

计算机等级考试二级 C 各章内容摘要7 PDF转换可能丢失图片或格式，建议阅读原文

[https://www.100test.com/kao\\_ti2020/133/2021\\_2022\\_\\_E8\\_AE\\_A1\\_E7\\_AE\\_97\\_E6\\_9C\\_BA\\_E7\\_c97\\_133722.htm](https://www.100test.com/kao_ti2020/133/2021_2022__E8_AE_A1_E7_AE_97_E6_9C_BA_E7_c97_133722.htm) 第7章 继承和派生

**【考点一】 继承与派生** 1.基本概念派生类从基类继承了各种成员的关系就称为继承。类的继承是新的类从已有类那里得到已有的特性。从已有的类产生新类的过程就是类的派生。在继承过程中，原有的类或已经存在的用来派生新类的类称为基类或父类，而由已经存在的类派生出的新类则称为派生类或子类。从派生类的角度，根据它所拥有的基类数目不同，可以分为单继承和多继承。一个类只有一个直接基类时，称为单继承；而一个类同时有多个直接基类时，则称为多继承。从上面的描述可知，任何一个类都可以派生出一个新类，派生类也可以再派生出新类，因此，基类和派生类是相对而言的，一个基类可以是另一个基类的派生类，从而形成了复杂的继承结构，出现了类的层次。基类与派生类之间的关系如下：(1)基类是对派生类的抽象，派生类是对基类的具体化。基类抽取了它的派生类的公共特征，而派生类通过增加信息将抽象的基类变为某种有用的类型，派生类是基类定义的延续。(2)派生类是基类的组合。多继承可以看作是多个单继承的简单组合。(3)公有派生类的对象可以作为基类的对象处理。这一点与类聚集(成员对象)是不同的，在类聚集(成员对象)中，一个类的对象只能拥有作为其成员的其他类的对象，但不能作为其他类对象而使用。 2派生类的定义与构成定义派生类的一般格式如下：`class <派生类名> : <继承方式1> <基类名1> , <继承方式2> <基类名2> , ..... , <继`

承方式n > <基类名n> { <派生类新定义成员> } ; 其中，<基类名>是已有的类的名称，<派生类名>是继承原有类的特性而生成的新类的名称。单继承时，只需定义一个基类；多继承时，需同时定义多个基类。<继承方式>即派生类的访问控制方式，用于控制基类中声明的成员在多大的范围内能被派生类的用户访问。每一个继承方式，只对紧随其后的基类进行限定。继承方式包括3种：公有继承(public)、私有继承(private)和保护继承(protected)。如果不显式地给出继承方式，缺省的类继承方式是私有继承private。派生类新定义成员是指除了从基类继承来的所有成员之外，新增加的数据成员和成员函数。在一个派生类中，其成员由两部分构成：一部分是从基类继承得到的，另一部分是自己定义的新成员，所有这些成员仍然分为公有(public)、私有(private)和保护(protected)三种访问属性。其中，从基类继承下来的全部成员构成派生类的基类部分，这部分的私有成员是派生类不能直接访问的，公有成员和保护成员则是派生类可以直接访问的，但是它们在派生类中的访问属性将随着派生类对基类的继承方式而改变。派生类继承了基类的全部数据成员和除了构造、析构函数之外的全部成员函数，但是这些成员在派生类中的访问属性在派生的过程中是可以调整的，继承方式控制了基类中具有不同访问属性的成员在派生类中的访问属性。基类的成员有公有(public)、保护(protected)和私有(private)三种访问属性，类的继承方式也有公有继承(public)、保护继承(protected)和私有继承(private)三种。不同的继承方式，导致具有不同访问属性的基类成员在派生类中有了新的访问属性。(1)基类中的私有成员在派生类中是隐藏的，只能在基

