

```
容器、算法、迭代器、仿函数、适配器、空间配置器
                     内建在C++编译器中,不需要安装额外内容
           STL优点
                    不需要了解具体实现内容, 只要熟练运用即可
                    高复用,可移植,高性能
STL概论
                     序列式容器
           容器划分
                     关联式容器,有个key起到索引作用
                  质变算法
           算法
                  非质变算法
           vector容器 vector<int>v
           vector<int>::iterator itBegin = v.begin();
                                              v.begin()起始迭代器,指向容器中第一个数据
           vector<int>::iterator itEnd = v.end();
                                           v. end()结束迭代器,指向的是容器中最后一个元素的下一个位置
STL初识
           利用算法遍历容器
           容器中存放自定义数据类型
           容器中存放自定义数据类型指针
           容器嵌套容器
           构造、赋值
          []、at 字符存取
                         []访问越界,直接挂掉,at抛出out_of_range异常
           字符串拼接+=、append
          查找find、rfind(从右往左查找)
          替换replace、比较compare
           子串substr
                      案例: 求email中的用户名
string
          插入insert、删除 erase
                                string转const char * .c_str();
          string和const char* 转换
                                const char * 转string, 利用string有参构造
                                const char *可以隐式转换为string, 反之不可以
                      小写转大写 toupper
           大小写转换
                      大写转小写 tolower
             单端数组
             动态数组:并不是在原有空间后续进行扩展,而是找一个更大的内存空间,将原有数据
             拷贝到新空间下,并且释放原有空间
             构造、赋值、交换swap、大小size 、是否为空empty
                            如果重置的比原来大,有默认值填充新位置
             重置大小resize
                           如果重置的比原来小,超出的部分删除掉
 vector容器
             front返回容器中第一个元素,back返回容器中最后一个元素
             插入insert(迭代器)、删除erase (迭代器)、尾插push_back、尾删pop_back、
             清空clear
             案例1:巧用swap收缩内存
             案例2: 巧用reserve预留内存
             逆序遍历reverse_iterator,非质变
             判断容器的迭代器是否支持随机访问
              符合先进后出的数据结构
栈容器 Stack
              入栈push、出栈pop、栈顶top、是否为空empty、栈大小size
               符合先进先出的数据结构
队列容器 Queue
               入队 push、出队pop、队头元素front、队尾元素back
               是否为空empty、队列大小 size
           双向循环链表
           构造、赋值、大小、重置大小、是否为空、反转reverse
                     如果容器的迭代器支持随机访问, 可以使用系统提供的标志算法
           排序sort
list容器
                     不支持随机访问的迭代器的容器,内部会提供对应的算法接口
           对于自定义数据类型,必须要指定排序规则
           如果利用remove删除自定义数据类型,需要重载 ==
           关联式容器: key就是value
          默认排好序: 从小到大
          插入insert、大小size、是否为空empty
                                       set.insert的返回值是个对组 ,pair<iterator, bool> bool代表插入是否成功
          查找: find, 返回值: 迭代器
          统计: count, 对于set的结果要么是0, 要么是1
          lower_bound(keyElem);返回第一个key>=keyElem元素的迭代器。
set容器
          upper_bound(keyElem);返回第一个key>keyElem元素的迭代器。
                                                               pair对组: pair<string, int> p("Tom", 10);
          equal_range(keyElem);返回容器中key与keyElem相等的上下限的两个迭代器。
                                                              pair<string, int> p2 = make_pair("Jerry", 18);
          multiset可以插入重复的key值
          可以指定set容器的排序规则,但是必须在插入前指定,利用仿函数的技术
                                                          对于自定义数据类型,set通常都会指定出排序规则
          关联式容器,默认按照key从小到大排序
                  m. insert (pair \langle int, int \rangle (1, 10));
                 m. insert(make_pair(2, 20));
          插入
                 m. insert(map<int, int>::value_type(3, 30));
                  m[4] = 40;
map容器
          查找: find, 返回值是迭代器
          统计: count
          lower_bound(keyElem);返回第一个key>=keyElem元素的迭代器。
          upper_bound(keyElem);返回第一个key>keyElem元素的迭代器。
          equal_range(keyElem);返回容器中key与keyElem相等的上下限的两个迭代器。
          利用仿函数,实现指定排序规则
```

Day1

STL容器使用时机