1 定义抽象数据类型

定义用于返回售出书籍平均价格的函数avg\_price，因为这个函数的目的并非通用，所以它应该属于类的实现的一部分，而非接口的一部分。

成员函数的声明必须在类的内部，它的定义则既可以在类的内部也可以在类的外部。

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。

定义在类内部的函数时隐式的inline函数。

this的目的总是指向这个对象，所以this是一个常量指针，我们不允许改变this中保存的地址。

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。

像这样使用const的成员函数被称作常量成员函数。

常量对象，以及常量对象的引用或指针都只能调用常量成员函数。

类外部定义的成员的名字必须包含它所属的类名。

编译器分两步处理类：首先编译成员的声明，然后才轮到成员函数体。

一般来说，如果非成员函数是类接口的组成部分，则这些函数的声明应该与类在同一个头文件中。

构造函数不能被声明为const。

如果类包含有内置类型或复合类型的成员，则只有当这些成员全都被赋予了类内的初始值时，这个类才适合使用合成的默认构造函数。

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。

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。

如果我们不主动定义这些操作，编译器将替我们合成它们。一般来说，编译器生成的版本将对对象的每个成员执行拷贝，赋值和销毁操作。

2，访问控制与封装

使用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。

类可以允许其他类或函数访问它的非公有成员，方法是令其他类或函数成为它的友元。

如果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

最好只在类外部定义的地方说明inline，这样可以使类更容易理解。

有时候我们希望能修改类的某个数据成员，即使在一个const成员函数内，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。

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，或者说它把它自己的一些职责委托给其他构造函数。

例子：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。需要多个实参的构造函数不能用于执行隐式转换。

聚合类使得用户可以直接访问其成员，并且具有特殊的初始化语法形式。当一个类满足如下条件时，它是聚合的：

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"}

数据成员都是字面值类型的聚合类是字面值常量。

字面值常量类（了解）。

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。

例子：double Account::interestRate = initRate()

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

静态数据成员的类型可以就是它所属的类类型，而非静态数据成员则受到限制，只能声明成它所属类的指针或引用。

静态成员和普通成员的另外一个区别是我们可以使用静态成员作为默认实参。