1. 基础1
2. main.cpp

#include <iostream>

#include <string>

**using** **namespace** std;

#define M 4

#define N 5

**int**\*\* apply()   //申请空间

{

**int** \*\*mat = **new** **int**\*[M];

**for**(**int** i = 0;i<M;i++)

    {

        mat[i] = **new** **int**[N];

    }

**return** mat;

}

**void** init(**int** \*\*mat)    //初始化

{

    cout<<"请依次输入"<<M<<"\*"<<N<<"的矩阵元素"<<endl;

**for**(**int** i = 0;i<M;i++)

    {

**for**(**int** j = 0;j<N;j++)

        {

**int** x;

            cin>>x;

            mat[i][j] = x;

        }

    }

}

**void** print(**int** \*\*mat)   //输出矩阵

{

    cout<<"矩阵如下："<<endl;

**for**(**int** i = 0;i<M;i++)

    {

**for**(**int** j = 0;j<N;j++)

        {

            cout<<mat[i][j]<<" ";

        }

        cout<<endl;

    }

}

**void** add(**int** \*\*mat1, **int** \*\*mat2, **int** \*\*mat) //mat1+mat2存入mat

{

**for**(**int** i = 0;i<M;i++)

    {

**for**(**int** j = 0;j<N;j++)

        {

            mat[i][j] = mat1[i][j]+mat2[i][j];

        }

    }

}

**void** sub(**int** \*\*mat1, **int** \*\*mat2, **int** \*\*mat)     //mat1-mat2存入mat

{

**for**(**int** i = 0;i<M;i++)

    {

**for**(**int** j = 0;j<N;j++)

        {

            mat[i][j] = mat1[i][j]-mat2[i][j];

        }

    }

}

**void** release(**int** \*\*mat)     //释放空间

{

**for**(**int** i = 0;i<M;i++) **delete** []mat[i];

**delete** []mat;

}

**int** main() {

**int** \*\*mat1 = apply();

**int** \*\*mat2 = apply();

**int** \*\*mat3 = apply();

    init(mat1);

    init(mat2);

    string s;

    cout<<"请输入add或sub进行矩阵相加或相减，quit退出"<<endl;

    cin>>s;

**do** {

**while**(cin.fail())

        {

            cout<<"输入不合法，请重新输入"<<endl;

            //清除不合法输入标志

            cin.clear();

            //清空输入缓冲区

            cin.ignore(numeric\_limits<std::streamsize>::max(),

                       '\n');

            //重新输入

            cin>>s;

        }

**if**(s=="add")

        {

            add(mat1, mat2, mat3);

            cout<<"相加所得的";

            print(mat3);

        } **else** **if**(s=="sub") {

            sub(mat1, mat2, mat3);

            cout<<"相减所得的";

            print(mat3);

        }

        cout<<"请输入add或sub进行矩阵相加或相减,quit退出"<<endl;

        cin>>s;

    } **while**(s!="quit");

    release(mat1);

    release(mat2);

    release(mat3);

**return** 0;

}

1. 基础2
2. main.cpp

#include <iostream>

#include <cmath>

#include "point.hpp"

#include "circle.hpp"

**int** main() {

**double** a,b,c;

    Circle c1,c2;

    cout<<"输入d以采用默认构造的圆1，输入其他任意键自定义"<<endl;

**char** ch,ch2;

    cin>>ch;

**if**(ch!='d')

    {

        cout<<"请输入圆1的圆心坐标及半径"<<endl;

        cin>>a>>b>>c;

        c1 = Circle(a,b,c);

    }

    cout<<"输入d以采用默认构造的圆2,输入其他任意键自定义"<<endl;

    cin>>ch2;

**if**(ch2!='d')

    {

        cout<<"请输入圆2的圆心坐标及半径"<<endl;

        cin>>a>>b>>c;

        c2 = Circle(a,b,c);

        //circle构造出一个新的圆，并调用拷贝赋值函数，之后新的圆析构

    }

**if**(c1.xiangjiao(c1,c2)) cout<<"两圆相交"<<endl;

**else** cout<<"两圆不相交"<<endl;

    Point p;

    ++p;

    p.output();

    --p;

    p.output();

    p++;

    p.output();

    p--;

    p.output();

**return** 0;

}

1. circle.hpp

#ifndef circle\_hpp

#define circle\_hpp

**class** Circle{

**private**:

    Point center;

**double** radius;

**public**:

    Circle():radius(1){cout<<"默认构造了一个圆心为（0，0），半径为1的圆"<<endl;}

    Circle(**double** a, **double** b, **double** c):center(a,b),radius(c){

        cout<<"你构造了一个圆心为（"<<a<<"，"<<b<<"），半径为"<<c<<"的圆"<<endl;

    }

    Circle(Circle &c):center(c.center),radius(c.radius){

        cout<<"拷贝构造圆"<<endl;

    }

    ~Circle(){cout<<"圆析构"<<endl;}

    Point get\_center(){**return** center;}

**double** get\_radius(){**return** radius;}

**bool** xiangjiao(Circle a, Circle b);

};

**bool** Circle::xiangjiao(Circle a, Circle b){

    //两次圆拷贝对应两次圆点析构

    Point centera(a.center), centerb(b.center);

    //两次点拷贝对应两次点析构

**double** r1 = a.radius,r2 = b.radius;

**double** dis = centera.get\_dis(centera, centerb);

    //两次点拷贝对应两次点析构

**if**(dis>r1+r2) **return** **false**;

**else** **return** **true**;

}

#endif /\* circle\_hpp \*/

1. point.hpp

#ifndef point\_hpp

#define point\_hpp

**using** std::cout;

**using** std::endl;

**using** std::cin;

**class** Point{

**private**:

**double** x;

**double** y;

**public**:

    Point():x(0),y(0){cout<<"默认构造了（0，0）点"<<endl;}

    Point(**double** a, **double** b):x(a),y(b){cout<<"有参数构造点"<<endl;}

    Point(Point &p):x(p.x),y(p.y){cout<<"拷贝构造点"<<endl;}

    ~Point(){cout<<"点析构"<<endl;}

**double** get\_dis(Point a, Point b);

**double** get\_x(){**return** x;}

**double** get\_y(){**return** y;}

**void** set\_x(**double** a){x = a;}

**void** set\_y(**double** a){y = a;}

**void** output(){cout<<"点("<<x<<","<<y<<")"<<endl;}

    Point **operator** ++();//前置

    Point **operator** --();

    Point **operator** ++(**int**);//后置

    Point **operator** --(**int**);

};

Point Point::**operator** ++(){//调用了拷贝构造函数

    ++x;

    ++y;

**return** \***this**;

}

Point Point::**operator** --(){

    --x;

    --y;

**return** \***this**;

}

Point Point::**operator** ++(**int**){

    x++;

    y++;

**return** \***this**;

}

Point Point::**operator** --(**int**){

    x--;

    y--;

**return** \***this**;

}

**double** Point::get\_dis(Point a, Point b)

{

**double** dis = 0;

    dis = sqrt(pow((a.x-b.x),2)+pow(a.y-b.y,2));

**return** dis;

}

#endif /\* point\_hpp \*/

1. 基础3
2. main.cpp

#include <iostream>

#include <cmath>

#include "shape.hpp"

#include "rectangle.hpp"

#include "circle.hpp"

#include "square.hpp"

**using** **namespace** std;

**int** main(**int** argc, **const** **char** \* argv[]) {

    Rectangle rec(1,2);

    cout<<rec.area()<<endl;

    Circle c(2);

    cout<<c.area()<<endl;

    Square s(2);

    cout<<s.area()<<endl;

    cout<<"\n下面是指针操作部分\n"<<endl;

    Shape \*ptr = &rec;

    cout<<ptr->area()<<endl;

    ptr = &c;

    cout<<ptr->area()<<endl;

    ptr = &s;

    cout<<ptr->area()<<endl;

**return** 0;

}

1. shape.hpp

#ifndef shape\_hpp

#define shape\_hpp

#include <stdio.h>

#include <iostream>

**using** **namespace** std;

**class** Shape{//抽象类

**public**:

    Shape(){cout<<"构造Shape类"<<endl;}

**virtual** ~Shape(){cout<<"析构Shape类"<<endl;}

//    double area() {return -1;}

**virtual** **double** area() = 0;//纯虚函数

};

#endif /\* shape\_hpp \*/

1. circle.hpp

#ifndef circle\_hpp

#define circle\_hpp

#include <stdio.h>

#include <iostream>

**using** **namespace** std;

**class** Shape;

**class** Circle : **public** Shape{

**private**:

**double** r;

**public**:

    Circle(){cout<<"无参数构造Circle类"<<endl;}

    Circle(**double** a):r(a){cout<<"有参数构造Circle类"<<endl;}

    ~Circle(){cout<<"析构Circle类"<<endl;}

**double** area();

};

**double** Circle::area(){

**return** M\_PI\*r\*r;

}

#endif /\* circle\_hpp \*/

1. rectangle.hpp

#ifndef rectangle\_hpp

#define rectangle\_hpp

#include <stdio.h>

#include <iostream>

**class** Shape;

**using** **namespace** std;

**class** Rectangle : **public** Shape{

**private**:

**double** height;

**double** width;

**public**:

    Rectangle(){cout<<"无参数构造Rectangle类"<<endl;}

    Rectangle(**double** a, **double** b):height(a),width(b){cout<<"有参数构造Rectangle类"<<endl;}

    ~Rectangle(){cout<<"析构Rectangle类"<<endl;}

**double** area();

};

**double** Rectangle::area(){

**return** width\*height;

}

#endif /\* rectangle\_hpp \*/

1. square.hpp

#ifndef square\_hpp

#define square\_hpp

#include <stdio.h>

#include <iostream>

**using** **namespace** std;

**class** Rectangle;

**class** Square : **public** Rectangle{

**private**:

**double** edge;

**public**:

    Square(){cout<<"无参数构造Square类"<<endl;}

    Square(**double** a):edge(a){cout<<"有参数构造Square类"<<endl;}

    ~Square(){cout<<"析构Square类"<<endl;}

**double** area();

};

**double** Square::area(){

**return** edge\*edge;

}

#endif /\* square\_hpp \*/

1. 基础4
2. main.cpp

#include <iostream>

#include <cmath>

#include <time.h>

#include <cstdlib>

**using** **namespace** std;

**void** guess() {

    //设置随机数种子

    srand((**unsigned**)time(**nullptr**));

    //生成随机数

**int** n = rand()%1000+1;

**int** x;

**do**{

        cout<<"请输入你猜的数字，范围在0到1000"<<endl;

        cin>>x;

**while**(cin.fail()){//如果输入不合法

            cout<<"输入不合法，请重新输入"<<endl;

            //清除不合法输入标志

            cin.clear();

            //清空输入缓冲区

            cin.ignore(numeric\_limits<std::streamsize>::max(),

                       '\n');

            //重新输入

            cin>>x;

        }

**if**(x>n) cout<<"大了"<<endl;

**else** **if**(x<n) cout<<"小了"<<endl;

**else** cout<<"回答正确"<<endl;

    }**while**(x!=n);

}

**int** main(**int** argc, **const** **char** \* argv[]) {

    guess();

**return** 0;

}

1. 基础5
2. 基础5.1

（1）main.cpp

#include <iostream>

#include <cmath>

#include "shape.hpp"

#include "rectangle.hpp"

#include "circle.hpp"

#include "square.hpp"

**using** **namespace** std;

**int** main(**int** argc, **const** **char** \* argv[]) {

    Rectangle rec(1,2);

    cout<<rec.area()<<endl;

    Circle c(2);

    cout<<c.area()<<endl;

    Square s(2);

    cout<<s.area()<<endl;

    cout<<"\n下面是指针操作部分\n"<<endl;

    Shape \*ptr = &rec;

    cout<<ptr->area()<<endl;

    ptr = &c;

    cout<<ptr->area()<<endl;

    ptr = &s;

    cout<<ptr->area()<<endl;

**return** 0;

}

（2）shape.hpp

#ifndef shape\_hpp

#define shape\_hpp

#include <stdio.h>

#include <iostream>

**using** **namespace** std;

**class** Shape{//抽象类

**public**:

    Shape(){cout<<"构造Shape类"<<endl;}

**virtual** ~Shape(){cout<<"析构Shape类"<<endl;}

//    double area() {return -1;}

**virtual** **double** area() = 0;//纯虚函数

};

#endif /\* shape\_hpp \*/

1. circle.hpp

#ifndef circle\_hpp

#define circle\_hpp

#include <stdio.h>

#include <iostream>

**using** **namespace** std;

**class** Shape;

**class** Circle : **public** Shape{

**private**:

**double** r;

**public**:

    Circle(){cout<<"无参数构造Circle类"<<endl;}

    Circle(**double** a):r(a){cout<<"有参数构造Circle类"<<endl;}

    ~Circle(){cout<<"析构Circle类"<<endl;}

**double** area();

};

**double** Circle::area(){

**return** M\_PI\*r\*r;

}

#endif /\* circle\_hpp \*/

（3）rectangle.hpp

#ifndef rectangle\_hpp

#define rectangle\_hpp

#include <stdio.h>

#include <iostream>

**class** Shape;

**using** **namespace** std;

**class** Rectangle : **public** Shape{

**private**:

**double** height;

**double** width;

**public**:

    Rectangle(){cout<<"无参数构造Rectangle类"<<endl;}

    Rectangle(**double** a, **double** b):height(a),width(b){cout<<"有参数构造Rectangle类"<<endl;}

    ~Rectangle(){cout<<"析构Rectangle类"<<endl;}

**double** area();

};

**double** Rectangle::area(){

**return** width\*height;

}

#endif /\* rectangle\_hpp \*/

（4）square.hpp

#ifndef square\_hpp

#define square\_hpp

#include <stdio.h>

#include <iostream>

**using** **namespace** std;

**class** Rectangle;

**class** Square : **public** Rectangle{

**private**:

**double** edge;

**public**:

    Square(){cout<<"无参数构造Square类"<<endl;}

    Square(**double** a):edge(a){cout<<"有参数构造Square类"<<endl;}

    ~Square(){cout<<"析构Square类"<<endl;}

**double** area();

};

**double** Square::area(){

**return** edge\*edge;

}

#endif /\* square\_hpp \*/

1. 基础5.2

（1）main.cpp

#include <iostream>

#include <cmath>

#include "point.hpp"

#include "circle.hpp"

**int** main() {

**double** a,b,c;

    Circle c1,c2;

    cout<<"输入d以采用默认构造的圆1，输入其他任意键自定义"<<endl;

**char** ch,ch2;

    cin>>ch;

**if**(ch!='d')

    {

        cout<<"请输入圆1的圆心坐标及半径"<<endl;

        cin>>a>>b>>c;

        c1 = Circle(a,b,c);

    }

    cout<<"输入d以采用默认构造的圆2,输入其他任意键自定义"<<endl;

    cin>>ch2;

**if**(ch2!='d')

    {

        cout<<"请输入圆2的圆心坐标及半径"<<endl;

        cin>>a>>b>>c;

        c2 = Circle(a,b,c);

        //circle构造出一个新的圆，并调用拷贝赋值函数，之后新的圆析构

    }

**if**(c1.xiangjiao(c1,c2)) cout<<"两圆相交"<<endl;

**else** cout<<"两圆不相交"<<endl;

    Point p;

    ++p;

    p.output();

    --p;

    p.output();

    p++;

    p.output();

    p--;

    p.output();

**return** 0;

}

（2）circle.hpp

#ifndef circle\_hpp

#define circle\_hpp

**class** Circle{

**private**:

    Point center;

**double** radius;

**public**:

    Circle():radius(1){cout<<"默认构造了一个圆心为（0，0），半径为1的圆"<<endl;}

    Circle(**double** a, **double** b, **double** c):center(a,b),radius(c){

        cout<<"你构造了一个圆心为（"<<a<<"，"<<b<<"），半径为"<<c<<"的圆"<<endl;

