**北京科技大学实验报告**

学院：计算机与通信工程学院 专业： 物联网工程 班级：

姓名： 学号： 实验日期： 2022年 10 月 21日

**实验名称：个人银行账户管理系统的改进**

**实验目的：**

完善银行账户系统，并实现以下功能：

(简述你的系统新增了哪些“银行1-6”没有的功能，每个功能用1句话简述)

1. 增加了日期构造异常处理机制

2. 增加了账户错误的相关异常处理

3. 增加了图形界面。

4. 图形界面有当月还款的提示和收入支出统计

5. 可以查询某个月按照时间排序的账户查询信息

**实验仪器：**

计算机：MacBook Pro (15-inch, 2018)

CPU：2.2GHz Intel Core i7 8750H

内存：16GB 2400MHz LPDDR4

硬盘：256GB

显卡：Intel Iris Plus Graphics 650 1536 MB & Radeon Pro 555X 4GB

操作系统：macOS Monterey 12.5.1

编译器：Apple clang version 14.0.0 (clang-1400.0.29.102)

**实验内容与步骤：**

**【**功能1：建立简单的储蓄账户**】**

1.设计

（为了实现功能1，你添加了哪些类，设计了什么样的算法？画出类图和流程图，并配以文字解释说明）

2.实现（粘贴关键代码，需要有注释）

|  |
| --- |
| class SavingsAccount //建立储蓄账户类  { private:  int id; //账号  double balance; //余额  double rate; //存款的年利率  int lastDate; //上次变更余额的时期  double accumulation; //余额按日累加之和  void record(int date, double amount);//获得到指定日期为止的存款金额按日累积值①  double accumulate(int date) const {  return accumulation + balance \* (date - lastDate);  } //计算利息 ②  public:  SavingsAccount(int date, int id, double rate); //构造函数  int getId() const { return id; }  double getBalance() const { return balance; }  double getRate() const { return rate; }  void deposit(int date, double amount); //存入现金  void withdraw(int date, double amount); //取出现金  void settle(int date); //结算利息，每年1月1日调用一次该函数  void show() const; //显示账户信息  };  //计算年息  void SavingsAccount::settle(int date) {  double interest = accumulate(date) \* rate / 365;  if (interest != 0)  record(date, interest);  accumulation = 0;  } |

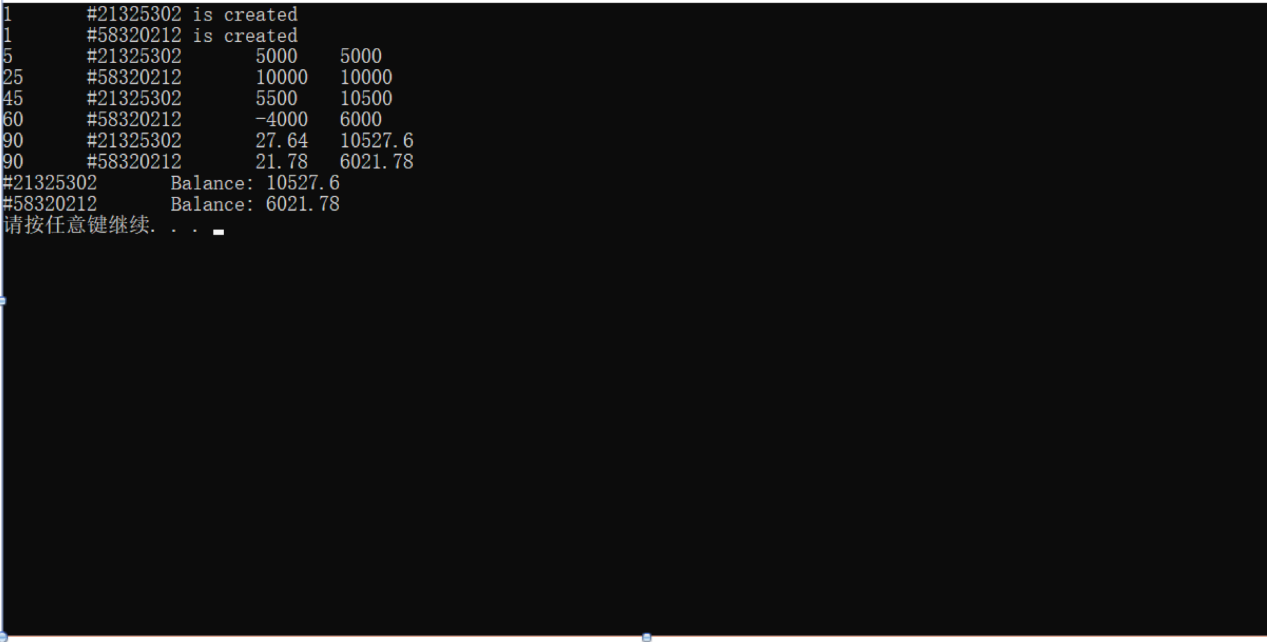
代码片段1 储蓄账户类的关键代码

(请用方框将代码圈住，并在方框下面写上这段代码的序号和名称，没写注释的同学注意在代码上加注释)

3.测试

（1）测试用例1：a s S3755217 0.015（用户输入正确数据）

得到的运行截图如下：



分析：设计这个测试用例的目的是什么？测出的结果表明什么？

（2）测试用例2：100 s S3755217 0.015（用户输入了错误的选项）

。。。

（3）测试用例5: a s S3755217 50000（用户输入了错误的利率）

(共填写3-5个测试用例)

**【**功能2：建立简单的储蓄账户**】**

……

……

……

【功能1：增加了日期构造异常处理机制】

1. 设计：定义了一个新的类用来处理日期的异常，当输入的日期不合法时，抛出这个异常。
2. 实现

|  |
| --- |
| class DateException : public exception{ public:  const char \*what() const noexcept override{  return "Not a right Date. Please input again.";  } };  下面是用来调用这个方法的：  Date::Date(int year, int month, int day):year(year), month(month), day(day) {  int days\_in\_month[13] = {0, 31, (28 + ((year % 4 == 0 && year % 100 != 0) || (year % 400 == 0))), 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};  if(year < 1900 || month > 12 || month < 1 || day < 1 || day > days\_in\_month[month]){  throw DateException();  }  totalDays = distance(); } |

