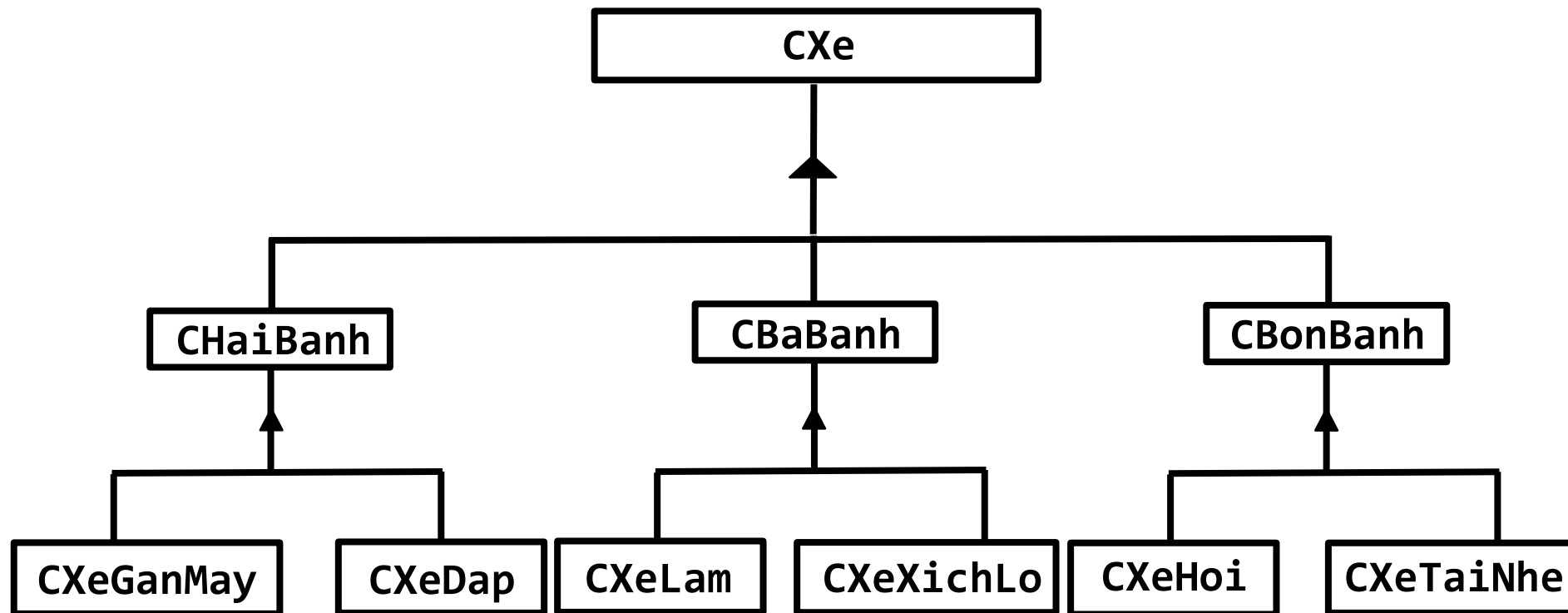
- + lớp đối tượng **CNgươi** và lớp đối tượng **CHeo** là trường hợp đặc biệt của lớp đối tượng **CDongVat** và
- + lớp đối tượng **CDongVat** là trường hợp tổng quát của lớp đối tượng **CNgươi** và lớp đối tượng **CHeo**.

CÂY KẾ THỪA

3. Cây kế thừa

- **Khái niệm:** Cây kế thừa là một cây đa nhánh thể hiện mối quan hệ đặc biệt hóa-tổng quát hóa giữa các lớp trong hệ thống, trong chương trình.
- **Ví dụ:** Hãy vẽ cây kế thừa cho các lớp đối tượng sau:
 - + Lớp CXdap
 - + Lớp CXeGanMay
 - + Lớp CXeHoi
 - + Lớp CXeHaiBanh
 - + Lớp CXeTaiNhe
 - + Lớp CXeLam
 - + Lớp CXe
 - + Lớp CXeBaBanh
 - + Lớp CXeBonBanh
 - + Lớp CXeXichLo

