武汉科技大学城市学院

课程设计

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

|  |  |  |
| --- | --- | --- |
| 课 | 程 | 面向对象课程设计 |
| 姓 | 名 |  |
| 院 | （系） | 信息工程学部 |

专业年级 计算机科学与技术 18 级 3 班

学 号指导教师

日 期 2019 年 6 月 3 日-2019 年 6 月 7 日

教务处制

**目 录**

1、需求与功能分析 2

2、系统总体框架 3

3、模块设计与分析 5

4、类的设计与分析 8

5、特色算法分析 9

6、功能测试 14

7、存在的不足与对策 15

8、程序源代码 25

9、使用说明 25

10、参考文献 25

**一、** 需求与功能分析

随着社会经济的发展，信息化程度的不断深入，银行的传统业务己愈来愈不能满足银行客户的需要。

现今，人们的金融意识、科技意识己经有了很大的提高，在紧张忙碌的生活中，己越来越来不习惯每月奔忙于各银行营业柜台之问去排队缴各种各样的费用了；同时，各种经营单位如电信、移动、供电、煤气、自来水、证券等等一是为了提高服务质量、方便客户，二是为了减轻自己日趋繁重的工作量，纷纷委托银行为其开展代收代付业务：同时，随着我国加入世贸组织的日益临近，我国的银行业将面临更加激烈的同业竞争，如何提供更多的金融产品和更优质的服务，如何吸引更多的客户，如何利用计算机技术加强银行帐户信息管理、进行银行业务再造，提高银行的工作效率和业务竟争能力是摆在各家银行面前的一个迫切需要解诀的问题。

近几年来，各商业银行加快了与社会各业的合作，利用自身的网点优势和业务特点，为其提供各种高效、快捷的代收代付业务，也就是中间代理业务，目前以与广大人民群众生活密切相关的各项缴费业务如水电费、电话费、手机费等代收业务为主，这些业务开展方式多种多样，但一般都离不开计算机的高效管理支持。

随着代理业务不断发展，业务品种越来越多，各项业务做法互有差异，这就对银行的电子化水平和相应的管理水平提出了更高的要求。如何利用电子化的手段构建一个高效统一的、通用灵活的系统来管理各种各样的业务，是每个商业银行所要研究的课题。支持决策系统的，需要在数据库的基础上，进行联机分析处理，每次处理的数据量大，响应时间长。

特别是银行每天要处理大量的存取款事件，做好存取款是银行工作重要的环节，然而要有效处理必须要有良好的程序和数据管理系统来建立一个良好的软件系统来实现快速、有效、准确、安全的处理银行事物。

主要用于银行的储蓄卡管理系统，它可以帮助我们有效、准确、并且高效实现的完成存取事件。此系统操作方便效率、安全性高，只要客户开户并设定好密码就可以轻松的实现存取款。

