

实验 2 商品交易信息系统

一、实验目的

通过此次实验，应达到如下要求：

1. 熟练使用一种 C++ 开发环境，包括 IDE 与编译器；掌握 C++ 程序的编写过程与调试；
2. 加深对 C\C++ 基本理论的理解，编程时不再纠结于 C++ 的语法；克服编程时的畏难情绪；
3. 熟悉貌似复杂系统从无到有的开发过程，具备初步系统分析、功能设计的能力。

二、实验任务

《商品交易信息系统：MyMarket》的面向对象程序设计与开发。

三、系统需求

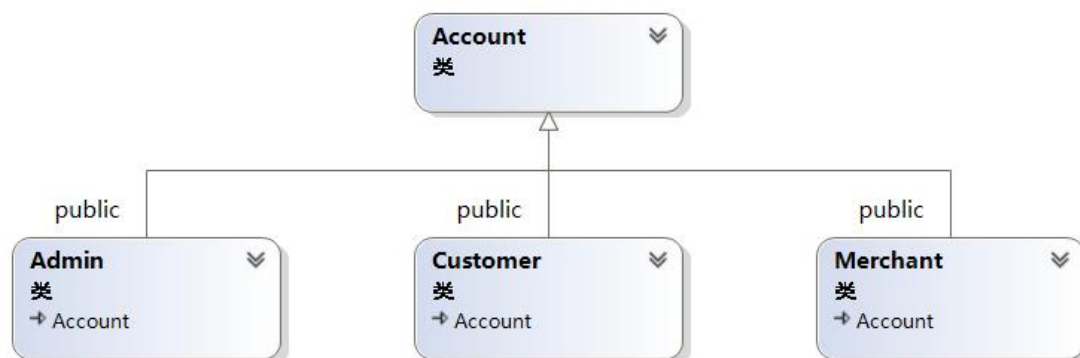
某平台要开发一套控制台程序下的内部商品交易系统，需求描述如下：

（一）管理信息内容

1. 账户信息（账户编号、登录名、密码、角色类型、账户金额）。
2. 买家基本信息（账户编号、姓名、性别、出生日期、地址、电话号码）；
3. 商家基本信息（账户编号、商户名、电话号码、交易总金额、评价次数，买家评分）；
4. 商品基本信息（商品编号、（商家）账户编号、商品名称、单价、库存数量、买家评分、评价次数、累计销量）；
5. 订单信息（订单编号、买家账户编号、商家账户编号、商品编号、商品名称（商品快照）、订单数量、订单日期、订单价格（商品快照）、收货姓名、收货地址、收货电话、订单状态）

上面几类信息中，账户角色类型包括管理员（ADMIN）、商家（MERCHANT）、买家（CUSTOMER）3 种，而订单状态包括（SUCCESS）、要求退款（REFUND_REQUIRED）、已退款（REFUNDED）、拒绝退款

(REFUND_REJECTED) 等 4 类。



(二) 系统功能

1. **账户登录模块。**用户输入用户名和密码，登录成功后，系统将根据登录用户角色类型显示相应操作菜单，其中，操作菜单共分为买家菜单、商家菜单、管理员菜单，分别实现下面功能 2-4。
2. **买家功能模块有：**
 - (1) 我要购物。我要购物子菜单包括：
 - ①搜索商品。输入商品名关键词，查询商品，显示商品编号、商品名称、单价、数量、评价、商家编号等信息，其中，查询得到的商品按商品价格排序。
 - ②查询商家。输入商家编号，若商家编号存在，则显示账户编号、商户名、电话号码、交易总金额，评价次数，买家评分。
 - ③查看商家商品。输入商家编号，若商家编号存在，显示商品编号、商品名称、单价、数量、评价、商家编号等信息，其中，查询得到的商品按商品销量排序。
 - ④购买。输入商品编号和购买数量，检查商品编号是否存在，以及商家库存和买家余额是否充足，若满足条件则提示“是否确认购买”，经输入密码验证后，确认后生成订单，通过更改账户信息表中相应账户的金额，实现买家对商家的付款同时，在订单信息表中创建订单记录。
 - ⑤返回上级菜单。
 - (2) 我的订单：我的订单子菜单包括：