1. 测试
2. 测试用例1:q 2008-01-01 2009-01-01 （用户输入正确的时间）

运行截图如下：

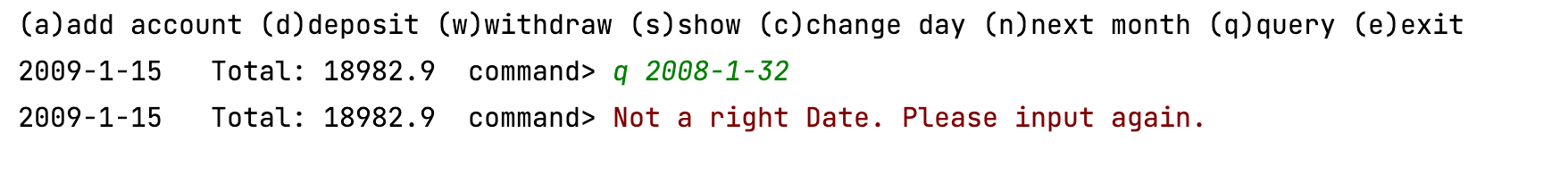
文本

中度可信度描述已自动生成

分析：该命令用于查询08年1月1日到09年1月1日的操作记录，结果表明可以成功运行

1. 测试用例 2:q 2008-01-32 2009-02-29 （用户输入了错误的时间）

运行截图如下：



分析：由于输入了错误的日期，导致throw了异常，并且监测到并输出异常信息。

1. 测试用例3:q qqqqqq（用户输入的不是日期）

运行截图如下：

图形用户界面, 文本, 应用程序

描述已自动生成

分析：由于输入的不是日期，依旧throw了该异常。

【功能2：增加了账户错误的相关异常处理】

1. 设计

当增加的账户不是一个合法的参数（s 或者c）时抛出一个异常。

当存入或者取出的账户编号（index）不存在时抛出一个异常。

当从账户取钱时，余额不足，抛出一个异常。

当改变日期时，如果输入的日期小于今天或者超出本月最大日期，抛出异常。

1. 实现

|  |
| --- |
| switch (cmd) {  case 'a'://增加账户  if(is\_now\_reading\_a\_file)  commands >> type >> id;  else {  try{  cin >> type >> id;  if(type == 's' || type == 'c'){  commands << endl << cmd << " ";  commands << type << " " << id << " ";  }  else throw runtime\_error("Not a saving or credit account.");  }  catch (runtime\_error& err) {  cerr << err.what() << endl;  }  }  if (type == 's') {  if(is\_now\_reading\_a\_file) commands >> rate;  else {  cin >> rate;  commands << rate;  }  account = new SavingsAccount(date, id, rate);  }  else {  if(is\_now\_reading\_a\_file) commands >> credit >> rate >> fee;  else {  cin >> credit >> rate >> fee;  commands << credit << " " << rate << " " << fee;  }  account = new CreditAccount(date, id, credit, rate, fee);  }  accounts.push\_back(account);  break;  case 'd'://存入现金  try{  if(is\_now\_reading\_a\_file){  commands >> index >> amount;  getline(commands, desc);  }  else{  cin >> index >> amount;  getline(cin, desc);  if(index < 0 || index >= accounts.size()){  throw runtime\_error("The index you input is not exist.");  }  commands << endl << cmd << " ";  commands << index << " " << amount << " " << desc;  }  accounts[index]->deposit(date, amount, desc);  }  catch (runtime\_error& err) {  cerr << err.what() << endl;  }  break;  case 'w'://取出现金  try{  if(is\_now\_reading\_a\_file){  commands >> index >> amount;  getline(commands, desc);  }  else{  cin >> index >> amount;  getline(cin, desc);  if(index < 0 || index >= accounts.size()){  throw runtime\_error("The index you input is not exist.");  }  }  accounts[index]->withdraw(date, amount, desc);  // 如果没有错误再进行写入文件  if(!is\_now\_reading\_a\_file) {  commands << endl << cmd << " ";  commands << index << " " << amount << " " << desc;  }  }  catch (AccountException& err){  cerr << err.what() << endl;  }  catch (runtime\_error& err) {  cerr << err.what() << endl;  }  break;  case 's'://查询各账户信息  for (size\_t i = 0; i < accounts.size(); i++) {  cout << "[" << i << "] ";  accounts[i]->show();  cout << endl;  }  break;  case 'c'://改变日期  if(is\_now\_reading\_a\_file) commands >> day;  else {  cin >> day;  }  if (day < date.getDay())  cerr << "You cannot specify a previous day" << endl;  else if (day > date.getMaxDay())  cerr << "Invalid day" << endl;  else {  date = Date(date.getYear(), date.getMonth(), day);  if(!is\_now\_reading\_a\_file) {  commands << endl << cmd << " ";  commands << day;  }  }  break;  case 'n'://进入下个月  if(!is\_now\_reading\_a\_file)  commands << endl << cmd << " ";  if (date.getMonth() == 12)  date = Date(date.getYear() + 1, 1, 1);  else  date = Date(date.getYear(), date.getMonth() + 1, 1);  for (auto & iter : accounts)  iter->settle(date);  break;  case 'q'://查询一段时间内的账目  try {  date1 = Date::read();  date2 = Date::read();  Account::query(date1, date2);  }  catch (DateException& err){  cerr << err.what() << endl;  if(cin.fail()) cin.clear();  }  break;  default:  cerr << "Not a right command." <<endl;  break; }  class AccountException : public exception{ private:  Account\* account; public:  explicit AccountException(Account \*account): account(account){};  const char \*what() const noexcept override{  string s1 = "Not enough money in this account:";  string s2 = account->getId();  string s3 = to\_string(account->getBalance());  static string s4 = s1+s2+"\tbalance:"+s3;  return s4.c\_str();  } };  void CreditAccount::withdraw(Date date, double amount, std::string desc) {  if(abs(amount) + abs(getBalance()) > credit){  throw AccountException(this);  }  else {  record(date, -amount, desc);  acc.change(date, getDebt());  } }  void SavingsAccount::withdraw(Date date, double amount, std::string desc) {  if(amount <= getBalance()) {  record(date, -amount, desc);  acc.change(date, getBalance());  }  else throw AccountException(this); } |

