

Lesson07---STL简介 (了解)

【本节目标】

- 1. 什么是STL
- 2. STL的版本
- 3. STL的六大组件
- 4. STL的重要性
- 5. 如何学习STL
- 6. STL的缺陷

1. 什么是STL

STL(standard template library-标准模板库): **是C++标准库的重要组成部分**, 不仅是一个可复用的组件库, 而且是一个包罗数据结构与算法的软件框架。

2. STL的版本

- **原始版本**

Alexander Stepanov、Meng Lee 在惠普实验室完成的原始版本, 本着开源精神, 他们声明允许任何人任意运用、拷贝、修改、传播、商业使用这些代码, 无需付费。唯一的条件就是也需要向原始版本一样做开源使用。HP 版本--所有STL实现版本的始祖。

- **P.J. 版本 VC 6.0**

由P.J. Plauger开发, 继承自HP版本, 被Windows Visual C++采用, 不能公开或修改, 缺陷: 可读性比较低, 符号命名比较怪异。

- **RW版本**

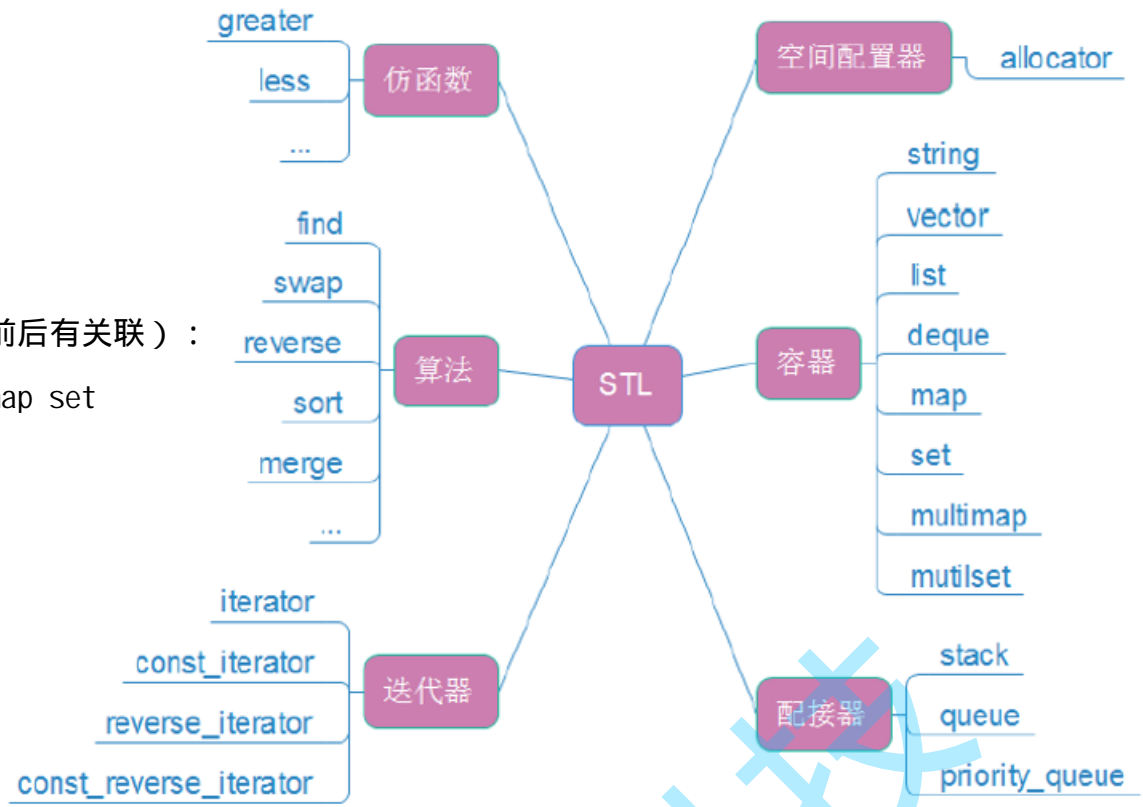
由Rouge Wage公司开发, 继承自HP版本, 被C++ Builder 采用, 不能公开或修改, 可读性一般。

- **SGI版本 STL的源码剖析**

由Silicon Graphics Computer Systems, Inc公司开发, 继承自HP版本。被GCC(Linux)采用, 可移植性好, 可公开、修改甚至贩卖, 从命名风格和编程 风格上看, 阅读性非常高。我们后面学习STL要阅读部分源代码, 主要参考的就是这个版本。

