

C++ 中智能指针的特点

1. 为什么要使用智能指针？ **智能指针是类，当超出类的作用域时，类会自动调用析构函数，析构函数会自动释放资源，很大程度上避免了忘记释放资源带来的内存泄漏**
2. `auto_ptr` (已弃用)： **问题是两个指针指向一个内存，会两次释放同一资源**
3. `unique_ptr`： **`unique_ptr`指针规定一个智能指针独占一块内存资源。当两个智能指针同时指向一块内存，编译报错。**
 1. 实现原理：将拷贝构造函数和赋值拷贝构造函数申明为`private`或`delete`。不允许拷贝构造函数和赋值操作符，但是支持移动构造函数，通过`std::move`把一个对象指针变成右值之后可以移动给另一个`unique_ptr`
4. `shared_ptr`： **共享指针可以实现多个智能指针指向相同对象，该对象和其相关资源会在引用为0时被销毁释放。**
 1. 实现原理：有一个引用计数的指针类型变量，专门用于引用计数，使用拷贝构造函数和赋值拷贝构造函数时，引用计数加1，当引用计数为0时，释放资源
5. `weak_ptr`： **当两个`shared_ptr`相互指着对方时，引用计数不会为0，所以引用`weak_ptr`，`weak_ptr`是弱引用，`weak_ptr`的构造和析构不会引起引用计数的增加或减少**