①查看历史订单。显示订单编号、商品名称、订单数量、订单日期、订单价格、收货姓名、收货地址、收货电话、订单状态等信息。

②评价商品。提交订单号，若订单号存在，则提示用户给商品评分，用户输入评分（0~100），将买家评分和评价次数更新至商品信息表和商家信息表。

③提交退货申请。买家输入订单编号，检查历史订单是否在 1 个月之内，若满足条件，将订单状态修改为“退货申请”。

④返回上一级菜单。

（3）我的账户。我的账户子菜单包括：

①浏览个人信息：显示个人姓名、性别、出生日期、地址、电话号码。

②修改个人信息：依次输入修改后的个人姓名、性别、出生日期、地址、电话号码，经提示确认后，修改至买家信息表，其中，出生日期需要进行合法性检查。

③修改密码；

④查看账户金额；

⑤返回上一级菜单。

（4）返回上一级菜单。

3. 商家用户功能模块有：

（1）商品信息管理。子菜单包括：

①浏览所有商品。显示商家在售的商品编号、商品名称、单价、库存数量、买家评分、评价次数、累计销量，按累计销量排序；

②录入新商品。依次输入商品名称、单价、库存数量，并提示是否继续录入。

③删除商品。输入商品编号，检查编号是否存在，若存在，提示是否确认删除商品，并提示是否继续删除。

④修改商品。输入商品编号，检查编号是否存在，若存在，依次输入修改商品的单价、库存数量，并提示是否继续修改。

⑤返回上一级子菜单。

(2) 订单管理。订单管理子菜单包括：

- ①查看所有订单；
- ②查看本年度订单；
- ③查看本月度账单；
- ④退款管理：输入订单编号，检查订单是否存在，若存在，则提示是否同意退款，经密码验证后，确认同意退款，通过修改账户信息表中的账户金额，退款给买家；通过修改商户信息中的交易总金额、商品信息中库存数量、累计销量等，恢复原始信息；通过修改订单状态，将“订单状态”修改为退款成功（若不同意退款，则将订单状态修改为拒绝退款）；
- ⑤返回上级菜单。

注：上述菜单功能中，对各订单记录，显示订单编号、商品名称、订单数量、订单日期、订单价格、收货姓名、收货地址、收货电话、订单状态等信息。

(3) 账户信息。账户信息子菜单包括：

- ①浏览商家信息：包括账户编号、商户名、电话号码、交易总金额、评价次数，买家评分等；
- ②修改商家信息：修改商户名、电话号码信息；
- ③修改密码；
- ④查看账户金额；
- ⑤返回上一级子菜单。

(4) 返回上一级菜单。

4. 系统管理员用户功能模块：

(1) 账户信息管理。功能子菜单包括：

- ①买家注册。填写登录名、姓名、性别、出生日期、地址、电话号码，其中，登录名只能输入数字、字母和下划线且不能与已有登录名重复，出生日期要进行合法性检查，同时随机生成初始密码，通过在账户信息表创建账户，在买家信息表新增买家信息完成买家注册。

②商户注册。填写登录名、商户名、电话号码，其中登录名只能输入数字、字母和下划线且不能与已有登录名重复，同时随机生成初始密码，通过在账户信息表创建账户，在商户信息表新增买家信息完成买家注册。

③查看买家信息。显示买家账户编号、姓名、性别、出生日期、地址、电话号码；

④查看商家信息。显示商家账户编号、商户名、电话号码、交易总金额，评价次数，买家评分；

⑤注销账户。输入登录名，查找账户信息，进行余额检查，若余额非 0，则提示需要提现再注销账户，若余额为 0，经过确认提示，输入用户密码和管理员密码确认后，通过在买家信息表/商户信息表和账户信息表删除相关信息，注销账户。

