**7 类**

**1，基础**

定义在类内部的函数是隐式的inline函数。

this是一个常量指针，不允许改变this中保存的地址。

struct和class的唯一区别就是默认访问权限不一样。class的成员默认是private的。

要在const成员函数内修改类的成员，通过在变量的声明中加入mutable关键字实现。

一个const成员函数如果以引用的形式返回\*this，那么它的返回类型将是常量引用。

**2，友元**

类可以允许其他类或函数访问它的非公有成员，方法是使用友元。

类想要把一个函数作为它的友元，只需要增加一条以friend关键字开始的函数声明即可。

一般来说，最好在类定义开始或结束前的位置集中声明友元。

可以把类设置成友元 friend class Window\_mgr

友元关系不存在传递性。

令成员函数作为友元 friend void Window\_mgr::clear(ScreenIndex);

**3，构造函数**

默认构造函数的声明 Sales\_data() = default;

Sales\_data(const std::string &s):bookNo(s){}冒号后的部分称为构造函数初始值列表。

委托构造函数：Sales\_data(std::string s, unsigned cnt, double price){}

和Sales\_data(std::string s):Sales\_data(s,0,0){}

隐式的类类型转换：

如果Sale\_data的构造函数只有一个参数，那么可以item.combine(string("Hello"))

抑制构造函数定义的隐式转换使用explicit声明，这个关键字只对一个实参的构造函数有效。

聚合类：

1，所有成员都是public的；2，没有定义任何构造函数；3，没有类内初始值；4，没有基类，也没有virtual函数。

例子：struct Data{int ival;string s};

Data val1 = { 0, "Anna"}

**4，静态成员**

在类的外部定义静态成员时，不能重复static关键字，该关键字只出现在类内部的声明语句。

使用类访问静态成员函数：Account::rate()