类内部访问。(2)派生类中的成员不能访问基类中的私有成员，可以访问基类中的公有成员和保护成员。此时派生类对基类中各成员的访问能力与继承方式无关，但继承方式将影响基类成员在派生类中的访问控制属性，基类中公有成员和保护成员在派生类中的访问控制属性将随着继承方式而改变：派生类从基类公有继承时，基类的公有成员和保护成员在派生类中仍然是公有成员和保护成员；派生类从基类私有继承时，基类的公有成员和保护成员在派生类中都改变为私有成员；派生类从基类保护继承时，基类的公有成员在派生类中改变为保护成员，基类的保护成员在派生类中仍为保护成员。

【考点二】派生类的构造函数和析构函数 1.派生类的数据成员由所有基类的数据成员与派生类新增的数据成员共同组成，如果派生类新增成员中包括其他类的对象(成员对象)，派生类的数据成员中实际上还间接包括了这些对象的数据成员。因此，构造派生类的对象时，必须对基类数据成员、新增成员对象的数据成员和新增的其他数据成员进行初始化。派生类的构造函数必须要以合适的初值作为参数，隐含调用基类和新增成员对象的构造函数，用以初始化它们各自的数据成员，然后再对新增的其他数据成员进行初始化。派生类构造函数的一般格式如下：<派生类名>::<派生类名>(<总参数表>):<基类名1>(参数表1),.....<基类名n>(<参数表n>),<成员对象名1>(<参数表n 1>),.....,<成员对象名m>(<参数表n m>){<派生类构造函数体>}派生类的构造函数名与类名相同。在构造函数的参数表中，给出了初始化基类数据、成员对象数据以及新增的其他数据成员所需要的全部参数。在参数表之后，列出需要使用参

数进行初始化的基类名和成员对象名以及各自的参数名，各项之间使用逗号分隔。注意对基类成员和新增成员对象的初始化必须在成员初始化列表中进行。当派生类有多个基类时，处于同一层次各个基类的构造函数的调用顺序取决于定义派生类时声明的顺序(自左向右)，而与在派生类构造函数的成员初始化列表中给出的顺序无关。如果派生类的基类也是一个派生类，则每个派生类只需负责它的直接基类的构造，依次上溯。当派生类中有多个成员对象时，各个成员对象构造函数的调用顺序也取决于在派生类中定义的顺序(自上而下)，而与在派生类构造函数的成员初始化列表中给出的顺序无关。建立派生类对象时，构造函数的执行顺序如下：(1)执行基类的构造函数，调用顺序按照各个基类被继承时声明的顺序(自左向右)；(2)执行成员对象的构造函数，调用顺序按照各个成员对象在类中声明的顺序(自上而下)；(3)执行派生类的构造函数。派生类的构造函数只有在需要的时候才必须定义。派生类构造函数提供了将参数传递给基类构造函数的途径，以保证在基类进行初始化时能够获得必要的的数据。因此，如果基类的构造函数定义了一个或多个参数时，派生类必须定义构造函数。如果基类中定义了缺省构造函数或根本没有定义任何一个构造函数(此时，由编译器自动生成缺省构造函数)时，在派生类构造函数的定义中可以省略对基类构造函数的调用，即省略"<基类名>( <参数表>)"。成员对象的情况与基类相同。当所有的基类和成员对象的构造函数都可以省略，并且也可以不在成员初始化列表中对其他数据成员进行初始化时，可以省略派生类构造函数的成员初始化列表。

2.派生类的析构函数与构造函数相同，派生类的析构函数

在执行过程中也要对基类和成员对象进行操作，但它的执行过程与构造函数严格相反，即：(1)对派生类新增普通成员进行清理。(2)调用成员对象析构函数，对派生类新增的成员对象进行清理。(3)调用基类析构函数，对基类进行清理。派生类析构函数的定义与基类无关，与没有继承关系的类中的析构函数的定义完全相同。它只负责对新增普通成员的清理工作，系统会自己调用基类及成员对象的析构函数进行相应的清理工作。 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](http://www.100test.com)