

# C++程序设计第九节课官方笔记

## 目录

- 一、 课件下载及重播方法
- 二、 本章/教材结构图
- 三、 本章知识点及考频总结
- 四、 配套练习题
- 五、 其余课程安排

### 一、课件下载及重播方法

### 二、教材节构图



### 三、本章知识点及考频总结

(一) 选择题 (共 15 道)

1. 通过已有的类建立新类的过程,叫作类的派生。原来的类称为基类,也称为父类或一般类;新类称为派生类,也称为子类或特殊类。派生类派生自基类,或继承于基类,也可以说基类派生了派生类。派生机制是 C++语言及面向对象程序设计方法的重要特征之一。派生类可以再作为基类派生新的派生类,由此基类和派生类的集合称作类继承层次结构。

使用基类派生新类时,除构造函数和析构函数外,基类的所有成员自动成为派生类的成员,包括基类的成员变量和成员函数。同时,派生类可以增加基类中没有的成员,这同样是指成员变量和成员函数。可以重新定义或修改基类中已有的成员,包括可以改变基类中成员的访问权限。当然派生类需要定义自己的构造函数和析构函数。

使用基类成员是一个重用的过程,在基类之上进行调整,不论是添加新成员还是改造已有的,都是扩充的过程。

若派生类中定义了一个与基类中同名的成员,则会出现基类与派生类有同名成员的情况,这是允许的。同名的成员既可以是成员变量,也可以是成员函数。这种情况下,若在派生类的成员函数中访问这个同名成员,或通过派生类对象访问这个同名成员时,除非特别指明,访问的就是派生类中的成员,这种情况叫“覆盖”,即派生类的成员覆盖基类的同名成员。覆盖也称为重定义或是重写。对于成员函数来说,派生类既继承了基类的同名成员函数,又在派生类中重写了这个成员函数。这称为函数重定义,也称为同名隐藏。“隐藏”的意思是指,使用派生类对象调用这个名字的成员函数时,调用的是派生类中定义的成员函数,即隐藏了基类中的成员函数。

## 2. 派生类的定义

在 C++语言中,从基类派生派生类的一般格式如下:

```
class 派生类名: 继承方式说明符 基类名
{
    类体
};
```

继承方式说明符指明如何控制基类成员在派生类中的访问属性,通常有 3 种方式,分别是 public (公有继承)、private (私有继承) 和 protected (保护继承)。一般情况下都使用 public。protected 或 private 方式很少用到。

3. 派生类对象中包含基类成员变量,而且基类成员变量的存储位置位于派生类对象新增的成员变量之前。派生类对象占用的存储空间大小,等于基类成员变量占用的存储空间大小加上派生类对象自身成员变量占用的存储空间大小。

4. 如果基类有友元类或友元函数,则其派生类不会因继承关系而也有此友元类或友元函数。如果基类是某类的友元,则这种友元关系是被继承的。即被派生类继承过来的成员函数,如果原来是某类的友元函数,那么它作为派生类的成员函数仍然是某类的友元函数。总之,基类的友元不一定是派生类的友元;基类的成员函数是某类的友元函数,则其作为派生类继承的成员函数仍是某类的友元函数。

5. 如果基类中的成员是静态的,则在其派生类中,被继承的成员也是静态的,即其静态属性随静态成员被继承。

如果基类的静态成员是公有的或是保护的,则它们被其派生类继承为派生类的静态成员。访问这些成员时,通常用“<类名>::<成员名>”的方式引用或调用。无论有多少个对象被创建,这些成员都只有一个拷贝,它为基类和派生类的所有对象所共享。

6. 派生类和基类中都可以定义自己的成员变量和成员函数,派生类中的成员函数可以访问基类中的公有成员变量,但不能直接访问基类中的私有成员变量。也就是说,不能在派生类的函数中,使用“基类对象名.基类私有成员函数(实参)”,或是“基类对象名.基类私有成员变量”,或是“基类名::基类私有成员”的形式访问基类中的私有成员。

7. 在基类和派生类有同名函数的情况下，如果通过派生类的对象想调用基类中的函数，可以通过**派生类对象名.基类名::函数名**的方式调用。
8. 定义类时，类成员可以使用 `protected` 访问范围说明符进行修饰，从而成为“保护成员”。保护成员的访问范围比私有成员的访问范围大，能访问私有成员的地方都能访问保护成员。此外，**基类中的保护成员可以在派生类的成员函数中被访问。**
9. 在基类中，一般都需将需要隐藏的成员说明为保护成员而非私有成员。将基类中成员变量的访问方式修改为 `protected` 后，在派生类中可以直接访问。
10. 多重继承情况下如果多个基类间成员重名时，按如下方式进行处理：对派生类而言，不加类名限时默认访问的是派生类的成员；而要访问基类重名成员时，要通过类名加以限定。
11. 如果派生类中**没有新增同名成员**，当满足访问权限时，使用“派生类对象名.成员名”或“派生类对象指针->成员名”方式时，系统无法判断到底是调用哪个基类的成员，从而产生二义性。为了**避免二义性**，必须通过**基类名和作用域分辨符**来标识成员。
12. 当类的继承方式为公有继承时，基类的公有成员和保护成员的访问属性在派生类中不变，而**基类的私有成员在基类外不可直接访问**。具体来说，基类的公有成员和保护成员仍作为派生类的公有成员和保护成员，派生类的成员函数可以直接访问它们，但不能直接访问基类中的私有成员。在派生类和基类之外，可以通过派生类的对象访问从基类继承的公有成员，且必须通过基类的公有成员函数访问基类的私有成员。
13. **类型兼容规则**是指在需要基类对象的任何地方，都可以使用公有派生类的对象来替代，也称为**赋值兼容规则**。通过公有继承，派生类得到了基类中除构造函数、析构函数之外的所有成员。这样，公有派生类实际上就具备了基类的所有功能，凡是基类能解决的问题，公有派生类都可以解决。

在公有派生的情况下，有以下 3 条类型兼容规则。

- 1) 派生类的对象可以赋值给基类对象。
- 2) 派生类对象可以用来初始化基类引用。
- 3) 派生类对象的地址可以赋值给基类指针，即派生类的指针可以赋值给基类的指针。

上述 3 条规则反过来是不成立的。例如，不能把基类对象赋值给派生类对象。

14. 当类的继承方式为私有继承时，基类中的公有成员和保护成员都以私有成员身份出现在派生类中，而基类的私有成员在派生类中不可以直接访问。
15. 保护继承中，基类的公有成员和保护成员都以保护成员的身份出现在派生类中，而基类的私有成员不可以直接访问。这样，派生类的其他成员可以直接访问从基类继承来的公有和保护成员，但在类外通过派生类的对象无法直接访问它们。

## (二) 主观题 (共 0 道)

## 四、配套练习题

1、C++中，不加说明时，默认的继承方式是【 】

A.public

B.private

C.protected

D.public 或 protected

2、当派生类私有继承基类时，基类中的所有公有成员和保护成员成为派生类的

【 】

A.public 成员

B.private 成员

C.protected 成员

D.友元

3、当派生类保护继承基类时，基类中的所有公有成员和保护成员成为派生类的

【 】

A.public 成员

B.private 成员

C.protected 成员

D.友元

[参考答案] BBC

## 五、其余课程安排