    }

    Circle(Circle &c):center(c.center),radius(c.radius){

        cout<<"拷贝构造圆"<<endl;

    }

    ~Circle(){cout<<"圆析构"<<endl;}

    Point get\_center(){**return** center;}

**double** get\_radius(){**return** radius;}

**bool** xiangjiao(Circle a, Circle b);

};

**bool** Circle::xiangjiao(Circle a, Circle b){

    //两次圆拷贝对应两次圆点析构

    Point centera(a.center), centerb(b.center);

    //两次点拷贝对应两次点析构

**double** r1 = a.radius,r2 = b.radius;

**double** dis = centera.get\_dis(centera, centerb);

    //两次点拷贝对应两次点析构

**if**(dis>r1+r2) **return** **false**;

**else** **return** **true**;

}

#endif /\* circle\_hpp \*/

（3）point.hpp

#ifndef point\_hpp

#define point\_hpp

**using** std::cout;

**using** std::endl;

**using** std::cin;

**class** Point{

**private**:

**double** x;

**double** y;

**public**:

    Point():x(0),y(0){cout<<"默认构造了（0，0）点"<<endl;}

    Point(**double** a, **double** b):x(a),y(b){cout<<"有参数构造点"<<endl;}

    Point(Point &p):x(p.x),y(p.y){cout<<"拷贝构造点"<<endl;}

    ~Point(){cout<<"点析构"<<endl;}

**double** get\_dis(Point a, Point b);

**double** get\_x(){**return** x;}

**double** get\_y(){**return** y;}

**void** set\_x(**double** a){x = a;}

**void** set\_y(**double** a){y = a;}

**void** output(){cout<<"点("<<x<<","<<y<<")"<<endl;}

    Point **operator** ++();//前置

    Point **operator** --();

    Point **operator** ++(**int**);//后置

    Point **operator** --(**int**);

};

Point Point::**operator** ++(){//调用了拷贝构造函数

    ++x;

    ++y;

**return** \***this**;

}

Point Point::**operator** --(){

    --x;

    --y;

**return** \***this**;

}

Point Point::**operator** ++(**int**){

    x++;

    y++;

**return** \***this**;

}

Point Point::**operator** --(**int**){

    x--;

    y--;

**return** \***this**;

}

**double** Point::get\_dis(Point a, Point b)

{

**double** dis = 0;

    dis = sqrt(pow((a.x-b.x),2)+pow(a.y-b.y,2));

**return** dis;

}

#endif /\* point\_hpp \*/

1. 综合1
2. main.cpp

#include <iostream>

#include <fstream>

#include <stdlib.h>

#include <time.h>

#include <stdio.h>

#include <vector>

#include <algorithm>

#include <math.h>

#include <string>

#include "account.hpp"

#include "account\_system.hpp"

#include "user.hpp"

#include "merchant.hpp"

#include "consumer.hpp"

#include "goods.hpp"

#include "book.hpp"

#include "food.hpp"

#include "clothing.hpp"

**int** main(**int** argc, **const** **char** \* argv[]) {

    AccountSystem account\_system;

    account\_system.mainMenu();

**return** 0;

}

1. account\_system.hpp

#ifndef account\_system\_hpp

#define account\_system\_hpp

#include <stdio.h>

#include <fstream>

#include <iostream>

#include <string>

#include <map>

#include "account.hpp"

#include "merchant.hpp"

#include "consumer.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Account;

**class** Goods;

**class** Merchant;

**class** Consumer;

**class** AccountSystem {

**public**:

    AccountSystem();

    ~AccountSystem();

**void** readFile();    //加载文件中存储的信息

**void** writeFile();   //保存进文件

**void** mainMenu();    //主界面,进行功能选择

**void** merchantOperate();

**void** consumerOperate();

**void** showGoods();   //展示商品

**void** showAccounts();

**void** login();   //用户登陆

**void** signin();  //用户注册

**void** signin(**const** string &id);

**private**:

    map<string, Account\*> id2account;    //账号到账户的映射

    vector<Account\*> accounts;

    map<string, Merchant\*> merchants;

    map<string, Consumer\*> consumers;

    //存储所有的商品

    vector<Goods\*> goods;

    Account \*online\_account;

};

#endif /\* account\_system\_hpp \*/

1. account\_system.cpp

#include <fstream>

#include <iostream>

#include <stdio.h>

#include "account.hpp"

#include "account\_system.hpp"

#include "user.hpp"

#include "merchant.hpp"

#include "consumer.hpp"

#include "goods.hpp"

#include "book.hpp"

#include "food.hpp"

#include "clothing.hpp"

**using** std::ifstream;

**using** std::ofstream;

**using** std::ios;

AccountSystem::AccountSystem() {

    readFile();

}

AccountSystem::~AccountSystem() {

    writeFile();

**for**(**int** i = 0; i<accounts.size(); i++) {

**delete** accounts[i];

    }

**for**(**int** i = 0; i<goods.size(); i++) {

**delete** goods[i];

    }

**for**(**auto** item : merchants) {

**if**(item.second!=**nullptr**) **delete** item.second;

    }

**for**(**auto** item : consumers) {

**if**(item.second!=**nullptr**) **delete** item.second;

    }

}

**void** AccountSystem::readFile() {

    ifstream fin\_account;

    fin\_account.open("/Users/tom/account.dic",ios::in);

**if** (!fin\_account.is\_open()) {

        cout << "文件打开失败"<<endl;

        exit(EXIT\_FAILURE);

    }

    string id, psw, type;

**double** balance;

**while**(fin\_account>>id>>psw>>balance>>type) {

        Account \*account = **new** Account(id,psw,balance,type);

        id2account[id] = account;

        accounts.push\_back(account);

        Merchant\* merchant;

        Consumer\* consumer;

**if**(type=="merchant") {

            merchant = **new** Merchant(account);

            merchants[account->get\_id()] = merchant;

        }

**else** {

            consumer = **new** Consumer(account);

            consumers[account->get\_id()] = consumer;

        }

    }//while

    fin\_account.close();

    ifstream fin\_goods;

    fin\_goods.open("/Users/tom/goods.dic",ios::in);

**if** (!fin\_goods.is\_open()) {

        cout << "文件打开失败"<<endl;

        exit(EXIT\_FAILURE);

    }

    string name, description;

**double** price;

**int** remaining;

**while**(fin\_goods>>name>>description>>price>>remaining>>type>>id) {

        Goods \*good;

**if**(type=="book") {

            good = **new** Book(name,description,price,remaining,type,merchants[id]);

        } **else** **if**(type=="food") {

            good = **new** Food(name,description,price,remaining,type,merchants[id]);

        } **else** {

            good = **new** Clothing(name,description,price,remaining,type,merchants[id]);

        }

        goods.push\_back(good);

    }//while

    fin\_goods.close();

}

**void** AccountSystem::writeFile() {

    ofstream fout\_account;

    fout\_account.open("/Users/tom/account.dic",ios::out|ios::trunc);

**if** (!fout\_account.is\_open()) {

        cout <<"文件打开失败"<<endl;

        exit(EXIT\_FAILURE);

    }

**for**(**auto** account : accounts) {

        fout\_account<<account->get\_id()<<" "<<account->get\_password()<<" "<<account->get\_balance()<<" "<<account->get\_type()<<endl;

    }

    fout\_account.close();

    ofstream fout\_goods;

    fout\_goods.open("/Users/tom/goods.dic",ios::out|ios::trunc);

**if** (!fout\_goods.is\_open()) {

        cout <<"文件打开失败"<<endl;

        exit(EXIT\_FAILURE);

    }

**for**(**auto** good : goods) {

        fout\_goods<<good->get\_name()<<" "<<good->get\_description()<<"  "<<good->get\_price()<<" "<<good->get\_remaining()<<" "<<good->get\_type()<<" "<<good->get\_merchant()->get\_account()->get\_id()<<endl;

    }

    fout\_goods.close();

}

**void** AccountSystem::mainMenu() {

**int** x;

    cout << endl<<"       请选择操作序号！(0~2)" << endl;

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    cout << "\*                                 \*" << endl;

    cout << "\*              0.退出              \*" << endl;

    cout << "\*              1.登陆              \*" << endl;

    cout << "\*              2.注册              \*" << endl;

    cout << "\*         3.查看所有账号信息         \*" << endl;

    cout << "\*                                 \*" << endl;

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    cin>>x;

**while**(x!=0)

    {

**switch**(x) {

**case** 1:

                login();

**break**;

**case** 2:

                signin();

**break**;

**case** 3:

                showAccounts();

**break**;

**default**:

                cout<<"请输入正确的数字哦～"<<endl;

**break**;

        }

        cout << endl<<"       请选择操作序号！(0~2)" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*              0.退出              \*" << endl;

        cout << "\*              1.登陆              \*" << endl;

        cout << "\*              2.注册              \*" << endl;

        cout << "\*         3.查看所有账号信息         \*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cin>>x;

    }

}

**void** AccountSystem::signin() {

    string id,psw,check;

    cout<<"请输入您要注册的账号"<<endl;

    cin>>id;

**while**(id2account.count(id)) {

        cout<<"该账号已存在，请更换后重新输入"<<endl;

        cin>>id;

    }

**do** {

        cout<<"请输入密码"<<endl;

        cin>>psw;

        cout<<"请确认密码"<<endl;

        cin>>check;

**if**(check!=psw) cout<<"两次输入密码不同"<<endl;

    } **while**(check!=psw);

    cout<<"请选择您要注册的账号类型(consumer或merchant）"<<endl;

    string type;

    cin>>type;

    Account \*new\_account = **new** Account(id,psw,type);

    //新注册账号对应的用户存入merchants或consumers

**if**(type=="merchant") {

        Merchant \*new\_merchant = **new** Merchant(new\_account);

        merchants[id] = new\_merchant;

    } **else** {

        Consumer \*new\_consumer = **new** Consumer(new\_account);

        consumers[id] = new\_consumer;

    }

    accounts.push\_back(new\_account);

    id2account[id] = new\_account;

    cout<<"恭喜您注册成功，您的账号是:"<<id<<endl;

}

**void** AccountSystem::signin(**const** string &id) {

    string psw,check;

    cout<<"您的账号是"<<id<<",";

**do** {

        cout<<"请输入密码"<<endl;

        cin>>psw;

        cout<<"请确认密码"<<endl;

        cin>>check;

**if**(check!=psw) cout<<"两次输入密码不同"<<endl;

    } **while**(check!=psw);

    cout<<"请选择您要注册的账号类型（consumer或merchant）"<<endl;

    string type;

    cin>>type;

    Account \*new\_account = **new** Account(id,psw,type);

    //新注册账号对应的用户存入merchants或consumers

**if**(type=="merchant") {

        Merchant \*new\_merchant = **new** Merchant(new\_account);

        merchants[id] = new\_merchant;

    } **else** {

        Consumer \*new\_consumer = **new** Consumer(new\_account);

        consumers[id] = new\_consumer;

    }

    accounts.push\_back(new\_account);

    id2account[id] = new\_account;

    cout<<"恭喜您注册成功，您的账号是:"<<id<<endl;

}

**void** AccountSystem::login() {

    string id,psw;

    cout<<"请输入您的账号"<<endl;

    cin>>id;

**int** x;

**while**(1){

**while**(id2account.find(id)==id2account.end()) {

            //如果没找到

            cout<<"账号不存在，注册请输入1，重新输入账户输入2"<<endl;

            cin>>x;

**while**(x!=1&&x!=2) {

                cout<<"请输入1或2"<<endl;

                cin>>x;

            }

**if**(x==1) {

                signin(id);

**break**;

            }

**else**

            {

                cout<<"请重新输入您的账号"<<endl;

                cin>>id;

            }

        }//while(没找到)

        cout<<"请输入密码"<<endl;

        cin>>psw;

**while** (psw!=id2account[id]->get\_password()) {

            cout<<"密码错误，重新输入请按1，更换账号请按2"<<endl;

            cin>>x;

**while**(x!=1&&x!=2) {

                cout<<"请输入1或2"<<endl;

                cin>>x;

            }

**if**(x==2) **break**;

            cout<<"请重新输入密码"<<endl;

            cin>>psw;

        }//while(密码错误)

**if**(psw==id2account[id]->get\_password()) {

            cout<<"登陆成功"<<endl;

            online\_account = id2account[id];

**break**;

        }

    }

**if**(online\_account->get\_type()=="merchant")

        merchantOperate();

**else** consumerOperate();

}

**void** AccountSystem::showGoods() {

**if**(goods.size()==0) {

        cout<<"当前系统中没有商品"<<endl;

**return**;

    }

    cout<<"当前系统中有如下商品："<<endl<<endl;

    //商品编号从1开始，和下标有1的偏差

**for** (**int** i = 0; i<goods.size(); i++) {

        cout<<"商品编号："<<i+1<<endl;

        goods[i]->show\_good();

        cout<<endl;

    }

}

**void** AccountSystem::showAccounts() {

**if**(accounts.size()==0) {

        cout<<"当前系统中没有账户"<<endl;

**return**;

    }

    cout<<"当前系统中有如下账户："<<endl<<endl;

    //账户编号从1开始，和下标有1的偏差

**for** (**int** i = 0; i<accounts.size(); i++) {

        cout<<"账户编号："<<i+1<<endl;

        accounts[i]->show\_account();

        cout<<endl;

    }

}

**void** AccountSystem::merchantOperate() {

**int** x;

    Merchant \*merchant = merchants[online\_account->get\_id()];

    cout<<"当前在线账号id为："<<online\_account->get\_id()<<endl;

    merchant->menu();

    showGoods();

    cin>>x;

**while**(x!=0) {

**switch**(x) {

**case** 1:

                merchant->set\_name(goods);

**break**;

**case** 2:

                merchant->set\_description(goods);

**break**;

**case** 3:

                merchant->set\_remaining(goods);

**break**;

**case** 4:

                merchant->set\_price(goods);

**break**;

**case** 5:

                merchant->add\_goods(goods);

**break**;

**case** 6:

                merchant->onsale(goods);

**break**;

**case** 7:

                merchant->findGood(goods);

**break**;

**case** 8:

                //更改密码

                merchant->changePassword();

**break**;

**case** 9:

                //余额查询

                merchant->getBalance();

**break**;

**case** 10:

                //充值

                merchant->recharge();

**break**;

**case** 11:

                //消费

                merchant->purchase(goods);

**break**;

**default**:

                cout<<"请输入正确的数字哦～"<<endl;

**break**;

        }//switch

        cout<<"当前在线账号id为："<<online\_account->get\_id()<<endl;

        merchant->menu();

        showGoods();

        cin>>x;

    }//while

    cout<<"您已成功退出商家管理系统"<<endl;

}

**void** AccountSystem::consumerOperate() {

    Consumer \*consumer = consumers[online\_account->get\_id()];

    cout<<"当前在线账号id为："<<online\_account->get\_id()<<endl;

    consumer->menu();

    showGoods();

**int** x;

    cin>>x;

**while**(x!=0) {

**switch**(x) {

**case** 1: {

                //修改密码

                consumer->changePassword();

**break**;

            }

**case** 2: {

                //余额查询

                consumer->getBalance();

**break**;

            }

**case** 3: {

                //充值

                consumer->recharge();

**break**;

            }

**case** 4: {

                //购买商品

                consumer->purchase(goods);

**break**;

            }

**case** 5: {

                //查询商品

                consumer->findGood(goods);

**break**;

            }

**case** 6: {

                //购物车管理

                consumer->cartManage(goods);

**break**;

            }

**default**: {

                cout<<"请输入正确的数字哦～"<<endl;

**break**;

            }

        }//switch

        cout<<"当前在线账号id为："<<online\_account->get\_id()<<endl;

        consumer->menu();

        showGoods();

        cin>>x;

    }//while

    cout<<"您已成功退出买家系统"<<endl;

