挑战30天C 入门极限：C 运算符重载转换运算符 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__E6_8C_91_E6_88_9830_E5_A4_c97_134494.htm 为什么需要转换运算符？ 我们大家知道对于内置类型的数据我们可以通过强制转换符的使用来转换数据，例如(int)2.1f.自定义类也是类型，那么自定义类的对象在很多情况下也需要支持此操作，C 提供了转换运算符重载函数,它使得自定义类对象的强转换成为可能。 转换运算符的生命方式比较特别，方法如下： operator 类名(). 转换运算符的重载函数是没有返回类型的，它和类的构造函数，析构函数一样是不遵循函数有返回类型的规定的，他们都没有返回值。 下面我看一个例子，看看它是如何工作的： C 代码 //例１ //程序作者:管宁 //站点:www.cndev-lab.com //所有稿件均有版权,如要转载,请务必著名出处和作者 #include iostream> using namespace std. class Test { public: Test(int a = 0) { coutthis Test::a = a. } Test(Test amp.temp) { cout Test::a = temp.a. } ~Test() { coutthisthis->a cin.get(). } operator int() { coutthisthis->a return Test::a. } public: int a. }. int main() { Test a(100),b(100),c. cout| b的内存地址" c=Test((int)a (int)b).//显示式转换 //c=a b.//隐式转换 cout cout system("pause"). } 代码中的c=a b.属于隐式转换，它的实现过程与c=Test((int)a (int)b).完全相同。 运行结果如下图示（注意观察内存地址，观察构造与析构过程，执行过程图中有解释）： 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