**面向对象技术（Java）大作业二**

**飞马星球飞马银行微系统设计与开发**

**（此作业文档仅做参考说明，大家可以试作飞马图书馆项目、飞马宾馆项目、飞马车管所项目等等）**

1. **场景概述**



IMG_256

在我们的飞马星球上，飞马人手中使用了一种飞马币（纸币）进行商业往来和货物交易，为了规范和方便流通使用，飞马星球设置了一个飞马银行，为大家提供个人金融服务，这些服务包括但不限于以下服务：

1. 开户，年满18岁的飞马人可以申请开设个人账号，飞马银行会存入2000飞马币作为星球福利赠送给每个新用户；
2. 销户，对于年满70岁的老人强制销户，或者因故离世的人员，经申请可以销户处理。
3. 查询余额，用户可以自由查询自己账户中的余额。
4. 取钱，用户可以支取账户上的余额，但支取后要保证账户余额必须大于等于0.
5. 存钱，用户可以往账号中存钱。
6. 转账，用户可以给另一个存在账号转钱，同样，要保证账户余额大于等于0的要求。
7. 修改账户信息，但账户id，身份id，账户余额等信息不允许修改。

针对银行管理，我们还要提供以下功能：

1. 可以导入xls文件进行批量开户
2. 可以将特定查询信息导出为xls文件
3. 年终可以生成一个pdf版综述报告，报告内容包括但不限于目前账户总数，今年新开户数，总余额等数据。

现在要求大家使用面向对象理论、网络理论、数据库技术、GUI技术（\*）等理论和技术开发一套微系统用来完成这些服务。简化一下系统原型，我们开发的系统主要操作用户为柜台人员，假设用户密码都可以通过特定键盘输入。约定数据都存储在飞马银行的中心服务器上，用户都通过网络传输数据和操作命令，我们约定每个账户的信息包括但不限于以下字段：

银行内用户Id：10位数字组成

用户名：填姓名（10个汉字长度）

密码：不少于4位

身份ID：填学号（12位数字）

手机号：正常手机号

性别：用一位字符（M:男，F:女）

出生日期：YYYY-MM-DD

账户余额：双精度范围即可

柜台操作人员的用户名和密码可以在系统初始化时单独设定，或者另作专门的管理程序设置。服务器端数据的存储可以用文件，也可以用数据库，建议用mysql数据库，客户端可以使用字符界面，亦可以使用GUI界面；如果学有能力，可以使用Java Web方式实现，建议用JSP（新版教材第12章内容）即可。其它要求参考作业一，大家都熟悉了。

**二、作业要求**

**1)一般要求**

* 要求使用华为云鲲鹏计算机完成，需要大家注册华为云账号，将自己的账号名发给班级课代表，格式参考文件模板。
* 使用项目名称：FM2【小组名称】。
* 使用面向对象设计理念，运用网络技术、流技术、数据库技术，鼓励用多线程技术和GUI技术，程序中要使用数据集合和有效性验证。但具体分析和设计实现可自由发挥，酌情会加分。
* 源码逻辑结构清晰、注释要具体得当。
* 要有word版或pdf版说明文档，说明文档中给出小组名称，成员等信息，对该项目的分工和工作权重，以及其它分析、代码逻辑实现和操作等说明，好文档会加分。

**2)上交要求**

* 将作业项目压缩后上传到学习通，压缩文件名为FM【小组名】.zip，同时提交一份给课代表，课代表检查一下，并填写一个记录表。（防止学习通上下载后为空文件）。
* 提交的内容包含：源码、依赖包、sql脚本（如果有数据库）、数据文件（如果有文件操作），说明文档（说明小组成员、分工，以及项目的分析和说明等）
* 提交的作业的截止时间为2022年6月25日晚12点前。
* 如果没有按时提交的作业，或者如果您的文件无法打开或以任何方式损坏，你将得到本次作业零分。

