Longxiajun30天读书笔记分享计划
Day 23: C++

2018.12.29 Saturday. By KuKuXia@github.com

EC条款38： 通过复合塑模出has-a或“根据某物实现出”

复合指某种类型的对象包含其他类型的对象

在引用域application domain，复合意味着has-a，有一个的关系

在实现域implementation domain，复合意味着is-implemented-in-terms-of，根据某物实现出

EC条款39： 明智而审慎地使用private继承

private继承意味着implemented-in-terms-of，根据某物实现出

尽可能使用复合，必要时才使用private继承

private通常比复合的级别低，但是当派生类需要访问protected基类的成员，或需要重新定义继承而来的virtual函数时，这么设计是合理的

和复合不同，private继承可以造成empty base最优化，这对致力于对象尺寸最小化的程序库开发者而言，可能很重要

C++ STL

binary_search

底层实现：调用lower_bound获取不小于value的迭代器，然后判断不等于last并且其值比*first大进行

bool binary_search(ForwardIt first, ForwardIt last, const T& value)

bool binary_search(ForwardIt first, ForwardIt last, const T& value, Compare comp)

检查等价于 value 的元素是否出现于范围 [first, last) 中

完全排序的范围满足这些判别标准
第一版本用 operator< 比较元素，第二版本用给定的比较函数 comp

equal_range

std::pair<ForwardIt,ForwardIt> equal_range(ForwardIt first, ForwardIt last,const T& value)

std::pair<ForwardIt,ForwardIt> equal_range(ForwardIt first, ForwardIt last,const T& value, Compare comp)

返回范围 [first, last) 中含有所有等价于 value 的元素的范围

完全排序的范围满足这些判别标准
以二个迭代器定义返回的范围，一个指向首个不小于 value 的元素，而另一个指向首个大于 value 的元素
第一版本用 operator< 比较元素，第二版本用给定的比较函数 comp

底层实现：调用lower_bound 与upper_bound

C++ STL

equal

底层实现：线性迭代判断

bool equal(InputIt1 first1, InputIt1 last1, InputIt2 first2)

bool equal(InputIt1 first1, InputIt1 last1, InputIt2 first2, BinaryPredicate p)

如果范围 [first1, last1) 和范围 [first2, first2 + (last1 - first1) 相等，返回 true ，否则返回 false

两个范围相等的条件是：对于范围 [first1, last1) 内的每个迭代器 i ， *i 等于 *(first2 + (i - first1))

重载形式 (1,2,5,6) 用 operator== 判定两个元素是否相等，而重载形式 (3,4,7,8) 用给定的谓词函数

利用equal实现回文字符串的检测，完美~

```
if(std::equal(s.begin(), s.begin() + s.size()/2, s.rbegin())) {
    std::cout << "\"" << s << "\" is a palindrome\n";
} else {
    std::cout << "\"" << s << "\" is not a palindrome\n";
}
```

lexicographical_compare

bool lexicographical_compare(InputIt1 first1, InputIt1 last1,InputIt2 first2, InputIt2 last2)

bool lexicographical_compare(InputIt1 first1, InputIt1 last1, InputIt2 first2, InputIt2 last2,Compare comp)

检查第一个范围 [first1, last1) 是否按字典序小于第二个范围 [first2, last2)
1. 用 operator< 比较元素。
2. 用给定的二元比较函数 comp 比较函数

字典序比较是拥有下列属性的操作：

逐元素比较二个范围。

首个不匹配元素定义范围是否按字典序小于或大于另一个。

若一个范围是另一个的前缀，则较短的范围小于另一个。

若二个范围拥有等价元素和相同长度，则范围按字典序相等。

空范围按字典序小于任何非空范围。

二个空范围按字典序相等。

底层实现：线性迭代判断