

C++语言基础

迂者 - 贺利坚

<http://blog.csdn.net/sxhelijian/>

<http://edu.csdn.net>





本节主题：

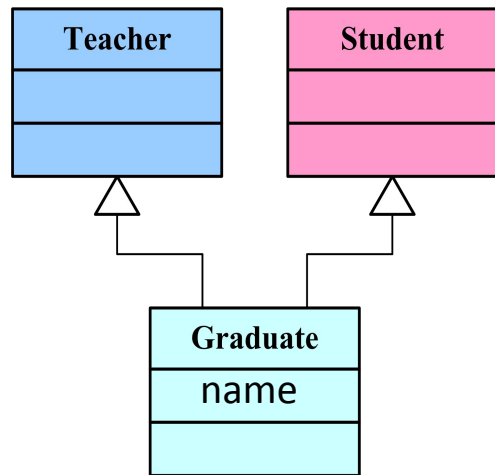
多重继承中二义性的解决

二义性

- 两个基类中数据成员名相同问题，怎么办？
 - 两个基类中用相同标识符表示同一事物——正常
 - 在共同的派生类中，这个标识符如何作解释和使用？

```
class student
{
    char *name;
    ...
};
```

```
class Teacher
{
    char *name;
    ...
};
```



- 在派生类中的解决方法
 - 在标识符前用基类名做前缀: `Teacher::name`和`Student::name`
 - 基类和派生类需要有一个完整的设计，不能随意而为。

二义性(1): 两个基类有同名成员

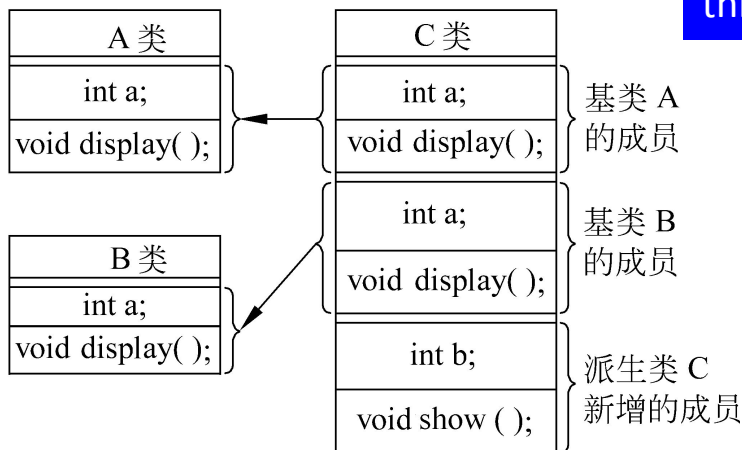
```
class A
{
public:
    int a;
    void display( );
};
```

```
class B
{
public:
    int a;
    void display( );
};
```

```
//错误写法
C c1;
c1.a=3;
c1.display();
```

```
//正确写法
//(1)可以用基类名来限定
c1.A::a=3;
c1.A::display();
//在派生类中针对当前对象
A::a=3;
A::display( );
this->B::display();
```

```
class C :public A,public B
{
public :
    int b;
    void show();
};
```



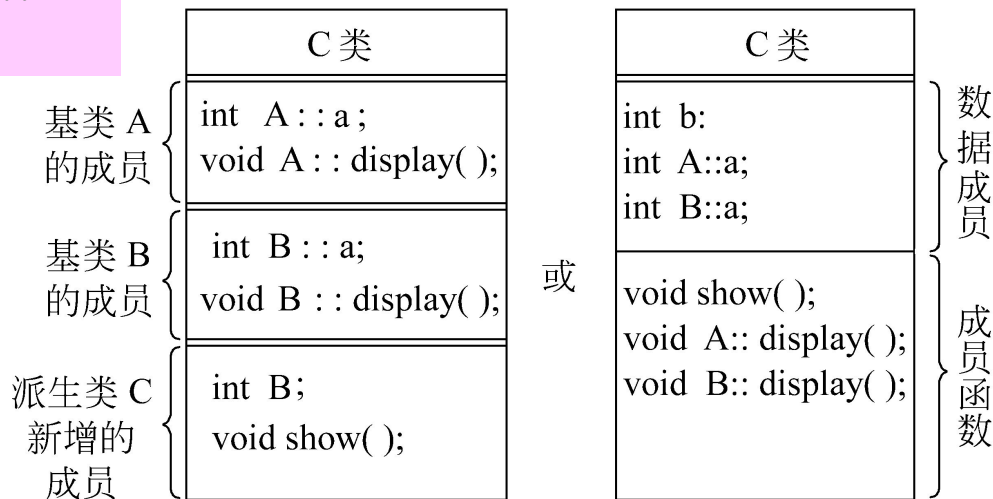
另一种更清晰的表达

```
class A
{
public:
    int a;
    void display( );
};
```

```
class B
{
public:
    int a;
    void display( );
};
```

```
c1.A::a=3;
c1.A::display();
A::a=3;
A::display( );
this->B::display();
```

```
class C :public A,public B
{
public :
    int b;
    void show();
};
```



