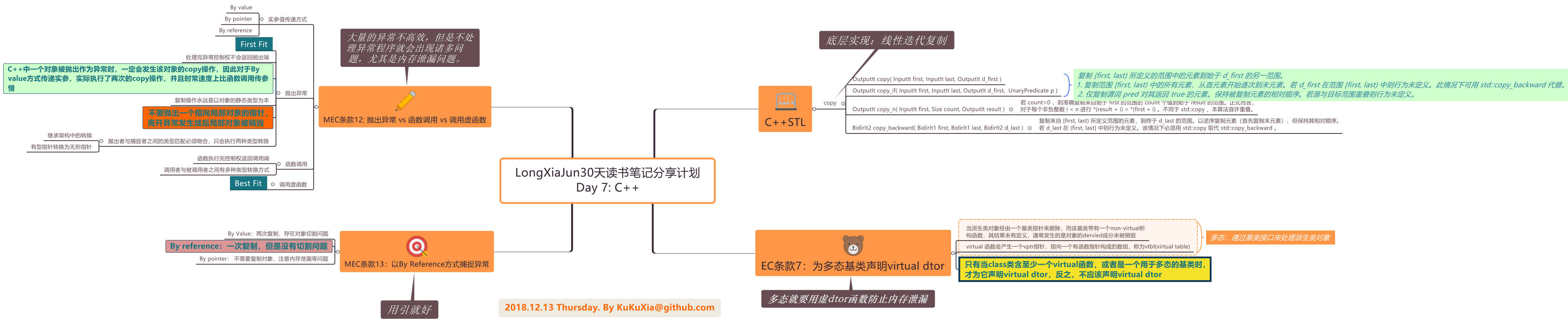

LongXiaJun30天读书笔记分享计划
Day 7: C++
MEC条款12: 抛出异常 vs 函数调用 vs 调用虚函数
大量的异常不高效，但是不处理异常程序就会出现诸多问题，尤其是内存泄漏问题。
实参值传递方式
By value
By pointer
By reference
抛出异常
First Fit
处理完异常控制权不会返回抛出端
C++中一个对象被抛出作为异常时，一定会发生该对象的copy操作，因此对于By value方式传递实参，实际执行了两次的copy操作，并且时常速度上比函数调用传参慢
复制操作永远是以对象的静态类型为本
不要抛出一个指向局部对象的指针，离开异常发生域后局部对象被销毁
抛出者与捕捉者之间的类型匹配必须吻合，只会执行两种类型转换
继承架构中的转换
有型指针转换为无形指针
函数调用
函数执行完控制权返回调用端
调用者与被调用者之间有多种类型转换方式
调用虚函数
Best Fit
MEC条款13：以By Reference方式捕捉异常
By Value：两次复制，存在对象切割问题
By reference：一次复制，但是没有切割问题
By pointer： 不需要复制对象，注意内存泄漏等问题
用引就好
C++STL
底层实现：线性迭代复制
copy
OutputIt copy(InputIt first, InputIt last, OutputIt d_first)
OutputIt copy_if(InputIt first, InputIt last, OutputIt d_first, UnaryPredicate p)
复制 [first, last) 所定义的范围中的元素到始于 d_first 的另一范围。
1. 复制范围 [first, last) 中的所有元素，从首元素开始逐次到末元素。若 d_first 在范围 [first, last) 中则行为未定义。此情况下可用 std::copy_backward 代替。
2. 仅复制谓词 pred 对其返回 true 的元素。保持被复制元素的相对顺序。若源与目标范围重叠则行为未定义。
OutputIt copy_n(InputIt first, Size count, OutputIt result)
若 count>0 ，则准确复制来自始于 first 的范围的 count 个值到始于 result 的范围。正式而言，
对于每个非负整数 i < n 进行 *(result + i) = *(first + i) 。不同于 std::copy ，本算法容许重叠。
BidirIt2 copy_backward(BidirIt1 first, BidirIt1 last, BidirIt2 d_last)
复制来自 [first, last) 所定义范围的元素，到终于 d_last 的范围。以逆序复制元素（首先复制末元素），但保持其相对顺序。
若 d_last 在 (first, last] 中则行为未定义。该情况下必须用 std::copy 取代 std::copy_backward 。
EC条款7：为多态基类声明virtual dtor
当派生类对象经由一个基类指针来删除，而该基类带有一个non-virtual析构函数，其结果未有定义，通常发生的是对象的dervied成分未被销毁
virtual 函数会产生一个vptr指针，指向一个有函数指针构成的数组，称为vtbl(virtual table)
多态：通过基类接口来处理派生类对象
只有当class类含至少一个virtual函数，或者是一个用于多态的基类时，才为它声明virtual dtor，反之，不应该声明virtual dtor
多态就要用虚dtor函数防止内存泄漏
2018.12.13 Thursday. By KuKuXia@github.com