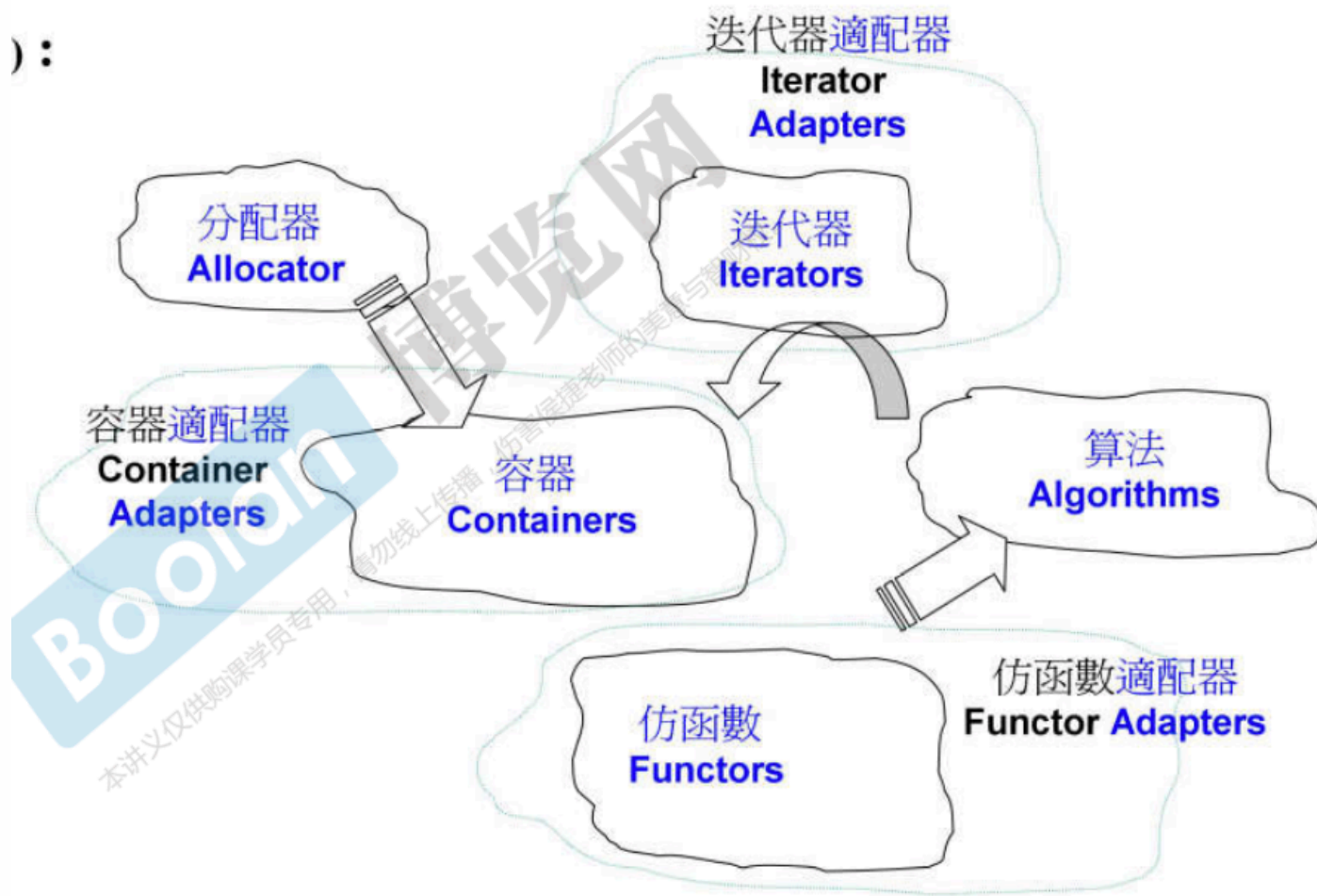


## 1 STL概述

1. 容器 Containers: STL内部封装好的数据结构, 一种class template, 常用的包括vector、list、deque、set、map、multiset、multimap等
2. 分配器 Allocators: 负责空间配置与管理。是一个实现了动态空间配置、空间管理、空间释放的class template。一般SGI STL为每一个容器都指定其缺省的空间配置器为alloc (SGI配置器)
3. 算法 Algorithms: 一种function template, 常用的有sort、search、copy、erase等
4. 迭代器 Iterators: 泛型指针, 是一种智能指针, 是一种将operator\*, operator->, operator++, operator-等指针相关操作予以重载的class template。所有STL容器都附带自己的迭代器
5. 适配器 Adapters: 一种用来修饰容器(container)或仿函数(functor)或迭代器(iterator)接口的东西。如queue和stack。它们的底部完全借助deque, 所有操作都由底层的deque供应。改变functor接口者, 称为functor adapter, 改变container接口者, 称为container adapter; 改变iterator接口者, 称为iterator adapter。
6. 仿函数 Functors: 行为类似函数, 就是使一个类的使用看上去象一个函数, 具有可配接性。它的具体实现就是通过在类中重载了operator(), 使这个类具有了类似函数的行为, 就是一个仿函数类了。一般函数指针、回调函数可视为狭义的仿函数。分为算术运算、关系运算、逻辑运算三大类。这部分内建的仿函数, 均放在头文件里, 使用时需引入头文件。



给出如下示例, 在例子中, 对应上述六大组件有:

容器: vector

分配器: allocator

迭代器: begin(),end()

算法: count\_if

适配器: not1,bind2nd

仿函数: less

```
1  void test_all_components()
2  {
3      int ia[7] = { 27, 210, 12, 47, 109, 83, 40 };
4      vector<int,allocator<int>> vi(ia,ia+7);
5
6      cout << count_if(vi.begin(), vi.end(),
7      not1(bind2nd(less<int>(), 40))); //5
8      cout << endl;
9  }
```