1. 测试
2. 测试用例1: a b S333 333 （用户输入了账户类型）

输出截图：

文本

描述已自动生成

分析：由于输入了错误的账户类型b，抛出了异常（账户类型错误）。

1. 测试用例2:d 19 810（用户输入错误的账户序号）

输出截图：

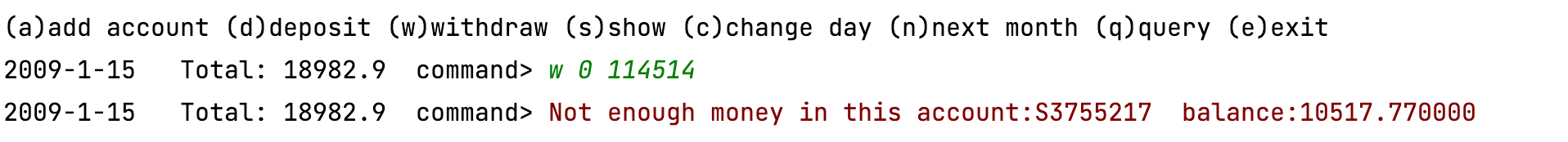
图片包含 文本

描述已自动生成

分析：由于输入了错误的账户序号（19），抛出了异常。

1. 测试用例3:w 0 114514（取钱超出余额）

输出截图：



分析：由于取的钱数超过了账户内的余额10517.77元，抛出异常

1. 测试用例4:c 50 （用户改变日期时，想要把当月天数改为50）

输出截图：

图片包含 文本

描述已自动生成

分析：由于50超出了1月的天数，抛出了异常。

【功能3：增加了图形页面】

1. 设计

采用qt5.12设计了图形页面，并且完成了相关的按钮，注册/登陆功能。

以下为主页面样式，包含了一个账户和密码的输入框和注册登陆的按钮。

电脑萤幕的截图

描述已自动生成

以下为登陆后页面的样式，包含欢迎语句和相关的功能输入框和按钮。

电脑萤幕的截图

描述已自动生成

当在输入框输入相应的内容，并且点按按钮，将会在下面的框内显示对应的信息。

1. 实现

|  |
| --- |
| *//main.cpp*  #include "Account.h"  #include "date.h"  #include "SavingsAccount.h"  #include "CreditAccount.h"  #include <iostream>  #include <vector>  #include <algorithm>  #include <fstream>  #include "stdexcept"  #include "mainwindow.h"  #include <QApplication>  #include "insidewindow.h"  *using* *namespace* std;  int **main**(int argc, char \*argv[])  {  QApplication a(*argc*, *argv*);  MainWindow w;  w.show();  *return* a.exec();  }  //mainwindow.cpp  #include "mainwindow.h"  #include "ui\_mainwindow.h"  #include "vector"  #include "bankusers.h"  #include "QLineEdit"  #include "QDebug"  #include "fstream"  #include "string"  #include "md5.h"  #include "iostream"  #include "QMessageBox"  #include "QDir"  #include "insidewindow.h"  *using* *namespace* std;  string MainWindow::Log\_on\_User;  MainWindow::**MainWindow**(QWidget \*parent)  : QMainWindow(*parent*)  , ui(*new* Ui::MainWindow)  {  ui->setupUi(*this*);  fstream users;  users.open("users.txt", ios\_base::in);  *while*(!users.eof() && users.is\_open()){  string id,password;  users >> id >> password;  bankUsers.push\_back(*new* BankUsers(id,password));  }  users.close();  }  MainWindow::~***MainWindow***()  {  *delete* ui;  }  void MainWindow::**on\_registerID\_clicked**()  {  fstream users;  users.open("users.txt", ios\_base::out | ios\_base::app);  *//QDir* *temdir("users.txt");*  *//QString* *filePath* *=* *temdir.absoluteFilePath("users.txt");*  *//qDebug()* *<<* *filePath;*  QString id = ui->inputID->text();  QString password = ui->inputPassword->text();  *//qDebug()* *<<* *users.is\_open();*  *for*(*auto* &iter : bankUsers){  *if*(iter->getID() == id.toStdString()){  QMessageBox::warning(*this*,tr("警告信息"),tr("系统中已有重名用户"));  *return*;  }  }  users << id.toStdString() << " " << MD5(password.toStdString()).toString() << endl;  *//qDebug()* *<<* *id* *<<* *"* *"* *<<* *password;*  bankUsers.push\_back(*new* BankUsers(id.toStdString(), MD5(password.toStdString()).toString()));  users.close();  *//std::cout* *<<* *id.toStdString()* *<<* *MD5(password.toStdString()).toString();*  }  void MainWindow::**on\_log\_onID\_clicked**()  {  QString id = ui->inputID->text();  QString password = ui->inputPassword->text();  *for*(*auto* &iter : bankUsers){  *if*(iter->getID() == id.toStdString()){  *if*(iter->getPassword() == MD5(password.toStdString()).toString()){  *//qDebug()* *<<* *1;*  Log\_on\_User = iter->getID();  *static* InsideWindow w1;  QObject::connect(*this*,SIGNAL(showInsideWindow()),&w1,SLOT(recvInsideWindow()));  *//qDebug()* *<<* *QString::fromStdString(* *Log\_on\_User* *);*  *this*->hide(); *//隐藏本页面*  emit showInsideWindow(); *//激活信号*  }  *else* {  QMessageBox::warning(*this*,tr("警告信息"),tr("密码错误"));  }  *return*;  }  }  QMessageBox::warning(*this*,tr("警告信息"),tr("系统中查无此户"));  *return*;  }  // insidewindow.h  #include "insidewindow.h"  #include "ui\_insidewindow.h"  #include "QString"  #include "QDebug"  #include "date.h"  #include "vector"  #include "Account.h"  #include "fstream"  #include "SavingsAccount.h"  #include "CreditAccount.h"  #include "stdexcept"  #include "iostream"  #include "QMessageBox"  *using* *namespace* std;  *struct* **deleter** {  *template* <*class* T> void *operator* () (T\* p) { *delete* p; }  };  InsideWindow \*InsideWindow::iw = *nullptr*;  Date date(2008, 11, 1); *//* *起始日期*  fstream commands;  vector<Account \*> accounts; *//* *创建账户数组，元素个数为0*  QString aa[5]; *//* *0.s* *or* *c* *1.账户名* *2.信用* *3.利率* *4.年费*  QString ddww[3]; *//* *0.账户名编号* *1.金额* *2.原因*  int day\_change;  Date q1,q2;  InsideWindow::**InsideWindow**(QWidget \*parent) :  MainWindow(*parent*),  ui(*new* Ui::InsideWindow)  {  ui->setupUi(*this*);  iw = *this*;  *//qDebug()* *<<* *QString::fromStdString(this->MainWindow::Log\_on\_User);*  QString theHello = "欢迎用户 "+ QString::fromStdString(*this*->MainWindow::Log\_on\_User) +" ，请进行接下来的操作：先在对应框内输入内容，再点击按钮，在下方会显示对应结果。";  ui->label->setText(theHello);  *//初始化只加载一次*  *//static* *bool* *is\_now\_reading\_a\_file* *=* *true;* *//* *1为正在读取文件，0为手动添加命令并且添加到文件内模式*  *//* *static* *bool* *only\_show\_once* *=* *false;*  char cmd;  commands.open("commands.txt", ios\_base::in | ios\_base::out);  *if*(!commands.is\_open()){  *//* *文件打开失败*  commands.open("commands.txt",ios\_base::out | ios\_base::app);  *//* *cout* *<<* *"File* *cannot* *open* *or* *there* *is* *no* *such* *a* *command.txt.* *It* *will* *create* *a* *new* *file."* *<<* *endl;*  *//is\_now\_reading\_a\_file* *=* *false;*  }  *else* {  *do*{  commands >> cmd;  InsideWindow::backMain(cmd, *true*);  }*while*(!commands.eof());  commands.clear();  }  InsideWindow::jisrBro2();  }  InsideWindow::~***InsideWindow***()  {  for\_each(accounts.begin(), accounts.end(), deleter());  commands.close();  *delete* ui;  }  void InsideWindow::**recvInsideWindow**(){  *this*->show();  }  void InsideWindow::**backMain**(char cmd, bool is\_now\_reading\_a\_file){  *//* *if* *(!is\_now\_reading\_a\_file* *&&* *!only\_show\_once)* *{*  *//* *cout* *<<* *"(a)add* *account* *(d)deposit* *(w)withdraw* *(s)show* *(c)change* *day* *(n)next* *month* *(q)query* *(e)exit"*  *//* *<<* *endl;*  *//* *only\_show\_once* *=* *true;*  *//* *}*  *//显示日期和总金额*  *if*(!is\_now\_reading\_a\_file){  date.show();  cout << "\tTotal: " << Account::getTotal() << "\tcommand> ";  }  char type;  unsigned int index; int day;  double amount, credit, rate, fee;  string user, id, desc;  Account\* account;  Date date1{}, date2{};  *//* *if(is\_now\_reading\_a\_file)* *commands* *>>* *cmd;*  *//* *else* *{*  *//* *cin* *>>* *cmd;*  *//* *//if(cmd* *!=* *'e'* *&&* *cmd* *!=* *'s')*  *//* *//* *commands* *<<* *endl* *<<* *cmd* *<<* *"* *";*  *//* *}*  *switch* (cmd) {  *case* 'a':*//增加账户*  *if*(is\_now\_reading\_a\_file)  commands >> type >> user >> id;  *else* {  *try*{  *//cin* *>>* *type* *>>* *user* *>>* *id;*  type = aa[0].toStdString().c\_str()[0];  user = MainWindow::Log\_on\_User;  id = aa[1].toStdString();  *if*(type == 's' || type == 'c'){  commands << endl << cmd << " ";  commands << type << " " << user << " " << id << " ";  }  *else* *throw* runtime\_error("Not a saving or credit account.");  }  *catch* (runtime\_error& err) {  cerr << err.*what*() << endl;  }  }  *if* (type == 's') {  *if*(is\_now\_reading\_a\_file) commands >> rate;  *else* {  *//cin* *>>* *rate;*  rate = aa[3].toDouble();  commands << rate;  }  account = *new* SavingsAccount(date, user , id, rate);  }  *else* {  *if*(is\_now\_reading\_a\_file) commands >> credit >> rate >> fee;  *else* {  *//cin* *>>* *credit* *>>* *rate* *>>* *fee;*  credit = aa[2].toDouble();  rate = aa[3].toDouble();  fee = aa[4].toDouble();  commands << credit << " " << rate << " " << fee;  }  account = *new* CreditAccount(date, user, id, credit, rate, fee);  }  *if*(user == MainWindow::Log\_on\_User) *//进行用户过滤*  iw->ui->textBrowser1->append(QString::fromStdString(user+"的"+id+"账号已创建."));  accounts.push\_back(account);  *break*;  *case* 'd':*//存入现金*  *try*{  *if*(is\_now\_reading\_a\_file){  commands >> index >> amount;  getline(commands, *desc*);  }  *else*{  *//cin* *>>* *index* *>>* *amount;*  *//getline(cin,* *desc);*  index = ddww[0].toUInt();  amount = ddww[1].toDouble();  desc = ddww[2].toStdString();  *if*(index < 0 || index >= accounts.size()){  *throw* runtime\_error("The index you input is not exist.");  }  commands << endl << cmd << " ";  commands << index << " " << amount << " " << desc;  InsideWindow::jisrBro2();  }  accounts[index]->*deposit*(date, amount, desc);  *if*(accounts[index]->getUser() == MainWindow::Log\_on\_User) *//* *进行用户过滤*  iw->ui->textBrowser1->append(QString::fromStdString(to\_string(date.getYear())+"年"+to\_string(date.getMonth())+"月"+to\_string(date.getDay())+"日"+accounts[index]->getId()+"存了"+to\_string(amount)+"元,原因："+desc));  }  *catch* (runtime\_error& err) {  cerr << err.