|  |  |
| --- | --- |
| 1. 转发参数包 | Forward  Template <class…Args>  Void emplace\_back(Args&&…);  Alloc.construct(first\_free++,std::forward<Args>(args)…);  Std::forward<Args>(args)…  Std::forward<Ti>(ti);  Std::forward<int>(10), std::forward<char>(c);  Forward<string>(string(“the end”)) |
| 1．组合使用可变参数模板与forward机制来编写函数其实参不变地传递给其他函数。  2．实现我们的emplace\_back，有一个可变参数成员模板，并且希望使用string的移动构造函数，所以需要保持传递给emplace\_back的实参的所有类型信息，就是接受左值右值，T&&，接受传递给它任意类型的实参，给emplace\_back可以是10，’c’，这种任意实参，只要能符合construct的构造函数就可，但是不能”ss”,”aa”,construct没有这种构造函数  3. 代码如上，就是会为每个参数成员都有std::forward<Ti>(ti)的转换，因为写法std::forward<Args>(args)…这样即扩展了模板参数包Args,也扩展了函数参数包args  4. 传递右值的话，还是能保持右值前面内容。  // 有点不懂  Forward<int>(10),forward<char>(c)给construct的右值引用有什么用？又不是指针实打实的指向值。需要右值引用到新的位置 | |