

江西财经大学

本科学生综合性实验报告

项目组长刘燕 学号0204284

成 员李激 钟琳

专 业软件工程 班级软件 203 班

实验项目名称小型书店进销存管理系统

指导教师及职称张玲 副教授

开课学期 2020 至 2021 学年 二 学期

上课时间 2021 年 3 月 2 日

目 录

一、实验综述.....	1
1、实验目的及要求.....	1
实验目的：	1
实验要求：	1
2、实验仪器、设备或软件.....	1
二、实验过程（实验步骤、记录、数据、分析）	1
1、实验分析.....	1
2、实验分工.....	4
3、实验过程.....	4
三、结论.....	19
1、实验结果.....	19
（1）程序初始界面.....	19
（2）程序主界面.....	21
（3）查看当前库存功能.....	21
（4） 增加、删除、修改书刊信息.....	22
（5） 增加书刊库存.....	24
（6） 书刊销售系统.....	24
（7） 当前财务统计.....	25
（8） 更改管理员 ID.....	25
（9） 切换登录账户.....	26
（10） 将库存信息输出至文件.....	26
（11） 退出书店系统.....	27
2、分析讨论.....	27
四、指导教师评语及成绩：	28

学生实验报告八（综合性实验）

学生姓名	李澈	学号	4201998	同组人 刘燕、钟琳
实验项目	小型书店进销存管理系统			
☒ 必修 ☐ 选修		☐ 演示性实验 ☐ 验证性实验 ☐ 操作性实验 ☒ 综合性实验		
实验地点			实验仪器台号	
指导教师	张玲		实验日期及节次	2021.6.1 - 6.15

一、实验综述

1、实验目的及要求

实验目的：

通过运用 C++ 编程解决一个比较复杂的实际问题，使学生对 C++ 编程加深理解，并对用 Visual C++ 进行应用程序开发有一个初步的认识。综合训练 C++ 知识，包括 C++ 程序的基本控制结构、类与对象、继承与派生、算法分析与设计等，从而为后续课程的学习打下扎实的编程基础。

