

C++语言基础

迂者 - 贺利坚

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

<http://edu.csdn.net>





本节主题：

基类与派生类的转换

基类与派生类的转换

- 不同类型的数据在赋值时的自动类型转换

```
int x=2;  
x=x+2.5;
```

- 公用派生类完整地继承了基类的功能，使基类与派生类对象之间有赋值兼容关系

```
a1=b1 ; //合法访问  
A &aa=b1 ;  
pa1=&b1;  
b1=a1; //非法访问  
B &bb=a1;  
pb1=&a1;
```

有：

```
class A  
{...}  
class B : public A  
{...}  
A a1, *pa1;  
B b1, *pb1;
```

基类
基类数据成员
基类成员函数

派生类
基类数据成员
基类成员函数
派生类数据成员
派生类成员函数

(1) 派生类对象向基类对象赋值

```
class A {...}
```

```
class B: public A {...}
```

```
A a1;
```

```
B b1;
```

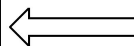
- 在赋值时，要舍弃派生类自己的成员。
- 赋值只是对数据成员赋值，对成员函数不存在赋值问题。
- 赋值后不能企图通过对象a1去访问派生类对象b1的成员

基类对象

基类
基类数据成员
基类成员函数

派生类对象

派生类
基类数据成员
基类成员函数
派生类数据成员
派生类成员函数



赋值时此
部分舍弃不用



```
a1=b1; //合法，用派生类对象对基类对象赋值  
//age是B中新增的成员  
b1.age=21; //正确  
a1.age=23; //错误
```

(2) 派生类对象替代基类对象.....

- ❏ 派生类对象替代基类对象可以向基类对象的引用进行赋值或初始化

```
class A {...}
```

```
class B: public A {...}
```

```
A a1;
```

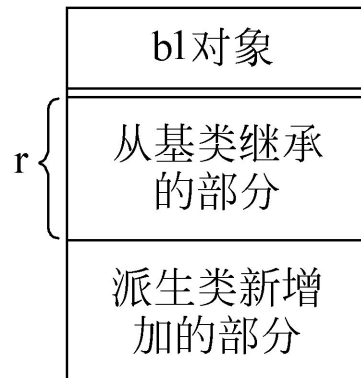
```
B b1;
```

```
A& r1=a1;
```

❏ 变量r1是a1的引用，r1和a1共享同一段存储单元

```
A& r=b1;
```

❏ 变量r是b1继承自A类的部分的别名，r和b1继承了A类的那一部分共享同一段存储单元



- ❏ 此时r并不是b1的别名，也不与b1共享同一段存储单元，只是b1中基类部分的别名，r与b1中基类部分共享同一段存储单元，r与b1具有相同的起始地址

(3) 子类对象当实参传递给基类形参(传值/传地址/引用)

```
class A {...}
```

```
class B: public A {...}
```

```
void fun(A& r)
```

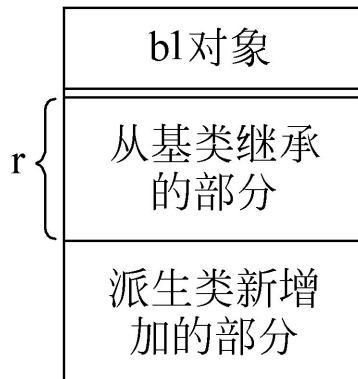
```
{
```

```
    cout<<r.num<<endl;
```

```
}
```

```
A a1;
```

```
B b1;
```



☞ 函数的参数是基类对象或基类对象的引用，相应的实参可以用子类对象

☞ 在调用fun函数时可以用派生类B的对象b1作实参:

```
fun(a1); fun(b1);
```

☞ 在fun函数中只能访问派生类中继承自基类成员的值。

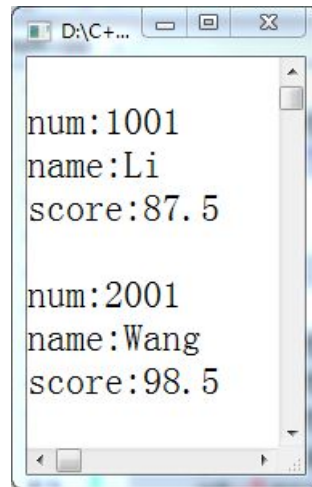
(4)指向基类对象的指针变量也可以指向派生类对象

❏ 派生类对象的地址可赋给指向基类对象的指针

```
class Student { ... };  
class Graduate:public Student { ... };  
int main()  
{  
    Student stud1(1001,"Li",87.5);  
    Graduate grad1(2001,"Wang",98.5,563.5);  
    Student *pt=&stud1;  
    pt->display( );  
    pt=&grad1;  
    pt->display( );  
    return 0;  
}
```

```
void Student::display( )  
{  
    cout<<"num:"<<num<<endl;  
    cout<<"name:"<<name<<endl;  
    cout<<"score:"<<score<<endl;  
}
```

- ❏ pt是指向Student类对象的指针变量，即使指向了grad1，只能访问从基类继承的部分
- ❏ 通过指向基类对象的指针，只能访问派生类中的基类成员，而不能访问派生类增加的成员
- ❏ 要通过使用基类指针能够调用基类和子类对象的成员，期待虚函数和多态性



```
void Graduate::display()  
{  
    Student::display();  
    cout<<"pay="<<pay<<endl;  
}
```

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/>

