武汉纺织大学

Java应用开发课程设计

**采用MyBatis框架实现**

**基于DAO设计模式的银行系统**

**学 院： 数学与计算机学院**

**班 级： 物联网11801**

**姓 名： 孙思萌**

**学 号： 1804240132**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年6月20日**

目 录

[一、需求分析 1](#_Toc43580102)

[1.1用户登陆 1](#_Toc43580103)

[1.2主菜单 1](#_Toc43580104)

[1.3各项功能 1](#_Toc43580105)

[1.3.1 锁定 1](#_Toc43580106)

[1.3.2 开户 1](#_Toc43580107)

[1.3.3 销户 1](#_Toc43580108)

[1.3.4 挂失 1](#_Toc43580109)

[1.3.5 解挂 1](#_Toc43580110)

[1.3.6 用户信息维护 1](#_Toc43580111)

[1.3.7 存款 1](#_Toc43580112)

[1.3.8 取款 1](#_Toc43580113)

[1.3.9 转账 1](#_Toc43580114)

[1.3.10 查询信息 2](#_Toc43580115)

[1.3.11 修改密码 2](#_Toc43580116)

[1.3.12 贷款 2](#_Toc43580117)

[1.3.13 还款 2](#_Toc43580118)

[二、系统实现 3](#_Toc43580119)

[2.1 项目结构 3](#_Toc43580120)

[2.2 配置文件 3](#_Toc43580121)

[2.2.1 jdbc.properties文件 3](#_Toc43580122)

[2.2.2 log4j.properties文件 4](#_Toc43580123)

[2.2.3 mybatis-config.xml文件 4](#_Toc43580124)

[2.3 VO类User.java 5](#_Toc43580125)

[2.4 DAO接口类IUserDAO.java 6](#_Toc43580126)

[2.5 接口映射文件UserMapper.xml 7](#_Toc43580127)

[2.6 工具包Util 7](#_Toc43580128)

[2.6.1 MybatisUtils.java 7](#_Toc43580129)

[2.6.2 MD5Util.java 8](#_Toc43580130)

[2.7 服务层UserService.java 9](#_Toc43580131)

[2.8 用户界面Driver.java 12](#_Toc43580132)

[三、系统总结 13](#_Toc43580133)

# 一、需求分析

设计一个银行系统，实现“登陆”、“锁定”、“开户”、“销户”、“挂失”、“解挂”、“用户信息维护”、“存款”、“取款”、“转账”、“查询信息”、“修改密码”、“贷款”、“还款”功能，具体要求如下：

