

move

```
#include <utility>
```

移动语义, 移动就是为了避免拷贝。

- `std::move`并不会真正移动对象, 真正的移动操作是在对象的移动构造函数、移动赋值操作符等完成, `std::move`只是将参数转换为右值引用而且(即`static_cast`)
- `std::move`无条件地将它的参数转换成一个右值, 然后调用对象相对应的移动构造或移动赋值函数, 如果未定义, 则会调用常规的拷贝构造或赋值函数。
- `std::move`的本质就是帮助编译器选择重载函数(右值引用版本的)

```
template<typename T>
typename remove_reference<T>::type && move(T &&param){
    using return_type=typename remove_reference<T>::type &&;
    return static_cast<return_type>(param);
}
```

移动语义的操作

移动构造函数, 移动赋值函数等, 各个类的实现不同。被移动的对象保持什么状态由各个类决定。如`string`类的移动赋值, 被移动后的原对象置为空(一个有效的状态)。