1 TS. GVC. VÕ XUÂN THỂ: voxuanthephd@gmail.com (công việc) tranhailua@yahoo.com (cá nhân)

|  |
| --- |
| **BÀI 4: TÍNH THỪA KẾ: Inheritance** |

**1. Lớp thừa kế (Inheritance class)**

Lớp thừa kế là lớp được xây dựng từ lớp có trước. Lớp có trước gọi là lớp cha/mẹ hay lớp cơ sở, lớp thừa  kế gọi là lớp con hoặc lớp dẫn xuất. Lớp con được *thừa hưởng* những thành phần của lớp cha và có thể có  *thêm* những thành phần riêng của nó.

**\* Khai báo**: lớp B thừa kế lớp A

class A

{…};

class B: **public** A //public gọi là kiểu thừa kế

{…};

**\* Chú ý:**

- Lớp con không thừa kế phương thức khởi tạo và phương thức hủy bỏ của lớp cha. - Phương thức khởi tạo của lớp con sẽ gọi phương thức khởi tạo của lớp cha để khởi tạo các dữ liệu thừa  kế **trước khi phương thức khởi tạo của lớp con được thực hiện**. Phương thức hủy bỏ của lớp con sẽ  gọi phương thức hủy bỏ của lớp cha **sau** khi thực hiện.

- Lớp con có thể định nghĩa lại phương thức của lớp cha nếu *không muốn thừa kế* phương thức đó. - Một đối tượng thuộc lớp con thì cũng có thể xem như là đối tượng thuộc lớp cha, KHÔNG nguợc lai.

B b; b cũng là 1 đối tượng của A (mặc nhiên tự hiểu: A b;)

A a; a Không phải là đối tượng của B (~~B a;~~ <= KHÔNG tự hiểu)

**\* Ví dụ:**

a) Xây dựng lớp hình chữ nhật có dữ liệu là cd,cr và có phương thức khởi tạo, phương thức tính diện tích,  phương thức tính chu vi.

b) Xây dựng lớp hình vuông thừa kế lớp hcn.

c) Viết chương trình sử dụng lớp hv, tính diện tích, chu vi hình vuông.

**HD:**

**a) file hcn.cpp**

**class hcn**

{

float cd,cr;

 public:

**hcn(float a=0, float b=0)** //phương thức khởi tạo

{ cd=a; cr=b; }

**float tinhdt()**

{ return cd\*cr; }

**float tinhcv()**

{ return (cd+cr)\*2; }

};

2 TS. GVC. VÕ XUÂN THỂ: voxuanthephd@gmail.com (công việc) tranhailua@yahoo.com (cá nhân) **b) file hv.cpp**

#include "hcn.cpp"

**class hv : public hcn**

{

public:

**hv(float a=0):hcn(a,a)** { } //a la canh hv,

 //phương thức khởi tạo **hv** gọi phương thức khởi tạo **hcn**

};

**Ghi chú**: phương thức khoi tạo hcn được thực hiện trước phương thức khởi tạo hv.   Phương thức khởi tạo hv không cần làm gì cả.

**c) file tinhdtcv.cpp**

#include <iostream>

#include "hv.cpp"

using namespace std;

**int main()**

{

float a;

cout<<"Nhap canh hv:"; cin>>a;

hv h(a);

cout<<"\ndien tich="<<h.tinhdt();

cout<<"\nchu vi="<< h.tinhcv();

}

|  |
| --- |
| **VIẾT BỔ SUNG HÀM** = nên làm trong class hcm -> trong class hv sẽ dùng được (ko ngược lại) **+ NHẬP/XUẤT (KIỂM TRA ĐIỀU KIỆN NHẬP)**  **+ CHIỀU DÀI ĐƯỜNG CHÉO = ĐK CỦA HÌNH TRÒN NGOẠI TIẾP** |

**Ghi chú**: có thể viết tất cả vào một file nhưng để có thể sử dụng lớp hcn, hv trong những trường hợp khác,  ta nên tách mỗi lớp là một file, cần lớp nào thì include file tương ứng.

