7 类

1 定义抽象数据类型

作为接口组成部分的非成员函数，它们的声明和定义都在类的外部。类外部定义的成员的名字必须包含它所属的类名。

std::string isbn() const{}中的const关键字的作用是修改隐式this指针的类型。这个成员函数不能改变调用它的对象的内容。使用const的成员函数被称作常量成员函数。

显式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。

除了定义类的对象如何初始化之外，类还需要控制拷贝，赋值和销毁对象时发生的行为。

2，访问控制与封装

struct和class定义类的唯一区别就是默认访问权限不太一样。class的成员默认是private的。

类要把一个函数作为它的友元，只需增加一条以friend关键字开始的函数声明语句即可。

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。

3，类的其他特性

将变量声明为mutable，即使在一个const成员函数内也可以进行修改。

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);

4 类的作用域

5 构造函数再探

如果成员是const，引用或者属于某种未提供默认构造函数的类类型，我们必须通过构造函数初始值列表为这些成员提供初值。

一个委托构造函数使用它所属类的其他构造函数执行它自己的初始化过程。

例子：Sales\_data(std::string s, unsigned cnt, double price){}

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

隐式的类类型转换：

Sale\_data的构造函数只有一个string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。需要多个实参的构造函数不能用于执行隐式转换。

当一个类满足如下条件时，它是聚合类：

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"}

6 类的静态成员

使用类访问静态成员函数：Account::rate()

当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。

必须在类的外部定义和初始化每个静态成员，和其他对象一样，一个静态数据成员只能定义一次。静态数据成员定义在任何函数之外。

可以为静态成员提供const整数类型的类内初始值，不过要求静态成员必须是字面值常量类型的constexpr。

例子：static constexpr int period = 30