系统实现的主要有储蓄卡开户管理、存取款管理、用户查询历史数据显示、

注销等功能。

储蓄卡开户管理模块：卡号、姓名、开户金额、身份证号、地址、电话、密码、确认密码和保存组成。（开户金额必须是数字，密码和确认密码必须一样是六位数字）

\*查询历史数据模块：存款记录、取款记录、余额显示等。

\*注销模块：一旦注销该卡号就不存在且余额提醒你取出。

通过该银行账户管理系统地运行，使办公人员可以轻松快捷的完成对账户管理的任务，提高账目管理效率，使银行的账目管理工作系统化、规范化、自动化。

该银行帐目管理信息系统，优点是设计过程思路清晰、模块划分简洁，设计各阶段分工明确。经过实践证明，该划分是合理的，极大得提高了本系统的实现。

二、 系统总体框架

存

钱页面

取

款页面

余

额页面

历

史记录

注

销页面

注销退出

主界页面

注册开户

登录界面

银行管理系统

图 1 -1

三、 模块设计与分析

储蓄卡开户管理模块：

* 1. 登录页面

输入卡号和密码，根据用户输入的卡号和密码，到后台文件查询，若正确则登录成功，并保存卡好信息已备后用。

若卡号存在和密码不匹配或无该卡号已经注销，提示该卡号已存在或已注销，保持界面三次不变等待用户的重新输入。

* 1. 主界面页面

主要导航栏有：欢迎使用储蓄卡管理（存款、取款、查询余额、查询历史记录、注销退出）

* 1. 储蓄卡开户页面

提供开户界面，由用户输入新储蓄卡的所有信息，将该信息保存到文件中， 并将开户金额作为该卡的第一次存款记录存人文件。

* 1. 存钱页面

从登录界面得到卡号，并提供数据输入界面，等待用户的输入存款金额。从后台数据库中找到该用户余额记录，修改余额，并将该存款事件的相应信息写人数据库，同时将存钱的金额显示在页面上。

* 1. 取款页面

从登录界面得到卡号，并提供数据输入界面，等待用户输入取款金额，如果取款金额大于该用户卡上的余额则不能取款，或者将新的余额写人文件，并将本次取款事件写入文件同时将取钱的金额显示在页面上，。

* 1. 查询余额页面

从登录界面得到卡好，把该卡上的余额显示到界面中。

* 1. 查询历史记录页面

从登录界面得到卡号，把该卡的所有存款取款信息查询出来，并显示在页面上。

* 1. 注销页面

从关闭文件，状态为注销，最后关闭整个系统。列举取款模块进行分析：

void Function::Drawing(User &u) {

system("cls"); cout<<"\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\t\t请输入取款金额："; string str1("\n\t\t现取RMB 钞- "); string str2;

int k=0; while(k<=3) {

cin>>SaveAndDrawMoney; u.Fmoney -=SaveAndDrawMoney; if(u.Fmoney>=0) {

system("cls");

cout<<"\n\n\n\t\t正在出炒，请稍等！"<<endl; strstream ss;

ss<<u.Accout; ss>>str2;

string frist="Record"+str2 +".txt"; const char \*RecordFile=frist.c\_str();

ofstream outfile(RecordFile,ios::out|ios::app); outfile<<str1<<SaveAndDrawMoney<<endl; outfile.close();

outfile.clear(); break;

}

else {

k++;

if(k>=3) { system("cls");

cerr<<"\n\n\n\t\t很抱歉，你重复输入错误多次"<<endl

<<"\n\t\t正在退卡中……"<<endl; exit(0);

}

system("cls");

u.Fmoney +=SaveAndDrawMoney;

cerr<<"\n\n\n\t\t对不起，你输入的金额超出有效金额。"<<endl

<<"\n\t\t 请重新输入: ";

}

}

cout<<"\n\n\t\t操作成功，单击任何键返回主菜单！"; getcher();

}

在取款页面中，输入取款金额并对取款金额与存款余额进行比较，如果三次都超过的话就输出：很抱歉，你重复输入错误多次，在退卡中……，如果取款金额小于存款余额的话就执行，并保存记录。

四、 类的设计与分析

1. User 类

class User 主要制定用户的成员数据，并进行登录与保存功能： 画出 User 类的类图



相关代码：

class User

{

public:

void get(User& ); friend class Function;

ostream &print(ostream &os);

friend istream & operator>>(istream & i, User & u); void PassWord();

bool Login(User &); void SaveFile(User &);

vector<string> StrFunction; User(){}

User(int A,string N,int F,string I,string Ad,int T):Accout(A),Name(N),Fmoney(F),ID(I),Address(Ad),Telephone(T){} private:

int Accout; string Name; int Fmoney; string ID; string Address; int Telephone;

char Pword1[9]; char Pword2[9];

};

表 1 User 类说明表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 名称 |  |  | 属性 |  |  | 说明 |  |
| Accout | | | 数据成员 | | | 用户账户数据 | | |
| Name | | | 数据成员 | | | 用户姓名 | | |
| Fmoney | | | 数据成员 | | | 用户首次金额 | | |
| ID | | | 数据成员 | | | 用户身份证号 | | |
| Address | | | 数据成员 | | | 用户地址 | | |
| Telephone | | | 数据成员 | | | 用户电话 | | |
| Pword1 | | | 数据成员 | | | 用户密码 | | |
| Pword2 | | | 数据成员 | | | 用户再一次确认密码 | | |
| Get() | | | 构造函数 | | | 默认构造函数 | | |
| Login() | | | 成员方法 | | | 登录方法 | | |
| Password() | | | 成员方法 | | | 密码验证 | | |
| SaveFile() | | | 成员方法 | | | 保存数据 | | |