}

1. account.hpp

#ifndef account\_hpp

#define account\_hpp

#include <stdio.h>

#include <iostream>

#include <string>

#include <vector>

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Account {

**public**:

    Account();

    Account(string id, string password, string type);

    Account(string id, string password, **double** balance, string type);

    ~Account();

    string get\_password(){**return** password;}

    string get\_id(){**return** id;}

**double** get\_balance(){**return** balance;}

    string get\_type(){**return** type;}

**void** set\_id(string id){**this**->id = id;}

**void** set\_password(string psw){password = psw;}

**void** recharge(**double** money);

**void** consume(**double** money);

**void** show\_account();

**void** transIn(**double** money);

**void** transOut(**double** money);

**protected**:

    string id;

    string password;

**double** balance;//余额

    string type;   //"consumer"表示消费者，"merchant"表示商家

};

#endif /\* account\_hpp \*/

1. account.cpp

#include "account.hpp"

Account::Account(){}

Account::Account(string id,string password, string type):

id(id),password(password),balance(0),type(type) {

}

Account::Account(string id, string password, **double** balance, string type):id(id),password(password),balance(balance),type(type) {

}

Account::~Account() {

}

**void** Account::recharge(**double** money) {

    balance += money;

    cout<<"您本次充值"<<money<<"元，账户余额"<<balance<<"元"<<endl;

}

**void** Account::consume(**double** money) {

    balance -= money;

    cout<<"您本次消费"<<money<<"元，账户余额"<<balance<<"元"<<endl;

}

**void** Account::show\_account() {

    cout<<"账号："<<id<<endl;

    cout<<"密码："<<password<<endl;

    cout<<"余额："<<balance<<endl;

    cout<<"类型："<<type<<endl;

}

**void** Account::transIn(**double** money) {

    balance += money;

    cout<<"账户："<<id<<" 转入 "<<money<<" 元"<<endl;

}

**void** Account::transOut(**double** money) {

    balance -= money;

    cout<<"账户："<<id<<" 转出 "<<money<<" 元"<<endl;

}

1. user.hpp

#ifndef user\_hpp

#define user\_hpp

#include <stdio.h>

#include <iostream>

#include <string>

#include <vector>

#include "account.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**using** std::vector;

**class** Account;

**class** Goods;

**class** User {

**public**:

    User();

    User(Account \*account);

**virtual** ~User();

**virtual** string getUserType() = 0;

    Account\* get\_account(){**return** account;}

**void** changePassword();

**void** getBalance();

**void** recharge();

**void** purchase(vector<Goods\*> &goods);

**void** findGood(vector<Goods\*> &goods);

**void** findGoodName(vector<Goods\*> &goods);

**void** findGoodPriceUp(vector<Goods\*> &goods);

**void** findGoodPriceDown(vector<Goods\*> &goods);

**void** findGoodRemainUp(vector<Goods\*> &goods);

**void** findGoodRemainDown(vector<Goods\*> &goods);

**void** findGoodType(vector<Goods\*> &goods);

**protected**:

    Account \*account;

};

#endif /\* user\_hpp \*/

1. user.cpp

#include "account.hpp"

#include "user.hpp"

#include "goods.hpp"

User::User():account(){

}

User::User(Account \*account):account(account) {

}

User::~User() {

}

**void** User::findGood(vector<Goods \*> &goods) {

**int** x;

    cout << endl<<"      请选择操作序号！(0~6)" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*              0.退出              \*" << endl;

        cout << "\*          1.筛选商品名称           \*" << endl;

        cout << "\*       2.筛选高于给定价格的商品      \*" << endl;

        cout << "\*       3.筛选低于给定价格的商品      \*" << endl;

        cout << "\*       4.筛选高于给定余量的商品      \*" << endl;

        cout << "\*       5.筛选低于给定余量的商品      \*" << endl;

        cout << "\*           6.筛选商品种类          \*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    cin>>x;

**while**(x!=0) {

**switch** (x) {

**case** 1:

                findGoodName(goods);

**break**;

**case** 2:

                findGoodPriceUp(goods);

**break**;

**case** 3:

                findGoodPriceDown(goods);

**break**;

**case** 4:

                findGoodRemainUp(goods);

**break**;

**case** 5:

                findGoodRemainDown(goods);

**break**;

**case** 6:

                findGoodType(goods);

**break**;

**default**:

                cout<<"请输入正确的数字哦～"<<endl;

**break**;

        }

        cout << endl<<"      请选择操作序号！(0~6)" << endl;

            cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

            cout << "\*                                 \*" << endl;

            cout << "\*              0.退出              \*" << endl;

            cout << "\*          1.筛选商品名称           \*" << endl;

            cout << "\*       2.筛选高于给定价格的商品      \*" << endl;

            cout << "\*       3.筛选低于给定价格的商品      \*" << endl;

            cout << "\*       4.筛选高于给定余量的商品      \*" << endl;

            cout << "\*       5.筛选低于给定余量的商品      \*" << endl;

            cout << "\*           6.筛选商品种类          \*" << endl;

            cout << "\*                                 \*" << endl;

            cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cin>>x;

    }

}

**void** User::findGoodName(vector<Goods\*> &goods) {

**int** flag = 0,x;

    string name;

**while** (1) {

        flag = 0;

        cout<<"请输入搜索的商品名称"<<endl;

        cin>>name;

        cout<<endl;

**for**(**int** i = 0;i<goods.size();i++) {

**if**(goods[i]->get\_name()==name) {

                flag = 1;

                //商品编号从1开始

                cout<<"商品编号："<<i+1<<endl;

                goods[i]->show\_good();

                cout<<endl;

            }

        }

**if**(flag==0) cout<<"没找到你要的商品"<<endl;

        cout<<"继续筛选商品名称请按1，退出请按0"<<endl;

        cin>>x;

**while**(x!=0&&x!=1) {

            cout<<"请输入0或1，继续筛选商品名称请按1，退出请按0"<<endl;

            cin>>x;

        }

**if**(x==0) **break**;

    }

}

//筛选高于某个价格的商品

**void** User::findGoodPriceUp(vector<Goods\*> &goods) {

**int** flag = 0,x;

**double** price;

**while** (1) {

        flag = 0;

        cout<<"请输入筛选的商品价格下限"<<endl;

        cin>>price;

        cout<<endl;

**for**(**int** i = 0;i<goods.size();i++) {

**if**(goods[i]->get\_price()>=price) {

                flag = 1;

                //商品编号从1开始

                cout<<"商品编号："<<i+1<<endl;

                goods[i]->show\_good();

                cout<<endl;

            }

        }

**if**(flag==0) cout<<"没找到你要的商品"<<endl;

        cout<<"继续筛选商品价格请按1，退出请按0"<<endl;

        cin>>x;

**while**(x!=0&&x!=1) {

            cout<<"请输入0或1，继续筛选商品价格请按1，退出请按0"<<endl;

            cin>>x;

        }

**if**(x==0) **break**;

    }

}

//筛选低于某个价格的商品

**void** User::findGoodPriceDown(vector<Goods\*> &goods) {

**int** flag = 0,x;

**double** price;

**while**(1) {

        flag = 0;

        cout<<"请输入筛选的商品价格上限"<<endl;

        cin>>price;

        cout<<endl;

**for**(**int** i = 0;i<goods.size();i++) {

**if**(goods[i]->get\_price()<=price) {

                flag = 1;

                //商品编号从1开始

                cout<<"商品编号："<<i+1<<endl;

                goods[i]->show\_good();

                cout<<endl;

            }

        }

**if**(flag==0) cout<<"没找到你要的商品"<<endl;

        cout<<"继续筛选商品价格请按1，退出请按0"<<endl;

        cin>>x;

**while**(x!=0&&x!=1) {

            cout<<"请输入0或1，继续筛选商品价格请按1，退出请按0"<<endl;

            cin>>x;

        }

**if**(x==0) **break**;

    }

}

**void** User::findGoodType(vector<Goods\*> &goods) {

**int** flag = 0,x;

    string type;

**while**(1) {

        flag = 0;

        cout<<"请输入筛选的商品种类"<<endl;

        cin>>type;

        cout<<endl;

**for**(**int** i = 0;i<goods.size();i++) {

**if**(goods[i]->get\_type()==type) {

                flag = 1;

                //商品编号从1开始

                cout<<"商品编号："<<i+1<<endl;

                goods[i]->show\_good();

                cout<<endl;

            }

        }

**if**(flag==0) cout<<"没找到你要的商品"<<endl;

        cout<<"继续筛选商品种类请按1，退出请按0"<<endl;

        cin>>x;

**while**(x!=0&&x!=1) {

            cout<<"请输入0或1，继续筛选商品种类请按1，退出请按0"<<endl;

            cin>>x;

        }

**if**(x==0) **break**;

    }

}

**void** User::findGoodRemainUp(vector<Goods\*> &goods) {

**int** flag = 0,x;

**int** remain;

**while**(**true**) {

        flag = 0;

        cout<<"请输入筛选的商品余量下限"<<endl;

        cin>>remain;

        cout<<endl;

**for**(**int** i = 0;i<goods.size();i++) {

**if**(goods[i]->get\_remaining()>=remain) {

                flag = 1;

                //商品编号从1开始

                cout<<"商品编号："<<i+1<<endl;

                goods[i]->show\_good();

                cout<<endl;

            }

        }

**if**(flag==0) cout<<"没找到你要的商品"<<endl;

        cout<<"继续筛选商品余量请按1，退出请按0"<<endl;

        cin>>x;

**while**(x!=0&&x!=1) {

            cout<<"请输入0或1，继续筛选商品余量请按1，退出请按0"<<endl;

            cin>>x;

        }

**if**(x==0) **break**;

    }

}

**void** User::findGoodRemainDown(vector<Goods\*> &goods) {

**int** flag = 0,x;

**int** remain;

**while**(**true**) {

        flag = 0;

        cout<<"请输入筛选的商品余量上限"<<endl;

        cin>>remain;

        cout<<endl;

**for**(**int** i = 0;i<goods.size();i++) {

**if**(goods[i]->get\_remaining()<=remain) {

                flag = 1;

                //商品编号从1开始

                cout<<"商品编号："<<i+1<<endl;

                goods[i]->show\_good();

                cout<<endl;

            }

        }

**if**(flag==0) cout<<"没找到你要的商品"<<endl;

        cout<<"继续筛选商品余量请按1，退出请按0"<<endl;

        cin>>x;

**while**(x!=0&&x!=1) {

            cout<<"请输入0或1，继续筛选商品余量请按1，退出请按0"<<endl;

            cin>>x;

        }

**if**(x==0) **break**;

    }

}

**void** User::changePassword() {

    string psw,check;

**do** {

        cout<<"请输入新的密码"<<endl;

        cin>>psw;

        cout<<"请确认密码"<<endl;

        cin>>check;

**if**(check!=psw) cout<<"两次输入密码不同"<<endl;

    } **while**(check!=psw);

    account->set\_password(psw);

    cout<<"密码修改成功"<<endl;

}

**void** User::getBalance() {

    cout<<"您账户余额还有："<<account->get\_balance()<<"元"<<endl;

}

**void** User::recharge() {

**double** money;

    cout<<"请输入您充值的数额："<<endl;

    cin>>money;

    account->recharge(money);

}

**void** User::purchase(vector<Goods\*> &goods) {

**if**(goods.size()==0) **return**;

    cout<<"请输入你想购买的商品编号："<<endl;

**int** n;

    cin>>n;

**if**(goods[n-1]->get\_remaining()==0) {

        cout<<"商品库存不足"<<endl;

**return**;

    }

**if**(goods[n-1]->get\_price()>account->get\_balance()) {

        cout<<"您账户中没有足够的余额，商品需要"<<goods[n-1]->get\_price()<<"元，您只有"<<account->get\_balance()<<"元"<<endl;

**return**;

    }

    account->consume(goods[n-1]->get\_price());

    goods[n-1]->set\_remaining(goods[n-1]->get\_remaining()-1);

    cout<<"您买下了"<<n<<"号商品："<<goods[n-1]->get\_name()<<endl;

    cout<<"谢谢惠顾"<<endl;

}

1. consumer.hpp

#ifndef consumer\_hpp

#define consumer\_hpp

#include <stdio.h>

#include <map>

#include "user.hpp"

#include "order.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**using** std::map;

**class** Goods;

**class** Account;

**class** Order;

**class** Consumer : **public** User {

**public**:

    Consumer();

    Consumer(Account \*account);

    ~Consumer();

    string getUserType(){**return** "consumer";}

**void** menu();

**void** cartManage(vector<Goods\*> &goods);

**void** cartMenu();

**void** showCart(**const** vector<Goods\*> &goods);

**void** addCart(**const** vector<Goods\*> &goods);

**void** rmCart(**const** vector<Goods\*> &goods);

**void** changeCart(**const** vector<Goods\*> &goods);

**void** select(**const** vector<Goods\*> &goods);

**void** unselect(**const** vector<Goods\*> &goods);

**void** generateOrder(vector<Goods\*> &goods);   //订单生成

**void** payOrder(vector<Goods\*> &goods);    //订单支付

**void** cancelOrder(vector<Goods\*> &goods); //订单取消

**void** showOrders(vector<Goods\*> &goods);

**private**:

    //商品编号需要-1才等于商品下标

    //商品编号作为key，商品数量作为value，需要映射到goods数组中获取商品具体信息

    map<**int**,**int**> cart;

    //商品编号作为key，是否选中作为value

    map<**int**,**int**> selected;

**double** total\_price;

    //除非订单取消，否则都会存在这里（待支付/已支付）

    vector<Order> orders;

};

#endif /\* consumer\_hpp \*/

1. consumer.cpp

#include "account.hpp"

#include "goods.hpp"

#include "consumer.hpp"

Consumer::Consumer() {

}

Consumer::Consumer(Account \*account) {

**this**->account = account;

    total\_price = 0;

}

Consumer::~Consumer() {

}

**void** Consumer::menu() {

    cout << endl<<"      请选择操作序号！(0~6)" << endl;

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    cout << "\*                                 \*" << endl;

    cout << "\*            0.退出登陆             \*" << endl;

    cout << "\*            1.修改密码             \*" << endl;

    cout << "\*            2.余额查询             \*" << endl;

    cout << "\*            3.充值                \*" << endl;

    cout << "\*            4.购买商品             \*" << endl;

    cout << "\*            5.查询商品             \*" << endl;

    cout << "\*            6.购物车管理           \*" << endl;

    cout << "\*                                 \*" << endl;

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

**void** Consumer::cartManage(vector<Goods\*> &goods) {

**int** x;

    showCart(goods);

    cartMenu();

    cin>>x;

**while**(x!=0) {

**switch** (x) {

**case** 1:

                addCart(goods);

**break**;

**case** 2:

                rmCart(goods);

**break**;

**case** 3:

                changeCart(goods);

**break**;

**case** 4:

                select(goods);

**break**;

**case** 5:

                unselect(goods);

**break**;

**case** 6:

                generateOrder(goods);

**break**;

**case** 7:

                payOrder(goods);

**break**;

**case** 8:

                cancelOrder(goods);

**break**;

**case** 9:

                showOrders(goods);

**break**;

**default**:

                cout<<"请输入正确的数字哦～"<<endl;

**break**;

        }

        showCart(goods);

        cartMenu();

        cin>>x;

    }

}

**void** Consumer::cartMenu() {

    cout << endl<<"      请选择操作序号！(0~6)" << endl;

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    cout << "\*                                 \*" << endl;

    cout << "\*            0.退出                \*" << endl;

    cout << "\*            1.增加商品             \*" << endl;

    cout << "\*            2.删除商品             \*" << endl;

    cout << "\*            3.修改数量             \*" << endl;

    cout << "\*            4.选中商品             \*" << endl;

    cout << "\*            5.取消选中             \*" << endl;

    cout << "\*            6.生成订单             \*" << endl;

    cout << "\*            7.支付订单             \*" << endl;

    cout << "\*            8.取消订单             \*" << endl;

    cout << "\*            9.查看全部订单         \*" << endl;

    cout << "\*                                 \*" << endl;