⑥返回上一级子菜单。

(2) 充值提现管理。

①充值管理。填写登录名，填写充值金额，经管理员密码确认后，通过修改账户信息表，实现充值。

②提现管理。填写登录名，填写充值金额，经管理员密码确认后，通过修改账户信息表，实现提现。

③返回上一级子菜单；

(3) 返回上一级菜单。

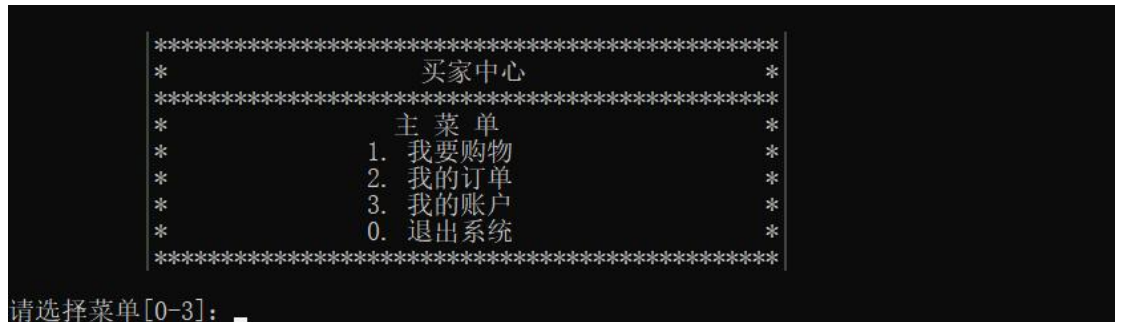
(三) 部分功能菜单界面（示例）

1. 登录界面

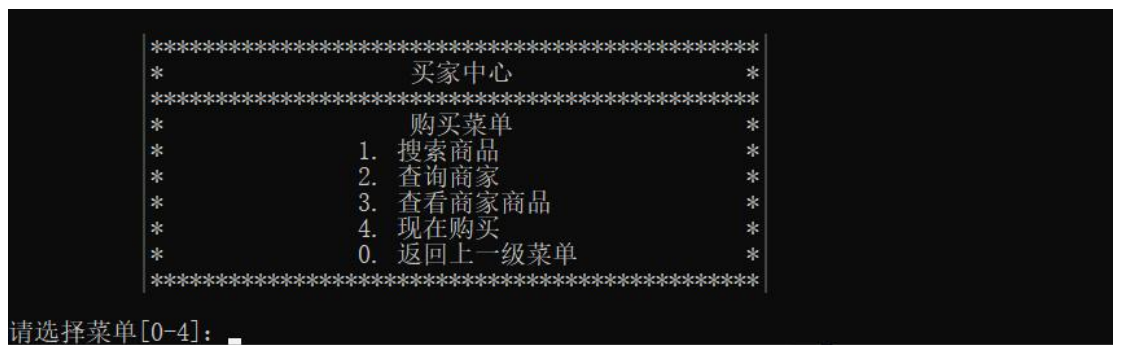


2. 买家功能

2.0. 主菜单



2.1. 购物菜单



2.2.1. 搜索商品菜单

请输入搜索关键词: huawei

商品编号	商品名	价格	库存	评分	评价次数	销量	商家编号
4965	huawei y6	1200	999951	100.00	21	92	85
11668	huawei y6	1200.00	999951	100.00	21	92	85
11669	huawei y6	1200.00	999951	100.00	21	92	85
15473	di jual huawei P30 lite ram 6 rom 128gb	4000.00	999959	99.00	25	63	86
15477	di jual huawei P30 lite ram 6 rom 128gb	4000.00	999959	99.00	25	63	86
15480	di jual huawei P30 lite ram 6 rom 128gb	4000.00	999959	99.00	25	63	86
15482	di jual huawei P30 lite ram 6 rom 128gb	4000.00	999959	99.00	25	63	86
15486	di jual huawei P30 lite ram 6 rom 128gb	4000.00	999959	99.00	25	63	86
6484	Huawei mate 30 pro 8 GB 256 GB segel new BNIB hu awei 2th	10100.00	4	100.00	4	6	84

2.2.2. 查询商家

账户编号	商户名	电话	销售额	评价次数	评分
86	Administrasi Jakarta Selatan	80076149683	281720341.30	38786	98.00

2.2.3. 查看商家商品

商品编号	商品名	价格	库存	评分	评价次数	销量	商家编号
16310	Xiaomi Redmi 6a Tam - RA NDOM	1820	6	97.00	302	3030	86
16392	Xiaomi Redmi 6a Tam - RA NDOM	1820.00	6	97.00	302	3030	86
16456	Xiaomi Redmi 6a Tam - RA NDOM	1820.00	6	97.00	301	3030	86
26009	ASUS ZENFONE MAX PRO M1 3/32GB ZB602KL - Hitam	1585.00	14	96.00	664	2107	86
26001	ASUS ZENFONE MAX PRO M1 3/32GB ZB602KL - Hitam	1615.00	46	96.00	626	2008	86
26002	ASUS ZENFONE MAX PRO M1 3/32GB ZB602KL - Hitam	1599.00	46	96.00	625	1991	86
3891	ASUS ZENFONE MAX PRO M1 3/32GB ZB602KL - Hitam	1499.00	189	96.00	607	1905	86
25966	ASUS ZENFONE LIVE L2 ZA5 50KL 2/16 RAM 2GB INTERN AL 16GB - Merah, 2GB 16G B	950.00	1	97.00	239	1679	86

2.2.4. 现在购买

商品编号	商品名	价格	库存	评分	评价次数	销量	商家编号
16310	Xiaomi Redmi 6a Tam - RA NDOM	1820	4	97.00	302	3032	86

请输入要购买的商品数量: 2
 商品共计3640.00元, 请输入密码确认购买: *****

购买成功!
 请按任意键继续. . .

2.2. 我的订单菜单

[illegible]

2.2.1. 查看历史订单

请选择菜单[0-3]: 1

订单号	商品名	数量	时间	价格	姓名	地址	电话	状态
21	Xiaomi Redmi 6a Tam - RANDOM	2	2021-2-12 19:43:53	3640	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	成功
20	Xiaomi Redmi 6a Tam - RANDOM	2	2021-2-12 19:40:11	3640	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	成功
19	Xiaomi Redmi 6a Tam - RANDOM	2	2021-2-12 19:37:45	3640	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	成功
18	Xiaomi Redmi 6a Tam - RANDOM	2	2021-2-12 19:2:40	3640	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	成功
17	茶杯	9000	2021-2-11 12:13:28	9000	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	成功
16	茶杯	20	2021-2-11 12:12:24	20	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	已退款
15	Iphone 5s 16gb G old	1	2021-2-10 1:59:31	2000	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	已退款
14	test2	1	2021-2-8 23:51:22	100	MacDonal d	River Road Bell City	13320202021	已退款

2.2.2. 评价商品

请输入要评价的订单编号: 21

订单号	商品名	数量	时间	价格	姓名	地址	电话	状态
21	Xiaomi Redmi 6a Tam - RANDOM	2	2021-2-12 19:43:53	3640	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	成功

请输入评分: 98
商品评价成功!

2.2.3. 申请退款

请选择菜单[0-3]: 3
请输入要退货的订单编号: 21

订单号	商品名	数量	时间	价格	姓名	地址	电话	状态
21	Xiaomi Redmi 6a Tam - RANDOM	2	2021-2-12 19:43:53	3640	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	成功

是否请求退款, 按Y键确认, 按任意键取消. Y
退款申请成功。

订单号	商品名	数量	时间	价格	姓名	地址	电话	状态
21	Xiaomi Redmi 6a Tam - RANDOM	2	2021-2-12 19:43:53	3640	李森	长沙市岳麓区大学 城科教园A栋9楼	13320202021	要求退款

