**《程序设计基础(2)》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | | **2020级计算机类01班** | | | **姓名** | **孙莹莹** |
| **实验题目** | **ATM模拟程序的设计与实现** | | | | | |
| **实验时间** | **2021.5.22** | | **实验地点** | **DS1408** | | |
| **实验成绩** |  | | **实验性质** | **□验证性 □设计性 ■综合性** | | |
| 教师评价：  □算法/实验过程正确；□源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | | | | |
| 一、实验目的  运用面向对象程序设计思想，基于命令行界面设计并实现一个ATM模拟程序，模拟常见的ATM功能。 | | | | | | |
| 二、实验项目内容  ATM模拟程序能够完成ATM的主要功能，包括：  1）显示欢迎词及提示信息；  2) 用户插卡，ATM验证用户账号及密码有效性，输入错误3次即被锁卡；  3) 余额查询：初始余额为10000元  4）取款功能：每次取款余额为100的整数倍，有单笔和单日金额限制；  5）转账功能：可将本账户中的存款转入其它账户，转入账户账号需两次输入确认；  6）修改密码：密码为6位数字，新密码需要两次输入确认；  7）退卡。  设计实现命令行界面，界面应友好、方便操作。参考界面如图1所示。  程序所涉及到的用户资料、银行帐户、存取款记录等信息保存在数据文件中。其中银行账户的格式如下：  账号 19位数字  姓名  身份证 18位字符  密码 6位数字  余额    图1 ATM模拟程序参考界面 | | | | | | |
| 1. 实验过程或算法（源程序）   1.源代码  #include <iostream>  #include <stdio.h>  #include <vector>  #include <string>  #include <cstring>  #include <fstream>  #include <sstream>  using namespace std;  ofstream outfile;  struct user  {      string account, name, ID, key;      int balance = 0;      bool state;  };  class ATM  {  private:      string account;      string name;      string ID;      string password;      int balance;      bool state;  public:      ATM() {};      void showbalance() const { cout << balance;}      void showpassword() const {cout << password;}      void operator=(user a);      string getaccount() const {return account;}      string getname() const {return name;}      string getID() const {return ID;}      string getpassword() const {return password;}      int getbalance() const {return balance;}      bool getstate() const {return state;}      void changestate(int f) {state = f;}      void showstate();      void changepassword();      void drawmoney(int i);      void savemoney(int i);      void transfermoney(string a, int b);  };  void ATM::operator=(user a)  {      account = a.account;      name = a.name;      password = a.key;      balance = a.balance;      state = a.state;      ID = a.ID;  }  void ATM::showstate()  {      if(state)          cout << "True" << endl;      else          cout << "False" << endl;  }  void ATM::changepassword()  {      cout << "Please enter your current password(3 times to lock):" << endl;      string key;      cin >> key;      int times = 0;      while(key != this->password)      {          cout << "wrong password, try again";          times += 1;          if(times == 3)          {              cout << "You have tried 3 times, your account locked";              return;          }          cin >> key;      }      cout << "right password, now please enter your new password(6 digits only): " << endl;      string newkey;      cin >> newkey;      while(newkey.size() != 6)      {          cout << "please enter only 6 digits! try again!" << endl;          cin >> newkey;      }      cout << "please confirm your password: " << endl;      string newkey1;      cin >> newkey1;      if(newkey == newkey1)      {          this->password = newkey;          outfile << "You have changed the password of " << name <<"'s account" << endl;          return;      }      else      {          cout << "Password cannot match!" << endl;          return;      }  }  void ATM::drawmoney(int i)  {      if(i > balance)      {          cout << "You have " << balance << " money only" << endl;          cout << "withdraw fail" << endl;          return ;      }      else      {          balance -= i;          cout << "success! You have withdraw " << i << " money from your account" << endl;          cout << "now you have " << balance << " money left" << endl;          return;      }  }  void ATM::savemoney(int i)  {      if(i < 0)      {          cout << "money cannot be negative" << endl;          return ;      }      else      {          balance += i;          return ;      }  }  ATM accns[100];  void regist(int i)  {      user a;      cout << "please enter Your account with 19 chars:  " << endl;      cin >> a.account;      cout << "please enter Your name: " << endl;      getchar();      getline(cin,a.name);      cout << "please enter your ID:" << endl;      cin >> a.ID;      cout << "please set your password with 6 numbers: " << endl;      cin >> a.key;      cout << "please confirm your password again: " << endl;      string b;      cin >> b;      if(a.key != b)      {          cout << "password not match, register fail" << endl;          return;      }      else      {          a.balance = 10000;          a.state = true;          accns[i] = a;          outfile << "You have registered an account " << endl;      }  }  void split\_str(char \* s, int i)  {      int j=0, k;      user a;      for(j=0; j<=18; j++)          a.account = a.account + s[j];      j = 20;      while(s[j] != '#')      {          a.name = a.name + s[j];          j += 1;      }      for(k=1; k<=18; k++)          a.ID = a.ID + s[j + k];      j = j + 19;      for(k=1; k<=6; k++)          a.key = a.key + s[j + k];      j += 8;      while (s[j] != ' ')      {          a.balance =  s[j] = 48 + a.balance \* 10;          j += 1;      }      if (s[j + 1] == 49)          a.state = true;      else          a.state = false;      accns[i] = a;  }  int read()  {      ifstream fs\_in;      int num = 0;      fs\_in.open("account.txt");      char info[100] = {0};      fs\_in.getline(info, 100);      while(strlen(info) > 1)      {          split\_str(info, num);          num += 1;          fs\_in.getline(info, 100);      }      return num;  }  int plug\_in(int i)  {      string s;      int j;      cout << "You have entered your card" << endl;      cout << "Your card account is:" << endl;      cin >> s;      for(j=0; j<i; j++)      {          if(accns[j].getaccount() == s)          {              s = accns[j].getpassword();              break;          }      }      if(j==i)      {          cout << "Your card haven't registered." << endl;          return -1;      }      if(!accns[j].getstate())      {          cout << "Your card has locked. Card withdrawed" << endl;          return -1;      }      cout << "Enter your password(3 times to lock): " << endl;      string key;      cin >> key;      int k = 0;      while(key != s)      {          k += 1;          if(k==3)          {              cout << "3 times tried. Your card has been locked" << endl;              cout << "Card withdrawed" << endl;              outfile << accns[j].getname() << "'s card has been locked" << endl;              accns[j].changestate(false);              return -1;          }          cout << "Wrong password. Try again" << endl;          cin >> key;      }      cout << "Welcome, dearly" << accns[j].getname() << endl;      outfile << "You have Plug-in a card named " << accns[j].getname() << endl;      return j;  }  void draw\_money(int i)  {      int money;      cout << "how much do you want to withdraw" << endl;      cin >> money;      bool f = true;      while(f)      {          accns[i].drawmoney(money);          f = false;      READ:          if(f)              cin >> money;      }      return;  }  void transfer\_money(int note, int num)  {      cout << "Please enter the account you what to tranfer to " << endl;      string s;      cin >> s;      int j;      for( j=0; j<num; j++)      {          if(accns[j].getaccount() == s)              break;      }      if(j == num)      {          cout << "His or her card hasn't registered " << endl;          return;      }      if(!accns[j].getstate())      {          cout << "His or her card has been locked. Transfer fail" << endl;          return;      }      cout << "Please confirm the account you want to transfer to" << endl;      string accoun;      cin >> accoun;      if(s == accoun)      {          cout << "You have" << accns[note].getbalance() << " money" << endl;          cout << "enter the money you want to transfer" << endl;          int money;          cin >> money;          bool f = true;          while(f)          {              if(money < 0)              {                  cout << "it cannot be negative" << endl;                  goto READ;              }              if(money > accns[note].getbalance())              {                  cout << "You don't have so much money" << endl;                  cout << "try again" << endl;                  goto READ;              }              cout << "Transfer successully!" << endl;              outfile << accns[note].getname() << "have transfer" << money << " $ to account named" << accns[j].getname() << endl;              accns[note].drawmoney(money);              accns[j].savemoney(money);              f = false;          READ:              if(f)                  cin >> money;          }          return ;      }      else      {          cout << "Account cannot match" << endl;          return;      }  }  void write\_infile(int num)  {      freopen("account.txt", "w", stdout); //输出重定向      for(int i=0; i<num; i++)      {          cout << accns[i].getaccount() << " " << accns[i].getname() << "#" << accns[i].getID() << " " << accns[i].getpassword() << " " << accns[i].getbalance() << " ";          if(accns[i].getstate())              cout << "1" <<endl;          else              cout << "0" << endl;      }      fclose(stdout);  //关闭重定向输出  }  void show()  {      cout << "what do you want to do?" << endl;      cout << "your account data will be reserved in account.txt, your log will be reserved in log.txt" << endl;      cout << "Main menu:" << endl;      cout << "     [1].register an account" <<endl;      cout << "     [2] plug in your card" << endl;      cout << "     [3] view my balance" << endl;      cout << "     [4] withdraw money" << endl;      cout << "     [5] transfer money" << endl;      cout << "     [6] change the password" << endl;      cout << "     [7] pop out the card" << endl;      cout << "     [8] quit" << endl;  }  int main()  {      outfile.open("log.txt", ios::app); //ios::app 追加模式打开文件夹， 以ios::app打开,如果没有文件，那么生成空文件；如果有文件，那么在文件尾 追加。      int num = read();      show();      int choice, note = -1;      cin >> choice;      while(choice != 8)      {          if(choice == 1)  //register an account          {              regist(num);              num += 1;              cout << "what do you want to do now?";              cin >> choice;              continue;          }          if(choice == 2) //plug in the card          {              if(note != -1)              {                  cout << "you cannot insert two cards at a time" << endl;                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }              else              {                  note = plug\_in(num);                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }          }          if(choice == 3) //view the balance          {              if(note != -1 && accns[note].getstate())              {                  cout << "You balance is " << accns[note].getbalance() << endl;                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }              else              {                  cout << "You Haven't plug-in," << endl                       << "Please Plug-in first" << endl;                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }          }          if(choice == 4) // withdraw money          {              if(note != -1 && accns[note].getstate())              {                  draw\_money(note);                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }              else              {                  cout << "You Haven't plug-in," << endl                       << "Please Plug-in first" << endl;                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }          }          if(choice == 5)  //transfer money          {              if(note != -1 && accns[note].getstate())              {                  transfer\_money(note, num);                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }               else              {                  cout << "You Haven't plug-in," << endl                       << "Please Plug-in first" << endl;                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }          }          if(choice == 6) //change the password          {              if(note != -1 && accns[note].getstate())              {                  accns[note].changepassword();                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }               else              {                  cout << "You Haven't plug-in," << endl                       << "Please Plug-in first" << endl;                  cout << "what do you want to do" << endl;                  cin >> choice;                  continue;              }          }          if(choice == 7) //pop out the card          {              cout << "Your card has been withdrawn." << endl;              outfile << "card withdrawed" << endl;              note = -1;              cout << "what do you want to do" << endl;              cin >> choice;              continue;          }      }      cout << "bye!" << endl;      write\_infile(num);      system("pause");      return 0;  }   1. 实验思路   ATM程序通过设计一个ATM类完成，类的私有成员包括客户的姓名卡号密码等信息，公有成员则主要为查看余额，取款等成员函数组成。程序通过一个类数组来存储已经注册的客户信息，因为在具体实现时需要传递/记录该客户在客户数组中的序号，所以在类的外面又定义了转账，取款，查看余额的非类成员函数，他们均通过调用成员函数来实现。 | | | | | | |
| 1. 实验结果及分析和（或）源程序调试过程 2. 当输入都是符合要求的时候：   客户可以创建一个自己的账号，在插卡登录之后进行转账，查看余额，取款，修改密码及退卡的操作。  IMG_20210522_143558  IMG_20210522_145552  IMG_20210522_143729  IMG_20210522_143813   1. 当用户的试图登录一个没有注册的账号，没有插卡就进行操作，取款、转账数目超过余额，转账时两次输入转入账户不一致时均会操作失败并终止该操作，而登录时三次输错密码账号会被冻结，修改密码两次输入新密码不同修改密码操作失败。   IMG_20210522_143924  IMG_20210522_144033  IMG_20210522_144115  IMG_20210522_144306  IMG_20210522_144400  3.因此该程序已经可以满足实验的基本要求，即实现ATM程序的基本操作。设计较为对客户友好合理。 | | | | | | |