1. Function 类

Function 类 主要实现用户的各种功能： 画出 User 类的类图：

相关代码：

class Function

{

public:

void Saving(User &); void Drawing(User &); void Balance(User &); void Record(User &);

private:

int SaveAndDrawMoney;

};

表 1 User 类说明表

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 名称 |  |  | 属性 |  |  | 说明 |  |
| SaveAndDrawMoney | | | 数据成员 | | | 用户存入与取出金额 | | |
| Saving() | | | 成员方法 | | | 实现存款功能与保存数据 | | |
| Drawing() | | | 成员方法 | | | 实现取款功能与保存数据 | | |
| Balance() | | | 成员方法 | | | 查询余额功能 | | |
| Record() | | | 成员方法 | | | 查询历史记录 | | |

五、 特色算法分析

1.User 类中的 PassWord 方法

功能：输入密码并进行确认，如果密码确认三次失败，则退出系统。流程图：

否

**While(n<3)**

是

输入：password1

Password2

否

strcmp(Pwor

d1,Pword2)

是

代码：

void User::PassWord() { int num=0;

while(num<3) {

cout<<"\n\t\t请输入位储蓄密码："; for(int i=0;i<8;i++) {

Pword1[i]=getch();cout<<"\*";} Pword1[8]='\0';

cout<<endl;

cout<<"\n\t\t请再一次确认密码：";

for(int i=0;i<8;i++)

{ Pword2[i]=getch();cout<<"\*";} cout<<endl;

Pword2[8]='\0'; if(strcmp(Pword1,Pword2)==0){

cout<<"\n\n\t\t密码确认成功，单击任何键进入主菜单"; getch();

return ;

}

else { num++;

cout<<"\n\t\t你输入的密码错误，请重新输入！\n"<<endl;

}

if(num==3){

system("cls");

cerr<<"\n\n\n\t\t对不起，你三次输入不正确\n"<<endl

<<"\t\t感谢你的使用，再见！ "<<endl;

exit(0);