实验要求：

运用 C++编程实现一个小型书店进销存管理系统，该系统需使用面向对象的设计思想，至少实现以下功能：

- （1）增加、删除书目
- （2）进货、销售书目
- （3）文件输出功能

2、实验仪器、设备或软件

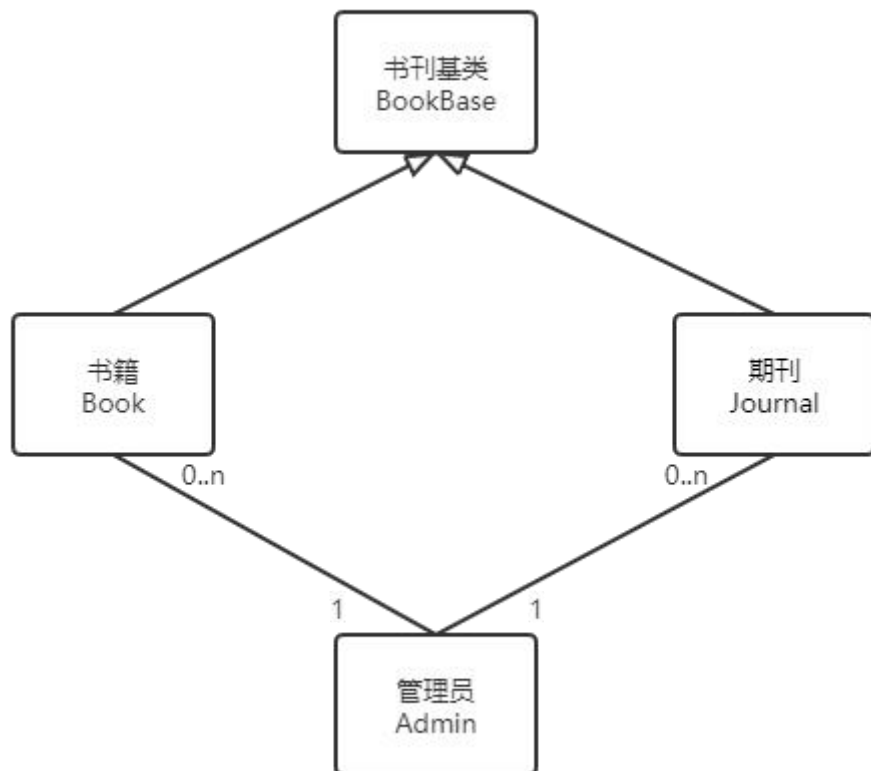
计算机、VC6.0、office、相关的操作系统等。

二、实验过程（实验步骤、记录、数据、分析）

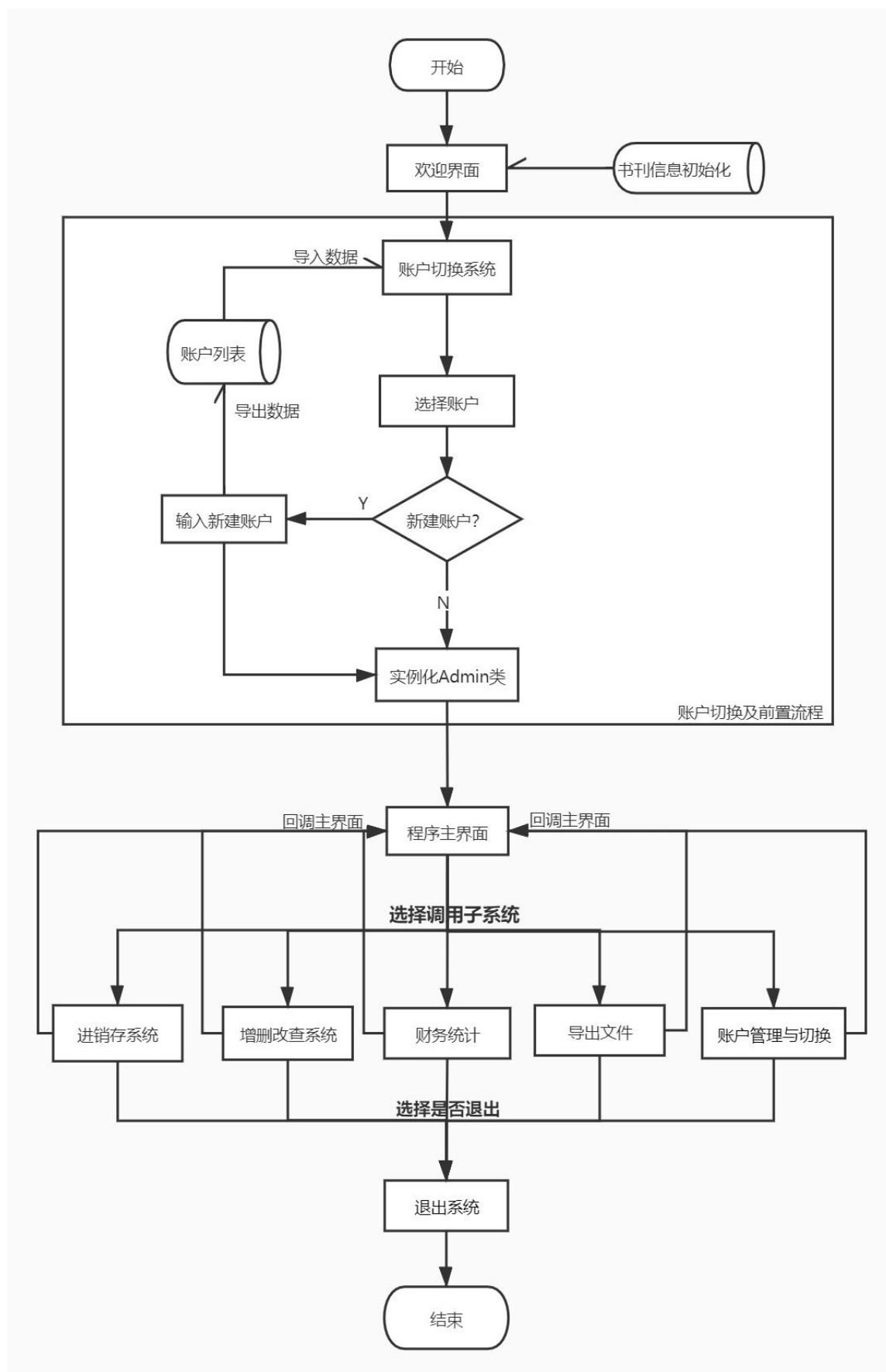
1、实验分析

（1）使用 C++ 编写面向对象的程序，需要对程序需求进行类的分析，形成程序的框架。该系统涉及到的类较少，经名词分析，得出符合我们设计目标的所需的

类有书刊基类、书籍类、期刊类、管理员类，所得 UML 类图如下：



(2) 对于功能的实现如进销存等，都使用成员函数来实现，对于有可能实现数据留存的功能，皆使用文件读写以保留数据。初步分析所有场景，作出目标程序的简化流程图如下，因 UML 建模非本次实验探讨重点，故省略泳道图：



(3) 本程序希望综合运用到本课程几乎每一节的重要知识点，充分巩固课堂所学并锻炼所学技能。

如：

对于类的关系，使用继承；

对于数据成员的保护，使用常成员函数；

对于对象数组的定义，使用宏变量定义等。

2、实验分工

姓名	角色	工作描述
刘燕 0204284	组长	统筹工作、分配任务、编写类相关的头文件、实现主要功能、撰写实验报告的实验分析、结果部分、总结所有工作
李澍 4201998	组员	面向对象分析与 UML 建模、编写源文件、撰写实验报告的实验过程部分、分析讨论部分、实现文件功能
钟琳 0204285	组员	收集资料、编写数据结构相关代码、编写实现交互的初版代码、撰写实验报告的主要框架

3、实验过程

1. 启动 Visual Studio Community 2019， 进入集成开发环境。

2. 创建新的基于 C++ 的 Win 32 控制台项目，开始输入源代码，其基本操作步骤为：“创建新项目” → “控制台应用” → 设置工作目录 → “创建”，在打开的解决方案资源管理器中，在源文件目录上右键选择 “添加” → “添加项” → “Visual C++” → “C++文件（.cpp）”，命名并点击添加。

