

条款14 如果函数不抛出异常,请使用 `noexcept`

① C++11中释放内存函数和析构函数都是隐式 `noexcept`,除非显式声明 `noexcept(false)`,如果一个对象的析构函数可能被标准库使用,且析构函数又可能抛出异常,那么程序的行为是未定义的.

② `std::vector` 添加新元素如果容量不够,那么 `vector` 会分配一个新的更大的内存用于存储元素,并且之前 `vector` 中的数据也一并复制或移动到新位置,但是如果移动操作为非 `noexcept`,则 `vector` 不敢使用移动操作.

一旦抛出异常,原有 `vector` 的数据已丢失,造成同一块连续内存空间割裂地分布在两个 `vector` 手中