}}

}

六、 功能测试

* 1. 首页：

图 6-1：



* 1. 储蓄卡开户页面

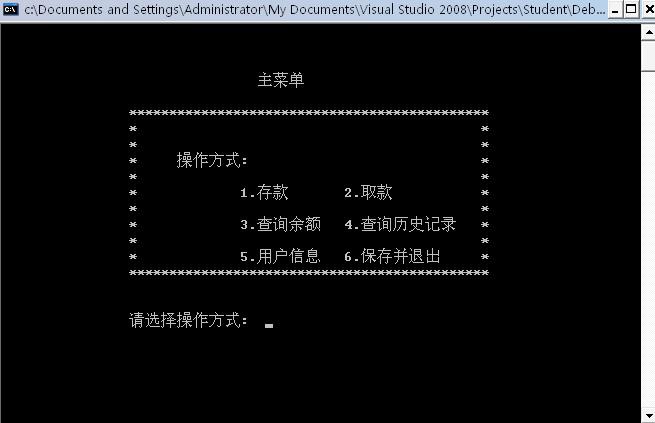
图 6-2：



* 1. 主界面页面

主要导航栏有：欢迎使用储蓄卡管理（存款、取款、查询余额、查询历史记录、注销退出）

图 6-3：



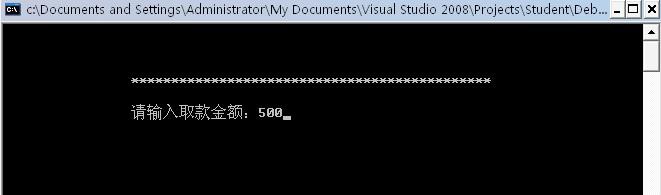
* 1. 存钱页面

图 6-4：

* 1. 取款页面

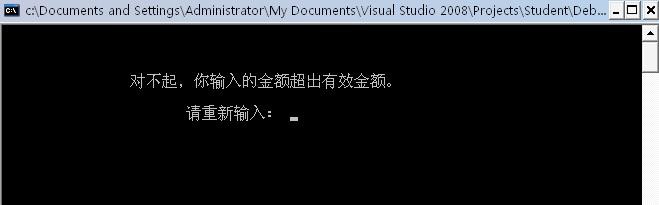
从登录界面得到卡号，并提供数据输入界面，等待用户输入取款金额，如果取款金额大于该用户卡上的余额则不能取款，或者将新的余额写人文件，并将本次取款事件写入文件同时将取钱的金额显示在页面上。

输入取款金额：

图 6-5：

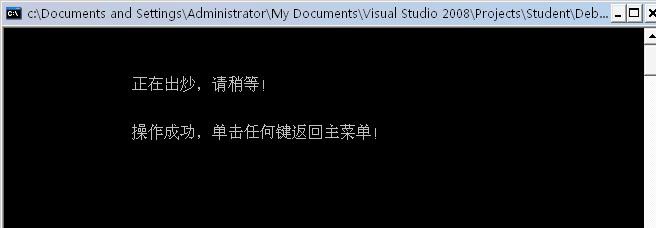
当取款金额大于现有金额：

图 6-6：



当取款金额小于或等于现有金额：

图 6-7：



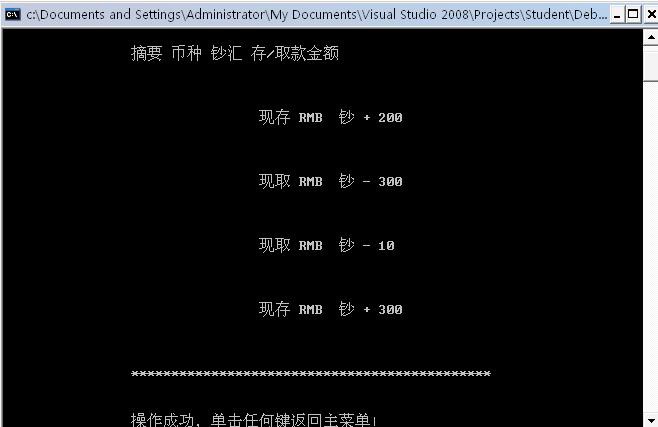
* 1. 查询余额页面

图 6-8：



* 1. 查询历史记录页面

图 6-9：



* 1. 用户信息

图 6-10：



* 1. 注销页面

从保存数据并关闭文件，状态为注销，最后关闭整个系统。

图 6-11：



七、 存在的不足与对策

程序存在的不足之处在于程序以文件方式储存，并且每个用户信息与用户的历史记录分别用一个文件来储存，如果这样的话一个用户就需要二个文件储存信息，那么一万个用户就需要用到二万个文件保存。这不但浪费内存而已在查找文件也会花费时间。

解决的对策是使用 stl 标准库或者数据库的方法进行解决，这里列举了一个用 stl 标准库里面用 vector 解决的例子：

#include <iostream> #include <fstream> #include <string> #include <vector> using namespace std; class Account

{

public:

string account; // 如果你要保存 0001 这样的格式，用 string 方便

double balance; // 余额

//自己补充数据成员

friend ostream & operator<<(ostream & o, const Account & t); friend istream & operator>>(istream & i, Account & t);

};

ostream & operator<<(ostream & o, const Account & t)

{

o<<t.account<<' '<<t.balance/\*自己补充各个数据成员\*/<<endl; return o;

}

istream & operator>>(istream & i, Account & t)

{

i>>t.account>>t.balance/\*自己补充各个数据成员\*/; return i;

}

int main()

{

ifstream ifs("account.txt"); vector<Account> vacc;

vacc.assign(istream\_iterator<Account>(ifs), istream\_iterator<Account>()); ifs.close();

for (vector<Account>::iterator it=vacc.begin(); it!=vacc.end(); ++it)

{

if (it->account == "0002"){

it->balance += 100; //余额增加 100 break;

}

}

ofstream ofs("account.txt");

copy(vacc.begin(), vacc.end(), ostream\_iterator<Account>(ofs));

return 0;