3. 输入程序代码如下（注：本程序使用多文件设计结构，为审阅方便特整合到单文件中，文件名在注释中标注）：

```
#include <time.h>
#include <iostream>
#include <string>
#include <iomanip>
#include <fstream>

//#include "Book.h"
//#include "Admin.h"

#define MAXSIZE 10000
using namespace std;

//全局变量
string ID_Value;
```

```
//存储新建账户名
```

```

string ID[5];           //存储 ID 列表
int ID_flag;           //存储 ID 操作序号
fstream file;          //文件对象
void launch();         //声明 RUNTIME 函数

//book.h
static int Bcount = 2;
static int Jcount = 2;
static double TTotal;
static int flag = 0;

class BookBase
{
private:
    string ISBN = "00000000";
    string Bname = "默认书籍";
    int stock = 0;
    double price = 0;
    double total = 0;
public:
    BookBase() {};
    BookBase(string is, string nam, int s, double p)
    {
        ISBN = is;
        Bname = nam;
        stock = s;
        price = p;
    }
    string getISBN() const { return ISBN; }
    string getName() const { return Bname; }
    int getStock() const { return stock; }
    double getPrice() const { return price; }
    double getTotal() const { return total; }
    void modISBN(string i) { ISBN = i; }
    void modName(string n) { Bname = n; }
    void modStock(int st) { stock = st; }
    void modPrice(double pr) { price = pr; }
    void addStock(int s)
    {
        stock += s;
        cout << "库存增加成功！目前库存为：" << getStock() << endl;
    }
    void outStock(int s)
    {
        if (s <= stock)
        {

```

```

        stock -= s;
        cout << "销售成功！目前库存为：" << getStock() << endl;
        total = s * price;
        TTotal += total;
        cout << "销售额为：" << getTotal() << endl;
        flag = 1;
    }
    else
    {
        cerr << "库存不足，请检查，目前库存仅有" << getStock() << "。" << endl;
        return;
    }
}

};

//期刊类
class Journal :public BookBase
{
private:
    string express;
public:
    Journal() {}
    Journal(string is, string nam, int s, double p, string exp) :BookBase(is, nam, s, p) { express =
exp; }
    friend ostream& operator<<(ostream& output, const Journal& j);
    string getExpress() const { return express; }
    void modExpress(string ex) { express = ex; }
};

//输出流运算符重载
ostream& operator<<(ostream& output, const Journal& j)
{
    cout.setf(ios_base::left);
    output << j.getISBN() << "\t" << setw(25) << j.getName() << setw(20) << j.getExpress() <<
setw(15) << j.getPrice() << setw(15) << j.getStock() << endl;
    return output;
}

//书籍类
class Book :public BookBase
{
private:
    string author;
public:
    Book() {}
    Book(string is, string nam, int s, double p, string au) :BookBase(is, nam, s, p) { author = au; }

```



```

    friend ostream& operator<<(ostream& output, const Book& b);
    string getAuthor() const { return author; }
    void modAuthor(string au) { author = au; }
};

//输出流运算符重载
ostream& operator<<(ostream& output, const Book& b)
{
    cout.setf(ios_base::left);
    output << b.getISBN() << "\t" << setw(25) << b.getName() << setw(20) << b.getAuthor() <<
    setw(15) << b.getPrice() << setw(15) << b.getStock() << endl;
    return output;
}

/*
 * 以上为书籍相关类及成员函数，以下为管理员类及其成员函数
 */
//Admin.h
class Admin
{
private:
    string ID;
public:
    Admin() {};
    string getID() const { return ID; }
    Admin(string id) { ID = id; cout << this->getID() << "，欢迎您登陆管理系统！所有权限已开启。
" << endl; }
    void changeID(string id) { ID = id; }
    void AddBook();
    void DelBook();
    void SearchBook();
    void inStock();
    void outStock();
    void printAll();
    void statics();
    void modify();
};

//预置书籍类数据
static Book Books[MAXSIZE] = {
    Book("9787302257646", "程序设计基础", 190, 25, "张三"),
    Book("9787302219972", "单片机技术及应用", 182, 32, "李四")
};

//预置期刊类数据
static Journal Journals[MAXSIZE] = {
    Journal("9787302257611", "电脑爱好者", 200, 32, "北京大学出版社"),

```

```

        Journal("9787302257622","无线电年刊",190,35,"江西财大出版社")
};
//输出所有 Journal
void JournalsPrint()
{
    cout << "----- 期刊类 -----" << endl;
    cout.setf(ios_base::left);
    cout << "序号\tISBN\t\t" << setw(25) << "期刊名" << setw(20) << "出版社" << setw(15) << "价格"
    << setw(15) << "库存" << endl;
    for (int i = 0; i < Jcount; i++) cout << i + 1 << "\t" << Journals[i];
    cout << endl;
}
//输出所有 Book
void BooksPrint()
{
    cout << "----- 书籍类 -----" << endl;
    cout.setf(ios_base::left);
    cout << "序号\tISBN\t\t" << setw(25) << "书名" << setw(20) << "作者" << setw(15) << "价格"
    << setw(15) << "库存" << endl;
    for (int i = 0; i < Bcount; i++) cout << i + 1 << "\t" << Books[i];
    cout << endl;
}
void Admin::statics()
{
    char c;
    if (flag)
    {
        cout << "-----" << endl;
        cout << setw(25) << internal << "销售统计" << endl;
        cout << "总销售额: " << TTotal << endl;
        cout << "是否将销售额导出至文件? (Y/N) "; cin >> c;
        if(c=='Y' || c=='y')
        {
            ofstream file;
            file.open("销售额.txt", ios::out);
            file << "总销售额: " << TTotal << endl;
            file.close();
            cout << "导出成功! " << endl;
        }
        cout << "-----" << endl;
    }
    else
    {
        cerr << "无任何销售记录! " << endl;
    }
}

