More Effective C++读书笔记

**【条款1】仔细区别pointers和references**

pointers和references的区别：

相同点：它们都使得你可以**间接参考其他对象**

不同点：

**1、有null pointer，没有null reference。**

null pointer是值为NULL的指针，即空指针，它不指向任何对象，因此不能对null pointer进行解引用，在对指针进行操作前需要判断该指针是否是空指针。

没有null reference，因此reference在定义时就必须引用到一个已存在的对象（即reference在定义时就必须进行初始化）。

**2、可以改变pointer的值，不能让reference去指向另一个对象**

改变pointer的值即是让pointer去指向另一个对象。

reference总是指向（代表）它最初获得的那个对象，一旦代表了某对象，就不能再改变。

什么时候使用pointers或references？

**当你需要指向（代表）某个已存在的对象，而且绝不会改变指向其他对象，或是你实现一个操作而其语法需求无法由pointers达成（比如operator[]），你就应该选用references。其他任何时候，请采用pointer。**

**【条款2】最好使用C++转型操作符**

C++引入了4个新的转型操作符：

(1)**static\_cast**

static\_cast是C++中适用范围最广的转型操作符，其行为类似于C旧式转型，以及相同的限制。

(2)**const\_cast**

只能用于移除对象的常量性。

(3)**dynamic\_cast**

在继承体系中执行安全的向下转型，即**把指向基类对象的指针或引用转型为指向派生类对象的指针或引用。如果转型对象是指针，当转型失败时会返回NULL；如果转型对象是引用，当转型失败时会抛出异常。**

注意，**dynamic\_cast只能用于有虚函数的类型身上**。

(4)**reinterpret\_cast**

如何选择各种新转型操作符？

如果你想为一个不涉及继承机制的类型执行转型动作，请用static\_cast

如果你仅仅是想移除一个对象的常量性，请用const\_cast