|  |  |
| --- | --- |
| 1. tuple类型 2. 定义和初始化tuple 3. 访问tuple的成员 4. 关系和相等运算符 | Tuple<size\_t,size\_t,size\_t> threeD;  Tuple<size\_t,size\_t,size\_t> threeD(1,2,3);  Auto it = make\_tuple(1,2,3);  Auto boo = get<0>(it);  Get<2>(it) = get<2>(it) \* 0.8;  Typedef decltype(it) trans;  Size\_t sz = tuple\_size<trans>::value;  Tuple\_element<1,trans>::type cnt = get<1>(item);  twoD < threeD |
| 1. tuple类型 2. 类似pair的模板，每个pair的成员类型都不相同，但每个pair都恰好有两个成员 3. 不同tuple类型的成员类型也不相同，一个tuple可以有任意数量的成员 4. 定义和初始化tuple 5. 可以使用tuple的默认构造函数，对每个成员进行值初始化 6. Tuple的构造函数是explicit的，所以我们必须使用直接初始化语法 7. Make\_tuple函数使用初始值的类型来判断tuple的类型 8. 访问tuple的成员 9. 用下标，用get的标准库函数模板，必须一个实参，指出需要第几个成员。返回一个指定成员的引用 10. 括号的值必须是一个整形常量表达式。0开始计数 11. 可以用辅助类模板来查询tuple成员的数量和类型   Size\_t sz = tuple\_size<trans>::value;  Tuple\_element<1,trans>::type cnt = get<1>(item);  需要一个tuple的类型用decltype，再使用tuple\_size,tuple\_element  Tuple\_size有一个名为value的public static数据成员  Tuple\_element模板除了一个tuple类型外，还接受一个索引值，有一个名为type的public类型成员   1. 关系和相等运算符 2. 要想tuple==运算符，tupe的每对成员使用==运算符必须都是合法的 3. 只有两个tuple具有相同数量的成员时才可以比较他们 | |