

第七章：图文例解C类的多重继承与虚拟继承 PDF转换可能丢失图片或格式，建议阅读原文

https://www.100test.com/kao_ti2020/134/2021_2022__E7_AC_AC

[_E4_B8_83_E7_AB_A0_EF_c97_134302.htm](https://www.100test.com/kao_ti2020/134/2021_2022__E7_AC_AC_E4_B8_83_E7_AB_A0_EF_c97_134302.htm) 在过去的学习中，我们始终接触的单个类的继承，但是在现实生活中，一些新事物往往会拥有两个或者两个以上事物的属性，为了解决这个问题，C引入了多重继承的概念，C允许为一个派生类指定多个基类，这样的继承结构被称做多重继承。举个例子，交通工具类可以派生出汽车和船连个子类，但拥有汽车和船共同特性水陆两用汽车就必须继承来自汽车类与船类的共同属性。由此我们不难想出如下的图例与代码：

当一个派生类要使用多重继承的时候，必须在派生类名和冒号之后列出所有基类的类名，并用逗好分隔。

```
#include iostream> using namespace std. class Vehicle { public: Vehicle(int weight = 0) { Vehicle::weight = weight. } void SetWeight(int weight) { cout Vehicle::weight = weight. } virtual void ShowMe() = 0. protected: int weight. }. class Car:public Vehicle//汽车 { public: Car(int weight=0,int aird=0):Vehicle(weight) { Car::aird = aird. } void ShowMe() { cout } protected: int aird. }. class Boat:public Vehicle//船 { public: Boat(int weight=0,float tonnage=0):Vehicle(weight) { Boat::tonnage = tonnage. } void ShowMe() { cout } protected: float tonnage. }. class AmphibianCar:public Car,public Boat//水陆两用汽车,多重继承的体现 { public: AmphibianCar(int weight,int aird,float tonnage):Vehicle(weight),Car(weight,aird),Boat(weight,tonnage) //多重继承要注意调用基类构造函数 { } void ShowMe() { cout } }. int
```

```
main() { AmphibianCar a(4,200,1.35f).//错误 a.SetWeight(3).//错
误 system("pause"). } 上面的代码从表面看，看不出有明显的语
发错误，但是它是不能够通过编译的。这有是为什么呢？这
是由于多重继承带来的继承的模糊性带来的问题。先看如下
的图示：在图中深红色标记出来的地方正是主要问题所在，
水陆两用汽车类继承了来自Car类与Boat类的属性与方法
，Car类与Boat类同为AmphibianCar类的基类，在内存分配
上AmphibianCar获得了来自两个类的SetWeight()成员函数，
当我们调用a.SetWeight(3)的时候计算机不知道如何选择分别
属于两个基类的被重复拥有了的类成员函数SetWeight()。由
于这种模糊问题的存在同样也导致了AmphibianCar
a(4,200,1.35f).执行失败，系统会产生Vehicle”不是基或成员的
错误。以上面的代码为例，我们要想让AmphibianCar类既获
得一个Vehicle的拷贝，而且又同时共享用Car类与Boat类的数
据成员与成员函数就必须通过C所提供的虚拟继承技术来实
现。我们在Car类和Boat类继承Vehicle类出，在前面加
上virtual关键字就可以实现虚拟继承，使用虚拟继承后，当系
统碰到多重继承的时候就会自动先加入一个Vehicle的拷贝，
当再次请求一个Vehicle的拷贝的时候就会被忽略，保证继承
类成员函数的唯一性。修改后的代码如下，注意观察变化
：#include iostream> using namespace std. class Vehicle { public:
Vehicle(int weight = 0) { Vehicle::weight = weight. cout } void
SetWeight(int weight) { cout Vehicle::weight = weight. } virtual void
ShowMe() = 0. protected: int weight. }. class Car:virtual public
Vehicle//汽车，这里是虚拟继承 { public: Car(int weight=0,int
aird=0):Vehicle(weight) { Car::aird = aird. cout } void ShowMe() {
```

```
cout } protected: int aird. }. class Boat:virtual public Vehicle//船,这里是虚拟继承 { public: Boat(int weight=0,float tonnage=0):Vehicle(weight) { Boat::tonnage = tonnage. cout } void ShowMe() { cout } protected: float tonnage. }. class AmphibianCar:public Car,public Boat//水陆两用汽车,多重继承的体现 { public: AmphibianCar(int weight,int aird,float tonnage):Vehicle(weight),Car(weight,aird),Boat(weight,tonnage) //多重继承要注意调用基类构造函数 { cout } void ShowMe() { cout } void ShowMembers() { cout } }. int main() { AmphibianCar a(4,200,1.35f). a.ShowMe(). a.ShowMembers(). a.SetWeight(3). a.ShowMembers(). system("pause"). }
```

注意观察类构造函数的构造顺序。虽然说虚拟继承与虚函数有一定相似的地方，但读者务必要记住，他们之间是绝对没有任何联系的！100Test 下载频道开通，各类考试题目直接下载。详细请访问 www.100test.com