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

**void** Consumer::showCart(**const** vector<Goods \*> &goods) {

**if**(cart.size()==0) {

        cout<<"当前购物车中没有商品"<<endl;

**return**;

    }

    cout<<"当前购物车中有如下商品："<<endl<<endl;

**for**(**auto** item : cart) {

**if**(item.second==0) **continue**;

        cout<<"商品编号："<<item.first<<endl;

        goods[item.first-1]->show\_good();

        cout<<"购物车中数量："<<item.second<<endl;

        cout<<"共计价格："<<item.second\*goods[item.first-1]->get\_price()<<endl;

**if**(selected[item.first]) cout<<"是否选中：已选中"<<endl;

**else** cout<<"是否选中：未选中"<<endl<<endl;

    }

    cout<<"选中商品价格共计："<<total\_price<<endl;

}

**void** Consumer::addCart(**const** vector<Goods \*> &goods) {

**int** no,amount;

    cout<<"请输入您要添加入购物车的商品编号,1~"<<goods.size()<<endl;

    cin>>no;

**while**(no>goods.size()||no<1) {

        cout<<"请输入正确的商品编号,1~"<<goods.size()<<endl;

        cin>>no;

    }

    cout<<"请输入您要添加的件数，1~"<<goods[no-1]->get\_remaining()-cart[no]<<endl;

    cin>>amount;

**while**(amount>goods[no-1]->get\_remaining()-cart[no]||amount<1) {

        cout<<"请输入正确的商品数量,1~"<<goods[no-1]->get\_remaining()-cart[no]<<endl;

        cin>>amount;

    }

    cart[no] += amount;

}

**void** Consumer::rmCart(**const** vector<Goods \*> &goods) {

**int** no,amount;

    cout<<"请输入您要从购物车移出的商品编号,1~"<<goods.size()<<endl;

    cin>>no;

**while**(no>goods.size()||no<1) {

        cout<<"请输入正确的商品编号,1~"<<goods.size()<<endl;

        cin>>no;

    }

    cout<<"请输入您要移出的件数,1~"<<cart[no]<<endl;

    cin>>amount;

**while**(amount>cart[no]||amount<1) {

        cout<<"请输入正确的商品数量,1~"<<cart[no]<<endl;

        cin>>amount;

    }

    cart[no] -= amount;

}

**void** Consumer::changeCart(**const** vector<Goods \*> &goods) {

**int** no,amount;

    cout<<"请输入您要更改数量的商品编号"<<endl;

    cin>>no;

**while**(no>goods.size()||no<1) {

        cout<<"请输入正确的商品编号,1~"<<goods.size()<<endl;

        cin>>no;

    }

    cout<<"请输入您要更改后的件数,0~"<<goods[no-1]->get\_remaining()<<endl;

    cin>>amount;

**while**(amount>cart[no]||amount<0) {

        cout<<"请输入正确的商品数量,0~"<<goods[no-1]->get\_remaining()<<endl;

        cin>>amount;

    }

    cart[no] = amount;

}

**void** Consumer::select(**const** vector<Goods \*> &goods) {

**int** no;

    cout<<"请输入您要选中的商品编号"<<endl;

    cin>>no;

**while**(no>goods.size()||no<1) {

        cout<<"请输入正确的商品编号,1~"<<goods.size()<<endl;

        cin>>no;

    }

    selected[no] = cart[no];

    total\_price += cart[no]\*goods[no-1]->get\_price();

}

**void** Consumer::unselect(**const** vector<Goods \*> &goods) {

**int** no;

    cout<<"请输入您要取消选中的商品编号"<<endl;

    cin>>no;

**while**(no>goods.size()||no<1) {

        cout<<"请输入正确的商品编号,1~"<<goods.size()<<endl;

        cin>>no;

    }

    selected[no] = 0;

    total\_price -= cart[no]\*goods[no-1]->get\_price();

}

//生成订单

**void** Consumer::generateOrder(vector<Goods\*> &goods) {

**int** x;

    cout<<"确认将选中商品下单？ 输入1确认，0返回"<<endl;

    cin>>x;

**while**(x!=1&&x!=0) {

        cout<<"请输入正确的数字，输入1确认，0返回"<<endl;

        cin>>x;

    }

**if**(x==0) **return**;

    Order temp(goods,selected);

    orders.push\_back(temp);

    //生成订单后购物车选中部分清空

**for**(**auto** &item : selected) {

        cart[item.first] -= item.second;

        item.second = 0;

    }

    total\_price = 0;

    cout<<"订单生成成功"<<endl;

//    cout<<"订单生成成功，是否立即支付？ 输入1确认，0返回"<<endl;

//    cin>>x;

//    while(x!=1&&x!=0) {

//        cout<<"请输入正确的数字，输入1确认，0返回"<<endl;

//        cin>>x;

//    }

//    if(x==0) return;

//    orders.pop\_back();

//    payOrder(goods);

//    orders.push\_back(temp);

}

**void** Consumer::payOrder(vector<Goods\*> &goods) {

    cout<<"您待支付的订单如下："<<endl;

**int** flag = 0;

**for**(**int** i = 0; i<orders.size(); i++) {

        cout<<"订单编号："<<i+1<<endl;

**if**(orders[i].get\_state()==0) {

            orders[i].showOrder(goods);

            flag = 1;

        }

    }

**if**(flag==0) {

        cout<<"没有待支付的订单"<<endl;

**return**;

    }

    cout<<"请选择您要支付的订单编号："<<endl;

**int** no;

    cin>>no;

**while**(no<1||no>orders.size()||orders[no-1].get\_state()==1) {

**if**(no<1||no>orders.size())

            cout<<"请输入正确的订单编号，1～"<<orders.size()<<endl;

**else** cout<<"该订单已支付，请重新输入要支付的订单编号"<<endl;

        cin>>no;

    }

    //如果账户余额足够

**while**(**this**->account->get\_balance()<orders[no-1].get\_price()) {

        cout<<"余额不足，是否立即充值？ 1确认，0返回"<<endl;

**int** x;

        cin>>x;

**while**(x!=0&&x!=1) {

            cout<<"请输入正确的数字，1确认，0返回"<<endl;

            cin>>x;

        }

**if**(x==1) recharge();

    }

**this**->account->transOut(orders[no-1].get\_price());

    orders[no-1].set\_state(1);

    orders[no-1].transfer(goods);

    orders[no-1].release(goods);

    cout<<"订单支付成功"<<endl;

}

//取消订单，释放冷冻量

**void** Consumer::cancelOrder(vector<Goods\*> &goods) {

    cout<<"您待支付的订单如下："<<endl;

**int** flag = 0;

**for**(**int** i = 0; i<orders.size(); i++) {

        cout<<"订单编号："<<i+1<<endl;

**if**(orders[i].get\_state()==0) {

            orders[i].showOrder(goods);

            flag = 1;

        }

    }

**if**(flag==0) {

        cout<<"没有待支付的订单"<<endl;

**return**;

    }

    cout<<"请选择您要取消的订单编号："<<endl;

**int** no;

    cin>>no;

**while**(no<1||no>orders.size()||orders[no-1].get\_state()==1) {

**if**(no<1||no>orders.size())

            cout<<"请输入正确的订单编号，1～"<<orders.size()<<endl;

**else** cout<<"该订单已支付，请重新输入要取消的订单编号"<<endl;

        cin>>no;

    }

    orders[no-1].release(goods);

    orders.erase(orders.begin()+no-1);

    cout<<"订单取消成功"<<endl;

}

**void** Consumer::showOrders(vector<Goods\*> &goods) {

    cout<<"所有订单如下："<<endl;

**for**(**int** i = 0; i<orders.size(); i++) {

        cout<<"订单编号："<<i+1<<endl;

        orders[i].showOrder(goods);

    }

}

1. merchant.hpp

#ifndef merchant\_hpp

#define merchant\_hpp

#include <stdio.h>

#include "user.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**using** std::vector;

**class** Account;

**class** Goods;

**class** User;

**class** Merchant : **public** User {

**public**:

    Merchant();

    Merchant(Account \*account);

    ~Merchant();

    string getUserType(){**return** "merchant";}

**void** menu();

    //商品名称管理

**void** set\_name(vector<Goods\*> &goods);

    //商品价格管理

**void** set\_price(vector<Goods\*> &goods);

    //商品剩余量管理

**void** set\_remaining(vector<Goods\*> &goods);

    //商品描述管理

**void** set\_description(vector<Goods\*> &goods);

    //添加商品

**void** add\_goods(vector<Goods\*> &goods);

    //对同一类商品打折

**void** onsale(vector<Goods\*> &goods);

};

#endif /\* merchant\_hpp \*/

1. merchant.cpp

#include "account.hpp"

#include "merchant.hpp"

#include "goods.hpp"

#include "book.hpp"

#include "food.hpp"

#include "clothing.hpp"

Merchant::Merchant() {

}

Merchant::Merchant(Account \*account) {

**this**->account = account;

}

Merchant::~Merchant() {

}

**void** Merchant::set\_name(vector<Goods\*> &goods) {

**if**(goods.size()==0) **return**;

    cout<<"请输入你想管理名称的商品编号"<<endl;

**int** n;

    cin>>n;

    //判断编号是否合法、商品是否所属该商家

**while**(n>goods.size()||n<1||

          (n<=goods.size()&&n>0&&goods[n-1]->get\_merchant()!=**this**)) {

**if**(n>goods.size()||n<1)

            cout<<"请输入正确的商品编号，1～"<<goods.size()<<endl;

**else** cout<<"您无权修改不属于您的商品，请重新输入"<<endl;

        cin>>n;

    }

    string name;

    cout<<"请输入更改后的名称"<<endl;

    cin>>name;

    goods[n-1]->set\_name(name);

    cout<<"名称修改成功"<<endl;

}

**void** Merchant::set\_price(vector<Goods\*> &goods) {

**if**(goods.size()==0) **return**;

    cout<<"请输入你想管理价格的商品编号"<<endl;

**int** n;

    cin>>n;

    //判断编号是否合法、商品是否所属该商家

**while**(n>goods.size()||n<1||

          (n<=goods.size()&&n>0&&goods[n-1]->get\_merchant()!=**this**)) {

**if**(n>goods.size()||n<1)

            cout<<"请输入正确的商品编号，1～"<<goods.size()<<endl;

**else** cout<<"您无权修改不属于您的商品，请重新输入"<<endl;

        cin>>n;

    }

**double** price;

    cout<<"请输入更改后的价格"<<endl;

    cin>>price;

    goods[n-1]->set\_price(price);

    cout<<"价格修改成功"<<endl;

}

**void** Merchant::set\_remaining(vector<Goods\*> &goods) {

    //商品编号从1开始

**if**(goods.size()==0) **return**;

    cout<<"请输入你想管理剩余量的商品编号"<<endl;

**int** n;

    cin>>n;

    //判断编号是否合法、商品是否所属该商家

**while**(n>goods.size()||n<1||

          (n<=goods.size()&&n>0&&goods[n-1]->get\_merchant()!=**this**)) {

**if**(n>goods.size()||n<1)

            cout<<"请输入正确的商品编号，1～"<<goods.size()<<endl;

**else** cout<<"您无权修改不属于您的商品，请重新输入"<<endl;

        cin>>n;

    }

**int** remaining;

    cout<<"请输入更改后的剩余量"<<endl;

    cin>>remaining;

    goods[n-1]->set\_remaining(remaining);

    cout<<"剩余量修改成功"<<endl;

}

**void** Merchant::set\_description(vector<Goods\*> &goods) {

**if**(goods.size()==0) **return**;

    cout<<"请输入你想管理描述的商品编号"<<endl;

**int** n;

    cin>>n;

    //判断编号是否合法、商品是否所属该商家

**while**(n>goods.size()||n<1||

          (n<=goods.size()&&n>0&&goods[n-1]->get\_merchant()!=**this**)) {

**if**(n>goods.size()||n<1)

            cout<<"请输入正确的商品编号，1～"<<goods.size()<<endl;

**else** cout<<"您无权修改不属于您的商品，请重新输入"<<endl;

        cin>>n;

    }

    string description;

    cout<<"请输入更改后的描述"<<endl;

    cin>>description;

    goods[n-1]->set\_description(description);

    cout<<"描述修改成功"<<endl;

}

**void** Merchant::add\_goods(vector<Goods\*> &goods) {

    Goods \*good;

**int** remaining;

    string name,description,type;

**double** price;

    cout<<"请输入你想添加的商品种类(book/food/clothing)"<<endl;

    cin>>type;

    cout<<"请输入商品名称(用下划线代替空格)"<<endl;

    cin>>name;

    cout<<"请输入商品描述(用下划线代替空格)"<<endl;

    cin>>description;

    cout<<"请输入商品价格"<<endl;

    cin>>price;

    cout<<"请输入商品剩余量"<<endl;

    cin>>remaining;

**if**(type=="book") {

        good = **new** Book(name,description,price,remaining,type,**this**);

    } **else** **if**(type=="food") {

        good = **new** Food(name,description,price,remaining,type,**this**);

    } **else** {

        good = **new** Clothing(name,description,price,remaining,type,**this**);

    }

    goods.push\_back(good);

    cout<<"商品增加成功"<<endl;

}

**void** Merchant::onsale(vector<Goods\*> &goods) {

**if**(goods.size()==0) **return**;

    cout<<"请输入你想打折的商品种类(book/food/clothing)"<<endl;

    string type;

    cin>>type;

**double** rate;

    cout<<"输入你想打折的折扣率"<<endl;

    cin>>rate;

**for**(**auto** &good : goods) {

        //只能管理属于自己的商品

**if**(good->get\_type()==type && good->get\_merchant()==**this**)

            good->set\_price(good->get\_price()\*rate);

    }

    cout<<"打折成功"<<endl;

}

**void** Merchant::menu() {

    cout << endl<<"      请选择操作序号！(0~6)" << endl;

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    cout << "\*                                  \*" << endl;

    cout << "\*           0.退出登陆              \*" << endl;

    cout << "\*           1.商品名称管理           \*" << endl;

    cout << "\*           2.商品描述管理           \*" << endl;

    cout << "\*           3.商品剩余量管理         \*" << endl;

    cout << "\*           4.商品价格管理           \*" << endl;

    cout << "\*           5.添加商品              \*" << endl;

    cout << "\*           6.同类商品打折           \*" << endl;

    cout << "\*           7.查询商品              \*" << endl;

    cout << "\*           8.更改密码              \*" << endl;

    cout << "\*           9.余额查询              \*" << endl;

    cout << "\*           10.充值                \*" << endl;

    cout << "\*           11.消费                \*" << endl;

    cout << "\*                                 \*" << endl;

    cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

}

1. goods.hpp

#ifndef goods\_hpp

#define goods\_hpp

#include <stdio.h>

#include <iostream>

#include <string>

#include <vector>

#include "merchant.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Merchant;

**class** Goods {

**public**:

    Goods();

    Goods(string name, string description, **double** price, **int** remaining, string type, Merchant \*mer);

**virtual** ~Goods();

    string get\_name(){**return** name;}

    string get\_description(){**return** description;}

**virtual** **double** get\_price() = 0;

**int** get\_remaining(){**return** remaining;}

    string get\_type(){**return** type;}

    Merchant\* get\_merchant() {**return** merchant;}

**void** set\_description(string description){

**this**->description = description;

    }

**void** set\_price(**double** price){**this**->price = price;}

**void** set\_remaining(**int** remain){remaining = remain;}

**void** set\_name(string name){**this**->name = name;}

**void** set\_type(string type){**this**->type = type;}

**void** set\_merchant(Merchant \*mer) {merchant = mer;}

**void** show\_good();

**protected**:

    string name;

    string description;