*what*() << endl;  }  *break*;  *case* 'w':*//取出现金*  *try*{  *if*(is\_now\_reading\_a\_file){  commands >> index >> amount;  getline(commands, *desc*);  }  *else*{  *//cin* *>>* *index* *>>* *amount;*  *//getline(cin,* *desc);*  index = ddww[0].toUInt();  amount = ddww[1].toDouble();  desc = ddww[2].toStdString();  *if*(index < 0 || index >= accounts.size()){  *throw* runtime\_error("The index you input is not exist.");  }  InsideWindow::jisrBro2();  }  accounts[index]->*withdraw*(date, amount, desc);  *if*(accounts[index]->getUser() == MainWindow::Log\_on\_User) *//* *进行用户过滤*  iw->ui->textBrowser1->append(QString::fromStdString(to\_string(date.getYear())+"年"+to\_string(date.getMonth())+"月"+to\_string(date.getDay())+"日"+accounts[index]->getId()+"取了"+to\_string(amount)+"元,原因："+desc));  *//* *如果没有错误再进行写入文件*  *if*(!is\_now\_reading\_a\_file) {  commands << endl << cmd << " ";  commands << index << " " << amount << " " << desc;  }  }  *catch* (AccountException& err){  cerr << err.*what*() << endl;  QMessageBox::warning(*iw*,tr("警告信息"),tr("你没这么多钱，你居然还取，乐死我了"));  }  *catch* (runtime\_error& err) {  cerr << err.*what*() << endl;  }  *break*;  *case* 's':*//查询各账户信息*  *static* unsigned int num;  num = 1;  *for* (size\_t i = 0; i < accounts.size(); i++) {  *//* *cout* *<<* *"["* *<<* *i* *<<* *"]* *";*  *//* *accounts[i]->show();*  *//* *cout* *<<* *endl;*  *if*(accounts[i]->getUser() == MainWindow::Log\_on\_User) *//* *进行用户过滤*  iw->ui->textBrowser1->append(QString::fromStdString("[" + to\_string(num++) + "] " + accounts[i]->*show*()));  }  *break*;  *case* 'c':*//改变日期*  *if*(is\_now\_reading\_a\_file) commands >> day;  *else* {  *//cin* *>>* *day;*  day = day\_change;  }  *if* (day < date.getDay()){  cerr << "You cannot specify a previous day" << endl;  QMessageBox::warning(*iw*,tr("警告信息"),tr("时光不能倒流。"));  }  *else* *if* (day > date.getMaxDay()){  cerr << "Invalid day" << endl;  QMessageBox::warning(*iw*,tr("警告信息"),tr("超出了当月最大日期"));  }  *else* {  date = Date(date.getYear(), date.getMonth(), day);  *if*(!is\_now\_reading\_a\_file) {  commands << endl << cmd << " ";  commands << day;  }  }  *break*;  *case* 'n':*//进入下个月*  *if*(!is\_now\_reading\_a\_file)  commands << endl << cmd << " ";  *if* (date.getMonth() == 12)  date = Date(date.getYear() + 1, 1, 1);  *else*  date = Date(date.getYear(), date.getMonth() + 1, 1);  *for* (*auto* & iter : accounts)  iter->*settle*(date);  *break*;  *case* 'q':*//查询一段时间内的账目*  *try* {  date1 = q1;  date2 = q2;  iw->ui->textBrowser1->append(QString::fromStdString(Account::query(*date1*, *date2*, MainWindow::Log\_on\_User)));  }  *catch* (DateException& err){  cerr << err.*what*() << endl;  *if*(cin.fail()) cin.clear();  }  *break*;  *default*:  cerr << "Not a right command." <<endl;  *break*;  }  }  void InsideWindow::**on\_pushButton\_vhhu\_clicked**()  {  aa[1] = ui->id->text();  *for*(*auto* &iter : accounts){  *if*(iter->getId() == aa[1].toStdString()){  QMessageBox::warning(*this*,tr("警告信息"),tr("系统中已有重名账号"));  *return*;  }  }  *if*(ui->checkBox->checkState() == Qt::*Checked*){  aa[0] = "c";  }  *else* aa[0] = "s";  aa[2] = ui->credit->text();  aa[3] = ui->rate->text();  aa[4] = ui->fare->text();  InsideWindow::backMain('a', *false*);  }  void InsideWindow::**on\_pushButton\_cyqm\_clicked**()  {  ddww[0] = ui->id\_cy->text();  *for*(unsigned int i=0; i < accounts.size(); i++){  *if*(accounts[i]->getId() == ddww[0].toStdString()){  *if*(accounts[i]->getUser() != MainWindow::Log\_on\_User){  QMessageBox::warning(*this*,tr("警告信息"),tr("非法操作：这不是你的账号"));  *return*;  }  ddww[0] = (char)i+48;  ddww[1] = ui->amount\_cy->text();  ddww[2] = ui->desc\_cy->text();  InsideWindow::backMain('d',*false*);  *return*;  }  }  QMessageBox::warning(*this*,tr("警告信息"),tr("系统中找不到此账号"));  }  void InsideWindow::**on\_pushButton\_quqm\_clicked**()  {  ddww[0] = ui->id\_qu->text();  *for*(unsigned int i=0; i < accounts.size(); i++){  *if*(accounts[i]->getId() == ddww[0].toStdString()){  *if*(accounts[i]->getUser() != MainWindow::Log\_on\_User){  QMessageBox::warning(*this*,tr("警告信息"),tr("非法操作：这不是你的账号"));  *return*;  }  ddww[0] = (char)i+48;  ddww[1] = ui->amount\_qu->text();  ddww[2] = ui->desc\_qu->text();  InsideWindow::backMain('w',*false*);  *return*;  }  }  QMessageBox::warning(*this*,tr("警告信息"),tr("系统中找不到此账号"));  }  void InsideWindow::**on\_pushButton\_gdbmriqi\_clicked**()  {  day\_change = ui->day\_change->text().toInt();  InsideWindow::backMain('c',*false*);  }  void InsideWindow::**on\_pushButton\_jbruxxgeyt\_clicked**()  {  InsideWindow::backMain('n',*false*);  }  void InsideWindow::**on\_pushButton\_iaxyxbxi\_clicked**()  {  InsideWindow::backMain('s',*false*);  }  void InsideWindow::**on\_pushButton\_iaxyvhmu\_clicked**()  {  q1 = Date(ui->dateEdit1->date().year(), ui->dateEdit1->date().month(), ui->dateEdit1->date().day());  q2 = Date(ui->dateEdit2->date().year(), ui->dateEdit2->date().month(), ui->dateEdit2->date().day());  InsideWindow::backMain('q',*false*);  }  *//wulmwh212bjwhgricvizo.veuiyigeuvyb.*  void InsideWindow::**jisrBro2**(){  double ykhr = 0;  double uzru = 0, viiu = 0;  *for*(unsigned int i=0; i < accounts.size(); i++){  *if*(accounts[i]->getUser() == MainWindow::Log\_on\_User){  *if*(accounts[i]->*s\_or\_c*()){*//信用卡*  ykhr += accounts[i]->*getDebt*();  }  }  }  *//cout* *<<* *date.getYear()<<date.getMonth()-1<<1*  uzru = Account::iaxy\_uzru(Date(date.getYear(),date.getMonth(),1),Date(date.getYear(),date.getMonth(),date.getMaxDay()),MainWindow::Log\_on\_User);  viiu = Account::iaxy\_viiu(Date(date.getYear(),date.getMonth(),1),Date(date.getYear(),date.getMonth(),date.getMaxDay()),MainWindow::Log\_on\_User);  iw->ui->textBrowser2->append(QString::fromStdString("本月应还："+to\_string(ykhr)+" 本月收入："+to\_string(uzru)+" 本月支出"+to\_string(viiu)));  }  //mainwindow.h  #ifndef MAINWINDOW\_H  #define MAINWINDOW\_H  #include <QMainWindow>  #include "vector"  #include "bankusers.h"  QT\_BEGIN\_NAMESPACE  *namespace* **Ui** { *class* **MainWindow**; }  QT\_END\_NAMESPACE  *class* **MainWindow** : *public* QMainWindow  {  Q\_OBJECT  *public*:  *static* string Log\_on\_User;  **MainWindow**(QWidget \*parent = *nullptr*);  ~***MainWindow***();  *private* slots:  void **on\_registerID\_clicked**();  void **on\_log\_onID\_clicked**();  signals:  void **showInsideWindow**();  *private*:  Ui::MainWindow \*ui;  *protected*:  vector<BankUsers \*> bankUsers;  };  #endif *//* *MAINWINDOW\_H*  //insidewindow.h  #ifndef INSIDEWINDOW\_H  #define INSIDEWINDOW\_H  #include <QMainWindow>  #include "mainwindow.h"  *namespace* **Ui** {  *class* **InsideWindow**;  }  *class* **InsideWindow** : *public* MainWindow  {  Q\_OBJECT  *public*:  *explicit* **InsideWindow**(QWidget \*parent = *nullptr*);  ~***InsideWindow***();  *private* slots:  void **recvInsideWindow**();  void **on\_pushButton\_vhhu\_clicked**();  void **on\_pushButton\_cyqm\_clicked**();  void **on\_pushButton\_quqm\_clicked**();  void **on\_pushButton\_gdbmriqi\_clicked**();  void **on\_pushButton\_jbruxxgeyt\_clicked**();  void **on\_pushButton\_iaxyxbxi\_clicked**();  void **on\_pushButton\_iaxyvhmu\_clicked**();  *private*:  Ui::InsideWindow \*ui;  *static* InsideWindow \*iw;  *static* void **backMain**(char cmd, bool is\_now\_reading\_a\_file);  *static* void **jisrBro2**();  };  #endif *//* *INSIDEWINDOW\_H*  // mainwindow.ui  <?