**《程序设计基础(2)》实验报告**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **年级、专业、班级** | |  | | | **姓名** |  |
| **实验题目** | **ATM模拟程序的设计与实现** | | | | | |
| **实验时间** | **2021.5.24** | | **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）   ATM\_main.cpp  #include <iostream>  #include "class.h"  #include "function.h"  int main()  {  ATM one;  string filename;  while (1)  {  one.showmeun();  cin >> filename;  filename=filename+".txt";  ifstream in;  in.open(filename);  if (!in)  {  cout<<"This account does not exist"<<endl;  in.close();  }  else  {  in.close();  break;  }  }  one.get\_password(filename);  one.get\_balance(filename);  one.Proofreadpassword();  while(one.showlock()==1)  {  int nu;  one.show();  cin >> nu;  switch(nu){  case 1:  {  one.showaccout();  break;  }  case 2:  {  int x;  cout<<"-------------------------------------------------"<<endl;  cout<<"Please enter the amount to be withdrawn (\*100): ";  cin >>x;  one.cutbalance(x,filename);  break;  }  case 3:  {  int x;  cout<<"-------------------------------------------------"<<endl;  cout <<"Please enter the amount to be deposit (\*100): ";  cin >>x;  one.addbalance(x,filename);  break;  }  case 4:  {  cout<<"-------------------------------------------------"<<endl;  cout <<"Please enter the transfer account number: ";  string filename1,filename2;  cin >> filename1;  filename1=filename1+".txt";  ifstream ine;  ine.open(filename1);  if (!ine)  {  cout<<"This account does not exist"<<endl;  ine.close();  }  else{  cout<<"Please enter the number of transfers (\*100): ";  int num;  cin >>num;  cout <<"Please enter the transfer account number again: ";  cin >>filename2;  filename2=filename2+".txt";  if (filename1!=filename2)  {  cout <<"The account number entered twice is different."<<endl;  }  else{  one.transfer(num,filename1,filename);  }  }  break;  }  case 5:  {  cout<<"-------------------------------------------------"<<endl;  one.changep(filename);  break;  }  case 6:  {  break;  }  default:  {  cout<<"-------------------------------------------------"<<endl;  cout<<"Please enter a valid number. ";  break;  }  }  }  cout <<"Thanks for using. "<<endl;  return 0;  }  Class.h  #ifndef \_CLASS\_H\_  #define \_CLASS\_H\_  #include <iostream>  #include <string>  #include <fstream>  using namespace std;  class ATM  {  private:  string accoutNunmber;  string password;  int balance;  char name[20];  int time=0;  int locks=0;  void wrongpw(){time+=1;};  public:  void showmeun();  void showaccout();  int showlock();  void show();  void unlock();  void lockagain();  void setpassword(string x) ;  void setbalance(int x);  void Proofreadpassword();  void jugepassword();  void get\_password(string filenamed);  void get\_balance(string filenamed);  void addbalance(int x,string filename);  void cutbalance(int x,string filename);  void write\_in\_add(string filename,int x,int y);  void write\_in\_cut(string filename,int x,int y);  void transfer(int x,string filename1,string filename2);  void changep(string filename);  };  #endif  Function.h  #include "class.h"  void ATM::showmeun()  {  cout<<"plesae input the account number: ";  }  void ATM::setpassword(string x)  {  password=string(x.begin(),x.begin()+6);  }  void ATM::setbalance(int x)  {  balance=x;  }  void ATM::unlock()  {  locks=1;  }  void ATM::lockagain()  {  locks=0;  }  void ATM::Proofreadpassword()  {  string number1;  while(time<=3){  cout << "please input password: ";  cin >> number1;  if (number1!=password)  {  cout <<"Password is wrong, you have "<< 3-time << " times can try again."<<endl;  wrongpw();  }  else  {  unlock();  break;  }  }  if (time>3)  {  cout <<"Your account is locked."