2.3. 我的账户菜单

* 买家中心 *

* 我的账户菜单 *
* 1. 浏览账户信息 *
* 2. 修改账户信息 *
* 3. 修改密码 *
* 4. 查询账户余额 *
* 0. 返回上一级菜单 *

请选择菜单[0-4]: _

2.3.1. 浏览账户信息

请选择菜单[0-4]: 1

账户编号	姓名	性别	生日	住址	电话
137	李森	男	1992-2-12	长沙市岳麓区大学城科教园A栋9楼	13320202021

2.3.2. 修改账户信息

请选择菜单[0-4]: 2
待修改的账户信息如下:

账户编号	姓名	性别	生日	住址	电话
137	李森	男	1992-2-12	长沙市岳麓区大学城科教园A栋9楼	13320202021

请输入姓名: 李森一
 请输入性别: 【0-男性/1-女性】 0
 请输入出生年: 1992
 请输入出生月: 2
 请输入出生日: 12
 请输入地址: 长沙市岳麓区大学城科教园B栋1楼
 请输入电话: 073188836110

账户编号	姓名	性别	生日	住址	电话
137	李森一	男	1992-2-12	长沙市岳麓区大学城科教园B栋1楼	73188836110

请确认修改为以上账户信息? 【按Y键确认, 按任意键取消】 y
 商家信息修改成功。

3. 商家功能

3.0. 商家功能主菜单

```

*****
*                      商家中心                      *
*****
*                      主 菜 单                      *
*                      1. 商品管理                    *
*                      2. 订单管理                    *
*                      3. 账户信息                    *
*                      0. 退出系统                    *
*****
  
```

请选择菜单[0-3]: _

3.1. 商品管理菜单

请选择菜单[0-4]:

请选择菜单[0-4]: 1

商品编号	商品名	价格	库存	评分	评价次数	销量	商家编号
16431	REDMI 6A TAM	1350.00	10	97.00	26	307	2
7917	Asus Zenfone 5 ZB620KL 4/64 - Midnight Blue	4000.00	999783	100.00	9	286	2
7914	Asus Zenfone 5 ZB620KL 4/64 - Midnight Blue	4000.00	999784	100.00	9	285	2
8378	Asus Zenfone Max Pro M2 4/64	3999.00	999847	100.00	9	253	2
22750	Samsung Galaxy J2 Prime - QA Licensed SamsungSM-G532 G52 - Black	1450.00	999906	100.00	58	176	2
5257	xiaomi mi 9T 4/64 - Black	4500.00	999911	100.00	72	131	2
6583	samsung galaxy a50	4000.00	999986	100.00	38	130	2
20846	Xiaomi 4x .. 3 / 32 .. Full Oprekan Rasa 4 / 64	2000.00	999976	100.00	37	119	2

```
请选择菜单[0-4]: 2
请输入商品名称: Huawei Mate40 Pro
请输入商品单价: 9999
请输入库存数量: 10000
```

商品编号	商品名	价格	库存	评分	评价次数	销量	商家编号
-1	Huawei Mate40 Pro	9999.00	10000	0.00	0	0	2

是否确认输入商品?【按Y键确认, 按N键取消】

Y
商品录入成功!

商品编号	商品名	价格	库存	评分	评价次数	销量	商家编号
50006	Huawei Mate40 Pro	9999.00	10000	0.00	0	0	2

是否继续录入?【按N键退出录入, 按任意键继续录入】

请按任意键继续. . .

3.1.3. 删除商品

3.2.4. 退款管理

请选择菜单[0-4]: 4
请输入要处理退货的订单编号: 22

订单号	商家号	商品名	数量	时间	价格	姓名	地址	电话	状态
22	2	REDMI 6A TAM	1	2021-2-13 11:38:10	1300.00	李森一	长沙市岳麓区大学城科教园B栋1楼	73188836110	要求退款

客户请求退款, 是否同意退款【按Y键同意退款, 按N键拒绝退款】? Y
请输入密码同意退款: *****
退款成功。

订单号	商家号	商品名	数量	时间	价格	姓名	地址	电话	状态
22	2	REDMI 6A TAM	1	2021-2-13 11:38:10	1300.00	李森一	长沙市岳麓区大学城科教园B栋1楼	73188836110	已退款

3.3. 账户信息菜单

```
*****
*                               *
*           商家中心           *
*                               *
*****
*           账户信息菜单       *
*                               *
*           1. 浏览账户信息     *
*           2. 修改账户信息     *
*           3. 修改密码         *
*           4. 查询账户余额     *
*           0. 返回上一级菜单   *
*                               *
*****
```

请选择菜单[0-4]:

3.3.1. 查看账户信息

账户编号	商户名	电话	销售额	评价次数	评分
2	Bandung Telecom Co.Ltd.	80006342108	19059907.00	1392	99.00

3.3.2. 修改账户信息

待修改的账户信息如下:

账户编号	商户名	电话	销售额	评价次数	评分
2	Bandung Telecom Co.Ltd	8006342018	19059907.00	1392	99.00

请输入修改后的商户名: Bandung Eletronics
请输入修改后的电话: 8006342018
请确认修改为以上账户信息? 【按Y键确认, 按任意键取消】Y
商家信息修改成功。

账户编号	商户名	电话	销售额	评价次数	评分
2	Bandung Eletronics	8006342018	19059907.00	1392	99.00

3.3.3. 修改密码

```
请输入旧密码: *****
请输入新密码: *****
请再次确认新密码: *****
是否确认修改密码? 【按Y键确认, 按任意键取消】 Y
密码修改成功。
```