3. Cây kế thừa

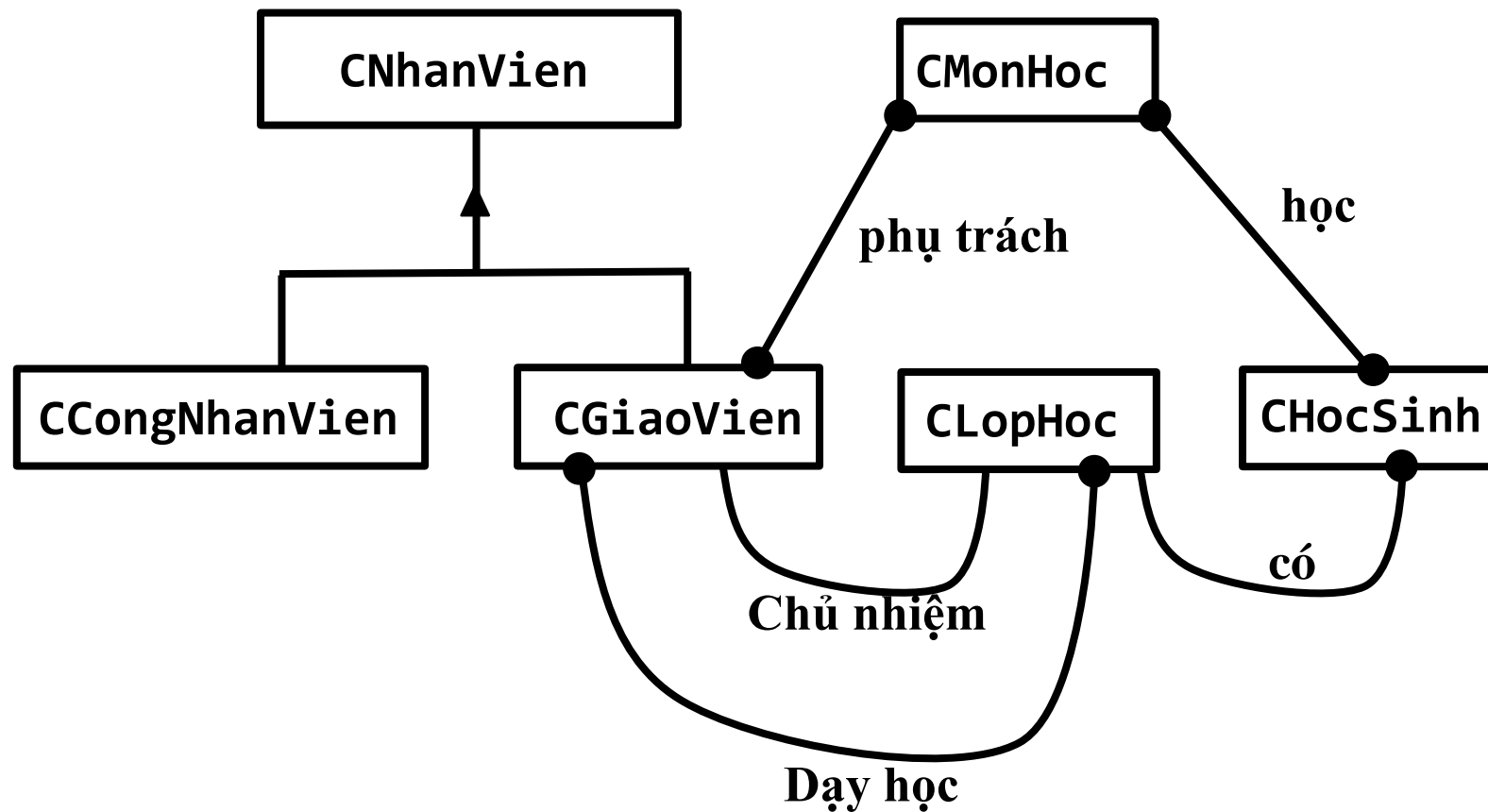


SƠ ĐỒ LỚP

Sơ đồ lớp

- **Khái niệm:** Sơ đồ lớp là sơ đồ thể hiện tất cả các mối quan hệ giữa các lớp trong hệ thống, trong chương trình.
- Xét ngữ cảnh trường PTTH Bạch Đằng trong niên học 2019-2020. Hãy vẽ sơ đồ lớp cho các lớp đối tượng sau:
 - + Lớp CGiaoVien
 - + Lớp CHocSinh
 - + Lớp CLopHoc
 - + Lớp CMonHoc
 - + Lớp CNhanVien
 - + Lớp CCongNhanVien
 - + Lớp CNhanVien: tất cả những nhân viên làm việc trong trường.
 - + Lớp CCongNhanVien: là các nhân viên làm việc trong nhà trường nhưng ko trực tiếp đứng lớp. Ví dụ: Bảo vệ, lao công, bảo mẫu.

Sơ đồ lớp



Access control

ĐIỀU KHIỂN TRUY XUẤT

Điều khiển truy xuất

— Một thuộc tính hay một phương thức khi được khai báo trong một lớp ta có thể khai báo trong 3 phạm vi khác nhau: **private**, **public** hoặc **protected**.

private
public
protected

— Ví dụ:

```
1.class A:
2.     _a = 2
3.     def _f(self):
4.         self._a = 15
5.     _b = 3
6.     def _g(self):