**xml** version="1.0" encoding="UTF-8"?>  <*ui* version="4.0">  <*class*>MainWindow</*class*>  <*widget* class="QMainWindow" name="MainWindow">  <*property* name="geometry">  <*rect*>  <*x*>0</*x*>  <*y*>0</*y*>  <*width*>800</*width*>  <*height*>600</*height*>  </*rect*>  </*property*>  <*property* name="windowTitle">  <*string*>银行系统——注册/登录页面</*string*>  </*property*>  <*widget* class="QWidget" name="centralwidget">  <*widget* class="QLabel" name="ID">  <*property* name="geometry">  <*rect*>  <*x*>50</*x*>  <*y*>80</*y*>  <*width*>41</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>账号：</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="Password">  <*property* name="geometry">  <*rect*>  <*x*>50</*x*>  <*y*>130</*y*>  <*width*>41</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>密码：</*string*>  </*property*>  </*widget*>  <*widget* class="QPushButton" name="registerID">  <*property* name="geometry">  <*rect*>  <*x*>170</*x*>  <*y*>320</*y*>  <*width*>113</*width*>  <*height*>32</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>注册</*string*>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="inputID">  <*property* name="geometry">  <*rect*>  <*x*>90</*x*>  <*y*>80</*y*>  <*width*>191</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="inputPassword">  <*property* name="geometry">  <*rect*>  <*x*>90</*x*>  <*y*>130</*y*>  <*width*>191</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  </*widget*>  <*widget* class="QPushButton" name="log\_onID">  <*property* name="geometry">  <*rect*>  <*x*>300</*x*>  <*y*>320</*y*>  <*width*>113</*width*>  <*height*>32</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>登录</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label">  <*property* name="geometry">  <*rect*>  <*x*>540</*x*>  <*y*>520</*y*>  <*width*>241</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>Made by: 王冠超 物联212 U202140874</*string*>  </*property*>  </*widget*>  </*widget*>  <*widget* class="QMenuBar" name="menubar">  <*property* name="geometry">  <*rect*>  <*x*>0</*x*>  <*y*>0</*y*>  <*width*>800</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  </*widget*>  <*widget* class="QStatusBar" name="statusbar"/>  </*widget*>  <*resources*/>  <*connections*/>  </*ui*>  //insidewindow.ui  <?**xml** version="1.0" encoding="UTF-8"?>  <*ui* version="4.0">  <*class*>InsideWindow</*class*>  <*widget* class="QMainWindow" name="InsideWindow">  <*property* name="geometry">  <*rect*>  <*x*>0</*x*>  <*y*>0</*y*>  <*width*>818</*width*>  <*height*>600</*height*>  </*rect*>  </*property*>  <*property* name="windowTitle">  <*string*>银行系统——用户界面</*string*>  </*property*>  <*widget* class="QWidget" name="centralwidget">  <*widget* class="QPushButton" name="pushButton\_vhhu">  <*property* name="geometry">  <*rect*>  <*x*>80</*x*>  <*y*>250</*y*>  <*width*>113</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>创建账户</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label">  <*property* name="geometry">  <*rect*>  <*x*>50</*x*>  <*y*>40</*y*>  <*width*>711</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>欢迎用户xxx，请进行接下来的操作：</*string*>  </*property*>  </*widget*>  <*widget* class="QTextBrowser" name="textBrowser1">  <*property* name="geometry">  <*rect*>  <*x*>40</*x*>  <*y*>300</*y*>  <*width*>481</*width*>  <*height*>241</*height*>  </*rect*>  </*property*>  </*widget*>  <*widget* class="QTextBrowser" name="textBrowser2">  <*property* name="geometry">  <*rect*>  <*x*>560</*x*>  <*y*>300</*y*>  <*width*>231</*width*>  <*height*>241</*height*>  </*rect*>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="id">  <*property* name="geometry">  <*rect*>  <*x*>130</*x*>  <*y*>100</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  </*widget*>  <*widget* class="QCheckBox" name="checkBox">  <*property* name="geometry">  <*rect*>  <*x*>130</*x*>  <*y*>130</*y*>  <*width*>111</*width*>  <*height*>20</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>是否为信用账户</*string*>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="rate">  <*property* name="geometry">  <*rect*>  <*x*>130</*x*>  <*y*>160</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="credit">  <*property* name="geometry">  <*rect*>  <*x*>130</*x*>  <*y*>190</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="fare">  <*property* name="geometry">  <*rect*>  <*x*>130</*x*>  <*y*>220</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_2">  <*property* name="geometry">  <*rect*>  <*x*>60</*x*>  <*y*>100</*y*>  <*width*>60</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>账户名：</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_3">  <*property* name="geometry">  <*rect*>  <*x*>10</*x*>  <*y*>130</*y*>  <*width*>111</*width*>  <*height*>20</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>打钩表示为信用卡</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_4">  <*property* name="geometry">  <*rect*>  <*x*>60</*x*>  <*y*>160</*y*>  <*width*>60</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>利率：</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_5">  <*property* name="geometry">  <*rect*>  <*x*>9</*x*>  <*y*>190</*y*>  <*width*>111</*width*>  <*height*>20</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>透支额度(信用卡)：</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_6">  <*property* name="geometry">  <*rect*>  <*x*>30</*x*>  <*y*>220</*y*>  <*width*>91</*width*>  <*height*>20</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>年费(信用卡)：</*string*>  </*property*>  </*widget*>  <*widget* class="QPushButton" name="pushButton\_cyqm">  <*property* name="geometry">  <*rect*>  <*x*>270</*x*>  <*y*>250</*y*>  <*width*>113</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>存钱</*string*>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="id\_cy">  <*property* name="geometry">  <*rect*>  <*x*>270</*x*>  <*y*>120</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="amount\_cy">  <*property* name="geometry">  <*rect*>  <*x*>270</*x*>  <*y*>170</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="desc\_cy">  <*property* name="geometry">  <*rect*>  <*x*>270</*x*>  <*y*>220</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_7">  <*property* name="geometry">  <*rect*>  <*x*>270</*x*>  <*y*>100</*y*>  <*width*>111</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>存钱账户名：</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_8">  <*property* name="geometry">  <*rect*>  <*x*>270</*x*>  <*y*>150</*y*>  <*width*>111</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>存钱金额：</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_9">  <*property* name="geometry">  <*rect*>  <*x*>270</*x*>  <*y*>200</*y*>  <*width*>111</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>存钱原因：</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_10">  <*property* name="geometry">  <*rect*>  <*x*>410</*x*>  <*y*>150</*y*>  <*width*>111</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>取钱金额：</*string*>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="id\_qu">  <*property* name="geometry">  <*rect*>  <*x*>410</*x*>  <*y*>120</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="amount\_qu">  <*property* name="geometry">  <*rect*>  <*x*>410</*x*>  <*y*>170</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="desc\_qu">  <*property* name="geometry">  <*rect*>  <*x*>410</*x*>  <*y*>220</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_11">  <*property* name="geometry">  <*rect*>  <*x*>410</*x*>  <*y*>200</*y*>  <*width*>111</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>取钱原因：</*string*>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_12">  <*property* name="geometry">  <*rect*>  <*x*>410</*x*>  <*y*>100</*y*>  <*width*>111</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>取钱账户名：</*string*>  </*property*>  </*widget*>  <*widget* class="QPushButton" name="pushButton\_quqm">  <*property* name="geometry">  <*rect*>  <*x*>410</*x*>  <*y*>250</*y*>  <*width*>113</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>取钱</*string*>  </*property*>  </*widget*>  <*widget* class="QLineEdit" name="day\_change">  <*property* name="geometry">  <*rect*>  <*x*>560</*x*>  <*y*>120</*y*>  <*width*>113</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*/>  </*property*>  </*widget*>  <*widget* class="QLabel" name="label\_13">  <*property* name="geometry">  <*rect*>  <*x*>560</*x*>  <*y*>100</*y*>  <*width*>111</*width*>  <*height*>16</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>改变日期(day)：</*string*>  </*property*>  </*widget*>  <*widget* class="QPushButton" name="pushButton\_gdbmriqi">  <*property* name="geometry">  <*rect*>  <*x*>680</*x*>  <*y*>120</*y*>  <*width*>113</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>确认</*string*>  </*property*>  </*widget*>  <*widget* class="QPushButton" name="pushButton\_jbruxxgeyt">  <*property* name="geometry">  <*rect*>  <*x*>560</*x*>  <*y*>160</*y*>  <*width*>113</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>进入下个月</*string*>  </*property*>  </*widget*>  <*widget* class="QPushButton" name="pushButton\_iaxyxbxi">  <*property* name="geometry">  <*rect*>  <*x*>680</*x*>  <*y*>160</*y*>  <*width*>113</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>查询账户信息</*string*>  </*property*>  </*widget*>  <*widget* class="QPushButton" name="pushButton\_iaxyvhmu">  <*property* name="geometry">  <*rect*>  <*x*>620</*x*>  <*y*>250</*y*>  <*width*>113</*width*>  <*height*>31</*height*>  </*rect*>  </*property*>  <*property* name="text">  <*string*>查询时间段账目</*string*>  </*property*>  </*widget*>  <*widget* class="QDateEdit" name="dateEdit1">  <*property* name="geometry">  <*rect*>  <*x*>560</*x*>  <*y*>220</*y*>  <*width*>110</*width*>  <*height*>25</*height*>  </*rect*>  </*property*>  <*property* name="toolTip">  <*string*>&lt;html&gt;&lt;head/&gt;&lt;body&gt;&lt;p&gt;www&lt;/p&gt;&lt;/body&gt;&lt;/html&gt;</*string*>  </*property*>  <*property* name="dateTime">  <*datetime*>  <*hour*>0</*hour*>  <*minute*>0</*minute*>  <*second*>0</*second*>  <*year*>2008</*year*>  <*month*>1</*month*>  <*day*>1</*day*>  </*datetime*>  </*property*>  <*property* name="displayFormat">  <*string*>yyyy-M-d</*string*>  </*property*>  <*property* name="calendarPopup">  <*bool*>true</*bool*>  </*property*>  </*widget*>  <*widget* class="QDateEdit" name="dateEdit2">  <*property* name="geometry">  <*rect*>  <*x*>680</*x*>  <*y*>220</*y*>  <*width*>110</*width*>  <*height*>25</*height*>  </*rect*>  </*property*>  <*property* name="dateTime">  <*datetime*>  <*hour*>0</*hour*>  <*minute*>0</*minute*>  <*second*>0</*second*>  <*year*>2008</*year*>  <*month*>1</*month*>  <*day*>1</*day*>  </*datetime*>  </*property*>  <*property* name="displayFormat">  <*string*>yyyy-M-d</*string*>  </*property*>  <*property* name="calendarPopup">  <*bool*>true</*bool*>  </*property*>  </*widget*>  </*widget*>  <*widget* class="QMenuBar" name="menubar">  <*property* name="geometry">  <*rect*>  <*x*>0</*x*>  <*y*>0</*y*>  <*width*>818</*width*>  <*height*>21</*height*>  </*rect*>  </*property*>  </*widget*>  <*widget* class="QStatusBar" name="statusbar"/>  </*widget*>  <*resources*/>  <*connections*/>  </*ui*>  // md5.h 用来给账户密码加密  #ifndef MD5\_H  #define MD5\_H  #include <string>  #include <fstream>  */\** *Type* *define* *\*/*  *typedef* unsigned char byte;  *typedef* unsigned int uint32;  *using* std::string;  *using* std::ifstream;  */\** *MD5* *declaration.* *\*/*  *class* **MD5**  {  *public*:  **MD5**();  **MD5**( *const* void \*input, size\_t length );  **MD5**( *const* string &str );  **MD5**( ifstream &in );  void **update**( *const* void \*input, size\_t length );  void **update**( *const* string &str );  void **update**( ifstream &in );  *const* byte\* **digest**();  string **toString**();  void **reset**();  *private*:  void **update**( *const* byte \*input, size\_t length );  void **final**();  void **transform**( *const* byte block[64] );  void **encode**( *const* uint32 \*input, byte \*output, size\_t length );  void **decode**( *const* byte \*input, uint32 \*output, size\_t length );  string **bytesToHexString**( *const* byte \*input, size\_t length );  */\** *class* *uncopyable* *\*/*  **MD5**( *const* MD5& );  MD5& *operator*=( *const* MD5& );  *private*:  uint32 \_state[4]; */\** *state* *(ABCD)* *\*/*  uint32 \_count[2]; */\** *number* *of* *bits,* *modulo* *2^64* *(low-order* *word* *first)* *\*/*  byte \_buffer[64]; */\** *input* *buffer* *\*/*  byte \_digest[16]; */\** *message* *digest* *\*/*  bool \_finished; */\** *calculate* *finished* *?* *\*/*  *static* *const* byte PADDING[64]; */\** *padding* *for* *calculate* *\*/*  *static* *const* char HEX[16];  *static* *const* size\_t BUFFER\_SIZE = 1024;  };  #endif*/\*MD5\_H\*/*  //md5.cpp 用来给账户密码加密  #include "md5.h"  *using* *namespace* std;  */\** *Constants* *for* *MD5Transform* *routine.* *\*/*  #define S11 7  #define S12 12  #define S13 17  #define S14 22  #define S21 5  #define S22 9  #define S23 14  #define S24 20  #define S31 4  #define S32 11  #define S33 16  #define S34 23  #define S41 6  #define S42 10  #define S43 15  #define S44 21  */\** *F,* *G,* *H* *and* *I* *are* *basic* *MD5* *functions.*  *\*/*  #define F(x, y, z) (((x) & (y)) | ((~x) & (z)))  #define G(x, y, z) (((x) & (z)) | ((y) & (~z)))  #define H(x, y, z) ((x) ^ (y) ^ (z))  #define I(x, y, z) ((y) ^ ((x) | (~z)))  */\** *ROTATE\_LEFT* *rotates* *x* *left* *n* *bits.*  *\*/*  #define ROTATE\_LEFT(x, n) (((x) << (n)) | ((x) >> (32-(n))))  */\** *FF,* *GG,* *HH,* *and* *II* *transformations* *for* *rounds* *1,* *2,* *3,* *and* *4.*  *Rotation* *is* *separate* *from* *addition* *to* *prevent* *recomputation.*  *\*/*  #define FF(a, b, c, d, x, s, ac) { \  (a) += F ((b), (c), (d)) + (x) + ac; \  (a) = ROTATE\_LEFT ((a), (s)); \  (a) += (b); \  }  #define GG(a, b, c, d, x, s, ac) { \  (a) += G ((b), (c), (d)) + (x) + ac; \  (a) = ROTATE\_LEFT ((a), (s)); \  (a) += (b); \  }  #define HH(a, b, c, d, x, s, ac) { \  (a) += H ((b), (c), (d)) + (x) + ac; \  (a) = ROTATE\_LEFT ((a), (s)); \  (a) += (b); \  }  #define II(a, b, c, d, x, s, ac) { \  (a) += I ((b), (c), (d)) + (x) + ac; \  (a) = ROTATE\_LEFT ((a), (s)); \  (a) += (b); \  }  *const* byte MD5::PADDING[64] = { 0x80 };  *const* char MD5::HEX[16] =  {  '0', '1', '2', '3',  '4', '5', '6', '7',  '8', '9', 'a', 'b',  'c', 'd', 'e', 'f'  };  */\** *Default* *construct.* *\*/*  MD5::**MD5**()  {  reset();  }  */\** *Construct* *a* *MD5* *object* *with* *a* *input* *buffer.* *\*/*  MD5::**MD5**( *const* void \*input, size\_t length )  {  reset();  update( input, length );  }  */\** *Construct* *a* *MD5* *object* *with* *a* *string.* *\*/*  MD5::**MD5**( *const* string &str )  {  reset();  update( str );  }  */\** *Construct* *a* *MD5* *object* *with* *a* *file.* *\*/*  MD5::**MD5**( ifstream &in )  {  reset();  update( *in* );  }  */\** *Return* *the* *message-digest* *\*/*  *const* byte\* MD5::**digest**()  {  *if* ( !\_finished )  {  \_finished = *true*;  final();  }  *return* \_digest;  }  */\** *Reset* *the* *calculate* *state* *\*/*  void MD5::**reset**()  {  \_finished = *false*;  */\** *reset* *number* *of* *bits.* *\*/*  \_count[0] = \_count[1] = 0;  */\** *Load* *magic* *initialization* *constants.* *\*/*  \_state[0] = 0x67452301;  \_state[1] = 0xefcdab89;  \_state[2] = 0x98badcfe;  \_state[3] = 0x10325476;  }  */\** *Updating* *the* *context* *with* *a* *input* *buffer.* *\*/*  void MD5::**update**( *const* void \*input, size\_t length )  {  update( ( *const* byte\* )input, length );  }  */\** *Updating* *the* *context* *with* *a* *string.* *\*/*  void MD5::**update**( *const* string &str )  {  update( ( *const* byte\* )str.c\_str(), str.length() );  }  */\** *Updating* *the* *context* *with* *a* *file.