4. 管理员功能

4.0. 管理员功能主菜单

```
*****
*                               *
*           管理员中心         *
*****
*                               *
*           主 菜 单           *
*           1. 账户管理         *
*           2. 充值提现管理     *
*           0. 退出系统         *
*****

请选择菜单[0-2]:
```

4.1. 账户管理菜单

```
*****
*                               *
*           管理员中心         *
*****
*                               *
*           账户管理           *
*           1. 买家注册         *
*           2. 商户注册         *
*           3. 查看买家信息     *
*           4. 查看商户信息     *
*           5. 注销账户         *
*           0. 返回上一级       *
*****

请选择菜单[0-6]:
```

4.1.1. 买家注册

请选择菜单[0-6]: 1
 请输入登录名: zhangsan
 请输入姓名: 张三
 请输入性别: 【0-男性/1-女性】 0
 请输入出生年: 1999
 请输入出生月: 1
 请输入出生日: 1
 请输入地址: 湖南省长沙市中南大学科技园B座4楼
 请输入电话: 073188831001

账户编号	姓名	性别	生日	住址	电话
-1	张三	男	1999-1-1	湖南省长沙市中南大学科技园B座4楼	73188831001

是否确认创建该买家【按Y键盘确认, 按任意键取消】: Y

账户编号	姓名	性别	生日	住址	电话
215	张三	男	1999-1-1	湖南省长沙市中南大学科技园B座4楼	73188831001

账户创建成功!
 账户初始密码为: 315792

4.1.2. 商家注册

请输入登录名: inteltrade
 请输入商户名: 英特尔贸易有限公司
 请输入电话: 80040080221

账户编号	商户名	电话	销售额	评价次数	评分
-1	英特尔贸易有限公司	80040080221	0.00	0	0.00

是否确认创建该商家【按Y键确认, 按任意键取消】: y

账户编号	商户名	电话	销售额	评价次数	评分
217	英特尔贸易有限公司	80040080221	0.00	0	0.00

账户创建成功!
 账户初始密码为: 671698

4.1.3. 查看买家信息

账户编号	姓名	性别	生日	住址	电话
137	李森一	男	1992-2-12	长沙市岳麓区大学城科教园B栋1楼	73188836110
203	TEST MAN	男	1930-2-3	HONG KONG	19292929999
209	马云	男	1940-12-1	杭州市西湖区胜利路	19832040888
215	张三	男	1999-1-1	湖南省长沙市中南大学科技园B座4楼	73188831001

4.1.4. 查看商家信息

账户编号	商户名	电话	销售额	评价次数	评分
1	Badung	80040429154	2250430.00	309	100.00
2	Bandung Eletronics	8006342018	19059907.00	1392	99.00
3	Bandung Barat	80028364393	1329342.00	183	99.00
4	Bangkalan	80093668057	2700086.00	216	99.00
5	Banjarnegara	80091687209	2769500.00	216	100.00
6	Bantul	80009413253	113160215.70	7756	99.00
7	Banyuasin	80084835517	20250.00	2	100.00
8	Banyumas	80092586937	20744931.06	1921	99.00
9	Banyuwangi	80048725448	4458580.00	369	99.00
10	Batang	80041118963	7307487.10	782	98.00
11	Batanghari	80078129166	144000.00	23	100.00
12	Bekasi	80067595509	101186625.50	13103	98.00
13	Bengkalis	80054677805	112480.00	18	95.00

4.1.5. 注销账户

```

请选择菜单[0-6]: 5
请输入要注销的账户名: inteltrade
请输入该账户密码确认删除账户!
*****
账户删除成功!

```

4.2. 充值提现管理菜单

```

*****
*                               *
*          管理员中心          *
*                               *
*          充值提现管理        *
*          1. 充值管理          *
*          2. 提现管理          *
*          0. 返回上一级子菜单 *
*****
请选择菜单[0-2]: 

```

4.2.1. 充值管理

```

请选择菜单[0-2]: 1
请输入要充值的账户名: TESTCUS
请输入要充值的金额: 1
请输入管理员密码: *****
请输入充值账户密码: *****

充值成功!
充值金额为1.00元。
当前账户余额为20001.00元。

```

4.2.2. 提现管理

```
请输入要提现的账户名: TESTCUS
请输入管理员密码: *****
请输入提现账户密码: *****
请输入要提现的金额: 101
```

```
提现成功!
提现金额为101.00元。
当前账户余额为19900.00元。
请按任意键继续
```

（四）系统操作菜单程序代码

见 src\initial_project 目录附带程序 MyMarket v0.1。

四、实验要求

（一）基本要求

本实验采用教师指导，学生自学，单人编程，采用面向对象程序设计思想实现《商品交易信息系统》。

1. 单人设计和调试程序。
2. 合理安排工作计划。
3. 数据管理可用文本文件或二进制文件的形式保存。
4. 程序要求运行正确无错误，界面美观、可操作性强、稳定性好。

（二）考核方式及评分标准

1. 程序演示（小计 60%）

程序的正确性：40%

程序的可读性：10%

界面的友好性：10%

2. 实验报告：30%
3. 工作态度：10%

五、实验指导

（一）数据

针对系统需求给出的管理信息内容，我们将5类管理信息分别存储在5个csv数据表之中（放在MyMarket工程下的data目录下，如图X所示）。每个数据表中的每行代表信息的一条记录，每列代表记录的属性。以商品信息表为例，每一行数据代表一件商品，这行数据的每一列，分别存储该商品的账户编号、商户名、

