|  |  |
| --- | --- |
| 1. 成员指针 2. 数据成员指针 3. 使用数据成员指针 4. 返回数据成员指针的函数 | 成员指针  Const string Screen::\*pdata  Auto pada = &Screen::contents;  Screen my,\*pScreen = &myScreen  Auto s = myScree.\*pdata  S = pScreen->\*pdata  (.\*),(->\*) |
| 1. 成员指针 2. 指向类的非静态成员的指针。 3. 成员指针的类型囊括了类的类型以及成员的类型。但是不指定该成员所属的对象，使用时才提供所指的对象 4. 数据成员指针 5. Const string Screen::\*pdata,一个指向screen类的const string 成员的指针。const string可以让指针指向任何screen对象的一个成员，但不能写入内容 6. 向成员指针赋值，指定所指的成员   Pdata = &screen::contents;  3) 最简单的方法  Auto pdata = &Screen::contents;   1. 使用数据成员指针   提供对象的信息  我们以上操作知识指定了类的成员，但是不是该类的对象的成员，所以当解引用成员指针时我们才提供对象的信息。   1. .\*和->\*,可以解引用指针并获得该对象的成员   Auto s = myScreen.\*pdata;  S = pScreen->\*pdata   1. 返回数据成员指针的函数 2. 就是当成员content是私有时，不能直接访问生成一个指向数据成员的指针，可以提供一个public的方法返回一个指向screen类的conststring成员的指针   Static const std::string screen::\*data(){return &Screen::contents;}  Const string screen::\*pdata = screen::data();  Auto s = myScreen.\*pdta; | |