|  |  |
| --- | --- |
| 1. 阻止拷贝 2. 定义删除的函数 3. 析构函数不能是删除的成员 4. 合成的拷贝控制成员可能是删除的 5. private控制拷贝 | Nocopy(const Nocopy&)=delete;  Private:  Nocopy(const Nocopy&); |
| 1. 阻止拷贝 2. 对某些类来说拷贝赋值和拷贝构造没有合理的意义，必须采用某种机制阻止 3. 如：iostream不能拷贝,若采取不定义方式是不可的，因为编译器会合成一个 4. 定义删除的函数 5. 删除的函数：虽然声明了他们，但不能以任何方式使用它们 6. 在函数参数列表后面加上=delete指明 7. =delete必须出现在第一次声明地方 8. 适合所有函数，不同于default只适合拷贝控制 9. 当我们希望引导函数匹配过程时，删除函数有时也是很有用的 10. 析构函数不能是删除的成员 11. 只是规定，非要定义删除的话，不允许定义该类型的变量或创建该类的临时对象 12. 但是可以动态分配，但是delete不掉 13. 合成的拷贝控制成员可能是删除的 14. 编译器可能会将这些合成的拷贝控制成员定义为删除的   1. 若类的某个成员的析构函数是删除的或者不可访问，这个类的析构函数被定义删除  2. 若类的某个成员的拷贝构造函数或者析构函数是删除的或者不可访问，这个类的拷贝构造函数被定义删除  3. 若类的某个成员的拷贝赋值运算符是删除的或者不可访问，或者有一个const或引用成员，则这个类合成的拷贝赋值运算符被定义删除的  4. 若类的某个成员的析构函数是删除的或者不可访问，或类有一个引用成员它没有类内初始化器，或者const成员 它没有类内初始化器(在类内指定默认值)且其类型未显示定义默认构造函数则该类的默认构造函数被定义为删除的  note：如果类的某个成员不能构造、拷贝 复制或销毁，则对应的成员函数被定义为删除   1. private控制拷贝 2. 新标准之前，通过将拷贝构造 赋值声明为private来阻止拷贝 3. 友元和成员函数仍旧可以拷贝对象，可以通过声明不定义方式，来让链接时错误 4. 尽量用delete，而不是private | |