

C++标准程序库读书笔记

STL迭代器

逆向迭代器需要专门的头文件，但是由于容器必须要定义它自己的逆向迭代器型别，所以容器本身已经包含了该头文件。

迭代器奉行一个纯抽象概念：任何东西，只要行为类似迭代器，就是一个迭代器。不同的迭代器具有不同的能力（行进和存取能力）。

迭代器分类及其能力

迭代器类型	能力	供应者
input迭代器	向前读取	istream
output迭代器	向前写入	ostream, inserter
forward迭代器	向前读取和写入	
bidirectional迭代器	向前和向后读取和写入	list set multiset map multimap
random access迭代器	随机存取，可读取也可写入	vector deque string array

关于输入迭代器的前置和后置递增操作符选用的讨论

应该尽可能优先选用前置式递增运算操作符（++iter）而不是后置式递增运算操作符（iter++），因为前者性能更好，前置式递增元素操作符不需传回旧值，所以也不必花费一个临时对象来保存旧值，因此，面对任何迭代器（以及任何抽象数据类型别），应该优先使用前置式，这条原则对递减运算操作符同样适用。（input迭代器并不提供递减运算操作符）。

forward iterator不具备output迭代器的全部功能，这里会有一个约束条件：

- 面对output迭代器，无需检查是否抵达序列尾端，便可直接写入数据。
- 对于forward迭代器，必须在存取数据之前确保它有效。

以下对象和型别支持random access迭代器：

- 可随机存取的容器（vector deque）
- strings(字符串 string wstring)
- 一般array(指针)