 [LƯU CÙNG TM hoặc KHAI BÁO using namespace <đường dẫn> HOẶC thiết lập Options]

**2. Các loại thừa kế**: Có hai loại thừa kế:

+ Thừa kế đơn (đơn thừa kế)

 Thừa kế đơn là thừa kế từ một lớp cha [như VD trên: hcn + hv].

+ Đa thừa kế : lớp con thừa kế từ 2 lớp cha/mẹ trở lên;

 Tức là, có thể thừa kế nhiều hơn 2, tức là có 3, 4,…lớp cha => đa thừa kế (2 cha trở lên)  ->nếu là 2 cha/mẹ thì gọi là: thừa kế bội/kép

**\* Khai báo**: lớp C thừa kế lớp A,B [thừa kế bội]

class A

{…};

class B

{…};

class C: public A, public B

{…};

3 TS. GVC. VÕ XUÂN THỂ: voxuanthephd@gmail.com (công việc) tranhailua@yahoo.com (cá nhân)

**\* Ví dụ:**

#include <iostream>

**class C1**

{ int a;

 public:

void set(int k) { a=k; }

int get() { return a; }

};

**class C2**

{

int b;

 public:

void set(int k) { b=k; }

int get() { return b; }

};

**class C: public C1, public C2** //C co hai set, hai get

{

public:

void output()

{

C1::set(1); C2::set(2);

 cout<<C1::get()<<" "<<C2::get();

}

};

**int main()**

{

C c;

 c.output();

}

|  |
| --- |
| **VIẾT BỔ SUNG HÀM** = nên làm trong class C1, C2  **+ NHẬP/XUẤT**  **+ Bổ sung thêm thuộc tính int c cho class C (private)**  **+ Hàm Set và Get cho class C (con thừa kế) trong đó : Set 3 giá trị (C1, C2, C)  Get : in ra cả 3 giá trị**  (tương tự như Bài tập : định vị một điểm trong không gian 3 chiều) |

 Không gian 3 chiều là có 3 trục x, y, z; vì vậy 1 điểm X trong không gian này có dạng:  X(x, y, z) VD X(6,9,3): mỗi trục tọa độ là 1 class

**\* Ghi chú**: do lớp C1, C2 đều có phương thức set(),get()

 nên trong lớp C khi sử dụng cần phải ghi rõ tên lớp, vd:C1::set(1); là gọi set() của C1.

**3. Kiểu thừa kế**

Có ba loại thuộc tính kế thừa là: private(mặc định), protected, public

**3.1 Thuộc tính truy cập**

Trong một lớp nếu một thành phần có thuộc tính là

- **private:** thì thành phần này chỉ được truy xuất trong lớp.

 Lớp con sẽ không được thừa kế thành phần này.

 Lớp/hàm friend ĐƯỢC truy xuất.

- **protected:** giống private nhưng: cho phép các lớp con thừa kế trực tiếp được truy xuất .  Lớp/hàm friend KHÔNG truy xuất được.

- **public:** mọi phương thức hoặc hàm đều có thể truy xuất. Đối với lớp con thì có thể truy xuất trực tiếp hoặc thông qua một đối tượng; phương thức trong lớp ngang hàng hoặc hàm ngoài lớp thì chỉ có thể  truy xuất qua một đối tượng.

4 TS. GVC. VÕ XUÂN THỂ: voxuanthephd@gmail.com (công việc) tranhailua@yahoo.com (cá nhân)

**Ví dụ:**

#include <iostream>

**class C1**

{ private protected:

int a;

void f()

{ a=1; //truy xuat truc tiep (trong lớp), đúng

C1 c;

c.a=1; //truy xuat thong qua doi tuong (trong lớp), đúng

}

};

**class C2 : C1** // thừa kế theo private (không ghi gì = mặc định)

{ void g()

{ a=1; //sai (trên kia private), do a la private, phương thức trong lớp con khong the truy xuat C1 c;

 c.a=1; //sai, do a la private, phương thức trong lớp con khong the truy xuat }

};

**int main()**

{ a=1; //sai, do a chua khai bao (không sử dụng a trong class C1 vì hàm main là ngoài class C1) C2 c;

c.a=1; //sai, do a la private, khong the truy xuat

}

|  |
| --- |
| **VIẾT BỔ SUNG HÀM** = nên làm trong class C1  **+ NHẬP/XUẤT**  **+ Hàm Set và Get cho class C1, C2 (con thừa kế)**  **+ thử ALL các trường hợp trong hàm main()**  **+ 3 file: private, protect, public** |

**Nhận xét:**

- Nếu cho a trong lớp C1 có thuộc tính là protected thì trong phương thức g() chỉ có lệnh c.a=1 là sai. - Nếu cho a trong lớp C1 có thuộc tính là public thì trong phương thức g() không còn lệnh sai. Trong hàm  main chỉ có lệnh a=1 là sai.

