**STL问题集合**

**1、为什么迭代器五种类型用对象来表现，而不是1，2，3，4，5来区分呢？iterator\_category对算法的影响?**

第一点：这样方便重载呀~

template<class I>

void display\_category(I itr)

{

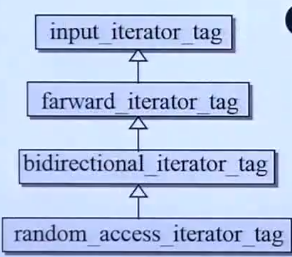
typename iterator\_traits<I>::iteator\_category cagy;

\_display\_category(cagy);

}

第二点：存在is-a 的关系

首先迭代器的五个类型存在继承关系，因此，我可能只需要提供input\_iterator和random\_access\_iterator 版本的函数就行了，当我萃取出迭代器类型farward\_iterator时，会自动匹配到input\_iterator



**2、stl是如何进行内存管理？**

使用了内存池来处理内存碎片问题，通过将内存大小以128bytes为界限进行区分，大于128bytes直接使用malloc函数申请，小于时，先从访问一个空闲内存链表，查询链表上是否有满足条件大小的内存，如果没有则向内存池申请。

释放内存时同理，大于128bytes直接释放，而小于128bytes则优先插入空闲链表。

**3、你是如何理解迭代器的？**

迭代器是存储容器信息的一个对象。每个容器都会实现针对容器本身的container\_iterator class。与容器类进行组合，作为成员变量出现。

迭代器最重要的工作是对 operator\*和operator->进行重载，这样通过\*iter就能得到迭代器所指对象的值，通过iter->能调用

**4、你是如何理解type\_traits的？**