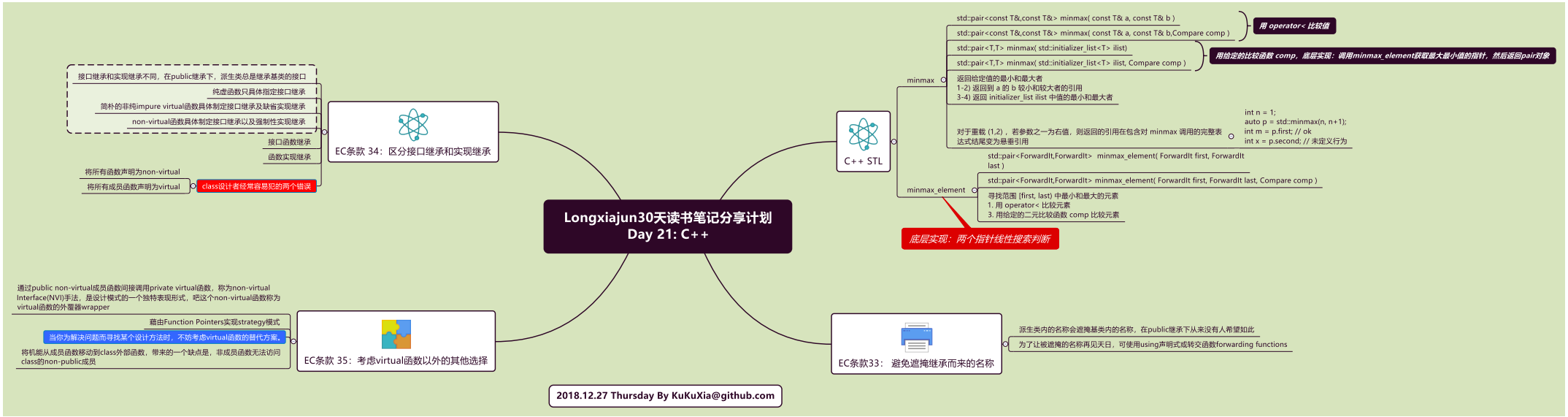
Longxiajun30天读书笔记分享计划
Day 21: C++
EC条款 34：区分接口继承和实现继承
接口继承和实现继承不同，在public继承下，派生类总是继承基类的接口
纯虚函数只具体指定接口继承
简朴的非纯impure virtual函数具体制定接口继承及缺省实现继承
non-virtual函数具体制定接口继承以及强制性实现继承
接口函数继承
函数实现继承
将所有函数声明为non-virtual
将所有成员函数声明为virtual
class设计者经常容易犯的两个错误
C++ STL
minmax
std::pair<const T&,const T&> minmax(const T& a, const T& b)
std::pair<const T&,const T&> minmax(const T& a, const T& b,Compare comp)
用 operator< 比较值
std::pair<T,T> minmax(std::initializer_list<T> ilist)
std::pair<T,T> minmax(std::initializer_list<T> ilist, Compare comp)
用给定的比较函数 comp，底层实现：调用minmax_element获取最大最小值的指针，然后返回pair对象
返回给定值的最小和最大者
1-2) 返回到 a 的 b 较小和较大者的引用
3-4) 返回 initializer_list ilist 中值的最小和最大者
对于重载 (1,2)，若参数之一为右值，则返回的引用在包含对 minmax 调用的完整表达式结尾变为悬垂引用
int n = 1;
auto p = std::minmax(n, n+1);
int m = p.first; // ok
int x = p.second; // 未定义行为
minmax_element
std::pair<ForwardIt,ForwardIt> minmax_element(ForwardIt first, ForwardIt last)
std::pair<ForwardIt,ForwardIt> minmax_element(ForwardIt first, ForwardIt last, Compare comp)
寻找范围 [first, last) 中最小和最大的元素
1. 用 operator< 比较元素
3. 用给定的二元比较函数 comp 比较元素
底层实现：两个指针线性搜索判断
EC条款 35：考虑virtual函数以外的其他选择
通过public non-virtual成员函数间接调用private virtual函数，称为non-virtual Interface(NVI)手法，是设计模式的一个独特表现形式，把这个non-virtual函数称为virtual函数的外覆器wrapper
藉由Function Pointers实现strategy模式
当你为解决问题而寻找某个设计方法时，不妨考虑virtual函数的替代方案。
将机能从成员函数移动到class外部函数，带来的一个缺点是，非成员函数无法访问class的non-public成员
EC条款33：避免遮掩继承而来的名称
派生类内的名称会遮掩基类内的名称，在public继承下从来没有人希望如此
为了让被遮掩的名称再见天日，可使用using声明式或转交函数forwarding functions
2018.12.27 Thursday By KuKuXia@github.com