***所以上交前检查你的文件三次！如果你不检查你的文件，要准备好这样做，并在收到一个零的风险。***

**三、项目实现建议和要求**



模式二

**数据库服务器数据库服务器**

**服务端程序**



模式一

**数据库服务器数据库服务器**



模式三

**数据库服务器数据库服务器**

**WEB服务端程序**

其中模式一比较简单，直接通过客户端使用jdbc访问远程数据库服务；模式二符合当下的很多软件的开发模式，建议大家使用，在服务端实现多线程，每一个客户端申请连接，就开一个线程提供服务，在服务端访问数据库服务，客户端看不到数据库，并且此模式下要主要对数据库访问时使用同步锁机制；模式三采用WEB方式，在服务端安装WEB服务器（建议用Tomcat），然后编写jsp网页程序，通过浏览器访问和操作，建议大家自学第12章内容。具体界面和交互逻辑可自由发挥。

**四、分级标准**

1.作业将分为100分。

2.分配将是在源代码和输出上进行的。

|  |  |
| --- | --- |
| 程序需求/评判标准 | 分数 |
| 项目基础分（根据实现功能）： | 70 |
| 能够正常开户 | 10 |
| 能够正确查询余额 | 10 |
| 能够正确存钱取钱 | 10 |
| 能够实现正确转账功能 | 10 |
| 能够修改密码和其它信息（除id和身份证（学号）） | 10 |
| 通过导入xsl批量开户 | 10 |
| 导出用户信息和生成报表 | 10 |
| 面向对象理论、网络、多线程和数据库等使用 | 20 |
| 使用数据库技术 | 5 |
| 熟练使用面向对象技术 | 5 |
| 使用网络服务端程序 | 5 |
| 使用多线程技术 | 5 |
| 逻辑清晰、代码规范、注释及其他 | 10 |

**五、提示和建议**

小组可以采用自己认可和熟悉的模式设计，但程序运行起来的外观应该是干净的和令人愉快的，界面友好，数据准确。在项目的建模设计、业务实现模式和界面的布局等方面有相当大的灵活性。

**1.连接数据库代码参考如下：**

public static Connection getConnection() throws SQLException, java.lang.ClassNotFoundException {

String url = "jdbc:mysql://www.lzucclab.net:3306/lzu2021db?characterEncoding=utf8"; // 连接MySQL中的数据库

Class.forName("com.mysql.jdbc.Driver");

String userName = "lzustudent"; // 登录用户名

String password = "Lzu1909"; // 密码

Connection con = DriverManager.getConnection(url, userName, password);

return con;

}

**2.xls数据导入代码参考：**

public static void main(String[] args) {

Workbook book = null;

try {

Connection con = getConnection();

Statement sql = con.createStatement();

/\* 创建表 \*/

sql.execute("drop table if exists student"); // 如果原来存在student表，则删除

sql.execute("create table student(姓名 varchar(12),学院 varchar(30),科目 varchar(30),成绩 int,订单编号 varchar(15)) ENGINE=InnoDB DEFAULT CHARSET=utf8;");

PreparedStatement pstmt = con.prepareStatement("insert into student values(?,?,?,?,?)");

/\* 读取本地Excel文件内容 \*/

book = Workbook.getWorkbook(new File("d://temp//readme.xls"));

// 获取Excel第一个选项卡对象

Sheet sheet = book.getSheet(0);

int cols = sheet.getColumns(); // 取到表格的列数

int rows = sheet.getRows();// 取到表格的行数

String[] contents = new String[cols];

for (int i = 1; i < sheet.getRows(); i++) {

for (int j = 0; j < cols; j++) {

contents[j] =sheet.getCell(j, i).getContents();

if (j == 3)

pstmt.setInt(4, Integer.parseInt(contents[j]));

else

pstmt.setString(j + 1, contents[j]);

System.out.print(contents[j]+ "\t");

}

pstmt.executeUpdate();