**double** price;

**int** remaining;

    string type;

    //所属商家(只会把对应的id存入文件）

    Merchant\* merchant;

};

#endif /\* goods\_hpp \*/

1. goods.cpp

#include "goods.hpp"

Goods::Goods() {

}

Goods::Goods(string name,string description, **double** price, **int** remaining,string type, Merchant \*mer):

    name(name),description(description),price(price),remaining(remaining),type(type),merchant(mer)

{

}

Goods::~Goods() {

}

**void** Goods::show\_good() {

    cout<<"名称："<<name<<endl;

    cout<<"描述："<<description<<endl;

    cout<<"价格："<<price<<endl;

    cout<<"剩余量："<<remaining<<endl;

    cout<<"种类："<<type<<endl;

    cout<<"所属商家id："<<merchant->get\_account()->get\_id()<<endl;

}

1. book.hpp

#ifndef book\_hpp

#define book\_hpp

#include <stdio.h>

#include "goods.hpp"

#include "merchant.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Goods;

**class** Merchant;

**class** Book : **public** Goods {

**public**:

    Book();

    Book(string name,string description,**double** price,**int** remaining, string type,Merchant \*mer);

    ~Book();

**double** get\_price(){**return** price;}

};

#endif /\* book\_hpp \*/

1. book.cpp

#include "goods.hpp"

#include "book.hpp"

Book::Book() {

}

Book::Book(string name,string description,**double** price,**int** remaining,string type, Merchant\* mer){

**this**->name = name;

**this**->description = description;

**this**->price = price;

**this**->remaining = remaining;

**this**->type = type;

**this**->merchant = mer;

}

Book::~Book() {

}

1. food.hpp

#ifndef food\_hpp

#define food\_hpp

#include <stdio.h>

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Goods;

**class** Merchant;

**class** Food : **public** Goods {

**public**:

    Food();

    Food(string name, string description,**double** price,**int** remaining,string type, Merchant \*mer);

    ~Food();

**double** get\_price() {**return** price;}

};

#endif /\* food\_hpp \*/

1. food.cpp

#include "goods.hpp"

#include "food.hpp"

Food::Food() {

}

Food::Food(string name,string description,**double** price,**int** remaining,string type, Merchant \*mer){

**this**->name = name;

**this**->description = description;

**this**->price = price;

**this**->remaining = remaining;

**this**->type = type;

**this**->merchant = mer;

}

Food::~Food() {

}

1. clothing.hpp

#ifndef clothing\_hpp

#define clothing\_hpp

#include <stdio.h>

#include "merchant.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Goods;

**class** Merchant;

**class** Clothing : **public** Goods {

**public**:

    Clothing();

    Clothing(string name, string description,**double** price,**int** remaining,string type, Merchant \*mer);

    ~Clothing();

**double** get\_price(){**return** price;}

};

#endif /\* clothing\_hpp \*/

1. clothing.cpp

#include <string>

#include "goods.hpp"

#include "clothing.hpp"

Clothing::Clothing() {

}

Clothing::Clothing(string name,string description,**double** price,**int** remaining,string type, Merchant\* mer){

**this**->name = name;

**this**->description = description;

**this**->price = price;

**this**->remaining = remaining;

**this**->type = type;

**this**->merchant = mer;

}

Clothing::~Clothing() {

}

1. order.hpp

#ifndef order\_hpp

#define order\_hpp

#include <stdio.h>

#include <map>

#include "goods.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**using** std::map;

**class** Goods;

**class** Order {

**public**:

    Order();

    Order(vector<Goods\*> &goods, **const** map<**int**,**int**> &selected);

    map<**int**,**int**> get\_goods() {**return** good\_list;}

**double** get\_price() {**return** total\_price;}

**int** get\_state() {**return** state;}

**void** set\_state(**int** x) {state = x;}

**void** release(vector<Goods\*> &goods);

**void** showOrder(vector<Goods\*> &goods);

**void** transfer(vector<Goods\*> &goods);    //把用户扣除的钱转移到商家账户中

**private**:

    //first是商品编号，second是商品数量

    map<**int**,**int**> good\_list;

**double** total\_price;

    //订单状态（0待支付/1已支付）

**int** state;

};

#endif /\* order\_hpp \*/

1. order.cpp

#include "order.hpp"

#include "goods.hpp"

Order::Order() {

    total\_price = 0;

}

Order::Order(vector<Goods\*> &goods, **const** map<**int**,**int**> &selected) {

**for**(**auto** item : selected) {

        good\_list[item.first] = item.second;

        total\_price += goods[item.first-1]->get\_price()\*item.second;

        state = 0;

        //冷冻商品

        goods[item.first-1]->set\_remaining(goods[item.first-1]->get\_remaining()-item.second);

    }

}

//释放冻结的商品余量

**void** Order::release(vector<Goods\*> &goods) {

**for**(**auto** item : good\_list) {

        goods[item.first-1]->set\_remaining(goods[item.first-1]->get\_remaining()+item.second);

    }

}

**void** Order::showOrder(vector<Goods\*> &goods) {

    cout<<"订单中商品如下："<<endl;

**for**(**auto** item : good\_list) {

        cout<<"商品名称："<<goods[item.first-1]->get\_name()<<"，商品单价："<<goods[item.first-1]->get\_price()<<"，商品件数："<<item.second<<endl;

    }

    cout<<"订单总金额："<<total\_price<<endl;

**if**(state==0) cout<<"订单状态：待支付"<<endl;

**else** cout<<"订单状态：已支付"<<endl;

}

**void** Order::transfer(vector<Goods\*> &goods) {

**for**(**auto** item : good\_list) {

**double** money = goods[item.first-1]->get\_price()\*item.second;

        //商品盈利转入商家账户

        goods[item.first-1]->get\_merchant()->get\_account()->transIn(money);

    }

}

1. 综合2

同综合1

1. 综合3
2. main.cpp

#include <iostream>

#include "server.h"

#include "client.hpp"

**using** **namespace** std;

**int** main(**int** argc, **const** **char** \* argv[]) {

**if** (argc <= 1)

        {

            std::cout << "使用方法:\n"

                      << "打开一个服务器 -s <port>\n"

                      << "打开一个客户端 -c <ip> <port>\n"

                      << "帮助信息 -h\n";

        }

**else**

        {

**if** (strcmp(argv[1], "-s") == 0)

            {

                    Server server;

                    server.start();

            }

**else** **if** (strcmp(argv[1], "-c") == 0)

            {

                    Client client;

                    client.start();

            }

**else**

            {

                std::cout << "Usage :\n"

                          << "Start a server : app -s <port>\n"

                          << "Run a client   : app -c <ip> <port>\n"

                          << "Help info      : app -h\n";

            }

        }

**return** 0;

    }

1. client.hpp

#ifndef client\_hpp

#define client\_hpp

#include <sys/types.h>

#include <sys/socket.h>

#include <stdio.h>

#include <unistd.h>

#include <string.h>

#include <arpa/inet.h>

#include <stdlib.h>

#include <netinet/in.h>

#include <iostream>

#include <string>

#include <vector>

#include "account.hpp"

#define PORT 9990

#define SIZE 1024

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**using** std::ifstream;

**using** std::ios;

**using** std::to\_string;

**class** Account;

**class** Client {

**public**:

    Client();

    ~Client();

**void** init();

**void** start();

**void** menu();

**void** find();

**private**:

**int** connect\_socket;

    string id;

    string type;

};

#endif /\* client\_hpp \*/

1. client.cpp

#include "client.hpp"

Client::Client() {

    id = "";

    type = "";

}

Client::~Client() {

}

**void** Client::init() {

    connect\_socket = socket(AF\_INET, SOCK\_STREAM, 0);   //创建和服务器连接套接字

**if**(connect\_socket == -1)

    {

        perror("socket");

**return**;

    }

**struct** sockaddr\_in addr;

    memset(&addr, 0, **sizeof**(addr));

    addr.sin\_family = AF\_INET;  /\* Internet地址族 \*/

    addr.sin\_port = htons(PORT);  /\* 端口号 \*/

    addr.sin\_addr.s\_addr = htonl(INADDR\_ANY);   /\* IP地址 \*/

    inet\_aton("127.0.0.1", &(addr.sin\_addr));

**int** addrlen = **sizeof**(addr);

**int** ret =  connect(connect\_socket,  (**struct** sockaddr \*)&addr, addrlen); //连接服务器

**if**(ret == -1) {

        perror("connect");

**return**;

    }

    printf("成功连接服务器\n");

}

**void** Client::menu() {

**if**(type.empty()) {

        cout << endl<<"       请选择操作序号！(0~2)" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*              0.退出              \*" << endl;

        cout << "\*              1.登陆              \*" << endl;

        cout << "\*              2.注册              \*" << endl;

        cout << "\*           3.查看所有账户           \*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    } **else** **if**(type == "merchant") {

        cout << endl<<"      请选择操作序号！(0~6)" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cout << "\*                                  \*" << endl;

        cout << "\*           0.退出登陆              \*" << endl;

        cout << "\*           1.商品名称管理           \*" << endl;

        cout << "\*           2.商品描述管理           \*" << endl;

        cout << "\*           3.商品剩余量管理         \*" << endl;

        cout << "\*           4.商品价格管理           \*" << endl;

        cout << "\*           5.添加商品              \*" << endl;

        cout << "\*           6.同类商品打折           \*" << endl;

        cout << "\*           7.查询商品              \*" << endl;

        cout << "\*           8.更改密码              \*" << endl;

        cout << "\*           9.余额查询              \*" << endl;

        cout << "\*           10.充值                \*" << endl;

        cout << "\*           11.消费                \*" << endl;

        cout << "\*           12.查看所有商品          \*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    } **else** **if**(type == "consumer") {

        cout << endl<<"      请选择操作序号！(0~14)" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*            0.退出登陆             \*" << endl;

        cout << "\*            1.修改密码             \*" << endl;

        cout << "\*            2.余额查询             \*" << endl;

        cout << "\*            3.充值                \*" << endl;

        cout << "\*            4.购买商品             \*" << endl;

        cout << "\*            5.查询商品             \*" << endl;

        cout << "\*            6.增加商品             \*" << endl;

        cout << "\*            7.删除商品             \*" << endl;

        cout << "\*            8.修改数量             \*" << endl;

        cout << "\*            9.选中商品             \*" << endl;

        cout << "\*            10.取消选中            \*" << endl;

        cout << "\*            11.生成订单            \*" << endl;

        cout << "\*            12.支付订单            \*" << endl;

        cout << "\*            13.取消订单            \*" << endl;

        cout << "\*            14.查看全部订单         \*" << endl;

        cout << "\*            15.查看购物车          \*" << endl;

        cout << "\*            16.查看所有商品        \*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

    }

}

**void** Client::find() {

        cout << endl<<"      请选择操作序号！(0~6)" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*              0.退出              \*" << endl;

        cout << "\*          1.筛选商品名称           \*" << endl;

        cout << "\*       2.筛选高于给定价格的商品      \*" << endl;

        cout << "\*       3.筛选低于给定价格的商品      \*" << endl;

        cout << "\*       4.筛选高于给定余量的商品      \*" << endl;

        cout << "\*       5.筛选低于给定余量的商品      \*" << endl;

        cout << "\*           6.筛选商品种类          \*" << endl;

        cout << "\*                                 \*" << endl;

        cout << "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*" << endl;

**int** x;

**char** buffer\_send[SIZE];

**char** buffer\_recv[SIZE];

        cin>>x;

**switch** (x) {

**case** 0:

                cout<<"您已退出商品查询"<<endl;

**break**;

**case** 1: {

                buffer\_send[0] = 13;

                string name;

                cout<<"请输入您要查找的名称"<<endl;

                cin>>name;

                memcpy(buffer\_send+1, name.c\_str(), name.size());

                send(connect\_socket, buffer\_send, name.size()+1, 0);

                recv(connect\_socket, buffer\_recv, SIZE, 0);

                cout<<buffer\_recv;

**break**;

            }

**case** 2: {

                buffer\_send[0] = 14;

**double** price;

                cout<<"请输入您要筛选的价格下限"<<endl;

                cin>>price;

                memcpy(buffer\_send+1, to\_string(price).c\_str(), to\_string(price).size());

                send(connect\_socket, buffer\_send, to\_string(price).size()+1,0);

                recv(connect\_socket, buffer\_recv, SIZE, 0);

                cout<<buffer\_recv;

**break**;

            }

**case** 3: {

                buffer\_send[0] = 15;

**double** price;

                cout<<"请输入您要筛选的价格上限"<<endl;

                cin>>price;

                memcpy(buffer\_send+1, to\_string(price).c\_str(), to\_string(price).size());

                send(connect\_socket, buffer\_send, to\_string(price).size()+1,0);

                recv(connect\_socket, buffer\_recv, SIZE, 0);

                cout<<buffer\_recv;

**break**;

            }

**case** 4: {

                buffer\_send[0] = 16;

**int** remain;

                cout<<"请输入您要筛选的余量下限"<<endl;

                cin>>remain;

                memcpy(buffer\_send+1, to\_string(remain).c\_str(), to\_string(remain).size());

                send(connect\_socket, buffer\_send,to\_string(remain).size()+1,0);

                recv(connect\_socket, buffer\_recv, SIZE, 0);

                cout<<buffer\_recv;

**break**;

            }

**case** 5: {

                buffer\_send[0] = 17;

**int** remain;

                cout<<"请输入您要筛选的余量上限"<<endl;

                cin>>remain;

                memcpy(buffer\_send+1, to\_string(remain).c\_str(), to\_string(remain).size());

                send(connect\_socket, buffer\_send,to\_string(remain).size()+1,0);

                recv(connect\_socket, buffer\_recv, SIZE, 0);

                cout<<buffer\_recv;

**break**;

            }

**case** 6: {

                buffer\_send[0] = 18;

                string ty;

                cout<<"请输入您要查找的类型"<<endl;

                cin>>ty;

                memcpy(buffer\_send+1, ty.c\_str(), ty.size());

                send(connect\_socket, buffer\_send, ty.size()+1,0);

                recv(connect\_socket, buffer\_recv, SIZE, 0);

                cout<<buffer\_recv;

**break**;

            }

**default**:

**break**;

    }

}

**void** Client::start()

{

    init();

**char** buffer\_recv[SIZE];

**char** buffer\_send[SIZE];

**int** ope;

    menu();

    printf("请输入您想进行的操作：");

**while**(cin>>ope) {

        memset(buffer\_send, 0, SIZE);

        memset(buffer\_recv, 0, SIZE);

**if**(type.empty()) {

**switch** (ope) {

**case** 0: {

                    close(connect\_socket);

                    cout<<"您已退出系统"<<endl;

**return**;

                }

**break**;

**case** 1: {

                    buffer\_send[0] = 1;

                    string id,psw;

                    cout<<"请输入您的账号"<<endl;

                    cin>>id;

                    cout<<"请输入您的密码"<<endl;

                    cin>>psw;

                    string tmp = id+" "+psw;

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    send(connect\_socket, buffer\_send, tmp.size()+1, 0);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]=='0') {

                        cout<<"该账号不存在"<<endl;

                    } **else** **if**(buffer\_recv[0]=='1') {

                        cout<<"密码错误"<<endl;

                    } **else** {

                        type = buffer\_recv+1;

**this**->id = id;

                        cout<<"登录成功，账号id为："<<id<<"类型为："<<type<<endl;

                    }

                }

**break**;

**case** 2: {

                    buffer\_send[0] = 0;

                    string id,psw,tmp\_type;

                    cout<<"请输入您要注册的账号"<<endl;

                    cin>>id;

                    cout<<"请输入您的密码"<<endl;

                    cin>>psw;

                    cout<<"请输入您的想注册的类型"<<endl;

                    cin>>tmp\_type;

                    string tmp = id+" "+psw+" "+tmp\_type;

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    send(connect\_socket, buffer\_send, tmp.size()+1, 0);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0) {

