条款23 理解std::move和std::forward (萃取器见条款9)

① std::move的实现 a.传入参数的类型

(伪码仿照) b.本质为类型转换 static_cast<type&&>(xxx);

C++11 C.如何去掉引用

C++14 std::remove_reference<T>type

② std::move实际作用：转化实参为右值(有建议命名为rvalue_cast)

对一T对象使用std::move就是告诉编译器，这T对象适合移动

对const右值引用，只能匹配拷贝构造

```
class Annotation{ public:
    explicit Annotation(const A a):
        a_param(std::move(a));
private:                    右值引用绑定将亡值时注意
    A a_param;              底层const
}
```

③初识std::forward

```
template<typename T>
void logAndProcess(T && param){
    process(xxx param xxx);}
```

希望将param的左右值信息进行保留，避免右值降级为左值

函数声明

a. process(const T &lval);

b. process(T &&rval);

函数调用

process(param); 只匹配a,仅识别左值

process(std::move(param)); 只匹配b,仅识别右值

process(std::forward<T>(param)); a.b均能匹配，保留了左右值信息

std::forward<T>本质：有条件的move，只有当实参用右值初始化时才为右值

std::move 本质：总是将左值转为右值