

13章 拷贝控制和资源管理

拷贝控制操作总共有5种：拷贝构造函数、拷贝赋值运算、移动构造函数、移动赋值运算、析构函数。

拷贝构造函数、拷贝赋值运算

```
1 Sales_data(const Sales_data&); //拷贝构造函数范例
2 Sales_data& operator=(const Sales_data &); //拷贝赋值运算范例
3
4 Sales_data(const Sales_data&) = default;
5 Sales_data& operator=(const Sales_data &) = default;
6
7 Sales_data(const Sales_data&) = delete; // 阻止拷贝，必须在类内声明
8 Sales_data& operator=(const Sales_data &) = delete; // 阻止赋值，必须在类内声明
9
10 // 发生情况
11 string s1(12,'a'); // 直接构造初始化
12 string s2(s1); // 我觉得是拷贝构造初始化，但书上写的是直接初始化（仅针对string）
13 string s3 = s1; // 拷贝赋值运算
14 string s4 = "nice"; //拷贝赋值运算
15 // 其他
16 // 将一个对象作为实参传递给一个非引用类型实参
17 // 用花括号列表初始化一个数组中的元素或者一个聚合类中的成员P266
```

析构函数

```
1 ~Sales_data(); //析构函数范例
```

析构函数调用时机

1. 变量离开其作用域时
2. 当一个对象被销毁时，成员也被销毁
3. 容器被销毁时，成员也被销毁
4. delete的
5. 临时对象，当创建它的完整表达式结束时被销毁

注1：通过拷贝控制，可以创建**行为像值的类**和**行为像指针的类**

注2：交换操作交换的是指针，没有创建值，因此快且不占内存

```
1 using std::swap;
2 swap(a,b); //这是先看自己有没有swap，自己的优先使用，没有再用std::swap
```

移动构造函数和移动赋值函数

左值持久，右值短暂，变量是左值

```
1 int i = 1;
2 int &r = i;    // r引用i
3 int &&rr = i;  // 错误，不能将一个右值引用绑定到一个左值上
4 int &r2 = i * 2;    // 错误，i*2 是个右值
5 const int &r3 = i * 2; // const可以
6 int &&r4 = i * 2;    //正确，将r4绑定到乘法结果上
```

```
1 Sales_data(Sales_data&& ) noexcept: 成员初始化 {}    // 移动构造函数范例
2 Sales_data& operator=(Sales_data&& ) noexcept{   } // 移动赋值函数范例，一直没看到什么时候会被调用
```

```
1 Sales_data x;
2 Sales_data y(x);    // 拷贝构造函数
3 Sales_data z(std::move(x)); // 移动构造函数，若没有，则未拷贝构造函数
```

。。。。。。 待续