7.         self._b = 23
8.     ...
```

Điều khiển truy xuất

— Một thuộc tính hay một phương thức khi được khai báo trong một lớp ta có thể khai báo trong 3 phạm vi khác nhau: **private**, **public** hoặc **protected**.

private
public
protected

— Ví dụ:

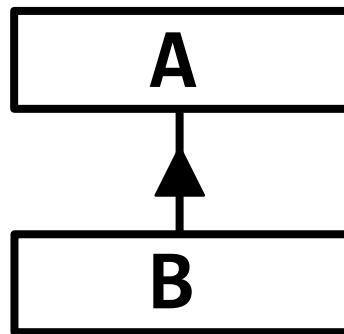
```
1.class A:  
2.    ...  
3.    c = 4  
4.    def h(self):  
5.        self.c = 23
```

Inheritance

KẾ THỪA – INHERITANCE

Thế giới thực

— Hình vẽ



- Trong hình vẽ trên ta nói **A** và **B** có quan hệ đặc biệt hoá, tổng quát hoá với nhau.
- Trong đó **B** là trường hợp đặt biệt của **A** và **A** là trường hợp tổng quát của **B**.

Lập trình hướng đối tượng

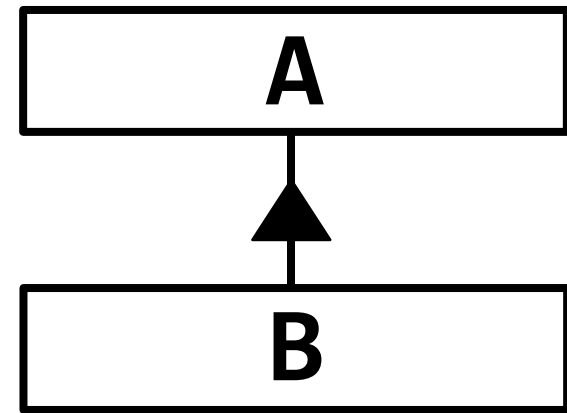
— Xét khai báo.

1. `class A:`

2. `...`

3. `class B(A):`

4. `...`



- Khai báo trên ta nói lớp đối tượng **B** kế thừa từ lớp đối tượng **A**.
- Lớp đối tượng **A** được gọi là lớp cơ sở.
- Lớp đối tượng **B** được gọi là lớp dẫn xuất từ lớp đối tượng **A**.