```

```

}
//添加新书籍
void NewBook()
{
    string ISBN;
    string Bname;
    int stock;
    double price;
    string author;
    cout << "-----" << endl;
    cout << "正在添加新的书籍信息，当前已有书籍数目:" << Bcount << endl;
    cout << "输入 ISBN 号： "; cin >> ISBN;
    cout << "请输入书名： "; cin >> Bname;
    cout << "请输入库存： "; cin >> stock;
    cout << "请输入价格： "; cin >> price;
    cout << "请输入作者名： "; cin >> author;
    Books[Bcount] = Book(ISBN, Bname, stock, price, author);
    Bcount++;
    cout << "添加成功！ " << endl;
}
//添加新期刊
void NewJournal()
{
    string ISBN;
    string Bname;
    int stock;
    double price;
    string express;
    cout << "正在添加新的期刊信息，当前已有期刊数目:" << Bcount << endl;
    cout << "输入 ISBN 号： "; cin >> ISBN;
    cout << "请输入期刊名： "; cin >> Bname;
    cout << "请输入库存： "; cin >> stock;
    cout << "请输入价格： "; cin >> price;
    cout << "请输入作者名： "; cin >> express;
    Journals[Jcount] = Journal(ISBN, Bname, stock, price, express);
    Jcount++;
    cout << "添加成功！ " << endl;
}
//增加书刊
void Admin::AddBook()
{
    int select = 1;
    cout << setw(20) << ">>>>>>>>增加书刊<<<<<<<<" << endl
        << setw(20) << "1.书籍类" << endl
        << setw(20) << "2.期刊类" << endl
        << setw(20) << "0.退出该操作" << endl;
}

```

```

while (select)
{
    cout << "要对哪类进行增加操作（输入 0 返回主菜单）:";
    cin >> select;
    if (select < 0 || select > 2)
    {
        cerr << "非法值，请重新输入。" << endl;
        continue;
    }
    else
    {
        switch (select)
        {
            case 1: NewBook(); continue;
            case 2: NewJournal(); continue;
            case 0: break;
        }
    }
}

//删除书刊
void Admin::DelBook()
{
    string s = "1";
    while (1) {
        cout << "请输入作者名、出版社名、期刊名、书名或 ISBN 号（输入 0 返回主菜单）: ";
        cin >> s;
        if (s == "0") break; //退出判断
        int flag = 0; //查找成功状态符
        //查找书籍类
        for (int i = 0; i < Bcount; i++)
            if (Books[i].getAuthor() == s || Books[i].getISBN() == s || Books[i].getName() == s)
            {
                flag = 1;
                cout << "发现目标书籍如下，是否删除该书籍信息？（Y 是 / N 否）"
                     << endl << "-----"
                     << endl << Books[i] << "-----" << endl << "请选择: ";
                char c; cin >> c;
                if (c == 'Y' || c == 'y')
                {
                    for (int j = i; j < Bcount - 1; j++)
                    {
                        Books[j] = Books[j + 1];
                    }
                    Bcount--;
                }
            }
    }
}

```

```

        else continue;
    }
    else continue;
//查找期刊类
for (int i = 0; i < Jcount; i++)
    if (Journals[i].getExpress() == s || Journals[i].getISBN() == s || Journals[i].getName()
== s)
    {
        flag = 1;
        cout << "发现目标期刊如下，是否删除该期刊信息？（Y 是 / N 否）"
        << endl << "-----"
        << endl << Journals[i] << "-----" << endl << "请选择： ";
        char c; cin >> c;
        if (c == 'Y' || c == 'y')
        {
            for (int j = i; j < Jcount - 1; j++)
            {
                Journals[j] = Journals[j + 1];
            }
            Jcount--;
        }
        else continue;
    }
    else continue;
if (flag)cout << "删除成功！" << endl;
else cerr << "未找到目标书籍。" << endl;
}
}
//入库
void Admin::inStock()
{
    int select = 1;
    cout << setw(20) << ">>>>>>>>>入库书刊<<<<<<<<<" << endl
    << setw(20) << "1.书籍类" << endl
    << setw(20) << "2.期刊类" << endl
    << setw(20) << "0.退出该操作" << endl;
    while (select)
    {
        cout << "要对哪类进行入库操作（输入 0 返回主菜单）:";
        cin >> select;
        if (select < 0 || select > 2)
        {
            cerr << "非法值，请重新输入。" << endl;
            continue;
        }
        else

```

```

{
    switch (select)
    {
        case 1:
        {
            BooksPrint();
            int a, b;
            cout << "要操作的书籍编号是: "; cin >> a;
            if (a<1 || a>Bcount)
            {
                cerr << "序号错误, 请检查." << endl;
                break;
            }
            cout << "要增加的库存量是: "; cin >> b;
            Books[a - 1].addStock(b);
            continue;
        }
        case 2:
        {
            JournalsPrint();
            int a, b;
            cout << "要操作的期刊编号是: "; cin >> a;
            if (a<1 || a>Jcount)
            {
                cerr << "序号错误, 请检查." << endl;
                break;
            }
            cout << "要增加的库存量是: "; cin >> b;
            Books[a - 1].addStock(b);
            continue;
        }
        case 0: break;
    }
}

}

//修改书籍类
void modifyBook()
{
    int select = 1;
    BooksPrint();
    int a, b;
    string s;
    int st;
    double pr;
    cout << "要操作的书籍编号是: "; cin >> a;

