|  |  |
| --- | --- |
| 1. 从左值引用函数参数推断类型 2. 从右值引用函数参数推断类型 3. 引用折叠和右值引用参数 4. 编写接受右值引用参数的模板函数 | Template <typename T> void f(T &p)  Template <typename T> void f1(const T&)  Template <typename T> void F3(t&&);  X& &、X& && 和X&& &都折叠成类型X&  类型X&& && 折叠为X&&  Template<typename T> void f4(T&& val)  {T t = val; }  Template <typename T> void f(T&&);// 非const右值  Template<typename T> void f(const T&);// z左值或const右值 |
| 1. 从左值引用函数参数推断类型   T时：是按值传递  T&时：是按类型还是地址传递？  Const只在函数参数是T忽略，T&不会忽略，const T&默认存在const   1. 若函数参数是左值引用时，T&，只能传递一个左值，const左值 2. 函数参数T&   若传左值，i是int，模板参数T是int  若传const左值，模板参数T是const int  不能传右值   1. 若函数参数是const T&，可以传递任何类型的实参   若传左值，那么T为int  若传为const int i，那么T为int  若传右值，那么T为int&&   1. 从右值引用函数参数推断类型 2. 函数参数是右值引用 T&&   传参数为42，为一个int类型的右值， T是int，直接匹配，不用为int&&   1. 引用折叠和右值引用参数 2. 通常不能一个右值引用绑定到左值引用上，定义一个引用的引用 3. 两个例外：   第一个影响右值引用绑定到左值引用推断如何进行   1. 将一个左值传递给函数的右值引用参数，T&&，编译器推断类型参数为实参的左值引用类型。T的类型是int&   第二个通过模板类型参数间接可以定义一个右值引用的引用   1. 这些重叠的引用会形成“折叠”，除了右值引用的引用会折叠成右值引用，其它的都折叠成左值引用类型 2. Note:引用折叠只能应用于间接创建的引用的引用，类型别名或模板参数 3. 如：给T&&传递左值f3(i)，则T为int&,传递f3(ci),T为const int&   Void f3<int&>(int& &&);   1. 总结两个重要结果：T&& 2. 它可以绑定到一个左值 3. 若实参是一个左值，则T为左值引用，且函数参数将被实例化为一个普通左值引用参数T& 4. 显然也可以传递右值，折叠T为类型   Note：T&&，可以传递任意类型的实参，若传左值，则T&   1. 编写接受右值引用参数的模板函数 2. 问题：T&&内模板的模板参数T是引用int&还是类型int，维护起来麻烦 3. T t = val 4. 若传左值i，则T为int&，t为一个引用 5. 若传字母值42，则T为int，t为一个int拷贝，因为T&& &&折叠为T&&，与函数模板T&&一样就为T为类型int 6. 在函数模板代码块因为T可能为类型，也可能为引用类型，所以编写代码通用困难 7. 模板引用用于：模板转发和模板重载 8. 通常定义两种带引用的函数参数还实现完整性     第一个版本可以绑定可修改的右值，第二个版本绑定到左值或const右值  这两个版本应该是确保T为int类型，不会是引用int&,这样编写维护代码容易？就是看谁优先匹配了？  如：传入左值引用，第二个版本优先匹配会选择这个吧？则T为int  传入右值引用，第一个版本优先匹配吧？则T为int  第一个版本：  非const右值：T为int  第二个版本： int,cons tint：T为int  cont右值：T为int&&也是可以操作不影响参数的值  但是我试了下  只有传const int才会绑定到第二个版本  Int，const右值，非const右值，都是给第一个版本！  const T&特点：除了传右值，T都为int  总：函数模板T&  传左值引用，T为int，(int&)  传const左值引用，T为const int,(const int&)  不能传右值引用  函数模板const T&  传左值引用，T为int, ( const int&)  传const左值引用，T为int, (const int&)  传右值引用(const或非const)，T int&&, 折叠后(const int&)  函数模板T&&  传左值引用，T为int&,折叠后 (int&),  传const左值引用，T为const int&,折叠后(const int&)  传右值引用，T为int，没有折叠(int&&)  只有间接创建的引用的引用？  说：能绑定非const右值    问题：  不知道能不能传const左值引用，是否能保持const  传递f3(ci),T为const int&，可以传，还会保持。 | |