Ví dụ kế thừa

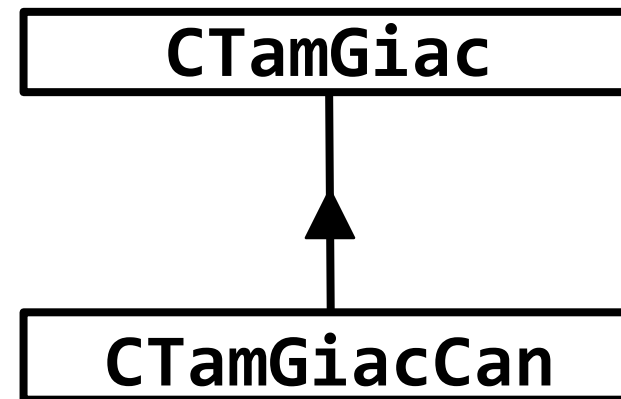
— Ví dụ 01: Khai báo lớp tam giác (CTamGiac) và lớp tam giác cân (CTamGiacCan).

1. `class CTamGiac:`

2. `...`

3. `class CTamGiacCan(CTamGiac):`

4. `...`



Ví dụ kế thừa

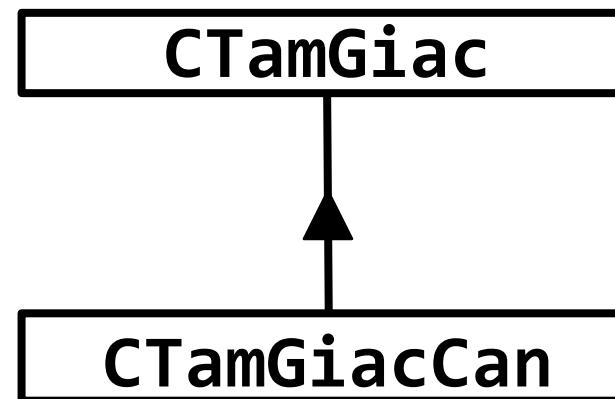
— Ví dụ 01: Khai báo lớp tam giác (CTamGiac) và lớp tam giác cân (CTamGiacCan).

1. `class CTamGiac:`

2. `...`

3. `class CTamGiacCan(CTamGiac):`

4. `...`



Dòng số 3 ta nói lớp đối tượng **CTamGiacCan** kế thừa từ lớp đối tượng **CTamGiac**

Ví dụ kế thừa

— Ví dụ 02: Khai báo lớp động vật, lớp heo và lớp người.

1. `class CDongVat:`

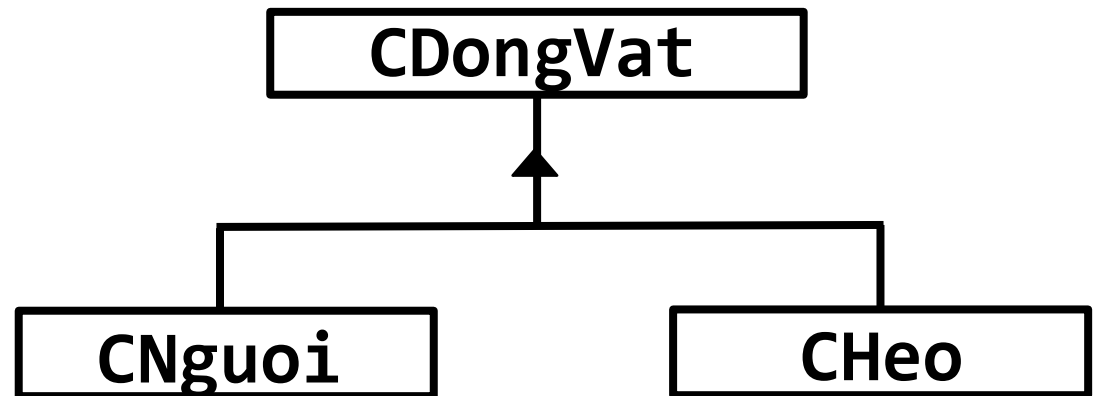
2. `...`

3. `class CHeo(CDongVat):`

4. `...`

5. `class CNgui(CDongVat):`

6. `...`



Ví dụ kế thừa

— Ví dụ 02: Khai báo lớp động vật, lớp heo và lớp người.

