二级C 多态性：多态性和虚函数 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__E4_BA_8C_E7_BA_A7C___E5_c97_134441.htm 1、 静态联编和动态联编：#8226. 静态联编：联编工作出现在编译连接阶段，这种联编过程在程序开始运行之前完成。 例如：一个静态联编的例子。 #include class Point { public: Point(double I,double j) { x=I.y=j.} double Area(){return 0.0.} private: double x,y. }. class Rectangle :public Point { public: Rectangle(double i,double j,double k,double l). Double Area() {return w*h.} private: double w,h. }. Rectangle::Rectangle(double i,double j ,double k,double l):point(i,j) { w=k.h=l. } void fun(point #8226. 动态联编：有时编译程序在编译阶段，并不能确切知道将要调用的函数，只有在程序执行时才能确定将要调用的函数，为此，要确切知道调用的函数，要求联编工作要在程序运行时进行，这种在程序运行时进行联编工作被称为动态联编。 动态联编是在虚函数的支持下实现的。 所以静态联编和动态联编也都属于多态性，它们是在不同阶段对不同实现进行的选择。 2、 虚函数：（是成员函数，且是非 static 的） #8226. 说明：①如果某类中的成员函数被说明 为虚函数，就意味着该成员函数在派生类中可能有不同的实现。当使用这个成员函数操作 指针或引用 所标识对象时，对该成员函数调用采用动态联编方式，即在运行时进行束定。 amp.s) // 被动态联编 { cout } void main() { Rectangle rec(3.0,5.2,15.0,25.0). Fun(rec). } 输出结果： 375 #8226. 与基类的虚函数有相同的参数个数。 #8226. 其返回值或者与基类虚函数相同，或者都返回指针或引用。 满足上述条件的派生类

的成员函数，自然是虚函数，可以不加 virtual. 例如：分析下列程序输出结果，并回答问题： #include class A { public: virtual void act1(). void act2() { act1(). } }. void A::act1() { cout class B:public A { public: void act1(). } void B::act1() {cout void main() { B b. b.act2(). } 回答问题：①该程序执行后的输出结果是什么？为什么？ 答： B::act1() called. 因为 B 是 A 的派生类， act1() 是 A 类的虚函数 , 类 B 中的 act1() 自然是虚函数。在 main() 函数中， b.act2(), 调用类 B 中的 act2() 函数， B 是派生类，实际上调用 A::act2(), 而 A::act2() 函数的实现中调用 act1(), 由于有两个 act1() 函数，并且是虚函数，产生了动态联编，根据运行情况，选择了 B::act1(). #8226. 如果将 A::act2() 的实现改为： void A::act2() { a::act1(). } 输出结果如何？ 答： A::act1() called. #8226. 只有类的成员函数才能 说明为虚函数。这是因为虚函数仅使用于有继承 关系的类对象，所以普通函数不能说明为虚函数。 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