

Longxiajun30天读书笔记分享计划  
Day 26: C++

EC条款44: 将于参数无关的代码抽离template



Templates生成多个classes和多个函数。所以任何template代码都不该与某个造成膨胀的template参数产生相依关系

因非类型模板参数non-type template parameters而造成的代码膨胀，往往可消除，做法是以函数参数或class成员变量替换template参数

因类型参数 type parameters而造成的代码膨胀，往往可降低，做法是让带有完全相同二进制表述binary representations的具现类型instantiation types共享实现码

EC条款45: 运动成员函数目标接受所有兼容类型



请使用member function templates成员函数模板生成“可接受所有兼容类型”的函数

如果你声明member template用于“泛化copy构造”或“泛化“assignment”操作，你还是需要声明正常的copy构造函数和copy assignment操作符

2019.1.1 Tuesday By KuKuXia@github.com



底层实现：先线性遍历找到不满足p的迭代器，然后继续判断后面的元素是否，满足p，如果是就返回false，不是则返回true

- is\_partitioned
  - bool is\_partitioned( InputIt first, InputIt last, UnaryPredicate p )
  - 若范围 [first, last) 中的所有满足 p 的元素都出现在所有不满足的元素前则返回 true。若 [first, last) 为空亦返回 true
  - ForwardIt partition( ForwardIt first, ForwardIt last, UnaryPredicate p )
- partition
  - 重排序范围 [first, last) 中的元素，使得谓词 p 对其返回 true 的元素前于谓词 p 对其返回 false 的元素。不保持相对顺序

快速排序的实现

底层实现：先找到不满足p的迭代器，然后用std::next()记录下一个迭代器并遍历后面的元素，进而用两个迭代器交换元素即可



底层实现：循环迭代判断赋值到两个容器中

- partition\_copy
  - std::pair<OutputIt1, OutputIt2> partition\_copy( InputIt first, InputIt last, OutputIt1 d\_first\_true, OutputIt2 d\_first\_false, UnaryPredicate p )
  - 从范围 [first, last) 复制元素到二个不同范围，取决于谓词 p 的返回值。复制满足谓词 p 的元素到始于 d\_first\_true 的范围。复制剩余元素到始于 d\_first\_false 的范围
- stable\_partition
  - BidirIt stable\_partition( BidirIt first, BidirIt last, UnaryPredicate p )
  - 重排序范围 [first, last) 中的元素，使得所有谓词 p 对其返回 true 的元素先于谓词 p 对其返回 false 的元素。保持元素的相对顺序
- partition\_point
  - ForwardIt partition\_point( ForwardIt first, ForwardIt last, UnaryPredicate p )
  - 检验（如同用 std::partition 的）划分范围 [first, last)，并定位第一划分的结尾，即首个不满足 p 的元素，或者所有元素满足 p 则为 last。