|  |  |
| --- | --- |
| 1. 效率与灵活性 2. 在运行时绑定删除器 3. 在编译时绑定删除器 | Shared\_ptr:del?del(p):delete p;  Unique\_ptr:del(p) |
| 1. 效率与灵活性 2. Shared\_ptr:创建一个可创建对象即可 3. Unique\_ptr:删除器的类型是一个unique\_ptr对象的类型一部分 4. 在运行时绑定删除器 5. Shared\_ptr:不是一个成员，是在运行时才知道，del(p)需要一次运行时跳转操作，并且可以用reset更换，shared\_ptr类成员的类型在运行时是不能改变的，所以不能直接保存删除器 6. 在编译时绑定删除器 7. 删除器类型是类类型的一部分，在编译时是知道的，没有运行时开销，更方便 | |