                        cout<<"账号已存在"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"账号注册成功"<<endl;

                    }

                }

**break**;

**case** 3: {

                    buffer\_send[0] = 30;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

                    cout<<buffer\_recv;

                }

**default**:

**break**;

            }

        } **else** **if**(type=="merchant") {

**switch** (ope) {

**case** 0: {

                    buffer\_send[0] = 29;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==1) {

                        id = "";

                        type = "";

                        cout<<"您已成功退出登录"<<endl;

                    }

                }

**break**;

**case** 1: {

                    buffer\_send[0] = 3;

                    cout<<"请输入你想更改的编号"<<endl;

**int** no;

                    cin>>no;

                    cout<<"请输入名称"<<endl;

                    string name;

                    cin>>name;

                    string tmp = std::to\_string(no)+" "+name;

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0){

                        cout<<"编号不合法"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"您无权管理该商品"<<endl;

                    } **else** {

                        cout<<"更改成功"<<endl;

                    }

                }

**break**;

**case** 2: {

                    buffer\_send[0] = 4;

                    cout<<"请输入你想更改的编号"<<endl;

**int** no;

                    cin>>no;

                    cout<<"请输入描述"<<endl;

                    string des;

                    cin>>des;

                    string tmp = std::to\_string(no)+" "+des;

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0){

                        cout<<"编号不合法"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"您无权管理该商品"<<endl;

                    } **else** {

                        cout<<"更改成功"<<endl;

                    }

                }

**break**;

**case** 3: {

                    buffer\_send[0] = 5;

                    cout<<"请输入你想更改的编号"<<endl;

**int** no;

                    cin>>no;

                    cout<<"请输入余量"<<endl;

**int** remain;

                    cin>>remain;

                    string tmp = std::to\_string(no)+" "+std::to\_string(remain);

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0){

                        cout<<"编号不合法"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"您无权管理该商品"<<endl;

                    } **else** {

                        cout<<"更改成功"<<endl;

                    }

                }

**break**;

**case** 4: {

                    buffer\_send[0] = 6;

                    cout<<"请输入你想更改的编号"<<endl;

**int** no;

                    cin>>no;

                    cout<<"请输入价格"<<endl;

**double** price;

                    cin>>price;

                    string tmp = std::to\_string(no)+" "+std::to\_string(price);

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0){

                        cout<<"编号不合法"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"您无权管理该商品"<<endl;

                    } **else** {

                        cout<<"更改成功"<<endl;

                    }

                }

**break**;

**case** 5: {

                    buffer\_send[0] = 7;

                    string ty,name,des;

**double** price;

**int** remain;

                    cout<<"请输入你想添加的商品种类(book/food/clothing)"<<endl;

                    cin>>ty;

                    cout<<"请输入商品名称(用下划线代替空格)"<<endl;

                    cin>>name;

                    cout<<"请输入商品描述(用下划线代替空格)"<<endl;

                    cin>>des;

                    cout<<"请输入商品价格"<<endl;

                    cin>>price;

                    cout<<"请输入商品剩余量"<<endl;

                    cin>>remain;

                    string tmp = ty+" "+name+" "+des+" "+to\_string(price)+" "+to\_string(remain);

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==1) {

                        cout<<"商品添加成功"<<endl;

                    }

                }

**break**;

**case** 6: {

                    buffer\_send[0] = 8;

                    string ty,tmp;

**double** rate;

                    cout<<"请输入你想打折的商品种类(book/food/clothing)"<<endl;

                    cin>>ty;

                    cout<<"输入你想打折的折扣率"<<endl;

                    cin>>rate;

                    tmp = ty+" "+to\_string(rate);

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==1) {

                        cout<<"商品打折成功"<<endl;

                    }

**break**;

                }

**case** 7: {

                    find(connect\_socket);

**break**;

                }

**case** 8: {

                    buffer\_send[0] = 9;

                    string psw,check;

**do** {

                        cout<<"请输入新的密码"<<endl;

                        cin>>psw;

                        cout<<"请确认密码"<<endl;

                        cin>>check;

**if**(check!=psw) cout<<"两次输入密码不同"<<endl;

                    } **while**(check!=psw);

                    memcpy(buffer\_send+1, psw.c\_str(), psw.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==1) {

                        cout<<"密码修改成功"<<endl;

                    }

**break**;

                }

**case** 9: {

                    buffer\_send[0] = 10;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

                    cout<<buffer\_recv;

**break**;

                }

**case** 10: {

                    buffer\_send[0] = 11;

**double** money;

                    cout<<"请输入您想充值的金额"<<endl;

                    cin>>money;

                    memcpy(buffer\_send+1, to\_string(money).c\_str(), to\_string(money).size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==1) {

                        cout<<"充值成功"<<endl;

                    }

**break**;

                }

**case** 11: {

                    buffer\_send[0] = 12;

                    cout<<"请输入你想购买的商品编号："<<endl;

**int** n;

                    cin>>n;

                    memcpy(buffer\_send+1, to\_string(n).c\_str(), to\_string(n).size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

                    cout<<buffer\_recv;

**break**;

                }

**case** 12: {

                    buffer\_send[0] = 2;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

                    cout<<buffer\_recv;

**break**;

                }

**default**:

**break**;

            }

        } **else** **if**(type=="consumer") {

**switch** (ope) {

**case** 0: {

                    buffer\_send[0] = 29;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==1) {

                        id = "";

                        type = "";

                        cout<<"您已成功退出登录"<<endl;

                    }

**break**;

                }

**case** 1: {

                    buffer\_send[0] = 9;

                    string psw,check;

**do** {

                        cout<<"请输入新的密码"<<endl;

                        cin>>psw;

                        cout<<"请确认密码"<<endl;

                        cin>>check;

**if**(check!=psw) cout<<"两次输入密码不同"<<endl;

                    } **while**(check!=psw);

                    memcpy(buffer\_send+1, psw.c\_str(), psw.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==1) {

                        cout<<"密码修改成功"<<endl;

                    }

**break**;

                }

**case** 2: {

                    buffer\_send[0] = 10;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

                    cout<<buffer\_recv;

**break**;

                }

**case** 3: {

                    buffer\_send[0] = 11;

**double** money;

                    cout<<"请输入您想充值的金额"<<endl;

                    cin>>money;

                    memcpy(buffer\_send+1, to\_string(money).c\_str(), to\_string(money).size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==1) {

                        cout<<"充值成功"<<endl;

                    }

**break**;

                }

**case** 4: {

                    buffer\_send[0] = 12;

                    cout<<"请输入你想购买的商品编号："<<endl;

**int** n;

                    cin>>n;

                    memcpy(buffer\_send+1, to\_string(n).c\_str(), to\_string(n).size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

                    cout<<buffer\_recv;

**break**;

                }

**case** 5:

                    find(connect\_socket);

**break**;

                //add cart

**case** 6: {

                    buffer\_send[0] = 19;

**int** no,amount;

                    cout<<"请输入您要添加入购物车的商品编号"<<endl;

                    cin>>no;

                    cout<<"请输入您要添加的件数"<<endl;

                    cin>>amount;

                    string tmp = to\_string(no)+" "+to\_string(amount);

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0) {

                        cout<<"商品编号不合法"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"商品数量不合法"<<endl;

                    } **else** {

                        cout<<"添加购物车成功"<<endl;

                    }

**break**;

                }

**case** 7: {

                    buffer\_send[0] = 20;

**int** no,amount;

                    cout<<"请输入您要从购物车移除的商品编号"<<endl;

                    cin>>no;

                    cout<<"请输入您要移除的件数"<<endl;

                    cin>>amount;

                    string tmp = to\_string(no)+" "+to\_string(amount);

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0) {

                        cout<<"商品编号不合法"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"商品数量不合法"<<endl;

                    } **else** {

                        cout<<"移出购物车成功"<<endl;

                    }

**break**;

                }

**case** 8: {

                    buffer\_send[0] = 21;

**int** no,amount;

                    cout<<"请输入您要更改数量的商品编号"<<endl;

                    cin>>no;

                    cout<<"请输入您要更改后的件数"<<endl;

                    cin>>amount;

                    string tmp = to\_string(no)+" "+to\_string(amount);

                    memcpy(buffer\_send+1, tmp.c\_str(), tmp.size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0) {

                        cout<<"商品编号不合法"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"商品数量不合法"<<endl;

                    } **else** {

                        cout<<"更改购物车成功"<<endl;

                    }

**break**;

                }

                //select

**case** 9: {

                    buffer\_send[0] = 22;

**int** no;

                    cout<<"请输入您要选中的商品编号"<<endl;

                    cin>>no;

                    memcpy(buffer\_send+1, to\_string(no).c\_str(), to\_string(no).size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0) {

                        cout<<"商品编号不合法"<<endl;

                    } **else** {

                        cout<<"选中成功"<<endl;

                    }

**break**;

                }

**case** 10: {

                    buffer\_send[0] = 23;

**int** no;

                    cout<<"请输入您要取消选中的商品编号"<<endl;

                    cin>>no;

                    memcpy(buffer\_send+1, to\_string(no).c\_str(), to\_string(no).size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0) {

                        cout<<"商品编号不合法"<<endl;

                    } **else** {

                        cout<<"取消选中成功"<<endl;

                    }

**break**;

                }

**case** 11: {

                    buffer\_send[0] = 24;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==1) {

                        cout<<"生成订单成功"<<endl;

                    }

**break**;

                }

**case** 12: {

                    buffer\_send[0] = 25;

**int** no;

                    cout<<"请输入您要支付的订单编号"<<endl;

                    cin>>no;

                    memcpy(buffer\_send+1, to\_string(no).c\_str(), to\_string(no).size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0) {

                        cout<<"订单编号不合法"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"账户余额不足"<<endl;

                    } **else** {

                        cout<<"支付订单成功"<<endl;

                    }

**break**;

                }

**case** 13: {

                    buffer\_send[0] = 26;

**int** no;

                    cout<<"请输入您要取消的订单编号"<<endl;

                    cin>>no;

                    memcpy(buffer\_send+1, to\_string(no).c\_str(), to\_string(no).size());

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(buffer\_recv[0]==0) {

                        cout<<"订单编号不合法"<<endl;

                    } **else** **if**(buffer\_recv[0]==1) {

                        cout<<"订单已支付"<<endl;

                    } **else** {

                        cout<<"取消订单成功"<<endl;

                    }

**break**;

                }

**case** 14: {

                    buffer\_send[0] = 27;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

                    cout<<buffer\_recv;

**break**;

                }

**case** 15: {

                    buffer\_send[0] = 28;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

                    cout<<buffer\_recv;

**break**;

                }

**case** 16: {

                    buffer\_send[0] = 2;

                    write(connect\_socket, buffer\_send, SIZE);

                    recv(connect\_socket, buffer\_recv, SIZE, 0);

                    cout<<buffer\_recv;

**break**;

                }

**default**:

**break**;

            }

        }

        menu();

        cout<<"请输入您想进行的操作"<<endl;

    }

    close(connect\_socket);

}

1. server.hpp

#ifndef server\_h

#define server\_h

#include <sys/types.h>

#include <sys/socket.h>

#include <stdio.h>

#include <unistd.h>

#include <arpa/inet.h>

#include <pthread.h>

#include <stdlib.h>

#include <netinet/in.h>

#include <iostream>

#include <fstream>

#include <string>

#include <vector>

#include <map>

#include <mutex>

#include "account.hpp"

#include "merchant.hpp"

#include "consumer.hpp"

#include "goods.hpp"

#include "book.hpp"

#include "clothing.hpp"

#include "food.hpp"

#define PORT 9990

#define SIZE 1024

**class** Account;

**class** Goods;

**class** Merchant;

**class** Consumer;

**class** Food;

**class** Goods;

**class** Clothing;

**class** Server {

**public**:

    Server();

    ~Server();

**void** initSocket();

**void** waitClient();

    //客户端传来的信息包括：用户ID，操作编号，其他数据（对应不同的操作有不同的数据）

**void** handleClient();

**void** start();

**void** readFile();

**void** writeFile();

**private**:

**int** listen\_socket;

**int** connect\_socket;

**struct** sockaddr\_in server\_addr;

    map<string, Account\*> id2account;    //账号到账户的映射

    vector<Account\*> accounts;

    map<string, Merchant\*> merchants;

    map<string, Consumer\*> consumers;

    //存储所有的商品

    vector<Goods\*> goods;

    Account \*online\_account;

};

#endif /\* server\_h \*/

1. server.cpp

#include <sstream>

#include <string>

#include "server.h"

**using** std::cout;

**using** std::cin;

**using** std::string;

**using** std::stringstream;

**using** std::ifstream;

**using** std::ios;

**using** std::ofstream;

**using** std::endl;

**using** std::to\_string;

Server::Server() {

    readFile();

}

Server::~Server() {

    writeFile();

}

**void** Server::initSocket() {

    listen\_socket = socket(AF\_INET, SOCK\_STREAM, 0);//TCP

**if**(listen\_socket==-1) {

        perror("socket");

**return**;

    }

**struct** sockaddr\_in server\_addr;

    memset(&server\_addr, 0, **sizeof**(server\_addr));

    server\_addr.sin\_family = AF\_INET;

    server\_addr.sin\_port = htons(PORT);

    server\_addr.sin\_addr.s\_addr = INADDR\_ANY;

**int** ret = bind(listen\_socket, (**struct** sockaddr\*)&server\_addr, **sizeof**(server\_addr));

**if**(ret==-1) {

        perror("bind");

**return**;

    }

    ret = listen(listen\_socket, 10);

**if**(ret==-1) {

        perror("listen");

**return**;

    }

    cout<<"服务器已创建"<<endl;

}

**void** Server::waitClient() {

**struct** sockaddr\_in client\_addr;

**int** client\_addr\_len = **sizeof**(client\_addr);

    connect\_socket = accept(listen\_socket, (**struct** sockaddr \*)&client\_addr, (socklen\_t\*)&client\_addr\_len);

**if**(connect\_socket==-1) {

        perror("connect");

**return**;

    }

    printf("成功连接到一个客户端：%s\n", inet\_ntoa(client\_addr.sin\_addr));

}

**void** Server::handleClient() {

**char** buffer\_recv[SIZE];

**char** buffer\_send[SIZE];

**int** len = 0;

**while**(1) {

        memset(buffer\_send, 0, SIZE);

        memset(buffer\_recv, 0, SIZE);

        len = 0;

**int** ret = (**int**)recv(connect\_socket, buffer\_recv, SIZE, 0);

**if**(ret==-1) {

            perror("read");

**break**;

        }

**if**(ret == 0) **break**;

**char** ope = buffer\_recv[0];

        stringstream ss(buffer\_recv+1);

**switch** (ope) {

            //sign in

**case** 0: {

                string id,psw,type;

                ss>>id>>psw>>type;

                //账号已存在

**if**(id2account[id]!=**nullptr**) {

                    buffer\_send[0] = 0;

                    len = 1;

                }

**else** {

                    Account \*new\_account = **new** Account(id,psw,type);

                    //新注册账号对应的用户存入merchants或consumers

**if**(type=="merchant") {

                        Merchant \*new\_merchant = **new** Merchant(new\_account);

                        merchants[id] = new\_merchant;

                    } **else** {

                        Consumer \*new\_consumer = **new** Consumer(new\_account);

                        consumers[id] = new\_consumer;

                    }

                    accounts.push\_back(new\_account);

                    id2account[id] = new\_account;

                    buffer\_send[0] = 1;

                    len = 1;

                }

            }

**break**;

            //login(0：账户不存在，1:密码错误，2:登录成功)

**case** 1: {

                string id,psw;

                ss>>id>>psw;

**if**(id2account.find(id)==id2account.end()) {

                    buffer\_send[0] = '0';

                    len = 1;

                } **else** **if**(psw!=id2account[id]->get\_password()) {

                    buffer\_send[0] = '1';

                    len = 1;