}

通过这个例子，我们就可以从程序中输入像

// account.txt 里面的内容示例

0001 104.7

0002 252.1

0004 4299.04

0003 723.2

八、 程序源代码#include<iostream> #include<fstream> #include<string> #include<vector> #include<algorithm> #include<strstream> #include<conio.h> using namespace std; class Function;

class User

{

public:

void get(User& ); friend class Function;

ostream &print(ostream &os);

//friend ostream & operator<<(ostream & o, const User & u); friend istream & operator>>(istream & i, User & u);

void PassWord(); bool Login(User &);

void SaveFile(User &); vector<string> StrFunction; User(){}

User(int A,string N,int F,string I,string Ad,int T):Accout(A),Name(N),Fmoney(F),ID(I),Address(Ad),Telephone(T){} private:

int Accout; string Name; int Fmoney; string ID; string Address; int Telephone;

char Pword1[9]; char Pword2[9];

};

class Function

{

public:

void Saving(User &); void Drawing(User &); void Balance(User &); void Record(User &);

private:

int SaveAndDrawMoney;

};

void Function::Saving(User &u)

{

system("cls"); cout<<"\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\t\t请输入存款金额："; string str1("\n\t\t现存RMB 钞+ "); cin>>SaveAndDrawMoney; u.Fmoney+=SaveAndDrawMoney;

strstream ss; string str2;

ss<<u.Accout; ss>>str2;

string frist="Record"+str2 +".txt"; const char \*RecordFile=frist.c\_str();

ofstream outfile(RecordFile,ios::out|ios::app); outfile<<str1<<SaveAndDrawMoney<<endl; outfile.close();

outfile.clear();

cout<<"\n\n\t\t操作成功，单击任何键返回主菜单！"; getch();

}

void Function::Drawing(User &u)

{

system("cls"); cout<<"\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\t\t请输入取款金额："; string str1("\n\t\t现取RMB 钞- "); string str2;

int k=0; while(k<=3)

{

cin>>SaveAndDrawMoney; u.Fmoney -=SaveAndDrawMoney; if(u.Fmoney>=0)

{

system("cls");

cout<<"\n\n\n\t\t正在出炒，请稍等！"<<endl; strstream ss;

ss<<u.Accout; ss>>str2;

string frist="Record"+str2 +".txt"; const char \*RecordFile=frist.c\_str();

ofstream outfile(RecordFile,ios::out|ios::app); outfile<<str1<<SaveAndDrawMoney<<endl; outfile.close();

outfile.clear(); break;

}

else

{

k++;

if(k>=3)

{

system("cls");

cerr<<"\n\n\n\t\t很抱歉，你重复输入错误多次"<<endl

<<"\n\t\t正在退卡中……"<<endl; exit(0);

}

system("cls");

u.Fmoney +=SaveAndDrawMoney;

cerr<<"\n\n\n\t\t对不起，你输入的金额超出有效金额。"<<endl

<<"\n\t\t 请重新输入: ";

}

}

cout<<"\n\n\t\t操作成功，单击任何键返回主菜单！"; getch();

}

void Function::Balance(User &u)

{

system("cls"); cout<<"\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\t\t你当前的余额是："; cout<<"\t\t"<<u.Fmoney<<endl;

cout<<"\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\n\t\t操作成功，单击任何键返回主菜单！"; getch();

}

void Function::Record(User &u)

{

dl;

string str1,str2; system("cls");

cout<<"\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<en

cout<<"\t\t摘要币种钞汇存/取款金额"<<endl; strstream ss;

ss<<u.Accout; ss>>str2;

string frist="Record"+str2 +".txt"; const char \*RecordFile=frist.c\_str(); ifstream infile(RecordFile,ios::in); if(!infile)

{

cout<<"\n\t\t没有任何历史记录！"<<endl; getch();

return ;

}

while(!infile.eof())

{

getline (infile,str1); cout<<"\n\t\t"<<str1<<endl;

}

cout<<"\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\n\t\t操作成功，单击任何键返回主菜单！"; getch();

