**C++学习笔记——万能引用与完美转发**

本文需要读者具备左值和右值引用的基础知识。

**1. 万能引用**

如果类型名称是&&，并且需要编译器推导引用所指向的类型，则属于万能应用（universal reference），或称转发引用（forwarding reference）。

从定义上来说，万能引用是一种类型。

万能引用最后被推导为左值引用还是右值引用，取决于实际传递的引用类型。

万能引用通常表现为“函数形参”或“auto&&”。见以下示例。

文本, 信件

描述已自动生成

以下情形中，&&不代表万能引用，而是代表右值引用：

(1) 使用具体类型，而不是模板类型。具体类型是不需要编译器推导的。

(2) 在形参的T&&前有const修饰，即“const T&&”。

文本

描述已自动生成

(3) 模板类本身有T类型的成员变量时，其构造函数参数列表中的T&&，代表右值引用，而非万能引用。

见如下代码：

文本

描述已自动生成

如果确实需要为模板类定义一个参数为万能引用的构造函数，那么在语法上并非不能实现，示例代码如下图所示。



但是，在开发实践中，极其不建议这样做。模板本身已经增加了理解难度；而将复制/移动构造函数合二为一，并且涉及两个表面上不同实则相同的模板类T和TT，更是严重降低代码可读性——要知道，绝大多数C++程序员根本不知道这种写法是什么意思。

最佳实践就是，分别声明和实现const左值引用和右值引用版本的构造函数。

2. std::move

将一个对象或变量无条件变为右值。多用于需要调用移动构造函数，以节约复制开销的场合。

3. std::forward<T>

T&&属于万能引用，可以接受左值或右值的实参。

但是，不管实参是左值还是右值，到了函数体内部，都会变为左值——右值引用本身也是左值。如果该函数还需要调用其他函数，并在调用时以该参数为实参（即“转发”变量），那么可能有必要保留参数本来的左右值属性。这种语义被称为“完美转发”。

std:forward<T>正是做到了完美转发。