                } **else** {

                    online\_account = id2account[id];

                    buffer\_recv[0] = '2';

                    memcpy(buffer\_send+1, online\_account->get\_type().c\_str(), online\_account->get\_type().size());

                    len = (**int**)online\_account->get\_type().size()+1;

                }

            }

**break**;

            //show\_goods

**case** 2: {

**if**(goods.size()==0) {

                    string tmp = "当前系统中没有商品";

                    memcpy(buffer\_send, tmp.c\_str(), tmp.size());

                    len = (**int**)tmp.size();

                }

**else** {

                    string tmp = "当前系统中有如下商品：\n";

                    //商品编号从1开始，和下标有1的偏差

**for** (**int** i = 0; i<goods.size(); i++) {

                        tmp += "商品编号："+std::to\_string(i+1)+"\n";

                        tmp += goods[i]->show\_good();

                    }

                    memcpy(buffer\_send, tmp.c\_str(), tmp.size());

                    len = (**int**)tmp.size();

                }

            }

**break**;

            //set\_name(0:编号不合法，1:商品不属于该账号,2:成功）

**case** 3: {

**int** no;

                string name;

                ss>>no>>name;

**if**(no>goods.size()||no<1) {

                    buffer\_send[0] = 0;

                } **else** **if**(goods[no-1]->get\_merchant()!=merchants[online\_account->get\_id()]) {

                    buffer\_send[0] = 1;

                } **else** {

                    goods[no-1]->set\_name(name);

                    buffer\_send[0] = 2;

                }

                len = 1;

            }

**break**;

            //set\_des

**case** 4: {

**int** no;

                string des;

                ss>>no>>des;

**if**(no>goods.size()||no<1) {

                    buffer\_send[0] = 0;

                } **else** **if**(goods[no-1]->get\_merchant()!=merchants[online\_account->get\_id()]) {

                    buffer\_send[0] = 1;

                } **else** {

                    goods[no-1]->set\_description(des);

                    buffer\_send[0] = 2;

                }

                len = 1;

            }

**break**;

            //set\_remain

**case** 5: {

**int** no,remain;

                ss>>no>>remain;

**if**(no>goods.size()||no<1) {

                    buffer\_send[0] = 0;

                } **else** **if**(goods[no-1]->get\_merchant()!=merchants[online\_account->get\_id()]) {

                    buffer\_send[0] = 1;

                } **else** {

                    goods[no-1]->set\_remaining(remain);

                    buffer\_send[0] = 2;

                }

                len = 1;

            }

**break**;

            //set\_price

**case** 6: {

**int** no;

**double** price;

                ss>>no>>price;

**if**(no>goods.size()||no<1) {

                    buffer\_send[0] = 0;

                } **else** **if**(goods[no-1]->get\_merchant()!=merchants[online\_account->get\_id()]) {

                    buffer\_send[0] = 1;

                } **else** {

                    goods[no-1]->set\_price(price);

                    buffer\_send[0] = 2;

                }

                len = 1;

            }

**break**;

            //add\_good

**case** 7: {

                string type,name,des;

**int** remain;

**double** price;

                Goods\* good;

                ss>>type>>name>>des>>price>>remain;

**if**(type=="book") {

                    good = **new** Book(name,des,price,remain,type,merchants[online\_account->get\_id()]);

                } **else** **if**(type=="food") {

                    good = **new** Food(name,des,price,remain,type,merchants[online\_account->get\_id()]);

                } **else** {

                    good = **new** Clothing(name,des,price,remain,type,merchants[online\_account->get\_id()]);

                }

                goods.push\_back(good);

                buffer\_send[0] = 1;

                len = 1;

            }

**break**;

            //discount

**case** 8: {

                string type;

**double** rate;

                ss>>type>>rate;

**for**(**auto** &good : goods) {

                    //只能管理属于自己的商品

**if**(good->get\_type()==type &&

                       good->get\_merchant()==merchants[online\_account->get\_id()])

                        good->set\_price(good->get\_price()\*rate);

                }

                buffer\_send[0] = 1;

                len = 1;

            }

**break**;

            //set\_psw

**case** 9: {

                string psw;

                ss>>psw;

                online\_account->set\_password(psw);

                buffer\_send[0] = 1;

                len = 1;

            }

**break**;

            //get\_balance

**case** 10: {

                string tmp = "您账户余额还有："+std::to\_string(online\_account->get\_balance())+"元\n";

                memcpy(buffer\_send,tmp.c\_str(),tmp.size());

                len = (**int**)tmp.size();

            }

**break**;

            //recharge

**case** 11: {

**double** money;

                ss>>money;

                online\_account->recharge(money);

                buffer\_send[0] = 1;

                len = 1;

            }

**break**;

            //purchase

**case** 12: {

                string tmp;

**int** no;

                ss>>no;

**if**(goods[no-1]->get\_remaining()==0) {

                    tmp = "商品库存不足";

                }

**else** **if**(goods[no-1]->get\_price()>online\_account->get\_balance())

                {

                    tmp = "您账户中没有足够的余额，商品需要"+

                        to\_string(goods[no-1]->get\_price())+"元，您只有"+

                        to\_string(online\_account->get\_balance())+"元\n";

                }

**else**

                {

                    online\_account->consume(goods[no-1]->get\_price());

                    goods[no-1]->set\_remaining(goods[no-1]->get\_remaining()-1);

                    tmp = "您买下了"+to\_string(no)+"号商品："+

                        goods[no-1]->get\_name()+"\n";

                }

                memcpy(buffer\_send, tmp.c\_str(), tmp.size());

                len = (**int**)tmp.size();

            }

**break**;

            //search\_name

**case** 13: {

                string name;

                string tmp = "";

                ss>>name;

**int** flag = 0;

**int** i = 0;

**for**(i = 0; i<goods.size(); i++) {

**if**(goods[i]->get\_name()==name) {

                        flag = 1;

                        tmp += "商品编号："+to\_string(i+1)+goods[i]->show\_good()+"\n";

                    }

                }

**if**(flag==0) {

                    tmp = "没有找到您查找的商品\n";

                }

                len = (**int**)tmp.size();

                memcpy(buffer\_send, tmp.c\_str(), tmp.size());

            }

**break**;

            //find by low price

**case** 14: {

**double** price;

                ss>>price;

                string tmp;

**int** flag = 0;

**int** i = 0;

**for**(i = 0; i<goods.size(); i++) {

**if**(goods[i]->get\_price()>=price) {

                        flag = 1;

                        tmp += "商品编号："+to\_string(i+1)+goods[i]->show\_good()+"\n";

                    }

                }

**if**(flag==0) {

                    tmp = "没有找到您查找的商品\n";

                }

                len = (**int**)tmp.size();

                memcpy(buffer\_send, tmp.c\_str(), tmp.size());

            }

**break**;

**case** 15: {

**double** price;

                ss>>price;

                string tmp;

**int** flag = 0;

**int** i = 0;

**for**(i = 0; i<goods.size(); i++) {

**if**(goods[i]->get\_price()<=price) {

                        flag = 1;

                        tmp += "商品编号："+to\_string(i+1)+goods[i]->show\_good()+"\n";

                    }

                }

**if**(flag==0) {

                    tmp = "没有找到您查找的商品\n";

                }

                len = (**int**)tmp.size();

                memcpy(buffer\_send, tmp.c\_str(), tmp.size());

            }

**break**;

            //find by remaining low

**case** 16: {

**int** remain;

                ss>>remain;

                string tmp;

**int** flag = 0;

**int** i = 0;

**for**(i = 0; i<goods.size(); i++) {

**if**(goods[i]->get\_price()>=remain) {

                        flag = 1;

                        tmp += "商品编号："+to\_string(i+1)+goods[i]->show\_good()+"\n";

                    }

                }

**if**(flag==0) {

                    tmp = "没有找到您查找的商品\n";

                }

                len = (**int**)tmp.size();

                memcpy(buffer\_send, tmp.c\_str(), tmp.size());

            }

**break**;

**case** 17: {

**int** remain;

                ss>>remain;

                string tmp;

**int** flag = 0;

**int** i = 0;

**for**(i = 0; i<goods.size(); i++) {

**if**(goods[i]->get\_price()<=remain) {

                        flag = 1;

                        tmp += "商品编号："+to\_string(i+1)+goods[i]->show\_good()+"\n";

                    }

                }

**if**(flag==0) {

                    tmp = "没有找到您查找的商品\n";

                }

                len = (**int**)tmp.size();

                memcpy(buffer\_send, tmp.c\_str(), tmp.size());

            }

**break**;

**case** 18: {

                string type;

                ss>>type;

                string tmp;

**int** flag = 0;

**int** i = 0;

**for**(i = 0; i<goods.size(); i++) {

**if**(goods[i]->get\_type()==type) {

                        flag = 1;

                        tmp += "商品编号："+to\_string(i+1)+goods[i]->show\_good()+"\n";

                    }

                }

**if**(flag==0) {

                    tmp = "没有找到您查找的商品\n";

                }

                len = (**int**)tmp.size();

                memcpy(buffer\_send, tmp.c\_str(), tmp.size());

            }

**break**;

            //add cart

**case** 19: {

**int** no, amount;

                ss>>no>>amount;

**if** (no>goods.size()||no<1) {

                    buffer\_send[0] = 0;

                }

**else** **if**(amount>goods[no-1]->get\_remaining()-

                    consumers[online\_account->get\_id()]->get\_cart\_value(no)

                        ||amount<1)

                {

                    buffer\_send[0] = 1;

                }

**else** {

                    consumers[online\_account->get\_id()]->addCart(no, amount);

                    buffer\_send[0] = 2;

                }

                len = 1;

            }

**break**;

            //remove cart

**case** 20: {

**int** no, amount;

                ss>>no>>amount;

**if** (no>goods.size()||no<1) {

                    buffer\_send[0] = 0;

                }

**else** **if**(amount>goods[no-1]->get\_remaining()-

                    consumers[online\_account->get\_id()]->get\_cart\_value(no)

                        ||amount<1)

                {

                    buffer\_send[0] = 1;

                }

**else** {

                    consumers[online\_account->get\_id()]->rmCart(no, amount);

                    buffer\_send[0] = 2;

                }

                len = 1;

            }

**break**;

            //change cart

**case** 21: {

**int** no, amount;

                ss>>no>>amount;

**if** (no>goods.size()||no<1) {

                    buffer\_send[0] = 0;

                }

**else** **if**(amount>goods[no-1]->get\_remaining()-

                    consumers[online\_account->get\_id()]->get\_cart\_value(no)

                        ||amount<1)

                {

                    buffer\_send[0] = 1;

                }

**else** {

                    consumers[online\_account->get\_id()]->changeCart(no, amount);

                    buffer\_send[0] = 2;

                }

                len = 1;

            }

**break**;

            //select

**case** 22: {

**int** no;

                ss>>no;

**if** (no>goods.size()||no<1) {

                    buffer\_send[0] = 0;

                }

**else** {

                    consumers[online\_account->get\_id()]->select(no, goods);

                    buffer\_send[0] = 1;

                }

                len = 1;

            }

**break**;

            //unselect

**case** 23: {

**int** no;

                ss>>no;

**if** (no>goods.size()||no<1) {

                    buffer\_send[0] = 0;

                }

**else** {

                    consumers[online\_account->get\_id()]->unselect(no, goods);

                    buffer\_send[0] = 1;

                }

                len = 1;

            }

**break**;

            //generate order

**case** 24: {

                consumers[online\_account->get\_id()]->generateOrder(goods);

                buffer\_send[0] = 1;

                len = 1;

            }

**break**;

            //pay order

**case** 25: {

**int** no;

                ss>>no;

**if** (no<1||no>consumers[online\_account->get\_id()]->get\_orders().size()) {

                    buffer\_send[0] = 0;

                }

**else** **if**(online\_account->get\_balance()<consumers[online\_account->get\_id()]->get\_order(no).get\_price()) {

                    buffer\_send[0] = 1;

                }

**else** {

                    consumers[online\_account->get\_id()]->payOrder(no, goods);

                    buffer\_send[0] = 2;

                }

                len = 1;

            }

**break**;

            //cancel order

**case** 26: {

**int** no;

                ss>>no;

**if** (no<1||no>consumers[online\_account->get\_id()]->get\_orders().size()) {

                    buffer\_send[0] = 0;

                }

**else** **if**(consumers[online\_account->get\_id()]->get\_order(no).get\_state()) {

                    buffer\_send[0] = 1;

                }

**else** {

                    consumers[online\_account->get\_id()]->cancelOrder(no, goods);

                    buffer\_send[0] = 2;

                }

                len = 1;

            }

**break**;

            //show orders

**case** 27: {

                string tmp = consumers[online\_account->get\_id()]->showOrder(goods);

                memcpy(buffer\_send, tmp.c\_str(), tmp.size());

                len = (**int**)tmp.size();

            }

**break**;

            //show cart

**case** 28: {

                string tmp = consumers[online\_account->get\_id()]->showCart(goods);

                memcpy(buffer\_send, tmp.c\_str(), tmp.size());

                len = (**int**)tmp.size();

            }

**break**;

**case** 29: {

                online\_account = **nullptr**;

                buffer\_send[0] = 1;

                len = 1;

            }

**break**;

**case** 30: {

**if**(accounts.size()==0) {

                    string tmp = "当前系统中没有账号";

                    memcpy(buffer\_send, tmp.c\_str(), tmp.size());

                    len = (**int**)tmp.size();

                }

**else** {

                    string tmp = "当前系统中有如下账号：\n";

**for**(**auto** item : id2account) {

                        tmp += item.first+"\n"+item.second->show\_account()+"\n";

                    }

//                    //商品编号从1开始，和下标有1的偏差

//                    for (int i = 0; i<accounts.size(); i++) {

//                        tmp += accounts[i]->show\_account()+"\n";

//                    }

                    memcpy(buffer\_send, tmp.c\_str(), tmp.size());

                    len = (**int**)tmp.size();

                }

            }

**break**;

**default**:

**break**;

        }

        send(connect\_socket, buffer\_send, len, 0);

    }

    close(connect\_socket);

}

**void** Server::start() {

    initSocket();

    waitClient();

    handleClient();

    close(listen\_socket);

    printf("与客户端断开连接");

}

**void** Server::readFile() {

    ifstream fin\_account;

    fin\_account.open("/Users/tom/account.dic",std::ios::in);

**if** (!fin\_account.is\_open()) {

        cout << "文件打开失败"<<endl;

        exit(EXIT\_FAILURE);

    }

    string id, psw, type;

**double** balance;

**while**(fin\_account>>id>>psw>>balance>>type) {

        Account \*account = **new** Account(id,psw,balance,type);

        id2account[id] = account;

        accounts.push\_back(account);

        Merchant\* merchant;

        Consumer\* consumer;

**if**(type=="merchant") {

            merchant = **new** Merchant(account);

            merchants[account->get\_id()] = merchant;

        }

**else** {

            consumer = **new** Consumer(account);

            consumers[account->get\_id()] = consumer;

        }

    }//while

    fin\_account.close();

    ifstream fin\_goods;

    fin\_goods.open("/Users/tom/goods.dic",std::ios::in);

**if** (!fin\_goods.is\_open()) {

        cout << "文件打开失败"<<endl;

        exit(EXIT\_FAILURE);

    }

    string name, description;

**double** price;

**int** remaining;

**while**(fin\_goods>>name>>description>>price>>remaining>>type>>id) {

        Goods \*good;

**if**(type=="book") {

            good = **new** Book(name,description,price,remaining,type,merchants[id]);

        } **else** **if**(type=="food") {

            good = **new** Food(name,description,price,remaining,type,merchants[id]);

        } **else** {

            good = **new** Clothing(name,description,price,remaining,type,merchants[id]);

        }

        goods.push\_back(good);

    }//while

    fin\_goods.close();

