

右值引用

左值和右值

- 有名字、或可以取地址的都是左值（在一定范围的取地址，因为右值严格上也可以取地址）
- 左值可以看作“对象”，右值可看作“值”
- 左值到右值的转换可看作“读出对象的值”
- `std::move`允许任何表达式以“值”的方式处理
- `std::forward`允许在处理的同时，保留表达式为“对象”还是“值”的特性

任何一个有价值的程序都是：

- 反复操作各种类型的“对象”
- 这些“对象”在运行时创建在明确的内存范围
- 这些“对象”存储着“值”

右值引用就是为了移动语义引入的引用类型。可以自由地将一个右值引用的资源“移动”到另一个对象中。

- 左值表达式表示的是一个对象的身份
- 右值表达式表示的是一个对象的值
- 左值持久；右值短暂

有以下需要注意的规则：

1. `const`左值引用可以绑定到右值上

```
const int &r=a+b;
```

2. 右值引用只可以绑定右值
3. 非`const`左值引用只可以绑定到左值上
4. 右值引用其本身是左值。

```
int &&r=a+b;
// r是右值引用类型
// r本身是左值，不是右值
// 即r是一个右值引用类型的左值
```