**ATM code**

**对数据库操作的类：**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Data.SqlClient;

using System.Data;

using System.Configuration; namespace ATM1 { /\*对users表进行操作的服务类\*/

class UsersService{

private static UsersService userService=null;

private UsersService(){}

public static UsersService getService(){

if (userService == null)

{//如果是第一次使用方法则实例化

userService = new UsersService(); }

return userService; }\* //得到连接

private void openConnection() {

if (conn == null)

{ conn = new SqlConnection(ConfigurationManager.ConnectionStrings["ATM"].ToString()); //conn = new SqlConnection(conStr); conn.Open(); } } /\*\* \* 设置连接命令\* command sql语句\* isQuery 是否是查询语句，是则不开启业务

\* 增删改则开启业\*/

private void setCommand(string command,bool isQuery) {

this.openConnection();

comm = new SqlCommand();

if (!isQuery)

{

tran = conn.BeginTransaction();

comm.Transaction = tran;

}

comm.CommandText = command;

comm.CommandType = CommandType.Text;

comm.Connection = conn;

}

/\* \*设置命令参数\*from 要替换的字符串\*to 替换为的字符串\*/ private void setParameter(string from,object to)

{

comm.Parameters.AddWithValue(from, to);

}

/\*\*\*提交结果提交成功 reture true \*提交失败 return false 并回滚\*对添，删，改有效 \*/

private bool commitCommand()

{

try

{

int numEffected;

numEffected= comm.ExecuteNonQuery();

if (numEffected >= 1)

{ //添加成功，提交

tran.Commit();

return true; }

else

{ //添加失败回滚 tran.Rollback();} }

catch (Exception e) {

Console.WriteLine(e.StackTrace); //出现异常回滚 tran.Rollback(); }

finally

{

tran.Dispose(); this.closeConnection(); }

return false; } /\*\* \*关闭连接\*/

private void closeConnection()

{

if (conn != null)

{

conn.Close();

conn.Dispose();

conn = null; } }

/\*\*\* 登陆\* \* 判断输入用户 不需要开启事务\* id 输入卡号\* passWd 密码\* 没有该用户return false\* 存在则 return true\* \*/

public bool verifyUser(string id,string passwd) {

this.setCommand("select count(\*) from users where cardNum=@cardNum and passWd=@passWd",true);

this.setParameter("@cardNum",id); this.setParameter("@passWd",passwd);

int num = (int)this.comm.ExecuteScalar();

if (1 == num)

{

return true;

}

return false;

}

/\*\*\* 开户 \* 创建用户服务 需要开启事务 \* user 要添加的用户 \* return \* true 添加成功 \* false 添加失败 \*/ public bool addUser(UserBean user)

{

this.setCommand("insert into users values(@cardNum,@passWd,@userName,@moneySum,@cardBalance,@identifyCardNum)",false);

this.setParameter("@cardNum",user.cardNum); this.setParameter("@passWd", user.passWd); this.setParameter("@userName", user.userName); this.setParameter("@moneySum", user.moneySum); this.setParameter("@cardBalance", user.cardBalance); this.setParameter("@identifyCardNum", user.identifyCardNum); return this.commitCommand(); }

/\*\*\* 改密\* \* 更改密码服务 需要开启事务\* id 要修改的卡号\* return true 修改成功\* reture fasle 修改失败\*/

public bool setPassWd(string id,string newPasswd)

{

this.setCommand("update users set passWd=@passWd where cardNum=@id",false);

this.setParameter("@passWd", newPasswd); this.setParameter("@id",id);

return this.commitCommand();

}

/\*\* \* 查询金额\* id 卡号 \* return 查询结果 \* \*/ public object getCardBalance(string id) {

this.setCommand("select cardBalance from users where cardNum=@cardNum",true);

this.setParameter("@cardNum",id);

return comm.ExecuteScalar();

}

/\*\*\* 改变金额(包括取现和存款) \* id 卡号 \* double 增加或减少的金额 \* 成功返回 tru \* 失败返回 false\*

public bool changeBalance(string id, decimal balance)

{//持卡人卡上金额

decimal ownerBalance; try

{ ownerBalance= (decimal)this.getCardBalance(id);

}

catch (Exception e)

{

Console.WriteLine(e.StackTrace);

return false; }

//如果转移的金额大于已有的金额，则转账失败

if ((ownerBalance+balance)<0)

{

return false; }

this.setCommand("update users set cardBalance=@balance where cardnum=@id", false); this.setParameter("@balance",(ownerBalance + balance)); this.setParameter("@id",id);

return this.commitCommand();

}

/\*\* 转账 \* from\_id 持卡人卡号\* to\_id 转账到的卡号 \* transferMenoy 转账金额\* 成功返回 true失败返回 false \*/ public bool transferBalance(string from\_id,string to\_id,decimal transferMenoy)

{

decimal ownerBalance =(decimal)this.getCardBalance(from\_id); //如果转移的金额大于已有的金额，或卡内金额小于100 则转账失败 if (transferMenoy > ownerBalance && ownerBalance<100) { return false; }

decimal otherBalance = (decimal)this.getCardBalance(to\_id); this.setCommand("update users set cardBalance=@fromOwonerBalance where cardNum=@from\_id;

update users set cardbalance=@toOwnerBalance where cardNum=@to\_id;", false);

this.setParameter("@from\_id", from\_id); this.setParameter("@to\_id", to\_id); this.setParameter("@fromOwonerBalance",(ownerBalance - transferMenoy)); this.setParameter("@toOwnerBalance", (otherBalance+transferMenoy)); return this.commitCommand(); }

private SqlConnection conn;

private SqlCommand comm;

private SqlDataReader reader;

private SqlTransaction tran;

private const string conStr = "server=172.7.41.166;database=ATM;uid=sa;pwd=123"; } }