电话号码、交易总金额、评价次数，买家评分等属性。在csv数据表格式中，各属性以逗号分隔。我们可以用文本编辑工具打开商品信息表（product.csv），观察订单信息表的格式，而其他信息表格式显然与此类似。

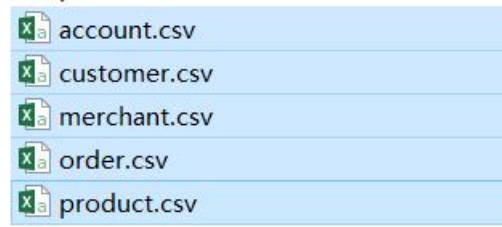


图1 系统数据表

（二）数据结构

针对上述5个信息表表示的数据，我们可以定义主要5种数据结构。我们的数据结构可以采用类或结构体为主的自定义数据类型实现。此处，我们采用结构体为例，定义对应5类信息的数据结构，采用类的方式本质上与此大同小异。对于5类数据结构，以商品信息为例，可以定义为：

```
class Product
{
    friend class DataLayer;
    friend class Account;
    friend class Customer;
    friend class Merchant;
public:
    Product() {}
    Product(int ID, int MerchantID, char na[], double p, int s):
        ProductID(ID), AccountID(MerchantID), price(p), stock(s)
    {
        strcpy(name, na);
    }
    // 将商品结构体变量写为文件的一行
    void PutProductRecord(ofstream& ofs)
    {
        ofs << ProductID << ',';
        ofs << AccountID << ',';
        ofs << name << ',';
        ofs << price << ',';
```

```

        ofs << stock << ',';
        ofs << rating << ',';
        ofs << nRatings << ',';
        ofs << nSales << endl;
    }

    static void PrintProducts(Product products[], int nProduct);
    void setMerchant(Merchant* pm) { pMerchant = pm; }
    void UpdateRating(int rating);
    static void InputProductInfo(char* name, double& price, int& stock);
    void UpdateProductInfo(char* na, double p, int s)
    { strcpy(name, na); price = p; stock = s; }

private:
    int ProductID;
    int AccountID;
    Merchant* pMerchant;
    char name[ProdNameBuffSize];
    double price;
    int stock;
    float rating;
    int nRatings;
    int nSales;
};

```

上面定义的结构体数据类型还是只能存储商品信息表中的1行数据。要存储信息表中的所有记录，我们可以采用数组方式实现。然而，各个数据表中记录的个数不仅是在程序编译之前难以预知的，而且在程序运行过程中是可变的。因此我们需要采用可变大小的结构体“数组”实现数据表中所有记录的存储。这里，我们推介采用C++标准库中的vector类实现动态数组功能。vector处于C++标准库的std命名空间之中，定义为容器模板类，即可支持任意数据类型，建立相关的动态数组。同学们可以参考cplusplus.com上面的官方文档，熟悉vector类的使用，其中，我们可以重点关注关于数组大小动态变化的相关函数，如在数组尾部新增元素、删除数组元素和获得数组动态大小等操作是如何实现的，具体文档可参考（重点熟悉文档给出的实例代码）：

vector类介绍- <http://cplusplus.com/reference/vector/vector/>

vector的初始化-

<http://cplusplus.com/reference/vector/vector/vector/>

在vector尾部新增元素（vector::push_back成员函数）-

http://cplusplus.com/reference/vector/vector/push_back/

删除vector元素（`vector::erase`） -

<http://cplusplus.com/reference/vector/vector/erase/>

获得vector动态大小（`vector::size`） -

<http://cplusplus.com/reference/vector/vector/size/>

除此之外，`vector`各元素值的操作完全可以按我们习惯的数组元素的操作方式进行（也就是可以将其忽略为一般数组了）。结合已知的系统需求，我们在熟悉上述操作后，足以实现系统功能了。

最终我们可以将信息表中的所有商品表示为`vector<Product>`形式。为了令我们的代码更为紧凑，我们可以通过类型重定义的方式定义一个`Product`数组类型：

```
typedef std::vector<Product> ProductArray;
```

其他如买家、商家、订单、账户等信息表内容，也完全可以按上述方式定义相关结构体，以及结构体数组，实现信息的存储。

我们可以把这些数据结构全部定义在1个头文件之中，如命名为`MyMarket.h`，按需通过包含头文件`MyMarket.h`调用这些数据结构。

（三）程序架构

有了数据和数据结构，我们可以开始着手实现系统需求中的功能。要实现系统需求功能，如果将所有代码都编写在同一个文件里，可能导致程序结构的混乱。因此，需要尽可能遵循模块化的设计理念，将功能特性相同的函数放在同一个文件里，实现高内聚低耦合的程序设计模式。

高内聚低耦合，是软件工程中的概念，是判断软件设计好坏的标准，高内聚低耦合将使程序模块的可重用性、移植性大大增强。内聚是从功能角度来度量模块内的联系，一个好的内聚模块应当恰好做一件事，它描述的是模块内的功能联系；耦合是软件结构中各模块之间相互连接的一种度量，耦合强弱取决于模块间接口的复杂程度、进入或访问一个模块的点以及通过接口的数据。