二义性 (2): 基类和派生类有同名成员

```
class A
{
public:
    int a;
    void display( );
};
```

```
class B
{
public:
    int a;
    void display( );
};
```

```
class C :public A,public B
{
public :
    int a;
    void display();
};
```

C 类
int a; int A : : a; int B : : a;
void display(); void A::display(); void B::display();

访问的是派生类C中的成员:

```
C c1;
c1.a=3;
c1.display( );
```

在派生类外访问基类A中的成员:

```
c1.A::a=3;
c1.A::display();
```

规则: 同名覆盖

- ☐ 基类的同名成员在派生类中被屏蔽，成为“不可见”的
- ☐ 对成员函数，限于函数名和参数个数相同、类型相匹配时；若只有函数名相同而参数不同，属于函数重载。

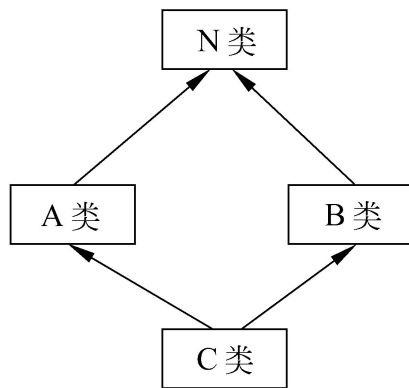
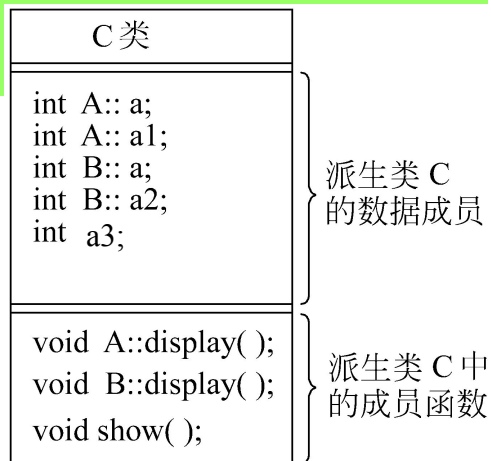
二义性(3): 两个基类从同一个基类派生

```
class N
{
public:
    int a;
    void display(){
        cout<<"A::a="<<a<<endl;
    }
};
```

```
class A:public N
{
public:
    int a1;
};
```

```
class B:public N
{
public:
    int a2;
};
```

```
class C :public A,public B
{
public :
    int a3;
    void show( ){
        cout<<"a3="<<a3<<endl;
    }
};
```



```
C c1;
//合法访问
c1.A::a=3;
c1.B::display();
//非法访问
c1.a=3;
c1.display( );或
c1.N::a=3;
c1.N::display();
```

THANKS

本课程由 迂者-贺利坚 提供

CSDN网站：www.csdn.net
企业服务：<http://ems.csdn.net/>
人才服务：<http://job.csdn.net/>
CTO俱乐部：<http://cto.csdn.net/>
高校俱乐部：<http://student.csdn.net/>
程序员杂志：<http://programmer.csdn.net/>

CODE平台：<https://code.csdn.net/>
项目外包：<http://www.csto.com/>
CSDN博客：<http://blog.csdn.net/>
CSDN论坛：<http://bbs.csdn.net/>
CSDN下载：<http://download.csdn.net/>

