|  |  |
| --- | --- |
| 1. 关联容器操作 2. 关联容器迭代器 3. Set的迭代器是const的 4. 遍历关联容器 5. 关联容器和算法 | Set:key\_type value\_type  Map: key\_type value\_type mapped\_type  set<string>::key\_type v1;  set<string>::value\_type v2;  map<string, int>::value\_type v3;  map<string, int>::key\_type v4;  map<string, int>::mapped\_type v5; |
| 1. 关联容器操作 2. Set的key\_type与value\_type是一样的 3. Map的value\_type是pair，first是关键字类型，second是value类型   Mapped\_type是值类型，key\_type是关键字类型   1. 关联容器迭代器 2. 解引用关联容器的迭代器或得到value\_type类型的值引用。对于map来说是pair类型。 3. Pair的关键字是const的不能改变，second是值，可以改变值不能改变关键字 4. Set的迭代器是const的 5. set的迭代器是const的，set的关键字也是const的 不能修改 6. 即使有iterator与constiterator但是都一样不能修改 7. 遍历关联容器 8. 遍历关联容器 可以用迭代器   Note:迭代器按关键字升序遍历元素   1. 关联容器和算法 2. 因为关键字是const的不能用于泛型算法中重排容器元素的算法 3. 关联容器不能通过关键字快速查找，使用泛型算法的find是个坏主意，可以使用关联容器自定义的find的成员，给定关键字直接获取元素。 4. 若想使用泛型算法，可以把关联容器当做原序列或目的位置，如：copy算法将元素从一个关联容器到另一个序列，与inserter插入器绑定到一个关联容器做目的位置 | |