## 1.1用户登陆

输入银行卡号和密码，输入正确之后进入银行管理系统

## 1.2主菜单

当用户名及密码正确时，进入系统主界面，如下：

1. 开户
2. 销户
3. 挂失
4. 解挂
5. 用户信息维护
6. 退出

输入序号：

如果输入1-6以外的选择，则显示提示信息“输入无效，只能输入1-6”后，返回主界面。

## 1.3各项功能

### 1.3.1 锁定

当用户输入错误三次之后，锁定账户。

### 1.3.2 开户

输入身份证号，密码，开通什么卡，身份是什么。然后在数据库中创建新的内容。

### 1.3.3 销户

输入身份证号和银行卡号，如果都是对的，那就在数据库中删除对应的记录。

### 1.3.4 挂失

输入身份证号和银行卡号，如果都是对的，那就在数据库中将对应记录的锁定。

### 1.3.5 解挂

输入身份证号和银行卡号，如果都是对的，那就在数据库中将对应记录的取消锁定。

### 1.3.6 用户信息维护

可以选择从excel表格导入数据，或者txt文件导入数据，或者直接键盘输入

### 1.3.7 存款

根据输入的金额，增加数据库中的数值。

### 1.3.8 取款

根据输入的金额，减少数据库中的数值。

### 1.3.9 转账

根据输入的金额和银行卡号，增加对应卡的数值。

### 1.3.10 查询信息

根据输入的银行卡号，查询数据库中对应的内容，显示出来。

### 1.3.11 修改密码

输入原密码，访问数据库，对比密码正不正确，如果正确，让用户输入新密码，然后修改数据库中密码为新密码。

### 1.3.12 贷款

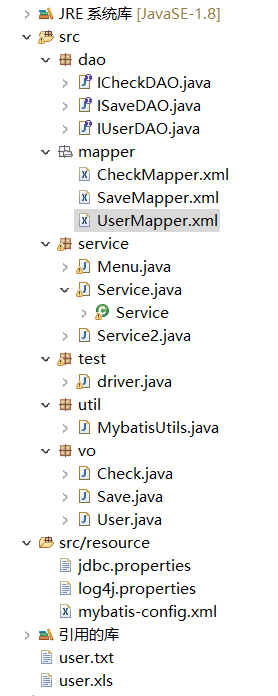
根据输入的金额，减少数据库中的数值。

### 1.3.13 还款

根据输入的金额，增加数据库中的数值。

# 二、系统实现

## 2.1 项目结构



## 2.2 配置文件

在项目下新建Source Folder，名为resources

### 2.2.1 jdbc.properties文件

该配置文件主要是以文件形式保存数据库的驱动类名称，连接数据库的URL地址，访问数据库的用户名及对应的密码，程序运行时会读取该文件相关信息，避免硬编码，当相关信息发生变化时，只需修改配置文件而不用修改源代码，增加程序的可扩展型。

DBDRIVER=com.mysql.jdbc.Driver

DBURL=jdbc:mysql://localhost:3306/bank?useUnicode=true&characterEncoding=utf-8&useSSL=false

DBUSER=root

PASSWORD=ssm666666

### 2.2.2 log4j.properties文件

日志配置

log4j.rootLogger=DEBUG, stdout

log4j.appender.stdout=org.apache.log4j.ConsoleAppender

log4j.appender.stdout.layout=org.apache.log4j.PatternLayout

log4j.appender.stdout.layout.ConversionPattern=%5p [%t] - %m%n

### 2.2.3 mybatis-config.xml文件

核心配置文件

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE configuration

PUBLIC "-//mybatis.org//DTD Config 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-config.dtd">

<!-- 根标签 -->

<configuration>

<!-- 指定资源文件 -->

<properties resource="jdbc.properties"></properties>

<!-- 设置Mybatis的log实现为LOG4J -->

<settings>

<setting name="logImpl" value="LOG4J" />

</settings>

<!-- 环境，可以配置多个，default：指定采用哪个环境 -->

<environments default="test">

<!-- id：唯一标识 -->

<environment id="test">

<!-- 事务管理器，JDBC类型的事务管理器 -->

<transactionManager type="JDBC" />

<!-- 数据源，池类型的数据源 -->

<dataSource type="POOLED">

<property name="driver" value="${DBDRIVER}" />

<property name="url" value="${DBURL}" />

<property name="username" value="${DBUSER}" />

<property name="password" value="${PASSWORD}" />

</dataSource>

</environment>

</environments>

<!-- 配置mapper映射文件 -->

<mappers>

<mapper resource="mapper/CheckMapper.xml" />

<mapper resource="mapper/SaveMapper.xml" />

<mapper resource="mapper/UserMapper.xml" />

</mappers>

</configuration>

## 2.3 VO类User.java

对应数据库中的表

public class User {

private String id;

private String number;

private String passwords;

private String locks;

private String type;

private String operator;

public User(String id, String number, String passwords, String locks, String type, String operator) {

super();

this.id = id;

this.number = number;

this.passwords = passwords;

this.locks = locks;

this.type = type;

this.operator = operator;

}

public User() {

super();

}

public String getId() {

return id;

}

public void setId(String id) {

this.id = id;

}

public String getnumber() {

return number;

}

public void setnumber(String number) {

this.number = number;

}

public String getPasswords() {

return passwords;

}

public void setPasswords(String passwords) {

this.passwords = passwords;

}

public String getLocks() {

return locks;

}

public void setLocks(String locks) {

this.locks = locks;

}

public String getType() {

return type;

}

public void setType(String type) {

this.type = type;

}

public String getOperator() {

return operator;

}

public void setOperator(String operator) {

this.operator = operator;

}

@Override

public String toString() {

return "User [id=" + id + ", number=" + number + ", passwords=" + passwords + ", locks=" + locks + ", type="

+ type + ", operator=" + operator + "]";