```

```

    if (a<1 || a>Bcount)
    {
        cerr << "序号错误，请检查。" << endl;
    }
    else
    {
        cout << "1.ISBN\t2.书籍名称\t3.作者\t4.库存量\t5.价格\t0.放弃修改" << endl << "要修改
的信息是："; cin >> b;
        switch (b)
        {
            case 1: cout << "输入修改后的信息："; cin >> s; Books[a - 1].modISBN(s); cout <<
Books[a - 1] << "修改成功！" << endl; break;
            case 2: cout << "输入修改后的信息："; cin >> s; Books[a - 1].modName(s); cout <<
Books[a - 1] << "修改成功！" << endl; break;
            case 3: cout << "输入修改后的信息："; cin >> s; Books[a - 1].modAuthor(s); cout <<
Books[a - 1] << "修改成功！" << endl; break;
            case 4: cout << "输入修改后的信息："; cin >> st; Books[a - 1].modStock(st); cout <<
Books[a - 1] << "修改成功！" << endl; break;
            case 5: cout << "输入修改后的信息："; cin >> pr; Books[a - 1].modPrice(pr); cout <<
Books[a - 1] << "修改成功！" << endl; break;
            case 0: break;
        }
    }
}
//修改期刊类
void modifyJournal()
{
    int select = 1;
    JournalsPrint();
    int a, b;
    string s;
    int st;
    double pr;
    cout << "要操作的书籍编号是："; cin >> a;
    if (a<1 || a>Bcount)
    {
        cerr << "序号错误，请检查。" << endl;
    }
    else
    {
        cout << "1.ISBN\t2.书籍名称\t3.出版社\t4.库存量\t5.价格\t0.放弃修改" << endl << "要修
改的信息是："; cin >> b;
        switch (b)
        {
            case 1: cout << "输入修改后的信息："; cin >> s; Journals[a - 1].modISBN(s); cout <<
Journals[a - 1] << "修改成功！" << endl; break;

```



```

    << setw(20) << "2.期刊类" << endl
    << setw(20) << "0.退出该操作" << endl;
while (select)
{
    cout << "要对哪类进行销售操作（输入 0 返回主菜单）:";
    cin >> select;
    if (select < 0 || select > 2)
    {
        cerr << "非法值，请重新输入。" << endl;
        continue;
    }
    else
    {
        switch (select)
        {
            case 1:
            {
                BooksPrint();
                int a, b;
                cout << "要操作的书籍序号是："; cin >> a;
                if(a<1 || a>Bcount)
                {
                    cerr << "序号错误，请检查。" << endl;
                    break;
                }
                cout << "销售的库存量是："; cin >> b;
                Books[a - 1].outStock(b);
                continue;
            }
            case 2:
            {
                JournalsPrint();
                int a, b;
                cout << "要操作的期刊序号是："; cin >> a;
                if (a<1 || a>Jcount)
                {
                    cerr << "序号错误，请检查。" << endl;
                    break;
                }
                cout << "要销售的库存量是："; cin >> b;
                Books[a - 1].outStock(b);
                continue;
            }
            case 0: break;
        }
    }
}

```

```

    }
}
//输出所有库存
inline void Admin::printAll()
{
    BooksPrint();
    JournalsPrint();
}
//输出 ID.txt
void ID_OUT()
{
    file.open("ID.txt", ios::out);
    if (!file)
    {
        cerr << "未检测到账户列表文件 ID.txt! 保存失败。" << endl;
        exit(1);
    }
    for (int i = 0; i < 5; i++)
        file << ID[i] << endl;
    file.close();
}
//读入 ID.txt
void ID_IN()
{
    file.open("ID.txt", ios::in);
    if (!file)
    {
        cerr << "未检测到预置账户列表文件 ID.txt, 请查看源代码末端的注释, 现已使用默认测试
账户登录。" << endl;
        ID_Value = "超级管理员";
        launch();
    }
    for (int i = 0; i < 5; i++)
        file >> ID[i];
    file.close();
}
//切换账户
void ID_Switch()
{
    ID_IN();
    for (int i = 0; i < 5; i++)
        cout << i + 1 << "\t" << ID[i] << endl;
    cout << "请选择要登录的账户: "; cin >> ID_flag;

    if (ID_flag == 5 && ID[ID_flag-1] == "选此项以新建账户")
    {