5 TS. GVC. VÕ XUÂN THỂ: voxuanthephd@gmail.com (công việc) tranhailua@yahoo.com (cá nhân)

**3.2 Kiểu thừa kế**

Khi lớp B thừa kế lớp A, thì có ba kiểu thừa kế: private(mặc định)/ protected/ public class A

{…};

class B: **[kiểu thừa kế]** A

{…};

***\* Bảng Thuộc tính của thành phần thừa kế***

|  |  |  |
| --- | --- | --- |
| **Thuộc tính của thành phần  trong lớp cha (lớp cơ sở) [1]** | **Kiểu thừa kế [2]** | **Thuộc tính của thành phần  thừa kế [3]** |
| Private | private/ protected/ public | không thừa kế = private all |
| Protected | private/protected/ public | private/protected/ protected |
| Public | private/protected/ public | private/protected/ public |

**Nhận xét:** Thuộc tính của thành phần thừa kế là thuộc tính yếu nhất giữa thuộc tính của thành phần trong  lớp cha và kiểu thừa kế

CHÚ Ý: Thuộc tính của thành phần : Thừa kế[3] = Cơ sở[1] ∧ Kiểu thừa kế[2]

 Với thứ tự thuộc tính thừa kế “mạnh: -> “yếu”: public -> protect -> private -------XEM LẠI RÕ HƠN MINH HỌA TRÊN trong trường hợp tô nền trong Bảng trên class A //lớp cơ sở

{ protect: //[1]

 int a;

 void f() { …}

 …

};

class B: **public** A //[2] kiểu thừa kế

{

 //tự động trong có do thừa kế các thành phần: thuộc tính a và phương thức f() của lớp A: theo dạng protected //[3]  };

**\* Ví dụ:** Xem chương trình sau:

#include <iostream>

**class C1**

{

int a;

};

**class C2: C1**

{

void set(int x)

{ a=x; }

};

**int main()**

{

C2 y;

y.a=5;

}

a) Lệnh nào sai, tại sao?

b) Nếu cho a có thuộc tính là public thi lenh nao sai, tai sao?

c) Nếu cho a có thuộc tính là public, và C2 thừa kế C1 theo kiểu protected thì còn sai không? Tại sao? d) Nếu cho a có thuộc tính là public, và C2 thừa kế C1 theo kiểu public thì còn sai không? Tại sao?

6 TS. GVC. VÕ XUÂN THỂ: voxuanthephd@gmail.com (công việc) tranhailua@yahoo.com (cá nhân)

**Đáp án**:

a) Vi a la private, chi su dung trong lop C1, nen lenh a=x va lenh y.a=5 sai.

b) Lenh y.a=5 la sai; vi a la public, C2 thua ke C1 theo kieu private nen a trong C2 la private, a chi su dung  trong C2, main khong the truy cap a.

c) Lenh y.a=5 van sai vi a la public, C2 thua ke C1 theo kieu protected nen a trong C2 la protected, a chi su  dung trong C2 va lop con cua C2, main khong the truy cap a.

d) Khong con lenh sai vi khi do a trong C2 la public nen y.a=5 dung.

|  |
| --- |
| **Cài đặt các files theo từng trường hợp thực nghiệm cho các đáp án của bài minh họa trên VIẾT BỔ SUNG HÀM**  **+ main()… & Hàm thiết lập (có tham số mặc định)**  **+ NHẬP() /XUẤT()**  **+ đầy đủ các hàm set() và get() cho class C1, C2 (con thừa kế)**  **+ thử các trường hợp trong hàm main()**  + 9 file: private, protect, public trong lớp “cha”C1 tương ứng 3 kiểu thừa kế của lớp “con” C2 |