对此，我们可以采用典型三层架构，实现程序模块化架构的设计。三层架构(3-tier application) 通常意义上的三层架构就是将整个业务应用划分为：表现层

(UI)、业务逻辑层(BLL)、数据访问层(DAL)。区分层次的目的即为了“高内聚，低耦合”的思想。

1. 表现层 (UI): 通俗讲就是展现给用户的界面，即用户在使用一个系统的时候他的所见所得。(负责展示而已)
2. 业务逻辑层 (BLL): 针对具体问题的操作，也可以说是对数据层的操作，对数据业务逻辑处理。(关键在于由原始数据抽象出逻辑数据) 能够提供 interface\API 层次上所有的功能。“中间业务层”的实际目的是将“数据访问层”的最基础的存储逻辑组合起来，形成一种业务规则
3. 数据访问层 (DAL): 该层所做事务直接操作数据库，针对数据的增添、删除、修改、查找等。(关键在于粒度的把握) 要保证“数据访问层”的中的函数功能的原子性！即最小性和不可再分。“数据访问层”只管负责存储或读取数据就可以了。

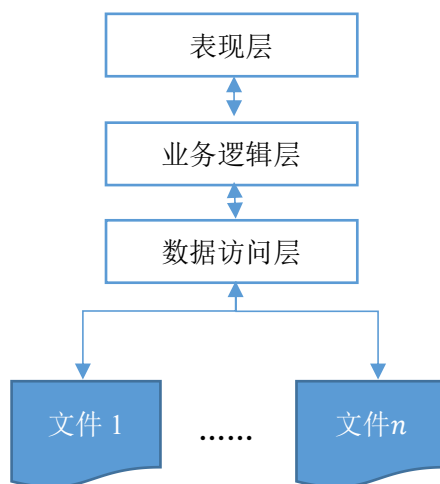


图2 程序的三层架构

此处需要指出的是，在软件工程中的经典三层架构之中，数据访问层一般是实现用于数据库的访问，我们由于背景知识限制，这里数据访问是通过读写文件的方式实现的，因此，图X的数据访问层实现的则是文件的读写（增、删、查、改）功能。未来，有余力的同学可以思考如何通过数据库这种更为高效、稳定、便捷的方式实现数据的存储与访问。

言归正传，我们现在开始实现三层结构。此处，数据访问层和表示层相对单

纯，下面我们不妨先实现它们。

（四）数据访问层

要实现数据访问层，我们需要重点实现数据的读取和写入。

为实现csv数据表的读取功能。因为数据表是以ASCII文件形式存储的，表的每一行代表一条记录，我们可以通过getline的方式按行读取数据表内容，以获得各条记录对应的字符串。对每一行数据的字符串，因为各个属性以逗号分隔，对此，我们需要将其分隔并转换为相应数据类型，得到该行记录所对应的结构体变量，放入相关类型的vector数组之中。按照上述方式，当处理完所有行后，我们将得到一个存储有csv数据表内容的vector数组。

反过来，给定这样一个vector数组，根据我们已有的C++知识，我们不难将它重写为csv数据表格式的文件。

对数据的读写类代码参考如下：

```
class DataLayer
{
private:
    CustomerArray CustomerList;
    MerchantArray MerchantList;
    ProductArray ProductList;
    OrderArray OrderList;
    AccountArray AccountList;
    Account* pAccount;
public:
    DataLayer(): pAccount(nullptr)
    {
        LoadAccountList();
        LoadCustomerList();
        LoadMerchantList();
        LoadProductList();
        LoadOrderList();
    }
    ~DataLayer()
    {
        if (pAccount != nullptr) delete pAccount;
        SaveOrderList(OrderList);
        SaveProductList(ProductList);
        SaveCustomerList(CustomerList);
        SaveMerchantList(MerchantList);
        SaveAccountList(AccountList);
    }
}
```

```

}
/** 登录提示 */
bool loginPrompt();
CustomerArray* getCustomerList() { return &CustomerList; }
MerchantArray* getMerchantList() { return &MerchantList; }
ProductArray* getProductList() { return &ProductList; }
OrderArray* getOrderList() { return &OrderList; }
AccountArray* getAccountList() { return &AccountList; }

bool AddOrder(const Order& order);
int AddOrder(int CustomerID, int MerchantID, int ProductID,
    const char* ProductName, int number, double price,
    const char* CustomerName, const char* address, UINT64 phone);
int CheckLoginName(const char* LoginName);

Account* GetAccount(int ID) const;
Customer* GetCustomer(int ID) const;
Merchant* GetMerchant(int ID) const;
Product* GetProduct(int ID) const;
Order* GetOrder(int ID) const;
private:
bool LoadCustomerList();
bool SaveCustomerList(const CustomerArray &CustomerList);
bool GetCustomerAddress(int ID, char *name, char *address, UINT64 &phone);

bool LoadMerchantList();
bool SaveMerchantList(const MerchantArray &MerchantList);
bool UpdateMerchantSales(int ID, double SaleChange);

bool LoadProductList();
bool SaveProductList(ProductArray &ProductList);

bool LoadOrderList();
bool SaveOrderList(const OrderArray &OrderList);

bool LoadAccountList();
bool SaveAccountList(const AccountArray &AccountList);

bool UpdateAccountMoney(int AccountID, double change);
bool GetAccountMoney(int AccountID, double &money);
bool GetAccountType(int AccountID, UserType &type);
bool CheckPassword(int AccountID, const char *Password);
bool ResetPassword(int AccountID, const char *OldPassword,
    const char *NewPassword);