* *\*/*  void MD5::**update**( ifstream &in )  {  *if* ( !in )  *return*;  std::streamsize length;  char buffer[BUFFER\_SIZE];  *while* ( !in.eof() )  {  in.read( *buffer*, BUFFER\_SIZE );  length = in.gcount();  *if* ( length > 0 )  update( buffer, length );  }  in.close();  }  */\** *MD5* *block* *update* *operation.* *Continues* *an* *MD5* *message-digest*  *operation,* *processing* *another* *message* *block,* *and* *updating* *the*  *context.*  *\*/*  void MD5::**update**( *const* byte \*input, size\_t length )  {  uint32 i, index, partLen;  \_finished = *false*;  */\** *Compute* *number* *of* *bytes* *mod* *64* *\*/*  index = ( uint32 )( ( \_count[0] >> 3 ) & 0x3f );  */\** *update* *number* *of* *bits* *\*/*  *if*( ( \_count[0] += ( ( uint32 )length << 3 ) ) < ( ( uint32 )length << 3 ) )  \_count[1]++;  \_count[1] += ( ( uint32 )length >> 29 );  partLen = 64 - index;  */\** *transform* *as* *many* *times* *as* *possible.* *\*/*  *if*( length >= partLen )  {  memcpy( *&\_buffer[index]*, input, partLen );  transform( \_buffer );  *for* ( i = partLen; i + 63 < length; i += 64 )  transform( &input[i] );  index = 0;  }  *else*  {  i = 0;  }  */\** *Buffer* *remaining* *input* *\*/*  memcpy( *&\_buffer[index]*, &input[i], length - i );  }  */\** *MD5* *finalization.* *Ends* *an* *MD5* *message-\_digest* *operation,* *writing* *the*  *the* *message* *\_digest* *and* *zeroizing* *the* *context.*  *\*/*  void MD5::**final**()  {  byte bits[8];  uint32 oldState[4];  uint32 oldCount[2];  uint32 index, padLen;  */\** *Save* *current* *state* *and* *count.* *\*/*  memcpy( *oldState*, \_state, 16 );  memcpy( *oldCount*, \_count, 8 );  */\** *Save* *number* *of* *bits* *\*/*  encode( \_count, *bits*, 8 );  */\** *Pad* *out* *to* *56* *mod* *64.* *\*/*  index = ( uint32 )( ( \_count[0] >> 3 ) & 0x3f );  padLen = ( index < 56 ) ? ( 56 - index ) : ( 120 - index );  update( PADDING, padLen );  */\** *Append* *length* *(before* *padding)* *\*/*  update( bits, 8 );  */\** *Store* *state* *in* *digest* *\*/*  encode( \_state, *\_digest*, 16 );  */\** *Restore* *current* *state* *and* *count.* *\*/*  memcpy( *\_state*, oldState, 16 );  memcpy( *\_count*, oldCount, 8 );  }  */\** *MD5* *basic* *transformation.* *Transforms* *\_state* *based* *on* *block.* *\*/*  void MD5::**transform**( *const* byte block[64] )  {  uint32 a = \_state[0], b = \_state[1], c = \_state[2], d = \_state[3], x[16];  decode( block, *x*, 64 );  */\** *Round* *1* *\*/*  FF ( a, b, c, d, x[ 0], S11, 0xd76aa478 ); */\** *1* *\*/*  FF ( d, a, b, c, x[ 1], S12, 0xe8c7b756 ); */\** *2* *\*/*  FF ( c, d, a, b, x[ 2], S13, 0x242070db ); */\** *3* *\*/*  FF ( b, c, d, a, x[ 3], S14, 0xc1bdceee ); */\** *4* *\*/*  FF ( a, b, c, d, x[ 4], S11, 0xf57c0faf ); */\** *5* *\*/*  FF ( d, a, b, c, x[ 5], S12, 0x4787c62a ); */\** *6* *\*/*  FF ( c, d, a, b, x[ 6], S13, 0xa8304613 ); */\** *7* *\*/*  FF ( b, c, d, a, x[ 7], S14, 0xfd469501 ); */\** *8* *\*/*  FF ( a, b, c, d, x[ 8], S11, 0x698098d8 ); */\** *9* *\*/*  FF ( d, a, b, c, x[ 9], S12, 0x8b44f7af ); */\** *10* *\*/*  FF ( c, d, a, b, x[10], S13, 0xffff5bb1 ); */\** *11* *\*/*  FF ( b, c, d, a, x[11], S14, 0x895cd7be ); */\** *12* *\*/*  FF ( a, b, c, d, x[12], S11, 0x6b901122 ); */\** *13* *\*/*  FF ( d, a, b, c, x[13], S12, 0xfd987193 ); */\** *14* *\*/*  FF ( c, d, a, b, x[14], S13, 0xa679438e ); */\** *15* *\*/*  FF ( b, c, d, a, x[15], S14, 0x49b40821 ); */\** *16* *\*/*  */\** *Round* *2* *\*/*  GG ( a, b, c, d, x[ 1], S21, 0xf61e2562 ); */\** *17* *\*/*  GG ( d, a, b, c, x[ 6], S22, 0xc040b340 ); */\** *18* *\*/*  GG ( c, d, a, b, x[11], S23, 0x265e5a51 ); */\** *19* *\*/*  GG ( b, c, d, a, x[ 0], S24, 0xe9b6c7aa ); */\** *20* *\*/*  GG ( a, b, c, d, x[ 5], S21, 0xd62f105d ); */\** *21* *\*/*  GG ( d, a, b, c, x[10], S22, 0x2441453 ); */\** *22* *\*/*  GG ( c, d, a, b, x[15], S23, 0xd8a1e681 ); */\** *23* *\*/*  GG ( b, c, d, a, x[ 4], S24, 0xe7d3fbc8 ); */\** *24* *\*/*  GG ( a, b, c, d, x[ 9], S21, 0x21e1cde6 ); */\** *25* *\*/*  GG ( d, a, b, c, x[14], S22, 0xc33707d6 ); */\** *26* *\*/*  GG ( c, d, a, b, x[ 3], S23, 0xf4d50d87 ); */\** *27* *\*/*  GG ( b, c, d, a, x[ 8], S24, 0x455a14ed ); */\** *28* *\*/*  GG ( a, b, c, d, x[13], S21, 0xa9e3e905 ); */\** *29* *\*/*  GG ( d, a, b, c, x[ 2], S22, 0xfcefa3f8 ); */\** *30* *\*/*  GG ( c, d, a, b, x[ 7], S23, 0x676f02d9 ); */\** *31* *\*/*  GG ( b, c, d, a, x[12], S24, 0x8d2a4c8a ); */\** *32* *\*/*  */\** *Round* *3* *\*/*  HH ( a, b, c, d, x[ 5], S31, 0xfffa3942 ); */\** *33* *\*/*  HH ( d, a, b, c, x[ 8], S32, 0x8771f681 ); */\** *34* *\*/*  HH ( c, d, a, b, x[11], S33, 0x6d9d6122 ); */\** *35* *\*/*  HH ( b, c, d, a, x[14], S34, 0xfde5380c ); */\** *36* *\*/*  HH ( a, b, c, d, x[ 1], S31, 0xa4beea44 ); */\** *37* *\*/*  HH ( d, a, b, c, x[ 4], S32, 0x4bdecfa9 ); */\** *38* *\*/*  HH ( c, d, a, b, x[ 7], S33, 0xf6bb4b60 ); */\** *39* *\*/*  HH ( b, c, d, a, x[10], S34, 0xbebfbc70 ); */\** *40* *\*/*  HH ( a, b, c, d, x[13], S31, 0x289b7ec6 ); */\** *41* *\*/*  HH ( d, a, b, c, x[ 0], S32, 0xeaa127fa ); */\** *42* *\*/*  HH ( c, d, a, b, x[ 3], S33, 0xd4ef3085 ); */\** *43* *\*/*  HH ( b, c, d, a, x[ 6], S34, 0x4881d05 ); */\** *44* *\*/*  HH ( a, b, c, d, x[ 9], S31, 0xd9d4d039 ); */\** *45* *\*/*  HH ( d, a, b, c, x[12], S32, 0xe6db99e5 ); */\** *46* *\*/*  HH ( c, d, a, b, x[15], S33, 0x1fa27cf8 ); */\** *47* *\*/*  HH ( b, c, d, a, x[ 2], S34, 0xc4ac5665 ); */\** *48* *\*/*  */\** *Round* *4* *\*/*  II ( a, b, c, d, x[ 0], S41, 0xf4292244 ); */\** *49* *\*/*  II ( d, a, b, c, x[ 7], S42, 0x432aff97 ); */\** *50* *\*/*  II ( c, d, a, b, x[14], S43, 0xab9423a7 ); */\** *51* *\*/*  II ( b, c, d, a, x[ 5], S44, 0xfc93a039 ); */\** *52* *\*/*  II ( a, b, c, d, x[12], S41, 0x655b59c3 ); */\** *53* *\*/*  II ( d, a, b, c, x[ 3], S42, 0x8f0ccc92 ); */\** *54* *\*/*  II ( c, d, a, b, x[10], S43, 0xffeff47d ); */\** *55* *\*/*  II ( b, c, d, a, x[ 1], S44, 0x85845dd1 ); */\** *56* *\*/*  II ( a, b, c, d, x[ 8], S41, 0x6fa87e4f ); */\** *57* *\*/*  II ( d, a, b, c, x[15], S42, 0xfe2ce6e0 ); */\** *58* *\*/*  II ( c, d, a, b, x[ 6], S43, 0xa3014314 ); */\** *59* *\*/*  II ( b, c, d, a, x[13], S44, 0x4e0811a1 ); */\** *60* *\*/*  II ( a, b, c, d, x[ 4], S41, 0xf7537e82 ); */\** *61* *\*/*  II ( d, a, b, c, x[11], S42, 0xbd3af235 ); */\** *62* *\*/*  II ( c, d, a, b, x[ 2], S43, 0x2ad7d2bb ); */\** *63* *\*/*  II ( b, c, d, a, x[ 9], S44, 0xeb86d391 ); */\** *64* *\*/*  \_state[0] += a;  \_state[1] += b;  \_state[2] += c;  \_state[3] += d;  }  */\** *Encodes* *input* *(ulong)* *into* *output* *(byte).* *Assumes* *length* *is*  *a* *multiple* *of* *4.*  *\*/*  void MD5::**encode**( *const* uint32 \*input, byte \*output, size\_t length )  {  *for*( size\_t i = 0, j = 0; j < length; i++, j += 4 )  {  output[j] = ( byte )( input[i] & 0xff );  output[j + 1] = ( byte )( ( input[i] >> 8 ) & 0xff );  output[j + 2] = ( byte )( ( input[i] >> 16 ) & 0xff );  output[j + 3] = ( byte )( ( input[i] >> 24 ) & 0xff );  }  }  */\** *Decodes* *input* *(byte)* *into* *output* *(ulong).* *Assumes* *length* *is*  *a* *multiple* *of* *4.*  *\*/*  void MD5::**decode**( *const* byte \*input, uint32 \*output, size\_t length )  {  *for*( size\_t i = 0, j = 0; j < length; i++, j += 4 )  {  output[i] = ( ( uint32 )input[j] ) | ( ( ( uint32 )input[j + 1] ) << 8 ) |  ( ( ( uint32 )input[j + 2] ) << 16 ) | ( ( ( uint32 )input[j + 3] ) << 24 );  }  }  */\** *Convert* *byte* *array* *to* *hex* *string.* *\*/*  string MD5::**bytesToHexString**( *const* byte \*input, size\_t length )  {  string str;  str.reserve( length << 1 );  *for*( size\_t i = 0; i < length; i++ )  {  int t = input[i];  int a = t / 16;  int b = t % 16;  str.append( 1, HEX[a] );  str.append( 1, HEX[b] );  }  *return* str;  }  */\** *Convert* *digest* *to* *string* *value* *\*/*  string MD5::**toString**()  {  *return* bytesToHexString( digest(), 16 );  } |

