

MoreEffectiveC : 防止资源泄漏 PDF转换可能丢失图片或格式  
， 建议阅读原文

[https://www.100test.com/kao\\_ti2020/257/2021\\_2022\\_MoreEffect\\_c67\\_257594.htm](https://www.100test.com/kao_ti2020/257/2021_2022_MoreEffect_c67_257594.htm) 如果你正在开发一个具有多媒体功能的通讯录程序。这个通讯录除了能存储通常的文字信息如姓名、地址、电话号码外，还能存储照片和声音(可以给出他们名字的正确发音)。为了实现这个通信录，你可以这样设计:

```
class Image
{ // 用于图像数据 public: Image(const string&
audioDataFileName). ... }.class PhoneNumber { ... }. // 用于存储电
话号码class BookEntry { // 通讯录中的条目 public:
BookEntry(const string& address = "", const string&
audioClipFileName = ""). ~BookEntry(). // 通过这个函数加入电
话号码 void addPhoneNumber(const PhoneNumber&
name,const string& imageFileName, Const string& name,
const string& imageFileName, const string&
audioClipFileName) : theName(name), theAddress(address),
theImage(0), theAudioClip(0){ try { // 这try block是新加入的 if
(imageFileName != "") { theImage = new Image(imageFileName). }
if (audioClipFileName != "") { theAudioClip = new
AudioClip(audioClipFileName). }}catch (...) { // 捕获所有异常
0delete theImage. // 完成必要的清除代码 0delete theAudioClip.
throw. // 继续传递异常}} 不用为BookEntry中的非指针数据成
员操心，在类的构造函数被调用之前数据成员就被自动地初
始化。所以如果BookEntry构造函数体开始执行，对象
的theName, theAddress 和 thePhones数据成员已经被完全构造
好了。这些数据可以被看做是完全构造的对象，所以它们将
```

被自动释放，不用你介入操作。当然如果这些对象的构造函数调用可能会抛出异常的函数，那么哪些构造函数必须去考虑捕获异常，在允许它们继续传递之前完成必需的清除操作。100Test 下载频道开通，各类考试题目直接下载。详细请访问 [www.100test.com](http://www.100test.com)