

类继承--从已有的类派生出新的类，派生类继承了基类的特征，包括方法。

1. 一个简单的基类

公有派生，如下所示

```
class RatePlayer : public TableTennisPlayer
{
    .....
}
```

(1)使用公有派生，基类的公有成员成为派生类的公有成员，而基类的私有成员也成为派生类的一部分，但只能通过基类的公有和保护方法访问；

(2)派生类构造函数必须使用基类构造函数，通过成员初始化列表语法来实现；

(3)有关派生类构造函数的要点如下：

- **首先创建基类对象；**
- **派生类构造函数应通过成员初始化列表将基类信息传递给基类构造函数；**
- **派生类构造函数应初始化派生类新增的数据成员。**

(4)派生类和基类间的特殊关系：

- **派生类对象可以使用基类的方法，条件是方法不是私有的；**
- **基类指针可以在不进行显示类型转换的情况下指向派生类对象；**
- **基类引用可以在不进行显示类型转换的情况下引用派生类对象；**
- **但基类指针或引用只能用于调用基类方法，不可以将基类对象和地址赋给派生类引用和指针；**
- **基类引用定义的函数或指针参数可以用于基类对象或派生类对象；**
- **引用兼容性也可以将基类对象初始化为派生类对象；**

2.继承：is-a关系 & 多态公有继承

公有继承是一种 is-a 关系；

同一个方法在派生类和基类中的行为是不同的，即同一个方法的行为随上下文而异，这就是多态，有两种机制可以用于实现多态公有继承：

- **在派生类中重新定义基类的方法；**
- **使用虚方法；**

(1) 虚方法是声明方法时使用了关键字**virtual**的方法，使用了该关键字，程序将根据引用或指针指向的对象的类型来选择方法；但如果没有使用，则程序根据指针类型或引用类型来选择方法。

(2) 使用**虚析构函数**可以确保正确的析构函数序列被调用，没有使用，则只调用对应于指针类型的析构函数；

3.静态联编和动态联编

函数名联编：将源代码中的函数调用解释为执行特定的函数代码块；

(1) 动态联编与通过指针和引用调用方法相关；

(2) 将派生类引用或指针转换为基类引用或指针被称为**向上强制转换**，这是不需要显示类型转换的；反过来，将基类引用或指针转换为派生类指针或引用是**向下强制转换**，需要显示类型转换。

(3) 编译器对非虚方法使用静态联编，因为静态联编的效率更高，因此被设置为C++的默认选择；所以仅将那些预期将被重新定义的方法声明为虚的。

(4) 编译器处理虚函数的方法是给每个对象添加了一个隐藏成员，而该隐藏成员中保存了一个指向函数地址数组的指针，该数组称为虚函数表；总之，在使用虚函数时，在内存和执行速度方面有一定的成本。

(5) 构造函数不能是虚函数，而析构函数应当是虚函数，除非类不用做基类；

友元不能是虚函数，因为它不是类成员，而只有成员函数才能是虚函数；

(6) 重新定义继承的方法，应确保与原来的原型完全相同，但如果返回类型是基类引用或指针，则可以修改为指向派生类的引用或指针，这种特性被称为**返回类型协变**，但这种例外只适用于返回值，而不适用于参数；

如果基类声明被重载了，则应在派生类中重新定义所有的基类版本；如果只重新定义其中的几个版本，而未被重新定义的版本都将被隐藏；

4.访问控制：protected

使用关键字**protected**表示，与**private**相似，其区别在于基类派生的类中才表现出来，派生类的成员可以直接访问基类的保护成员，但不能直接访问基类的私有成员；

最好对类数据成员采用私有访问控制，而对成员函数，则可以使用保护访问控制；

5.抽象基类

抽象基类必须至少包含一个纯虚函数，而纯虚函数是声明结尾处为=0，该函数可以有定义也可以不用定义；当类声明包含纯虚函数时，则不能创建该类的对象。

6.继承和动态内存分配

(1) 派生类不使用new的情况下，派生类可以不用定义**显示析构函数、复制构造函数和赋值运算符**；但派生类使用new的情况下，则需要定义，其中，复制构造函数必须使用成员初始化列表来调用基类的复制构造函数，而使用定义赋值运算符时也需要显示调用基类的赋值运算符；

(2) 友元函数要访问基类的成员，必须使用强制类型转换，以便匹配原型时能够选择正确的函数；

7.类设计回顾

(1) 编辑器生成的成员函数

- 默认构造函数：要么是没有参数，要么是所有参数都有默认值；
- 复制构造函数：

在下述情况下，将使用复制构造函数：

- 将新对象初始化为一个同类对象；
- 按值将对象传递给函数；
- 函数按值返回对象；
- 编译器生成临时对象。

- 赋值运算符

(2) 其他类方法

- 构造函数：构造函数会创建新的对象，因而不可以被继承；
- 析构函数：必须定义显示析构函数来释放使用new分配的内存，对于基类，应提供一个虚析构函数；
- 转换函数：只提供了一个参数的构造函数定义了从参数类型到类类型的转换，使用explicit关键字可以禁止进行隐式转换；

要将类对象转换为其他类型，应定义转换函数（参见第11章）

- 按值传递对象与传递引用：一般使用对象作为参数的函数时，应按引用而不是按值来传递对象；
- 返回对象与返回引用：如果函数返回在函数中创建的临时对象，则不要使用引用，否则一般返回引用；
- 使用const：使用const来确保方法不会修改调用它的对象；