

Longxiajun30天读书笔记分享计划
Day 22: C++

EC条款36: 绝不重新定义继承而来的non-virtual函数



绝对不要重新定义继承而来的non-virtual函数

EC条款37: 绝不重新定义继承而来的缺省函数值



- virtual函数系动态绑定, 而缺省函数值却是静态绑定
- 对象的静态类型, 就是它在程序中被声明时所采用的类型
- 对象动态类型是指目前所指对象的类型
- 动态类型可以再程序执行过程中改变



C++ STL

- is_permutation
 - bool is_permutation(ForwardIt1 first1, ForwardIt1 last1, ForwardIt2 first2)
 - bool is_permutation(ForwardIt1 first1, ForwardIt1 last1, ForwardIt2 first2, BinaryPredicate p)
 - bool is_permutation(ForwardIt1 first1, ForwardIt1 last1, ForwardIt2 first2, ForwardIt2 last2, BinaryPredicate p)
 - C++11 起
 - C++20 前
 - 若存在范围 [first1, last1) 中元素的排列, 使得该范围等于 [first2, last2), 则返回 true, 若不给定, 则其中 last2 代表 first2 + (last1 - first1)
 - 1,3) 用 operator== 比较元素。若它不是等价关系则行为未定义
 - 2,4) 用给定的二元谓词 p 比较元素。若它不是等价关系则行为未定义
- next_permutation
 - bool next_permutation(BidirIt first, BidirIt last)
 - bool next_permutation(BidirIt first, BidirIt last, Compare comp)
 - 变换范围 [first, last) 为来自所有按相对于 operator< 或 comp 的字典序的下个排列。若这种排列存在则返回 true, 否则变换范围为首个排列 (如同用 std::sort(first, last)) 并返回 false
- prev_permutation
 - bool prev_permutation(BidirIt first, BidirIt last)
 - bool prev_permutation(BidirIt first, BidirIt last, Compare comp)
 - 变换范围 [first, last) 为来自于相对于 operator< 或 comp 的字典序的所有排列集合的上个排列。若这种排列存在则返回 true, 否则变换范围为未排列 (如同用 std::sort(first, last); std::reverse(first, last);) 并返回 false



C++ STL

- lower_bound
 - ForwardIt lower_bound(ForwardIt first, ForwardIt last, const T& value)
 - ForwardIt lower_bound(ForwardIt first, ForwardIt last, const T& value, Compare comp)
 - 返回指向范围 [first, last) 中首个不小于 (即大于或等于) value 的元素的迭代器, 或若找不到这种元素则返回 last
 - 1. 用 operator< 比较元素
 - 2. 用给定的比较函数 comp
- upper_bound
 - ForwardIt upper_bound(ForwardIt first, ForwardIt last, const T& value)
 - ForwardIt upper_bound(ForwardIt first, ForwardIt last, const T& value, Compare comp)
 - 返回指向范围 [first, last) 中首个大于 value 的元素的迭代器, 或若找不到这种元素则返回 last
 - 1. 用 operator< 比较元素
 - 2. 用给定的比较函数 comp