```

```

        string tmp;
        cout << "请输入新账户的 ID: "; cin >> tmp;
        ID[4] = ID_Value = tmp;
        cout << "新建账户成功并已写入账户列表！您现在拥有所有权限。" << endl;
        ID_OUT();
    }
    else if (ID_flag < 1 || ID_flag > 5)
    {
        cerr << "该账户不存在！程序将退出。" << endl;
        abort();
    }
    else ID_Value = ID[ID_flag - 1];
    cout << "-----" << endl;
}
//导出库存信息
void fileOut()
{
    file.open("书籍.txt", ios::out);
    file << "ISBN\t" << setw(25) << "书名" << setw(20) << "作者" << setw(15) << "价格" <<
    setw(15) << "库存" << endl;
    for (int i = 0; i < Bcount; i++) file << Books[i];
    file.close();
    file.open("期刊.txt", ios::out);
    cout.setf(ios_base::left);
    file << "ISBN\t" << setw(25) << "期刊名" << setw(20) << "出版社" << setw(20) << "价格" <<
    setw(15) << "库存" << endl;
    for (int i = 0; i < Jcount; i++) file << Journals[i];
    cout << "已成功输出为'书籍.txt'与'期刊.txt'" << endl;
    file.close();
}
//欢迎
void welcome()
{
    cout << "=====
    << endl << setw(30) << "欢迎使用小型书店进销存管理系统"
    << endl << "=====
    << endl << "注：该程序仅用于江西财经大学张老师《C++程序设计语言》课程之期末综合
性实验报告，作者：刘燕、李澍、钟琳"
    << endl << "-----"
    << endl << "预置账户列表： " << endl;
    ID_Switch();
    launch();
}
//运行中函数
void launch()
{

```

```

Admin admin(ID_Value);
int select = 1;
while (select)
{
    cout << setw(20) << "-----" << endl
        << "当前用户: " << admin.getID() << endl
        << "1.查看当前库存" << endl
        << "2.增加书刊信息" << endl
        << "3.修改书刊信息" << endl
        << "4.删除书刊信息" << endl
        << "5.增加书刊库存" << endl
        << "6.书刊销售系统" << endl
        << "7.当前财务统计" << endl
        << "8.更改管理员 ID" << endl
        << "9.切换登录账户" << endl
        << "10.将库存信息输出至文件" << endl
        << "0.退出书店系统" << endl
        << setw(20) << "-----" << endl;
    cout << "请输入操作序号: ";
    cin >> select;
    if (select < 0 || select > 10)
    {
        cerr << "非法值，请重新输入。" << endl;
        continue;
    }
    else
    {
        switch (select)
        {
            case 1: admin.printAll(); continue;
            case 2: admin.AddBook(); continue;
            case 3: admin.modify(); continue;
            case 4: admin.DelBook(); continue;
            case 5: admin.inStock(); continue;
            case 6: admin.outStock(); continue;
            case 7: admin.statics(); continue;
            case 8:
            {
                string id;
                cout << "输入目标 ID: ";
                cin >> id;
                admin.changeID(id);
                ID[ID_flag - 1] = id;
                cout << "更改成功! 当前 ID 为" << id << endl;
                ID_OUT();
                continue;
            }
        }
    }
}

```

```

    }
    case 9: system("cls"); welcome(); break;
    case 10: fileOut(); continue;
    case 0: break;
    }
}
}
}
//主函数
int main()
{
    welcome();
    return 0;
}

```

/* 本程序使用外部文件实现了预置账户切换功能模块，该文件如下：

** 可将下方附件置于编译完成的 **exe** 文件目录中，但由于做了异常处理，缺失文件不会导致程序不可用

** ID.txt

超级管理员

刘燕

李澈

钟琳

选此项以新建账户

*/



ID.txt

4. 检查所有文件，对输入的程序进行调试、编译、链接，该过程中的调试不再赘述。其中曾出现的主要问题为，虚构造函数的重载问题，搜集资料后成功解决。

三、结论

1、实验结果

(1) 程序初始界面

程序运行的初始界面展示了系统名称、程序作者以及用户选择，界面如下

```

=====
欢迎使用小型书店进销存管理系统
=====
注：该程序仅用于江西财经大学张老师《C++程序设计语言》课程之期末综合性实验报告，作者：刘燕、李淑、钟琳
-----
预置账户列表：
1      超级管理员
2      刘燕
3      李淑
4      钟琳
5      选此项以新建账户
请选择要登录的账户：

```

账户选择功能由文件读写流实现，预置的 ID 被保存于同目录下的 ID.txt 中，运行时读该文件并使用字符串数组保存以展示账户列表。选择哪个账户便会以该用户名为参数实例化一个 Admin 对象进入书店管理系统。

若选择第 5 项新建账户，则会进入新建账户分支，并在输入 ID 后自动实例化对象、将新 ID 保存至数组并写入 ID.txt 以实现数据留存。

```

请选择要登录的账户：
5
请输入新账户的ID：测试_张老师
新建账户成功并已写入账户列表！您现在拥有所有权限。
-----
测试_张老师，欢迎您登陆管理系统！所有权限已开启。

```

以下为正常测试下的状态：

```

预置账户列表：
1      超级管理员
2      刘燕
3      李淑
4      钟琳
5      选此项以新建账户
请选择要登录的账户：1
-----
超级管理员，欢迎您登陆管理系统！所有权限已开启。

```

若 ID.txt 文件不存在，如考虑到教师批阅时复制代码直接运行环境，特增加了文件检测失败的异常处理，避免程序不可用。在该情况下，会直接以默认账户登录。

```

=====
欢迎使用小型书店进销存管理系统
=====
注：该程序仅用于江西财经大学张老师《C++程序设计语言》课程之期末综合性实验报告，作者：刘燕、李淑、钟琳
-----
预置账户列表：
未检测到预置账户列表文件ID.txt，请将其放入可执行文件目录，现已使用默认测试账户登录。
超级管理员，欢迎您登陆管理系统！所有权限已开启。