}

ostream &User::print(ostream &os) //mark

{

system("cls"); os<<"\n\n\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

os<<"\n\t\t用户卡号："<<Accout<<endl;

os<<"\n\t\t姓名： "<<Name <<endl;

os<<"\n\t\t账户金额："<<Fmoney<<endl;

os<<"\n\t\t身份证号："<<ID<<endl;

os<<"\n\t\t地址： "<<Address <<endl;

os<<"\n\t\t联系电话："<<Telephone <<endl;

os<<"\n\t\t密码： "<<Pword1<<endl; os<<"\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\n\t\t操作成功，单击任何键返回主菜单！"; getch();

return os;

}

void User::SaveFile(User &u)

{

int filename=u.Accout; string frist;

string last(".txt"); strstream ss; ss<<filename; ss>>frist; frist+=last;

const char \*UserID=frist.c\_str();

ofstream outfile(UserID,ios::out |ios::trunc );

//outfile<<(\*this);

outfile<<Accout<<' '<<Name<<' '<<Fmoney<<' '<<ID<<' '<<Address<<' '<<Telephone<<' '<<Pword1<<endl;

outfile.close();

}

bool User::Login(User &u)

{

int acc; string pwd;

cout<<"\n\t\t登陆用户："<<endl;

cout<<"\t\t卡号："; cin>>acc;

cout<<"\t\t密码："; cin>>pwd;

int filename=acc; string frist;

string last(".txt"); strstream ss; ss<<filename; ss>>frist; frist+=last;

const char \*UserID=frist.c\_str(); ifstream infile(UserID,ios::in ); while (infile >> (\*this))

{

if (u.Accout == acc )

{

string str; str=Pword1; if(str== pwd)

{

return true;

}

}

}

return false;

}

/\*ostream & operator<<(ostream & o, const User & u)

{

o<<u.Accout <<' '<<u.Name<<' '<<u.Fmoney<<' ' <<u.ID<<' '

<<u.Address <<' '<<u.Telephone<<' ' <<u.Pword1 <<endl; return o;

}\*/

istream & operator>>(istream & i, User & u)

{

i>>u.Accout >>u.Name >>u.Fmoney >>u.ID >>u.Address >>u.Telephone >>u.Pword 1 ;

return i;

}

void User::PassWord()

{

int num=0;

while(num<3)

{

cout<<"\n\t\t请输入位储蓄密码："; for(int i=0;i<8;i++)

{ Pword1[i]=getch();cout<<"\*";} Pword1[8]='\0';

cout<<endl;

cout<<"\n\t\t请再一次确认密码："; for(int i=0;i<8;i++)

{ Pword2[i]=getch();cout<<"\*";} cout<<endl;

Pword2[8]='\0'; if(strcmp(Pword1,Pword2)==0)

{

cout<<"\n\n\t\t密码确认成功，单击任何键进入主菜单"; getch();

return ;

}

else

{

}

num++;

cout<<"\n\t\t你输入的密码错误，请重新输入！\n"<<endl;

if(num==3)

{

system("cls");

cerr<<"\n\n\n\t\t对不起，你三次输入不正确\n"<<endl

<<"\t\t感谢你的使用，再见！ "<<endl;

exit(0);

}

}

}

void User::get(User& u)

{

system("cls");

cout<<"\n\n\n\t\t 请 输 入 用 户 信 息 ：\n"<<endl; cout<<"\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\n\t\t卡号："; cin>>Accout;

cout<<"\n\t\t姓名："; cin>>Name;

cout<<"\n\t\t账户金额："; cin>>Fmoney;

cout<<"\n\t\t身份证号：";

cin>>ID;

cout<<"\n\t\t地址："; cin>>Address;

cout<<"\n\t\t联系电话："; cin>>Telephone; u.PassWord();

cout<<"\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

}

void Sign()

{

system("cls");

cout<<"\n\n\n\n\n\n\t\t\t谢谢您使用银行管理系统\t \n\n"<<endl; cout<<"\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl;

cout<<"\t\t 姓名: 谢道尔 \n\n"

<<"\t\t 班级: 计算机班\n\n"