1. 测试
2. 测试用例1:在登陆页面输入账号：wgc 密码：123456点击注册两次再点击登陆。

输出截图：

图形用户界面, 应用程序

描述已自动生成

分析：点击第二次时，由于wgc这个账户已经创建过了，所以不能再注册，弹出提示框。

图形用户界面

描述已自动生成

分析：点击登陆，wgc成功登陆，显示用户页面。

1. 测试用例2：账户名：wgc1，不是信用账户，利率为0.015

输出截图：

图形用户界面, 应用程序

描述已自动生成

分析：如图所示，创建了wgc1的储蓄账户。

1. 测试用例3:账户名：wgc2 是信用账户，利率为0.1 透支额度114514元，年费为100元

输出截图：

图形用户界面, 应用程序

描述已自动生成

分析：如图所示，wgc2账号已创建

1. 测试用例4:再输入一个同名的wgc1账号，点击“创建账户”。

输出截图：

图形用户界面, 应用程序

描述已自动生成

分析：由于系统中有重名账户，弹出提示框

1. 测试用例5:存钱账户名：wgc1，存钱金额114514，存钱原因：hello，点击存钱。

输出截图：

图形用户界面

描述已自动生成

分析：如图。

1. 测试用例6:存钱账户名：wgc1，存钱金额：23333，存钱原因：world。点击存钱

图形用户界面

描述已自动生成

分析：如图。

1. 测试用例7:从wgc1取4000元，原因，buy a company

输出截图：

图形用户界面

描述已自动生成

分析：如图。

1. 测试用例8:从wgc2账户取500000元，取钱原因：233

输出截图：

图形用户界面, 网站

描述已自动生成

分析：账户余额加上透支额度不够取的金额，弹出提示框。

1. 测试用例9:点击“查询账户信息”

输出截图：

**图形用户界面

描述已自动生成**

分析：如图。

1. 测试用例10:点击进入下个月：

输出截图：

**图形用户界面

描述已自动生成**

分析：进入下个月后结算了上个月的利息，进入下一年后，扣除了信用卡的年费。

1. 测试用例11:2008-1-1～2009-12-12 点击查询时间段账目

输出截图：



分析：如图所示

【功能4：图形界面有当月还款的提示和收入支出统计】

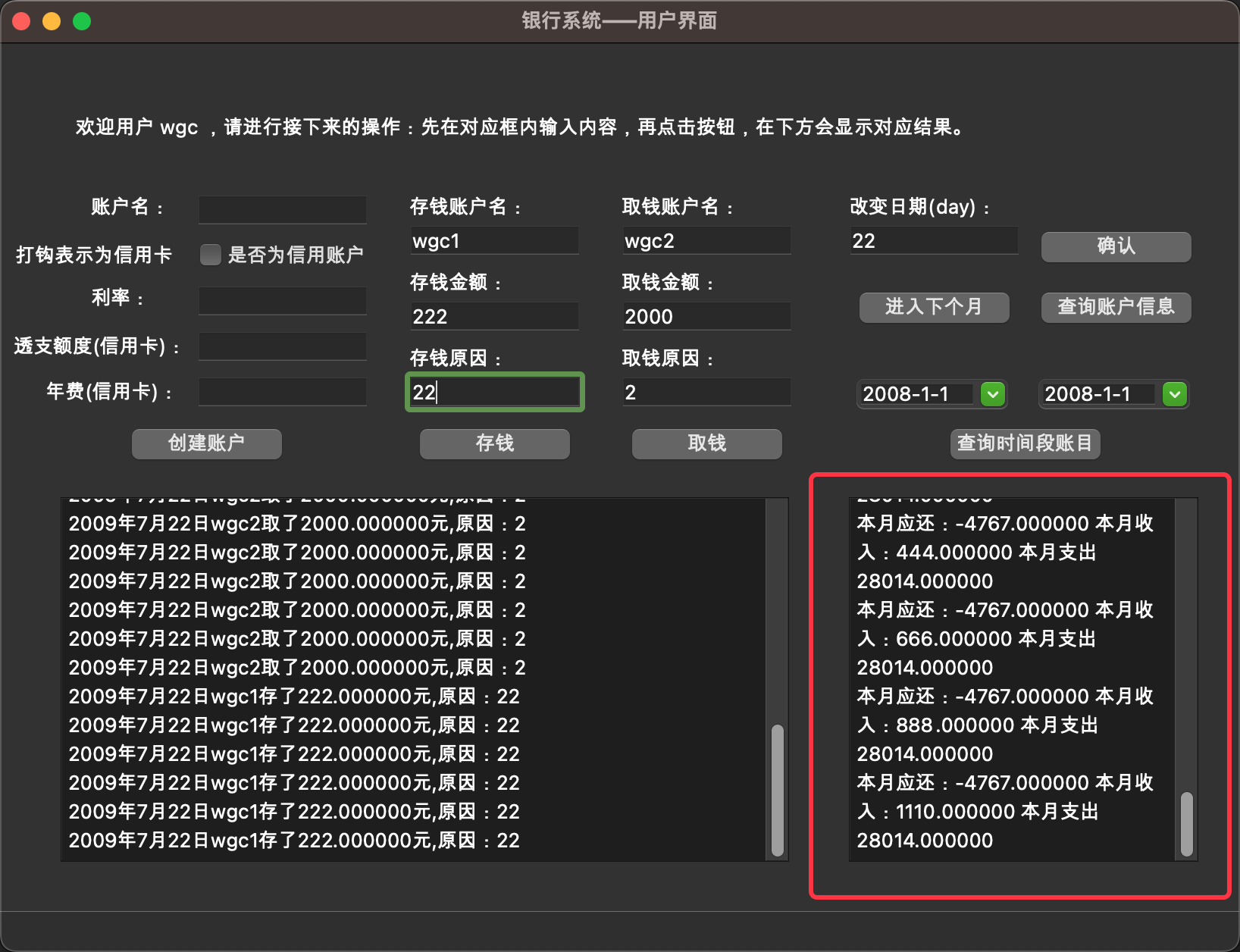
1. 设计

在右侧框显示本月支出和本月收入，用一个函数来统计，并且调用qt的方法来显示在文本框上。

1. 实现

|  |
| --- |
| 1. void InsideWindow::**jisrBro2**(){ 2. double ykhr = 0; 3. double uzru = 0, viiu = 0; 4. *for*(unsigned int i=0; i < accounts.size(); i++){ 5. *if*(accounts[i]->getUser() == MainWindow::Log\_on\_User){ 6. *if*(accounts[i]->*s\_or\_c*()){*//信用卡* 7. ykhr += accounts[i]->*getDebt*(); 8. } 9. } 10. } 11. *//cout* *<<* *date.getYear()<<date.getMonth()-1<<1* 12. uzru = Account::iaxy\_uzru(Date(date.getYear(),date.getMonth(),1),Date(date.getYear(),date.getMonth(),date.getMaxDay()),MainWindow::Log\_on\_User); 13. viiu = Account::iaxy\_viiu(Date(date.getYear(),date.getMonth(),1),Date(date.getYear(),date.getMonth(),date.getMaxDay()),MainWindow::Log\_on\_User); 14. iw->ui->textBrowser2->append(QString::fromStdString("本月应还："+to\_string(ykhr)+" 本月收入："+to\_string(uzru)+" 本月支出"+to\_string(viiu))); 15. } |

3.测试



分析：如图所示。

【功能5：可以查询某个月按照时间排序的账户查询信息】

1. 设计：与银行1～6的类似，但是时间输入改为了从qt提供的时间选择控件输入，然后点击查询时间段账目，即可。



1. 实现

|  |
| --- |
| 1. void InsideWindow::**on\_pushButton\_iaxyvhmu\_clicked**() 2. { 3. q1 = Date(ui->dateEdit1->date().year(), ui->dateEdit1->date().month(), ui->dateEdit1->date().day()); 4. q2 = Date(ui->dateEdit2->date().year(), ui->dateEdit2->date().month(), ui->dateEdit2->date().day()); 5. InsideWindow::backMain('q',*false*); 6. } |

1. 测试：查询了刚刚的7月的信息；

输出截图：

图形用户界面

描述已自动生成

分析：如图所示。

**实验结论与感想**

这个程序在macOS上是有黑暗模式切换的，并且在Windows下也是可以编译运行的。

通过本次实验，我对C++的类的运用有了更深刻的认识。同时学到了一小部分的qt的图形页面的编写和制作。同时对于我自己的搜索引擎搜索能力有了进一步的提升，会更加懂得自己去上网搜索对于自己有用的功能添加进来，来学习更多的知识。