类是C++面向对象提出的解决方案之一。是C++的核心。

类可以解释为用户定义的类型。

类用于指定对象的形式，包含了**数据表示方法**和**数据处理**的方法。

类中的(数据)和(方法）称为类的成员。

函数在一个类中被称为类成员，（一般函数定义类的行为。）

类定义了其中的对象包括了什么，可以在这个对象上执行哪些操作。如：

class Box

{

public:

double length; // 盒子的长度

double breadth; // 盒子的宽度

double height; // 盒子的高度

};

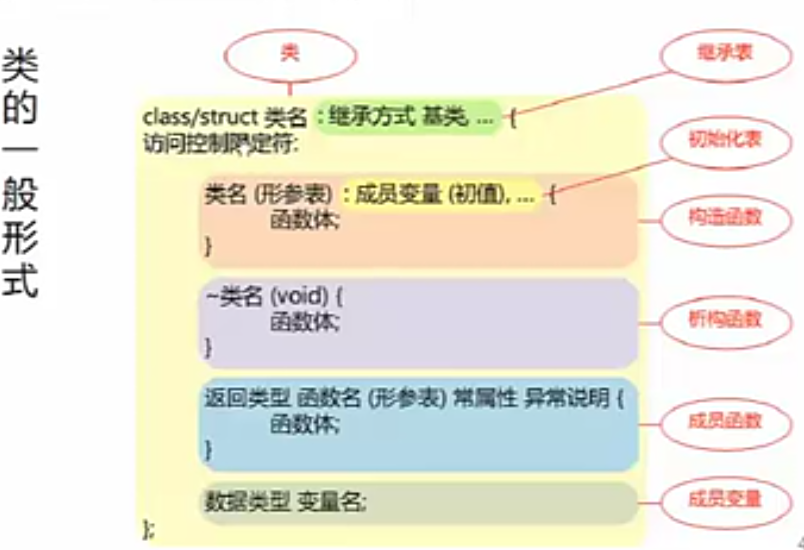
关键字 **public** 确定了类成员的访问属性。在类对象作用域内，公共成员在类的外部是可访问的。您也可以指定类的成员为 **private** 或 **protected**

上面只是声明了一个类，并没有定义具体的对象。可以如下进行定义相应类的对象：

Box box1；

Box box2；

**类的一般形式:**



class/struct(可以兼容C程序) : 继承方式/基类(**在后面说明**)

{

**访问控制限定符**: **public** /**protected** /**private**

**构造函数**,和类相同. 它没有返回类型，代表一个对象的出生的过程。并且可能包含了一个”:”冒号的(初始化值)的**初始化表.**

**析构函数**,也没有返回类型.

描述一个对象的死亡的过程.

**成员函数**, 描述了对象的行为. **常属性** **异常说明**

**{**

**函数体;**

**}**

**成员变量**,描述对象属性.

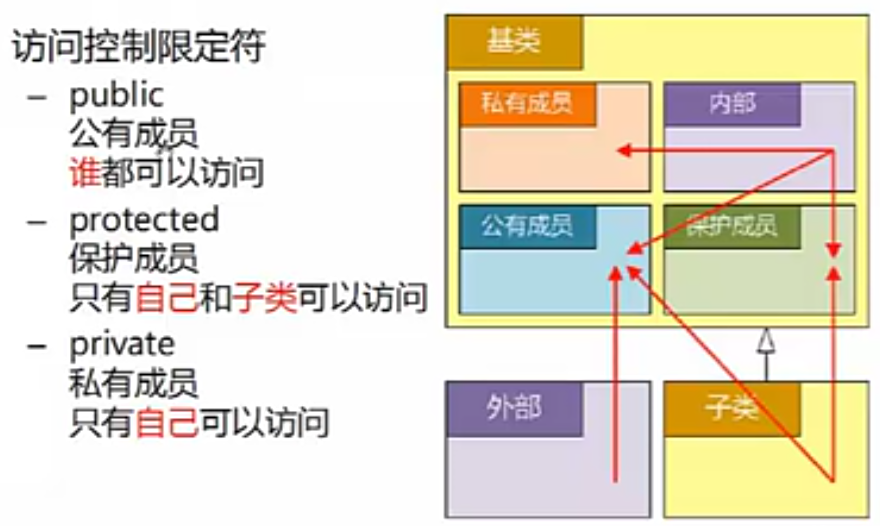
}

---------------------------- 继承 基类-------------------------

继承的概念：

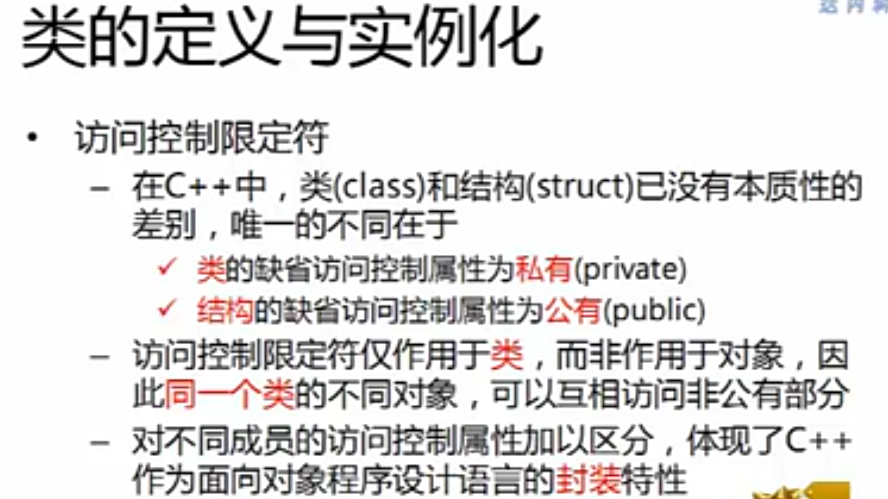
继承方式如果为protected，那么在只能在子类里面访问类的protected成员，在main()中不能直接访问protected成员。

**访问控制限定符:**



类内部, 是指类中的其他部分.

箭头指向说明了它可以访问的内容.(体现了数据的封装性, 私有成员变量不允许其他对象访问), 提高了数据的安全性。



- private限制的是类的内外，如定义了同一个类的两个不同实例，s1和s2，那么s2的参数属性可以被s1的函数改变。

student s1，s2；

s1.setName(s2); //这个函数修改了s2的私有变量。名字。

因为C++是静态语言，是在编译的时候确定是否能够访问类的属性，编译完成后就不能再改变。

示例：



-------------- **构造函数**的**初始化列表** -----------

类的构造函数是类的一种特殊的成员函数，它会在每次创建类的新对象时执行。

构造函数的名称与类的名称是完全相同的，并且不会返回任何类型，也不会返回 void。构造函数可用于为某些成员变量设置初始值。

构造函数可以不带参数，也可以带参数。参数一般是类内部的参数。

也可以使用初始化列表，进行初始化：





* new与malloc不同，new会分配空间，然后调用构造函数。
* 构造函数，可以定义构造形参。在定义对象的时候可以使用构造实参赋值给构造形参，进行类对象的初始化。

**类的析构函数**

类的析构函数是类的一种特殊的成员函数，它会在每次删除所创建的对象时执行。

析构函数的名称与类的名称是完全相同的，只是在前面加了个波浪号（~）作为前缀，它不会返回任何值，也不能带有任何参数。析构函数有助于在跳出程序（比如关闭文件、释放内存等）前释放资源。

示例：



拷贝构造函数：