<<endl;  }  }  void ATM::show()  {  cout<<"Main menu:"<<endl;  cout<<" 1 - View my balance"<<endl;  cout<<" 2 - Withdraw cash"<<endl;  cout<<" 3 - Desposit funds"<<endl;  cout<<" 4 - Transfer accounts"<<endl;  cout<<" 5 - Change password"<<endl;  cout<<" 6 - Exit"<<endl;  cout<<"Enter a choice:";  }  void ATM::get\_password(string filename)  {  ifstream in;  in.open(filename);  char inpassword[20];  int line=1;  while (in.getline(inpassword,20))  {  if (line==4)  {  break;  }  line++;  }  string password1;  password1=inpassword;  setpassword(password1);  in.close();  }  void ATM::get\_balance(string filename)  {  ifstream in;  in.open(filename);  int balance1,line=1;  string y;  while (getline(in,y))  {  if (line==5)  {  break;  }  line++;  }  balance1=atoi(y.c\_str());  setbalance(balance1);  in.close();  }  void ATM::addbalance(int x,string filename)  {  balance=balance+x\*100;  write\_in\_add(filename,x,balance);  cout <<"balance is: "<<balance<<endl;  }  void ATM::cutbalance(int x,string filename)  {  if (balance<x\*100)  {  cout << "Sorry, your credit is running low."<<endl;  }  else  {  balance=balance-x\*100;  write\_in\_cut(filename,x,balance);  }  }  void ATM::write\_in\_add(string filename,int x,int y)  {  ifstream ind;  ind.open(filename);  string strFileData = "";  int line = 1;  char lineData[1024] = {0};  while(ind.getline(lineData, sizeof(lineData)))  {  if (line==5)  {  strFileData += to\_string(y);  strFileData += "\n";  }  else{  string n;  n=lineData;  strFileData += n;  strFileData += "\n";  }  line++;  }  strFileData =strFileData + "add deposit is: "+ to\_string(x\*100) +"\n"+"balance becomes: "+to\_string(y)+"\n";  ind.close();  ofstream out;  out.open(filename);  out.flush();  out<<strFileData;  out.close();  }  void ATM::write\_in\_cut(string filename,int x,int y)  {  ifstream ind;  ind.open(filename);  string strFileData = "";  int line = 1;  char lineData[1024] = {0};  while(ind.getline(lineData, sizeof(lineData)))  {  if (line==5)  {  strFileData += to\_string(y);  strFileData += "\n";  }  else  {  string n;  n=lineData;  strFileData += n;  strFileData += "\n";  }  line++;  }  strFileData =strFileData + "take out deposit is: "+ to\_string(x\*100) +"\n"+"balance becomes: "+to\_string(y)+"\n";  ind.close();  ofstream out;  out.open(filename);  out.flush();  out<<strFileData;  out.close();  cout <<"balance is: "<<y<<endl;  }  void ATM::showaccout()  {  cout <<"you accout is: "<<balance<<endl;  }  int ATM::showlock()  {  return locks;  }  void ATM::transfer(int x,string filename1,string filename2)  {  filename1=filename1;  ifstream ind;  ind.open(filename1);  string strFileData = "";  int line = 1,y;  char lineData[1024] = {0};  while(ind.getline(lineData, sizeof(lineData)))  {  if (line==5)  {  y=atoi(lineData);  y=x\*100+y;  }  line++;  }  ind.close();  write\_in\_add(filename1,x,y);  if (balance<x\*100)  {  cout << "Sorry, your credit is running low."<<endl;  }  else  {  balance=balance-x\*100;  write\_in\_cut(filename2,x,balance);  }  }  void ATM::changep(string filename)  {  int x,y;  cout <<"Please input your new password: ";  cin >> x;  while(true)  {  if (x<100000||x>999999)  {  cout <<"Please enter six digits. "<<endl;  }  else  {  break;  }  cout <<"Please input your new password: ";  cin >> x;  }  cout <<"Please input your new password again: ";  cin >> y;  if (x!=y)  {  cout <<"The two passwords are different. "<<endl;  }  else  {  ifstream ind;  ind.open(filename);  string strFileData = "";  int line = 1;  char lineData[1024] = {0};  while(ind.getline(lineData, sizeof(lineData)))  {  if (line == 4)  {  strFileData +=to\_string(x);  strFileData += "\n";  }  else  {  string n;  n=lineData;  strFileData += n;  strFileData += "\n";  }  line++;  }  ind.close();  ofstream out;  out.open(filename);  out.flush();  out<<strFileData;  out.close();  cout <<"Password modified successfully"<<endl;  }  } | | | | | | |
| 1. 实验结果及分析和（或）源程序调试过程   截屏2021-05-16 下午7.58.03  截屏2021-05-16 下午7.58.32  截屏2021-05-16 下午7.58.41 | | | | | | |

备注：

1、学生应按照要求正确地撰写实验报告：

* 1. 将实验所涉及的源程序文件内容（实验操作步骤或者算法）填写在“实验过程或算法（源程序）”栏目中。
  2. 将实验所涉及源程序调试过程（输入数据和输出结果）或者实验的分析内容填写在“**实验结果及分析和（或）源程序调试过程**”栏目中。
  3. 在实验报告页脚的“报告创建时间：”处插入完成实验报告时的日期和时间。
  4. 学生提交实验报告时，每个实验一个电子文档，如果实验中有多个电子文档（如源程序或图形等），则压缩成一个.zip压缩包文档提交，压缩包文件名同实验报告文件名（见下条）。
  5. 提交的实验报告电子文档命名为：“年级（四位数字不要“级”字）专业（缩写：计算机科学与技术专业（计科）、信息安全专业（信息）、物联网工程（物联网））学号（八位数字）姓名＋实验序号（一位数字）．doc。如张三（学号20165676）完成第2个Project，专业为“计算机科学与技术”专业，完成的课程设计报告命名为：**2016计科20165676张三2**，以后几次实验的报告名称以此类推。