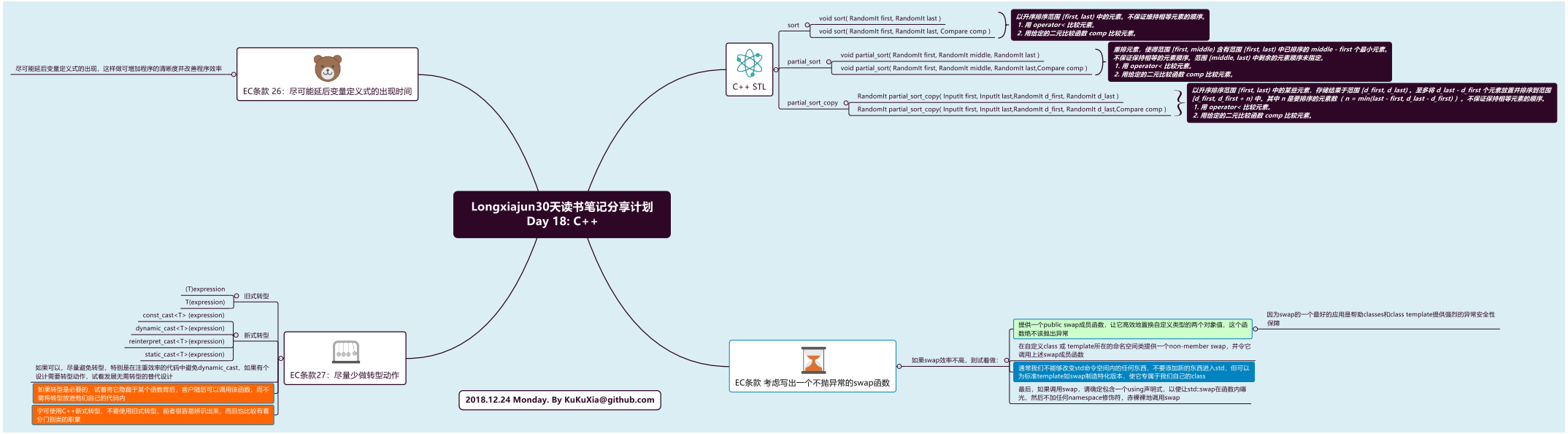
Longxiajun30天读书笔记分享计划
Day 18: C++
2018.12.24 Monday. By KuKuXia@github.com
EC条款 26：尽可能延后变量定义式的出现时间
尽可能延后变量定义式的出现，这样做可增加程序的清晰度并改善程序效率
C++ STL
sort
void sort(RandomIt first, RandomIt last)
void sort(RandomIt first, RandomIt last, Compare comp)
以升序排序范围 [first, last) 中的元素。不保证维持相等元素的顺序。
1.用 operator< 比较元素。
2.用给定的二元比较函数 comp 比较元素。
partial_sort
void partial_sort(RandomIt first, RandomIt middle, RandomIt last)
void partial_sort(RandomIt first, RandomIt middle, RandomIt last,Compare comp)
重排元素，使得范围 [first, middle) 含有范围 [first, last) 中已排序的 middle - first 个最小元素。不保证保持相等的元素顺序。范围 [middle, last) 中剩余的元素顺序未指定。
1.用 operator< 比较元素。
2.用给定的二元比较函数 comp 比较元素。
partial_sort_copy
RandomIt partial_sort_copy(InputIt first, InputIt last,RandomIt d_first, RandomIt d_last)
RandomIt partial_sort_copy(InputIt first, InputIt last,RandomIt d_first, RandomIt d_last,Compare comp)
以升序排序范围 [first, last) 中的某些元素，存储结果于范围 [d_first, d_last)，至多将 d_last - d_first 个元素放置并排序到范围 [d_first, d_first + n) 中。其中 n 是要排序的元素数（ n = min(last - first, d_last - d_first) ）。不保证保持相等元素的顺序。
1.用 operator< 比较元素。
2.用给定的二元比较函数 comp 比较元素。
EC条款27：尽量少做转型动作
旧式转型
(T)expression
T(expression)
新式转型
const_cast<T> (expression)
dynamic_cast<T>(expression)
reinterpret_cast<T>(expression)
static_cast<T>(expression)
如果可以，尽量避免转型，特别是在注重效率的代码中避免dynamic_cast，如果有个设计需要转型动作，试着发展无需转型的替代设计
如果转型是必要的，试着将它隐藏于某个函数背后，客户随后可以调用该函数，而不需将转型放进他们自己的代码内
宁可使用C++新式转型，不要使用旧式转型，前者很容易辨识出来，而且也比较有着分门别类的职掌
EC条款 考虑写出一个不抛异常的swap函数
如果swap效率不高，则试着做：
提供一个public swap成员函数，让它高效地置换自定义类型的两个对象值，这个函数绝不该抛出异常
因为swap的一个最好的应用是帮助classes和class template提供强烈的异常安全性保障
在自定义class 或 template所在的命名空间类提供一个non-member swap，并令它调用上述swap成员函数
通常我们不能够改变std命令空间内的任何东西，不要添加新的东西进入std，但可以为标准template如swap制造特化版本，使它专属于我们自己的class
最后，如果调用swap，请确定包含一个using声明式，以便让std::swap在函数内曝光，然后不加任何namespace修饰符，赤裸裸地调用swap