```

(2) 程序主界面

选择账户登录后，程序会读取预置的数据并进入主界面：

```
超级管理员，欢迎您登陆管理系统！所有权限已开启。

当前用户：超级管理员
1.查看当前库存
2.增加书刊信息
3.修改书刊信息
4.删除书刊信息
5.增加书刊库存
6.书刊销售系统
7.当前财务统计
8.更改管理员ID
9.切换登录账户
10.将库存信息输出至文件
0.退出书店系统
```

这部分程序封装于 `launch()` 函数中，首先实例化 `Admin` 对象，再利用 `while + switch` 使界面不断检测用户输入的操作序号，实现操作的可重用性。

同样的，该界面也加入了异常处理，由于并未系统学习 C++ 的异常处理部分，只能以此告终，实际留有遗憾。

```
请输入操作序号：111
非法值，请重新输入。
```

(3) 查看当前库存功能

该功能会遍历当前的两个用以存储相应对象的对象数组，并以精心调制过的输出格式显示在控制台中，效果如下图：

```
请输入操作序号：1
书籍类
序号  ISBN      书名              作者      价格      库存
1      9787302257646  程序设计基础      张三       25       190
2      9787302219972  单片机技术及应用  李四       32       182

期刊类
序号  ISBN      期刊名            出版社      价格      库存
1      9787302257611  电脑爱好者        北京大学出版社  32       200
2      9787302257622  无线电年刊        江西财大出版社  35       190
```

同一个类的对象存储于对象数组中，实际上可理解为书籍信息是采用两个顺序表数据结构进行存储，顺序查找的时间复杂度为 $O(n)$ ，由于并非本实验重点，并未实现二分查找等效率更高的查找方式。

本模块的缺点：未判断对象数组空时的输出，但由于预置了数据，本程序出现该情况的可能性较小。

(4) 增加、删除、修改书刊信息

该功能的作用为对书籍类与期刊类的对象进行增加与删除，增加即为实例化新对象并追加到数组中，删除则为查找并删除某对象。

增加功能：

```
请输入操作序号：2
>>>>>>>>增加书刊<<<<<<<<<<
1.书籍类
2.期刊类
0.退出该操作
要对哪类进行增加操作（输入0返回主菜单）：1
-----
正在添加新的书籍信息，当前已有书籍数目：2
输入ISBN号：9787302257612
请输入书名：C++程序设计
请输入库存：200
请输入价格：79.8
请输入作者名：张玲
添加成功！
要对哪类进行增加操作（输入0返回主菜单）：0
```

该界面仍具有合理的界面设计与提示，使用输入的参数进行对象实例化后，会将相应类的 `count` 变量加 1。

删除功能：


```

请输入操作序号: 3
请输入作者名、出版社名、期刊名、书名或ISBN号 (输入0返回主菜单): 张三
发现目标书籍如下, 是否删除该书籍信息? (Y 是 / N 否)
-----
9787302257646    程序设计基础                张三                25                190
-----
Y
删除成功!
请输入作者名、出版社名、期刊名、书名或ISBN号 (输入0返回主菜单): 0

请输入作者名、出版社名、期刊名、书名或ISBN号 (输入0返回主菜单): 电脑爱好者
发现目标期刊如下, 是否删除该期刊信息? (Y 是 / N 否)
-----
9787302257611    电脑爱好者                北京大学出版社    32                200
-----
请选择: Y
删除成功!

```

仅需输入某一个完全匹配的字符串，该模块会查找两个对象数组中符合条件的对象，并返回其对应的“书籍类”或“期刊类”的信息，确认是否删除。删除后的库存信息：

```

请输入操作序号: 1
----- 书籍类 -----
序号    ISBN                书名                作者                价格                库存
1        9787302219972        单片机技术及应用    李四                32                182

----- 期刊类 -----
序号    ISBN                期刊名                出版社                价格                库存
1        9787302257622        无线电年刊          江西财大出版社    35                190

```

若未找到对应信息，会提示未找到：

```

请输入作者名、出版社名、期刊名、书名或ISBN号 (输入0返回主菜单): 1
未找到目标书籍。

```

修改功能：

该界面实现了每个书籍对象的具体信息的修改，实现较为简单，大部分封装在基类中，使用成员函数访问：

```

>>>>>>>>>修改书刊信息<<<<<<<<<<<<
1.书籍类
2.期刊类
0.退出该操作
要对哪类进行修改操作（输入0返回主菜单）:1
      书籍类
序号   ISBN       书名           作者           价格           库存
1       9787302257646 程序设计基础     张三           25             203
2       9787302219972 单片机技术及应用 李四           32             182

要操作的书籍编号是: 1
1.ISBN  2.书籍名称  3.作者  4.库存量  5.价格  0.放弃修改
要修改的信息是: 3
输入修改后的信息: 王五
9787302257646  程序设计基础           王五           25             203
修改成功!