```

```
int CheckLoginInfo(const char* LoginName, const char* password);
};
```

在后续的业务逻辑层中，我们就可以（通过包含头文件Datalayer.h）调用相关函数接口实现对商品信息表的访问与操作。按照上述方式，我们不难定义完成买家、商家、订单、账户等一系列信息的读写和访问。例如，我们可以完成一系列数据访问层的函数，如上面头文件DataLayer.h所示，供业务逻辑层调用。

（五）表示层

在完成了数据访问层后，我们跨过稍微复杂的业务逻辑层，先实现表示层。对此，我们试图通过将数据打印成表格的形式实现数据在控制台下的显示。要实现商品信息的打印，我们可以将表格打印分解为表头打印和内容打印两部分。

```
class UI
{
public:
    static void GetPass(char password[]);
    static int GetNumber();
    static double GetFloatValue();
    static double GetCashValue();
    static char GetKey();
    static void pause();
};
```

对于表头打印和内容打印，其每一列的宽度是固定的，因此我们可以定义类，控制表格的打印。

```
struct FieldPrintSetting
{
    char PrintFieldName[16]; // 需要打印的字段名
    int PrintWidth; // 打印的字段宽度
};
```

基于上述数据结构，对商品信息表，我们可以设计如下结构体数组作为表头：

```
class Utility
{
public:
    static int ParseString(const char *str, char *word[], const char delim = ',');

    static void StringToDate(const char *str, Date &date);

    static void StringToTime(const char *str, Time &time);
```



```

static UINT64 StringToPhone(const char *str);

static void ReverseOrders(Order orders[], int nOrders);

static void GetCurrentTime(Time &time);

static bool CheckDate(const Date &date);

static char* TimeToString(const Time &time, char *buffer, int BufferSize = 20);

static char* DateToString(const Date &date, char *buffer, int BufferSize = 11);

static void GeneratePassword(char *pwd, int len);

// 根据字段格式，打印表格的行分割线
static void PrintLine(FieldPrintSetting PrintSet[], int nFields);
static void PrintLine(int fieldsWidth, int nFields);
// 根据字段列宽设置，打印截断后剩余的字符串
static void PrintRemaining(FieldPrintSetting PrintSet[], int nFields, string strs[]);
static void PrintTableHead(FieldPrintSetting PrintSet[], int nFields);
////////////////////////////////////
// 实现字符串的截断
// src -源字符串
// dst -截断的目标字符串
// width -截断宽度，超出该宽度则字符串截断
// remaining -截断后剩余的字符串
////////////////////////////////////
static char* TrimString(const char* src, char* dst, int width, string& remaining);
};

```

（六）业务逻辑层

有了数据访问层和表示层，实现系统需求分析的功能开始逐渐变得简单和清晰了。下面我们简要介绍一个在业务逻辑层实现系统功能的例子。

例如要实现商家用户的商品信息管理菜单功能：①浏览所有商品。显示商家在售的商品编号、商品名称、单价、库存数量、买家评分、评价次数、累计销量，按累计销量排序；可以通过编写如下类实现：

```

class Merchant : public Account
{
    friend class DataLayer;
private:
    //int ID;

```

```

    char name[UsrNameBuffSize];
    UINT64 phone;
    double TotalSales;
    int nRatings;
    float rating;
public:
    Merchant() {}
    Merchant(const Account&a):Account(a) {}
    ~Merchant();
    virtual void ShowMenu();
    void PutMerchantRecord(ofstream& ofs)
    {
        ofs << ID << ',';
        ofs << name << ',';
        ofs << phone << ',';
        ofs << setiosflags(ios::fixed) << setprecision(2) << TotalSales << ',';
        ofs << nRatings << ',';
        ofs << rating << endl;
    }
    void UpdateMerchantSales(double SaleChange) { TotalSales += SaleChange; }
    void UpdateRating(int rating);

    int AddProduct(int AccountID, const char* name, double price, int stock);

    static void PrintMerchants(Merchant merchants[], int nMerchants);
    bool MerchantSignIn();
    void RemoveAllProducts();
private:
    bool IsCurrentMerchant(const Order& order);
    bool IsCurrentMerchantThisYear(const Order& order);
    bool IsCurrentMerchantThisMonth(const Order& order);

    bool SelectMerchantMenu();

    void ShowProductManageMenu();
    bool SelectProductManageMenu();

    void ShowOrderManageMerchantMenu();
    bool SelectOrderManagewMerchantMenu();

    void ShowAccountInfoMerchantMenu();
    bool SelectAccountInfoMerchantMenu();

    void BrowseAllProducts();

```

```
void AddNewProduct();

void RemoveProduct();

void ModifiyProduct();

void BrowseAllOrders();

void BrowseOrdersThisYear();

void BrowseOrdersThisMonth();

void refund();

virtual void MyAccountInfo();

virtual void ModifyAccountInfo();
};
```

上面的所有代码只是抛砖引玉，启发同学们的思维，剩下的功能实现，就看你们的了！