}

}

## 2.4 DAO接口类IUserDAO.java

提供一个接口，然后后面用Mapper具体实现

public interface IUserDAO {

public boolean Insert(User user) throws Exception;

public boolean Update(User user) throws Exception;

public boolean Delete(String cardnum) throws Exception;

public ArrayList<User> Select(String attribute,String content) throws Exception;

}

## 2.5 接口映射文件UserMapper.xml

接口的具体实现

<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE mapper

PUBLIC "-//mybatis.org//DTD Mapper 3.0//EN"

"http://mybatis.org/dtd/mybatis-3-mapper.dtd">

<mapper namespace="dao.IUserDAO">

<insert id="Insert" parameterType="vo.User">

insert into user(id,number,passwords,locks,type,operator)

values(#{id},#{number},#{passwords},#{locks},#{type},#{operator})

</insert>

<delete id="Delete">

delete from user where number=#{number}

</delete>

<update id="Update" parameterType="vo.User">

update user

<set>

<if test="id != null">id=#{id},</if>

<if test="passwords != null">passwords=#{passwords},</if>

<if test="locks != null">locks=#{locks},</if>

<if test="type != null">type=#{type},</if>

<if test="operator != null">operator=#{operator}</if>

</set>

where number=#{number}

</update>

<select id="Select" resultType="vo.User">

select \* from user where ${arg0} like "%"#{arg1}"%"

</select>

</mapper>

## 2.6 工具包Util

### 2.6.1 MybatisUtils.java

统一管理数据库的连接，创建会话工厂

/\*\*

\* 为服务层提供sqlSession对象

\*/

**public** **class** MybatisUtils {

**private** **static** SqlSessionFactory *sqlSessionFactory*;

**public** MybatisUtils() {

**super**();

}

/\*

\* SqlSessionFactory一旦被创建,应该在应用执行期间都存在.在应用运行期间不要重复创建多次

\*/

**static** {

String resourse = "mybatis-config.xml";

**try** {

InputStream inputStream = Resources.*getResourceAsStream*(resourse);

*sqlSessionFactory* = **new** SqlSessionFactoryBuilder().build(inputStream);

} **catch** (IOException e) {

e.printStackTrace();

}

}

**public** **static** SqlSession getSqlSession() {

// 如果不设置参数或者参数为false就是手动提交事务，参数设置为true就是自动提交事务

SqlSession sqlSession = *sqlSessionFactory*.openSession();

**return** sqlSession;

}

}

### 2.6.2 MD5Util.java

该类主要实现对数据库中的用户密码进行MD5加密，描述该文件作用，并对里面的关键代码进行解释

**public** **class** MD5Util {

**public** **static** String md5(String str) {

**byte**[] digest = **null**;

**try** {

MessageDigest md5 = MessageDigest.*getInstance*("md5");

digest = md5.digest(str.getBytes("utf-8"));

} **catch** (NoSuchAlgorithmException e) {

e.printStackTrace();

} **catch** (UnsupportedEncodingException e) {

e.printStackTrace();

}

//16是表示转换为16进制数

String md5Str = **new** BigInteger(1, digest).toString(16);

**return** md5Str;

}

**public** **static** **void** main(String[] args) {

System.***out***.println(*md5*("admin"));

}

}

## 2.7 服务层UserService.java

Service里面是所有的业务逻辑的具体实现

public class Service2 {

static Scanner scan = new Scanner(System.in);

public static void changePassword(User user) throws Exception{

String old = null;

String news = null;

System.out.println("输入原密码：");

old = scan.next();

while(!old.equals(user.getPasswords())) {

System.out.println("密码错误，重新输入：");

old = scan.next();

}

if(old.equals(user.getPasswords())) {

System.out.println("请输入新密码：");

news = scan.next();

}

Service.changePassword(user, news);

}

public static void close() throws Exception {

String number = null;

String id = null;

User user = new User(id, number, null, "0", null,null);

System.out.println("输入卡号：");

number = scan.next();

System.out.println("输入身份证号：");

id = scan.next();

Service.closeAccount(user);

}

public static void cunkuan(User user) throws Exception {

System.out.println("输入存款金额：");

double money = scan.nextDouble();

Service.deposit(user, money);

}

public static void find() throws Exception {

System.out.println("输入银行卡号：");

String number = scan.next();

System.out.println("输入身份证号：");

String id = scan.next();