System.out.println();

}

System.out.println("共有" + (rows - 1) + "记录导入。");

sql.close();

pstmt.close();

con.close();

} catch (java.lang.ClassNotFoundException e) {

System.err.println("ClassNotFoundException:" + e.getMessage());

} catch (SQLException ex) {

System.err.println("SQLException:" + ex.getMessage());

} catch (BiffException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

1. **导出xls参考代码：**

import java.io.File;

import java.util.\*;

import jxl.\*;

import jxl.write.\*;

public class TestExportExcel {

public static void excelExport(Customer[] list, String path) {

WritableWorkbook book = null;

try {

// 创建一个Excel文件对象

book = Workbook.createWorkbook(new File(path));

// 创建Excel第一个选项卡对象

WritableSheet sheet = book.createSheet("第一页", 0);

// 设置表头，第一行内容

// Label参数说明：第一个是列，第二个是行，第三个是要写入的数据值，索引值都是从0开始

Label label1 = new Label(0, 0, "姓名");// 对应为第1列第1行的数据

Label label2 = new Label(1, 0, "年龄");// 对应为第2列第1行的数据

Label label3 = new Label(2, 0, "手机号码");// 对应第3列第1行的数据

Label label4 = new Label(3, 0, "住址");// 对应为第4列第1行的数据

// 添加单元格到选项卡中

sheet.addCell(label1);

sheet.addCell(label2);

sheet.addCell(label3);

sheet.addCell(label4);

// 遍历集合并添加数据到行，每行对应一个对象

for (int i = 0; i < list.length; i++) {

Customer customer = list[i];

// 表头占据第一行，所以下面行数是索引值+1

// 跟上面添加表头一样添加单元格数据，这里为了方便直接使用链式编程

sheet.addCell(new Label(0, i + 1, customer.getName()));

sheet.addCell(new Label(1, i + 1, customer.getAge().toString()));

sheet.addCell(new Label(2, i + 1, customer.getTelephone()));

sheet.addCell(new Label(3, i + 1, customer.getAddress()));

}

// 写入数据到目标文件

book.write();

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

// 关闭

book.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

public static void main(String[] args) {

Customer[] list = new Customer[10];

// 创建数据

for (int i = 0; i < 10; i++) {

Customer customer = new Customer();

customer.setName("云中飞马" + i);

customer.setAge(i);

customer.setTelephone("1301234976" + i);

customer.setAddress("甘肃省兰州市七里河区");

list[i] = customer;

}

String path = "D:\\temp\\customer.xls";

System.out.println("开始导出...");

long s1 = new Date().getTime();

// 开始导出

excelExport(list, path);

long s2 = new Date().getTime();

long time = s2 - s1;

System.out.println("导出完成！消耗时间：" + time + "毫秒");

}

}

1. **生成报表pdf参考代码：**

//https://link.csdn.net/?target=http%3A%2F%2Fitextpdf.com%2Fbook%2Fexamples.php

//https://link.csdn.net/?target=http%3A%2F%2Fitextpdf.com%2F

//https://github.com/itext/itext7

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.Date;

import com.itextpdf.text.Document;

import com.itextpdf.text.DocumentException;

import com.itextpdf.text.Paragraph;

import com.itextpdf.text.pdf.PdfWriter;

public class HelloWorldPdf {

public static final String RESULT = "d://temp//hello.pdf";

public static void main(String[] args)

throws DocumentException, IOException {

new HelloWorldPdf().createPdf(RESULT);

}

public void createPdf(String filename) throws DocumentException, IOException {

Document document = new Document();

PdfWriter.getInstance(document, new FileOutputStream(filename));

document.open();

document.add(new Paragraph("Hello World!"));

document.add(new Paragraph("sqrt(2.0)="+Math.sqrt(2.0)));

Date time=new Date();

Paragraph p=new Paragraph("The time is:"+time);

document.add(p);

document.close();

}

}