|  |  |
| --- | --- |
| 1. 析构函数 2. 析构函数完成什么工作 3. 什么时候会调用析构函数 4. 合成析构函数 | ~Foo();  Int \*p = new int(3);  Delete p;  ~Sales\_data(){} |
| 1. 析构函数 2. 析构函数释放对象使用的资源，并且销毁对象的非static数据成员 3. 是一个成员函数，由波浪号接类名构成，没有返回值，不接受参数 4. 析构函数不接受参数，所以不能被重载。一个类只有唯一的析构函数 5. 析构函数完成什么工作 6. 首先执行函数体，然后销毁成员，是初始化顺序的逆序销毁的 7. 析构函数体可以执行类设计者希望执行的任何收尾工作。通常，是释放对象在生存期分配的所有资源 8. 不存在初始化列表的东西控制成员如何销毁，析构部分是隐式的 9. 完成依赖于成员的类型，类类型的成员需要执行成员自己的析构函数，内置类型没有析构函数，所以说都不需做   Note: 所以需要显示销毁一个指针，智能指针是类类型，有析构函数，会自动析构   1. 什么时候会调用析构函数 2. 离开作用域 3. 对象被销毁，其成员被销毁 4. 容器被销毁，其元素被销毁 5. 动态分配的对象，delete时 6. 临时对象，表达式结束   由于1）所以可以定义普通对象，但是如果是动态分配的对象，需要显示delete,才会执行析构函数  Note:当指向一个对象的引用或指针离开作用域时，不会自动执行析构   1. 合成析构函数 2. 析构函数被用来阻止该类型的对象被销毁？不是这种情况，函数体为空 3. 空析构函数体执行完成后，成员会被自动销毁。如：执行完析构函数体，执行stirng类型的析构函数销毁bookno 4. 析构函数体自身并不直接销毁成员，是在析构函数体之后隐含的析构阶段中被销毁的 | |