}

**void** Server::writeFile() {

    ofstream fout\_account;

    fout\_account.open("/Users/tom/account.dic",std::ios::out|std::ios::trunc);

**if** (!fout\_account.is\_open()) {

        cout <<"文件打开失败"<<endl;

        exit(EXIT\_FAILURE);

    }

**for**(**auto** account : accounts) {

        fout\_account<<account->get\_id()<<" "<<account->get\_password()<<" "<<account->get\_balance()<<" "<<account->get\_type()<<endl;

    }

    fout\_account.close();

    ofstream fout\_goods;

    fout\_goods.open("/Users/tom/goods.dic",std::ios::out|std::ios::trunc);

**if** (!fout\_goods.is\_open()) {

        cout <<"文件打开失败"<<endl;

        exit(EXIT\_FAILURE);

    }

**for**(**auto** good : goods) {

        fout\_goods<<good->get\_name()<<" "<<good->get\_description()<<"  "<<good->get\_price()<<" "<<good->get\_remaining()<<" "<<good->get\_type()<<" "<<good->get\_merchant()->get\_account()->get\_id()<<endl;

    }

    fout\_goods.close();

}

1. account.hpp

#ifndef account\_hpp

#define account\_hpp

#include <stdio.h>

#include <iostream>

#include <string>

#include <vector>

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Account {

**public**:

    Account();

    Account(string id, string password, string type);

    Account(string id, string password, **double** balance, string type);

    ~Account();

    string get\_password(){**return** password;}

    string get\_id(){**return** id;}

**double** get\_balance(){**return** balance;}

    string get\_type(){**return** type;}

**void** set\_id(string id){**this**->id = id;}

**void** set\_password(string psw){password = psw;}

**void** recharge(**double** money);

**void** consume(**double** money);

    string show\_account();

**void** transIn(**double** money);

**void** transOut(**double** money);

**protected**:

    string id;

    string password;

**double** balance;//余额

    string type;   //"consumer"表示消费者，"merchant"表示商家

};

#endif /\* account\_hpp \*/

1. account.cpp

#include "account.hpp"

Account::Account(){}

Account::Account(string id,string password, string type):

id(id),password(password),balance(0),type(type) {

}

Account::Account(string id, string password, **double** balance, string type):id(id),password(password),balance(balance),type(type) {

}

Account::~Account() {

}

**void** Account::recharge(**double** money) {

    balance += money;

    cout<<"您本次充值"<<money<<"元，账户余额"<<balance<<"元"<<endl;

}

**void** Account::consume(**double** money) {

    balance -= money;

    cout<<"您本次消费"<<money<<"元，账户余额"<<balance<<"元"<<endl;

}

string Account::show\_account() {

    string tmp = "";

    tmp += "账号："+id+"\n";

    tmp += "密码："+password+"\n";

    tmp += "余额："+std::to\_string(balance)+"\n";

    tmp += "类型："+type+"\n";

**return** tmp;

}

**void** Account::transIn(**double** money) {

    balance += money;

    cout<<"账户："<<id<<" 转入 "<<money<<" 元"<<endl;

}

**void** Account::transOut(**double** money) {

    balance -= money;

    cout<<"账户："<<id<<" 转出 "<<money<<" 元"<<endl;

}

1. user.hpp

#ifndef user\_hpp

#define user\_hpp

#include <stdio.h>

#include <iostream>

#include <string>

#include <vector>

#include "account.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**using** std::vector;

**class** Account;

**class** Goods;

**class** User {

**public**:

    User();

    User(Account \*account);

**virtual** ~User();

**virtual** string getUserType() = 0;

    Account\* get\_account(){**return** account;}

**protected**:

    Account \*account;

};

#endif /\* user\_hpp \*/

1. user.cpp

#include "account.hpp"

#include "user.hpp"

User::User():account(){

}

User::User(Account \*account):account(account) {

}

User::~User() {

}

1. consumer.hpp

#ifndef consumer\_hpp

#define consumer\_hpp

#include <stdio.h>

#include <map>

#include "user.hpp"

#include "order.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**using** std::map;

**using** std::vector;

**class** Goods;

**class** Account;

**class** Order;

**class** Consumer : **public** User {

**public**:

    Consumer();

    Consumer(Account \*account);

    ~Consumer();

    string getUserType(){**return** "consumer";}

    string showCart(**const** vector<Goods\*> &goods);

    string showOrder(**const** vector<Goods\*> &goods);

**void** addCart(**int** no, **int** amount);

**void** rmCart(**int** no, **int** amount);

**void** changeCart(**int** no, **int** amount);

**void** select(**int** no, vector<Goods\*> &goods);

**void** unselect(**int** no, vector<Goods\*> &goods);

**void** generateOrder(vector<Goods\*> &goods);   //订单生成

**void** payOrder(**int** no, vector<Goods\*> &goods);    //订单支付

**void** cancelOrder(**int** no, vector<Goods\*> &goods); //订单取消

**int** get\_cart\_value(**int** no) {**return** cart[no];}

    vector<Order> get\_orders() {**return** orders;}

    Order get\_order(**int** no) {**return** orders[no-1];}

**private**:

    //商品编号需要-1才等于商品下标

    //商品编号作为key，商品数量作为value，需要映射到goods数组中获取商品具体信息

    map<**int**,**int**> cart;

    //商品编号作为key，是否选中作为value

    map<**int**,**int**> selected;

**double** total\_price;

    //除非订单取消，否则都会存在这里（待支付/已支付）

    vector<Order> orders;

};

#endif /\* consumer\_hpp \*/

1. consumer.cpp

#include "account.hpp"

#include "goods.hpp"

#include "consumer.hpp"

Consumer::Consumer() {

}

Consumer::Consumer(Account \*account) {

**this**->account = account;

    total\_price = 0;

}

Consumer::~Consumer() {

}

string Consumer::showCart(**const** vector<Goods \*> &goods) {

    string ret = "";

**if**(cart.size()==0) {

        ret = "当前购物车中没有商品\n";

**return** ret;

    }

    ret += "当前购物车中有如下商品：\n";

**for**(**auto** item : cart) {

**if**(item.second==0) **continue**;

        ret += "商品编号："+to\_string(item.first)+"\n";

        goods[item.first-1]->show\_good();

        ret += "购物车中数量："+to\_string(item.second)+"\n";

        ret += "共计价格："+to\_string(item.second\*goods[item.first-1]->get\_price())+"\n";

**if**(selected[item.first]) ret += "是否选中：已选中\n";

**else** ret += "是否选中：未选中\n";

    }

    ret += "选中商品价格共计："+to\_string(total\_price)+"\n";

**return** ret;

}

string Consumer::showOrder(**const** vector<Goods \*> &goods) {

    string tmp = "";

**if**(orders.size()==0) {

        tmp = "当前账户中没有订单";

    }

**else** {

        tmp = "当前账户中有如下订单：\n";

        //商品编号从1开始，和下标有1的偏差

**for** (**int** i = 0; i<orders.size(); i++) {

            tmp += "订单编号："+std::to\_string(i+1)+"\n";

            tmp += orders[i].showOrder(goods);

        }

    }

**return** tmp;

}

**void** Consumer::addCart(**int** no, **int** amount) {

    cart[no] += amount;

}

**void** Consumer::rmCart(**int** no, **int** amount) {

    cart[no] -= amount;

}

**void** Consumer::changeCart(**int** no, **int** amount) {

    cart[no] = amount;

}

**void** Consumer::select(**int** no, vector<Goods\*> &goods) {

    selected[no] = cart[no];

    total\_price += cart[no]\*goods[no-1]->get\_price();

}

**void** Consumer::unselect(**int** no, vector<Goods\*> &goods) {

    selected[no] = 0;

    total\_price -= cart[no]\*goods[no-1]->get\_price();

}

//生成订单

**void** Consumer::generateOrder(vector<Goods\*> &goods) {

    Order temp(goods,selected);

    orders.push\_back(temp);

    //生成订单后购物车选中部分清空

**for**(**auto** &item : selected) {

        cart[item.first] -= item.second;

        item.second = 0;

    }

    total\_price = 0;

}

**void** Consumer::payOrder(**int** no, vector<Goods\*> &goods) {

**this**->account->transOut(orders[no-1].get\_price());

    orders[no-1].set\_state(1);

    orders[no-1].transfer(goods);

    orders[no-1].release(goods);

}

//取消订单，释放冷冻量

**void** Consumer::cancelOrder(**int** no, vector<Goods\*> &goods) {

    orders[no-1].release(goods);

    orders.erase(orders.begin()+no-1);

}

1. merchant.hpp

#ifndef merchant\_hpp

#define merchant\_hpp

#include <stdio.h>

#include "user.hpp"

**class** User;

**class** Merchant : **public** User {

**public**:

    Merchant();

    Merchant(Account \*account);

    ~Merchant();

    string getUserType(){**return** "merchant";}

};

#endif /\* merchant\_hpp \*/

1. merchant.cpp

#include "merchant.hpp"

Merchant::Merchant() {

}

Merchant::Merchant(Account \*account) {

**this**->account = account;

}

Merchant::~Merchant() {

}

1. goods.hpp

#ifndef goods\_hpp

#define goods\_hpp

#include <stdio.h>

#include <iostream>

#include <string>

#include <vector>

#include "merchant.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Merchant;

**class** Goods {

**public**:

    Goods();

    Goods(string name, string description, **double** price, **int** remaining, string type, Merchant \*mer);

**virtual** ~Goods();

    string get\_name(){**return** name;}

    string get\_description(){**return** description;}

**virtual** **double** get\_price() = 0;

**int** get\_remaining(){**return** remaining;}

    string get\_type(){**return** type;}

    Merchant\* get\_merchant() {**return** merchant;}

**void** set\_description(string description){

**this**->description = description;

    }

**void** set\_price(**double** price){**this**->price = price;}

**void** set\_remaining(**int** remain){remaining = remain;}

**void** set\_name(string name){**this**->name = name;}

**void** set\_type(string type){**this**->type = type;}

**void** set\_merchant(Merchant \*mer) {merchant = mer;}

    string show\_good();

**protected**:

    string name;

    string description;

**double** price;

**int** remaining;

    string type;

    //所属商家(只会把对应的id存入文件）

    Merchant\* merchant;

};

#endif /\* goods\_hpp \*/

1. goods.cpp

#include "goods.hpp"

Goods::Goods() {

}

Goods::Goods(string name,string description, **double** price, **int** remaining,string type, Merchant \*mer):

    name(name),description(description),price(price),remaining(remaining),type(type),merchant(mer)

{

}

Goods::~Goods() {

}

string Goods::show\_good() {

    string ret;

    ret += "名称："+name+"\n";

    ret += "描述："+description+"\n";

    ret += "价格："+std::to\_string(price)+"\n";

    ret += "剩余量："+std::to\_string(remaining)+"\n";

    ret += "种类："+type+"\n";

    ret += "所属商家id："+merchant->get\_account()->get\_id()+"\n";

**return** ret;

}

1. book.hpp

#ifndef book\_hpp

#define book\_hpp

#include <stdio.h>

#include "goods.hpp"

#include "merchant.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Goods;

**class** Merchant;

**class** Book : **public** Goods {

**public**:

    Book();

    Book(string name,string description,**double** price,**int** remaining, string type,Merchant \*mer);

    ~Book();

**double** get\_price(){**return** price;}

};

#endif /\* book\_hpp \*/

1. book.cpp

#include "goods.hpp"

#include "book.hpp"

Book::Book() {

}

Book::Book(string name,string description,**double** price,**int** remaining,string type, Merchant\* mer){

**this**->name = name;

**this**->description = description;

**this**->price = price;

**this**->remaining = remaining;

**this**->type = type;

**this**->merchant = mer;

}

Book::~Book() {

}

1. food.hpp

#ifndef food\_hpp

#define food\_hpp

#include <stdio.h>

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Goods;

**class** Merchant;

**class** Food : **public** Goods {

**public**:

    Food();

    Food(string name, string description,**double** price,**int** remaining,string type, Merchant \*mer);

    ~Food();

**double** get\_price() {**return** price;}

};

#endif /\* food\_hpp \*/

1. food.cpp

#include "goods.hpp"

#include "food.hpp"

Food::Food() {

}

Food::Food(string name,string description,**double** price,**int** remaining,string type, Merchant \*mer){

**this**->name = name;

**this**->description = description;

**this**->price = price;

**this**->remaining = remaining;

**this**->type = type;

**this**->merchant = mer;

}

Food::~Food() {

}

1. clothing.hpp

#ifndef clothing\_hpp

#define clothing\_hpp

#include <stdio.h>

#include "merchant.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**class** Goods;

**class** Merchant;

**class** Clothing : **public** Goods {

**public**:

    Clothing();

    Clothing(string name, string description,**double** price,**int** remaining,string type, Merchant \*mer);

    ~Clothing();

**double** get\_price(){**return** price;}

};

#endif /\* clothing\_hpp \*/

1. clothing.cpp

#include <string>

#include "goods.hpp"

#include "clothing.hpp"

Clothing::Clothing() {

}

Clothing::Clothing(string name,string description,**double** price,**int** remaining,string type, Merchant\* mer){

**this**->name = name;

**this**->description = description;

**this**->price = price;

**this**->remaining = remaining;

**this**->type = type;

**this**->merchant = mer;

}

Clothing::~Clothing() {

}

1. order.hpp

#ifndef order\_hpp

#define order\_hpp

#include <stdio.h>

#include <map>

#include "goods.hpp"

**using** std::cout;

**using** std::cin;

**using** std::endl;

**using** std::string;

**using** std::map;

**using** std::to\_string;

**class** Goods;

**class** Order {

**public**:

    Order();

    Order(vector<Goods\*> &goods, **const** map<**int**,**int**> &selected);

    map<**int**,**int**> get\_goods() {**return** good\_list;}

**double** get\_price() {**return** total\_price;}

**int** get\_state() {**return** state;}

**void** set\_state(**int** x) {state = x;}

**void** release(vector<Goods\*> &goods);

    string showOrder(**const** vector<Goods\*> &goods);

**void** transfer(vector<Goods\*> &goods);    //把用户扣除的钱转移到商家账户中

**private**:

    //first是商品编号，second是商品数量

    map<**int**,**int**> good\_list;

**double** total\_price;

    //订单状态（0待支付/1已支付）

**int** state;

};

#endif /\* order\_hpp \*/

1. order.cpp

#include "order.hpp"

#include "goods.hpp"

Order::Order() {

    total\_price = 0;

}

Order::Order(vector<Goods\*> &goods, **const** map<**int**,**int**> &selected) {

**for**(**auto** item : selected) {

        good\_list[item.first] = item.second;

        total\_price += goods[item.first-1]->get\_price()\*item.second;

        state = 0;

        //冷冻商品

        goods[item.first-1]->set\_remaining(goods[item.first-1]->get\_remaining()-item.second);

    }

}

//释放冻结的商品余量

**void** Order::release(vector<Goods\*> &goods) {

**for**(**auto** item : good\_list) {

        goods[item.first-1]->set\_remaining(goods[item.first-1]->get\_remaining()+item.second);

    }

}

string Order::showOrder(**const** vector<Goods\*> &goods) {

    string ret = "";

**for**(**auto** item : good\_list) {

        ret += "商品名称："+goods[item.first-1]->get\_name()+"，商品单价："+to\_string(goods[item.first-1]->get\_price())+"，商品件数："+to\_string(item.second)+"\n";

    }

    ret += "订单总金额："+to\_string(total\_price)+"\n";

**if**(state==0) ret += "订单状态：待支付\n";

**else** ret += "订单状态：已支付\n";

**return** ret;

}

**void** Order::transfer(vector<Goods\*> &goods) {

**for**(**auto** item : good\_list) {

**double** money = goods[item.first-1]->get\_price()\*item.second;

        //商品盈利转入商家账户

        goods[item.first-1]->get\_merchant()->get\_account()->transIn(money);

    }

}