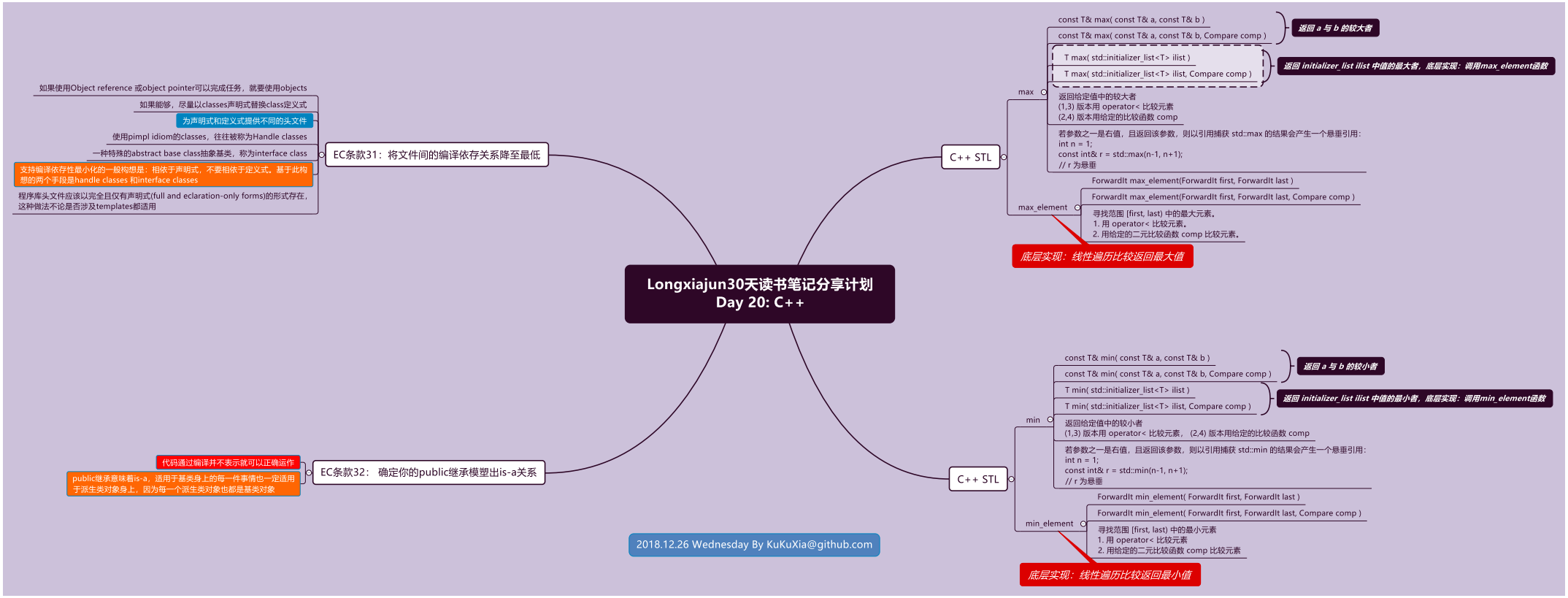
Longxiajun30天读书笔记分享计划
Day 20: C++
2018.12.26 Wednesday By KuKuXia@github.com
EC条款31：将文件间的编译依存关系降至最低
如果使用Object reference 或object pointer可以完成任务，就要使用objects
如果能够，尽量以classes声明式替换class定义式
为声明式和定义式提供不同的头文件
使用pimpl idiom的classes，往往被称为Handle classes
一种特殊的abstract base class抽象基类，称为interface class
支持编译依存性最小化的一般构想是：相依于声明式，不要相依于定义式。基于此构想的两个手段是handle classes 和interface classes
程序库头文件应该以完全且仅有声明式(full and eclaration-only forms)的形式存在，这种做法不论是否涉及templates都适用
EC条款32： 确定你的public继承模塑出is-a关系
代码通过编译并不表示就可以正确运作
public继承意味着is-a，适用于基类身上的每一件事情也一定适用于派生类对象身上，因为每一个派生类对象也都是基类对象
C++ STL
max
const T& max(const T& a, const T& b)
const T& max(const T& a, const T& b, Compare comp)
返回 a 与 b 的较大者
T max(std::initializer_list<T> ilist)
T max(std::initializer_list<T> ilist, Compare comp)
返回 initializer_list ilist 中值的最大者，底层实现：调用max_element函数
返回给定值中的较大者
(1,3) 版本用 operator< 比较元素
(2,4) 版本用给定的比较函数 comp
若参数之一是右值，且返回该参数，则以引用捕获 std::max 的结果会产生一个悬垂引用:
int n = 1;
const int& r = std::max(n-1, n+1);
// r 为悬垂
max_element
ForwardIt max_element(ForwardIt first, ForwardIt last)
ForwardIt max_element(ForwardIt first, ForwardIt last, Compare comp)
寻找范围 [first, last) 中的最大元素。
1. 用 operator< 比较元素。
2. 用给定的二元比较函数 comp 比较元素。
底层实现：线性遍历比较返回最大值
C++ STL
min
const T& min(const T& a, const T& b)
const T& min(const T& a, const T& b, Compare comp)
返回 a 与 b 的较小者
T min(std::initializer_list<T> ilist)
T min(std::initializer_list<T> ilist, Compare comp)
返回 initializer_list ilist 中值的最小者，底层实现：调用min_element函数
返回给定值中的较小者
(1,3) 版本用 operator< 比较元素， (2,4) 版本用给定的比较函数 comp
若参数之一是右值，且返回该参数，则以引用捕获 std::min 的结果会产生一个悬垂引用:
int n = 1;
const int& r = std::min(n-1, n+1);
// r 为悬垂
min_element
ForwardIt min_element(ForwardIt first, ForwardIt last)
ForwardIt min_element(ForwardIt first, ForwardIt last, Compare comp)
寻找范围 [first, last) 中的最小元素
1. 用 operator< 比较元素
2. 用给定的二元比较函数 comp 比较元素
底层实现：线性遍历比较返回最小值