1. `class CDongVat:`

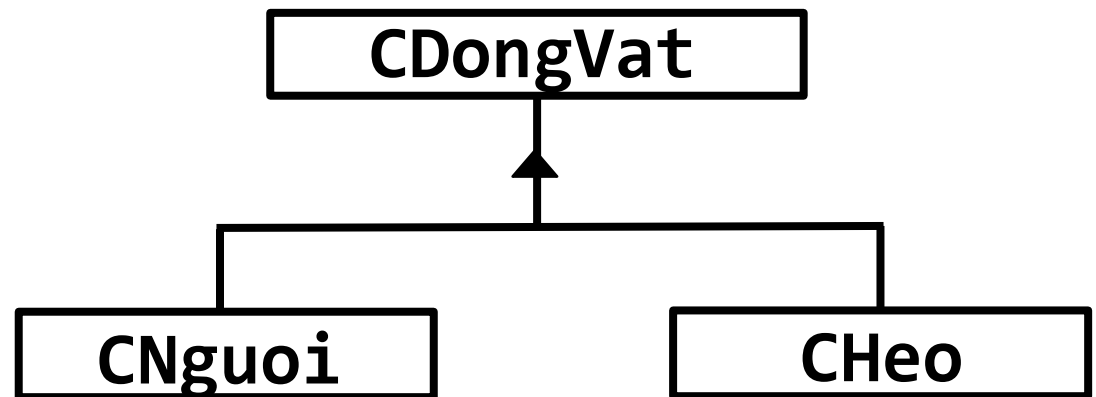
2. `...`

3. `class CHeo(CDongVat):`

4. `...`

5. `class CNgui(CDongVat):`

6. `...`



Dòng số 3 ta nói lớp đối tượng **CHeo** kế thừa từ lớp đối tượng **CDongVat**

Ví dụ kế thừa

— Ví dụ 02: Khai báo lớp động vật, lớp heo và lớp người.

1. `class CDongVat:`

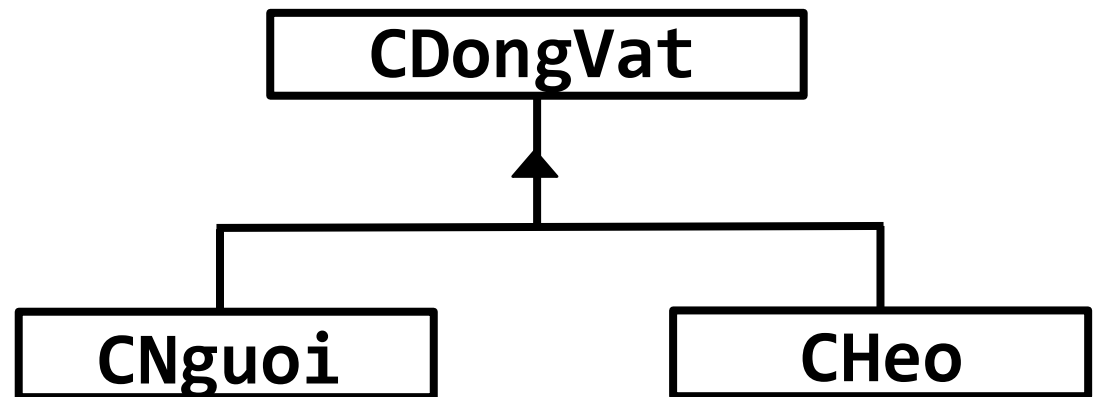
2. `...`

3. `class CHeo(CDongVat):`

4. `...`

5. `class CNgnoi(CDongVat):`

6. `...`



Dòng số 5 ta nói lớp đối tượng **CNgnoi** kế thừa từ lớp **CDongVat**

QUI TẮC KẾ THỪA

QUI TẮC KẾ THỪA

- Lớp con có thể sử dụng toàn bộ dữ liệu khai báo ở mức độ **protected** và **public** ở lớp cha. Riêng với **private** thì không được, vì đó là mức độ bảo mật cao nhất, chỉ sử dụng bên trong nội bộ của lớp cha.

Cảm ơn quý vị đã lắng nghe

Nhóm tác giả

TS. Nguyễn Tấn Trần Minh Khang