3. STL的六大组件

STL容器：
序列式（前后有关联）：
关联式：map set



4. STL的重要性

1. 在笔试中

[二叉数层序打印](#)

[重建二叉树](#)

[两个栈实现一个队列](#)

2. 在面试中

涂鸦移动面经 9.9 下午16:30 半个小时

【比特科技制作】 【186-2904-3527】

- 1.自我介绍
- 2.你项目里有空间配置器,你给我讲讲空间配置器和智能指针有什么联系吗?
- 3.智能指针了解多少,讲讲auto_ptr.
- 4.为什么C++11删掉了auto_ptr,他有什么缺陷吗?
- 5.C++11里有nullptr,这和NULL有什么区别吗?
- 6.讲讲vector和list,再讲讲两个区别.
- 7.讲讲虚函数和纯虚函数.
- 8.vector的capacity的增长.
- 9.接下来开始算法题,将思路即可
- 9.1 单链表逆置
- 【比特科技制作】 【186-2904-3527】
- 9.2 单链表带环判断
- 9.3 2~n中所有素数,n很大(欧拉筛)
- 9.4 一个整数中比特位为1的个数,整数在操作系统范围内.

一面 (电话面试)

【比特科技制作】 【18629043527】

- 1.先自我介绍一下
- 2.讲一下你理解的数组吧。数组可以越界访问吗？（我说了编译的时候不会检查，运行可能崩溃）
- 3.讲一下指针吧（我说了指针的本质、指针的初始化、野指针的问题、然后高级指针、他问了函数指针用在什么地方，我说了回调函数、多态）
- 4.指针和引用的区别
- 5.讲一下栈内存和堆内存，栈里面放什么？（我说了栈里面临时变量、又问还有什么？还有函数的参数列表）
- 6.用github多么，你用svn么，git和svn有啥区别。（我说只用git没用过svn，区别不知道）
- 7.讲一下动态内存，还有内存泄漏，后来扯到了智能指针，我说了shared_ptr，引用计数，问了为什么要有引用计数，怎么实现？
- 8.平时自己使用过库里面的智能指针吗？
- 9.说一下链表吧，如何删除一个单链表的结点，链表的结点在内存中是固定的吗？
- 10.链表的迭代器失效，怎么解决？
- 11.平时使用STL中哪些容器比较多，我说list vector，后来就问map的底层实现是什么，map和哈希表的差别？（这个问题我觉得好奇怪）
- 12.第一个项目做了多长时间（我写的是huffman压缩）
- 你说一下压缩的原理，如何保证你的压缩率，huffman压缩如何优化？你了解过ZIP压缩吗？
- 讲一下进程间通信的方式有哪些，针对共享内存说到了内存分页（说错了，就不问了）
- 13.问愿意做后台开发还是服务端开发（我不知道啥区别，就说后台开发，坑）
- 说公司主要做window环境，有什么样的想法

3. 在工作中

网上有句话说：“不懂STL，不要说你会C++”。STL是C++中的优秀作品，有了它的陪伴，许多底层的数据结构以及算法都不需要自己重新造轮子，站在前人的肩膀上，健步如飞的快速开发。

5. 如何学习STL

我个人于1998年开始潜心研究泛型技术和STL, 本书英文版《*The C++ Standard Library*》甫一出版便成为我学习 C++ 标准程序库的最重要案头工具书之一。小有心得之后, 我写过数篇相关技术文章, 从来离不开本书的影响和帮助。我曾经把 STL (代表泛型技术目前最被广泛运用的一个成熟产品, 也是 C++ 标准程序库的绝大成分) 的学习比喻为三个境界 (或层次):

- 第一境界: 熟用 STL
- 第二境界: 了解泛型技术的内涵与 STL 的学理乃至实作
- 第三境界: 扩充 STL

不论哪一个阶段, 你都能够从本书获得不同程度的帮助。



简单总结一下: 学习STL的三个境界: 能用, 明理, 能扩展。

6.STL的缺陷

1. STL库的更新太慢了。这个得严重吐槽, 上一版靠谱是C++98, 中间的C++03基本一些修订。C++11出来已经相隔了13年, STL才进一步更新。
2. STL现在都没有支持线程安全。并发环境下需要我们自己加锁。且锁的粒度是比较大的。
3. STL极度的追求效率, 导致内部比较复杂。比如类型萃取, 迭代器萃取。
4. STL的使用会有代码膨胀的问题, 比如使用vector/vector/vector这样会生成多份代码, 当然这是模板语法本身导致的。