```

本模块的缺点：由于缺乏相关知识，希望在删除信息功能中实现模糊查询但无力实现，也未对同作者、同名等情况进行处理。

（5）增加书刊库存

该功能可将库存增加，实质上是对库存数据的简单增加操作，实现较为简单，效果如下图：

```

>>>>>>>>>入库书刊<<<<<<<<<<<<
      1.书籍类
      2.期刊类
      0.退出该操作
要对哪类进行入库操作（输入0返回主菜单）:1
      书籍类
序号   ISBN       书名           作者           价格           库存
1       9787302257646 程序设计基础     张三           25             190
2       9787302219972 单片机技术及应用 李四           32             182

要操作的书籍编号是: 1
要增加的库存量是: 13
库存增加成功! 目前库存为: 203

```

输入合法性校验：

```

要操作的书籍编号是: 4
序号错误, 请检查。

```

（6）书刊销售系统

该功能用于销售书刊，实质上是对库存的减少操作以及将销售信息传给统计功能，实现较为简单，效果如下图：

```

请输入操作序号: 5
>>>>>>>>>销售书刊<<<<<<<<<<
    1.书籍类
    2.期刊类
    0.退出该操作
要对哪类进行销售操作(输入0返回主菜单):1
    书籍类
序号  ISBN      书名                作者      价格      库存
1      9787302257646  程序设计基础      张三      25      190
2      9787302219972  单片机技术及应用  李四      32      182

要操作的书籍序号是: 1
销售的库存量是: 100
销售成功! 目前库存为: 90
销售额为: 2500

```

也同样对输入的合法性进行了校验:

```

序号  ISBN      书名                作者      价格      库存
1      9787302257646  程序设计基础      张三      25      90
2      9787302219972  单片机技术及应用  李四      32      182

要操作的书籍序号是: 4
序号错误, 请检查。

```

```

要操作的书籍序号是: 2
销售的库存量是: 1000
库存不足, 请检查, 目前库存仅有182。

```

(7) 当前财务统计

该功能会简单统计当前程序生命周期内的总销售额, 并可选择导出至销售额.txt

```

请输入操作序号: 6
-----
                销售统计
总销售额: 2500
是否将销售额导出至文件? (Y/N) Y
导出成功!
-----

```

效果:



销售额.txt - 记事本

文件(F) 编辑(E) 格式(O) 查找

总销售额: 2500

该模块缺点: 实现功能过少, 主要由于时间与知识储备不足。

(8) 更改管理员 ID

该功能可以更改当前登录的管理员的 ID，使用常成员函数实现，并重新输出主程序信息演示更改结果。

```
请输入操作序号: 7
输入目标ID: 测试管理员
更改成功! 当前ID为测试管理员

当前用户: 测试管理员
1.查看当前库存
2.增加书刊信息
3.删除书刊信息
4.增加书刊库存
5.书刊销售系统
6.当前财务统计
7.更改管理员ID
8.切换登录账户
9.将库存信息输出至文件
0.退出书店系统
```

(9) 切换登录账户

该功能可以实现简单的账户切换功能，使用 `system("cls");` 语句调用命令行命令清屏，并重新运行 `welcome()` 函数。

```
=====
欢迎使用小型书店进销存管理系统
=====
注: 该程序仅用于江西财经大学张老师《C++程序设计语言》课程之期末综合性实验报告, 作者: 刘燕、李澍、钟琳
=====
预置账户列表:
1      超级管理员
2      刘燕
3      李澍
4      钟琳
5      测试_张老师
请选择要登录的账户:
```

若之前新建了账户，该功能会读取新的 ID.txt，并显示新账户 ID。

(10) 将库存信息输出至文件

该功能会将当前更改过的库存信息导出至期刊.txt 与书籍.txt 两个文件，实现操作的数据留存，对库存、书刊信息的增删都会保留。

```
请输入操作序号: 9
已成功输出为'书籍.txt'与'期刊.txt'
```

实现效果如下：

期刊.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

ISBN	期刊名	出版社	价格	库存
9787302257611	电脑爱好者	北京大学出版社	32	200
9787302257622	无线电年刊	江西财大出版社	35	190
9173875813859	电脑报世界	甲搜得就	3	100

期刊.txt

书籍.txt - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

ISBN	书名	作者	价格	库存
9787302257646	程序设计基础	张三	25	90
9787302219972	单片机技术及应用	李四	32	182
9787302257647	C++程序设计	张玲	79.8	2000

书籍.txt

该模块的缺点：输出的格式难以控制，对 `fstream` 与 `iostream` 的掌握不深。

(11) 退出书店系统

该操作会结束该程序。

2、分析讨论

(1) 运用面向对象的方法设计程序，会使思路及编程方法更加贴近直觉。不仅方便了功能的分类、接口的编写，也使类继承等能实际用于程序的结构实现。本程序中的部分功能是为了探索类对于现实实用情况下的实现，如管理员相关功能，这些功能可能在本程序中实用性不足，但对我们认识面向对象程序设计有很大的帮助。

(2) 本次实验编写的代码仍有许多不足，如：

i) 未实现账户密码验证，因缺失加密相关知识，而明文保存密码完全无意义，导致账户登录模块似乎也空有其表；

ii) 未对对象数组索引边界值进行处理，可能导致越界而出现程序乱码等。

(3) 同时，本次实验也有一些我们认为较为优秀的实现：

i) 基本实现了该系统的可用、实用，提供了向导性良好、交互友好的界面；

- ii) 对可能出现的错误输入基本实现了错误检查与处理，提高了程序的稳定性；
- iii) 充分运用了所学知识，践行了实验目的等。
- iv) 功能模块较多，可更好地体现编程水平。

四、指导教师评语及成绩：

评语：

成绩：

指导教师签名：

批阅日期：