More Effective C++ 阅读笔记

1. **基础议题**
2. **仔细区别pointers和references**
3. **最好使用C++转型操作符**
4. **绝对不要以多态(polymorphically)方式处理数组**
5. **非必要不提供default constructor**
6. **操作符**
7. **对定制的“类型转换函数”保持警觉**
8. **区别increment/decrement操作符的前置和后置形式**
9. **千万不要重载&&, ||和,操作符**
10. **了解各种不同意义的new和delete**
11. **异常**
12. **利用destructors避免泄露资源**
13. **在constructors内阻止资源泄露**
14. **禁止异常流出destructors之外**
15. **了解“抛出一个exception”与“传递一个参数”或“调用一个虚函数”之间的差异**
16. **以by reference方式捕捉exceptions**
17. **明智运用exception specifications**
18. **了解异常处理的成本**
19. **效率**
20. **谨记80-20法则**
21. **考虑使用lazy evaluation**
22. **分期摊还预期的计算成本**
23. **了解临时对象的来源**
24. **协助完成“返回值优化（RVO）”**
25. **利用重载技术避免隐式类型转换**
26. **考虑以操作符复合形式(op=)取代其独身形式(op)**
27. **考虑使用其他程序库**
28. **了解virtual functions, multiple inheritance, virtual base classes, runtime type identification的成本**
29. **技术**
30. **将constructor和non-member functions虚化**
31. **限制某个class所能产生的对象数量**
32. **要求（或禁止）对象产生于heap之中**
33. **Smart Pointers（智能指针）**
34. **Reference counting（引用计数）**
35. **Proxy classes（替身类、代理类）**
36. **让函数根据一个以上的对象类型来决定如何虚化**
37. **杂项讨论**
38. **在未来时态下发展程序**
39. **将非尾端类设计为抽象类**
40. **如何在同一个程序中结合C++和C**
41. **让自己习惯于标准C++语言**