User user = new User(id, number, null, "0", null,null);

while(!Service.find(user)) {

System.out.println("银行卡号或身份证号错误，重新输入：");

System.out.println("输入银行卡号：");

number = scan.next();

System.out.println("输入身份证号：");

id = scan.next();

}

}

public static void loan(User user) throws Exception {

System.out.println("输入贷款金额：");

double money = scan.nextDouble();

if(!Service.loan(user, money)) {

System.out.println("额度不足");

}

}

public static void lock(User user) throws Exception{

if(Service.Lock(user.getnumber()))

System.out.println("银行卡已锁定");

else

System.out.println("卡号错误");

System.exit(-1);

}

public static User login() throws Exception{

String number = null;

String password = null;

User user = new User();

System.out.println("输入卡号：");

number = scan.next();

System.out.println("输入密码：");

password = scan.next();

user.setnumber(number);

for(int i = 0; i <3; i++) {

if(Service.Login(number, password)!=null) {

user = Service.Login(number, password);

if(user.getLocks().equals("1")) {

user = null;

System.out.println("被锁定，无法登陆");

}

}

else {

System.out.println("密码错误，重新输入:");

password = scan.next();

}

}

return user;

}

public static void lose() throws Exception{

String number =null;

String id = null;

User user = new User(id, number, null, "0", null,null);

System.out.println("输入银行卡号：");

number = scan.next();

System.out.println("输入身份证号：");

id = scan.next();

Service.loss(user);

}

public static void open() throws Exception {

String ID= null;

String password = null;

String type = null;

Double lines = 0.0;

String number = null;

String operator = null;

User user = null;

System.out.println("输入身份证：");

ID = scan.next();

System.out.println("设置密码：");

password = scan.next();

System.out.println("开通储蓄卡还是信用卡：");

type = scan.next();

if(type.equals("信用卡")) {

System.out.println("请设置额度：");

lines = scan.nextDouble();

}

java.util.Date date = new java.util.Date();

SimpleDateFormat simpleDateFormat = new SimpleDateFormat("yyyyMMddhhMMss");

String str = simpleDateFormat.format(date);

number = "66"+str;

System.out.println("是客户还是工作人员：");

operator = scan.next();

user = new User(ID, number, password, "0", type,operator);

Service.openAccount(user,lines);

}

public static void query(User user) throws Exception{

Service.query(user);

}

public static void repay(User user) throws Exception{

double money = 0;

System.out.println("输入还款金额：");

money = scan.nextDouble();

Service.repay(user, money);

}

public static void transfer() throws Exception {

String number = null;

double money = 0;

System.out.println("输入转账卡号：");

number = scan.next();

System.out.println("输入转账的金额：");

money = scan.nextDouble();

Service.transfer(number, money);

}

public static void withdraw(User user) throws Exception{

double money = 0;

System.out.println("输入取款金额：");

money = scan.nextDouble();

Service.withdraw(user, money);

}

}

## 2.8 用户界面Driver.java

运行程序

public class driver {

public static void main(String[] args) throws Exception {

Service service = new Service();

User user = Service2.login();

if(user.getPasswords()==null)

Service2.lock(user);

else {

if(user.getOperator().equals("工作人员"))

Menu.stuff(user);

else if(user.getType().equals("信用卡"))

Menu.check(user);

else

Menu.save(user);

}

}

}

# 系统总结

时间过得太快，没有想到最后以这样一种方式结束了整个学期的学习。说实话，这学期采用的线上教学方式，有利有弊，好的一点是我们有了更多自主学习的机会，可随之而来的就是对每个人自觉性的考验，不少人抓住了这个机会，学到了很多东西，而也有很大一部分人无法在家中很好地管理自己的时间，我在想，我究竟是前者还是后者呢？可能我并没有能够很好地管理自己，但还是学到一些东西吧。

其实这次的课程设计对于我来说难度还是比较大的，因为在家中我无法保证自己全神贯注地听课，完成作业也不够积极，最后在写课设的时候就遇到了很多问题，问了不少学的比较好的同学，查了很多才能完成。自评成绩的话，给自己一个合格吧！

最后还是要说一句，老师您辛苦了！在这种特殊时期，以这样的方式学习，这无疑对老师和学生都是极大的挑战。感谢聂老师对大家的付出，以后还是要跟着聂老师继续学习深造的！