<<"\t\t 学号: 200710214217\n\n"<<endl;

cout<<"\n\t\t 已经成功退出"<<endl

<<"\n\t\t 欢迎你的光临！" <<endl;

}

char custom(User &u)

{

char x; int k=0;

system("cls"); cout<<"\n\n\n\t\t※※※※※※※※※※※※※※※※※※※※※※※※※※※\n "

|  |  |  |
| --- | --- | --- |
| <<"\t\t※ |  | ※\n" |
| <<"\t\t※ | 欢迎使用储蓄管理系统 | ※\n" |
| <<"\t\t※ |  | ※\n" |
| <<"\t\t※ | 请用户在使用前选择开户： | ※\n" |
| <<"\t\t※ |  | ※\n" |
| <<"\t\t※ | 请选择： | ※\n" |
| <<"\t\t※ | 1：注册开户 | ※\n" |
| <<"\t\t※ | 2：用户登录 | ※\n" |
| <<"\t\t※ | 其他：退出 | ※\n" |
| <<"\t\t※ |  | ※\n" |
| <<"\t\t※ |  | ※\n" |

ndl;

<<"\t\t※※※※※※※※※※※※※※※※※※※※※※※※※※※\n"<<e

cout<<"\n\t\t请选择操作方式："; cin>>x;

if(x=='1')

{ u.get(u);}

else if(x=='2')

{

while(k<3)

{

if(u.Login(u)) //判断登陆是否成功

{ return 0;} else

{

system("cls");

cerr<<"\n\n\n\t\t\t输入有误!\n"<<endl

<<"\t\t\t请重新输入："<<endl;

k++;

}

while(k>=3)

{

system("cls");

cerr<<"\n\n\n\t\t对不起，你三次输入不正确\n"<<endl

<<"\n\t\t感谢你的使用，再见！\n"<<endl; exit(0);

} } }

else

{ Sign(); exit(0); } return 0;

}

char menu()

{

int k=0; char a[3],c,i;

system("cls");

cout<<"\n\n\n\t\t 主菜单 \n\n" ; cout<<"\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"

<<"\t\t\* \*\n"

<<"\t\t\* \*\n"

<<"\t\t\* 操作方式: \*\n"

<<"\t\t\* \*\n"

<<"\t\t\* 1.存款 2.取款 \*\n"

<<"\t\t\* \*\n"

<<"\t\t\* 3.查询余额 4.查询历史记录 \*\n"

<<"\t\t\* \*\n"

<<"\t\t\* 5.用户信息 6.保存并退出 \*\n"

<<"\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n"<<endl; while(k<3)

{

cout<<"\n\t\t请选择操作方式: ";

cin>>a; if(strlen(a)==1)

{

}

else

{

for(i='1';i<='6';i++)

{ c=i;

if(c==a[0]) return c;

}

system("cls");

cerr<<"\n\n\n\t\t\t输入有误!\n"<<endl

<<"\t\t\t请重新输入："<<endl;

k++;

}

while(k>=3)

{

system("cls");

cerr<<"\n\n\n\t\t对不起，你三次输入不正确\n"<<endl

<<"\n\t\t感谢你的使用，再见！\n"<<endl; exit(0);

}

}

return c;

}

char Choose( User &u,Function &F)

{

char c=menu(); while(c!='6')

{

switch(c)

{

case '1': F.Saving(u);c=menu();break;

case '2':F.Drawing(u);c=menu();break;

case '3':F.Balance(u);c=menu();break;

case '4':F.Record(u);c=menu();break; case '5':u.print(cout);c=menu();break;

}

}

* 1. aveFile(u); Sign();

return 0;

}

int main()

{

User u; Function F;

vector<Function> FunctionUser; custom(u);

Choose(u,F);

}

九、 使用手册

* + 1. Nicolai M.Josuttis C++标准程序库自修教程与参考手册

十、 参考文献

1. 美.利普曼 加.劳乔 C++ Primer (4th Edition)： 人民邮电出版社，2006：

1-900

1. 谭浩强 C++面向对象程序设计： 清华大学出版社，2008：1-165
2. 叶至军 C++ STL 开发技术导引： 人民邮电出版社 2008：1-422