iterator、const\_iterator、reverse\_iterator和const\_reverse\_iterator。

* **五类迭代器**是迭代器可以执行的操作的通用分类，它们定义了迭代器的功能级别。
* **四种迭代器类型**是特定容器支持的迭代器接口，它们是实现五类迭代器概念的具体系列化，并且与容器的特定操作（如正向或反向遍历）相关联。

第26条：iterator优于const\_iterator、reverse\_iterator和const\_reverse\_iterator。

图示

描述已自动生成

某些版本的插入和擦除需要迭代器。如果你想调用这些函数，你就必须生成迭代器。常量和反向迭代器都不行。

不可能隐式地将const\_iterator转换为迭代器，并且第27项中描述的用于从const\_iterator生成迭代器的技术既不普遍适用也不能保证是有效的。

从reverse\_iterator到iterator的转换可能需要在转换后进行迭代器调整。第28章什么时候和为什么

避免混合使用，可能导致编译错误，或者隐式转换。

条款27：使用distance和advance将容器的const\_iterators转换为迭代器。

Item 28：理解如何使用reverse\_iterator的基迭代器。

Item 29：考虑itreambuf\_iterators用于逐个字符的输入。