武汉纺织大学

Java应用开发课程设计

**EMS物流下单程序**

**学 院： 数学与计算机学院**

**班 级： 物联网11801**

**姓 名： 盛亦凡**

**学 号： 1804240424**

**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年6月20日**

目 录

[1 需求分析 1](#_Toc24230)

[1.1主菜单 1](#_Toc20921)

[1.2各项功能 1](#_Toc31639)

[1.2.1导出图片功能 1](#_Toc29676)

[1.2.2导出PDF功能 1](#_Toc27539)

[1.2.3退出 1](#_Toc23387)

[2 系统设计 1](#_Toc23995)

[2.1用户用例图 1](#_Toc8094)

[2.2 ER图 2](#_Toc1117)

[2.3 UML类图（Class Diagram） 2](#_Toc1689)

[2.4 UML时序图（Sequence Diagram） 4](#_Toc8046)

[2.4.1 生成图片 4](#_Toc9092)

[2.5 UML活动图（Activity Diagram） 5](#_Toc25592)

[2.5.1 生成图片 5](#_Toc21290)

[3.1 项目结构 8](#_Toc28632)

[3.3 VO类Information.java 8](#_Toc10029)

[3.4 Excel处理类JXLControl.java 12](#_Toc13437)

[3.5 二维码生成类DAndQRCode.java 13](#_Toc8271)

[实现根据信息生成二维码，一维码的图像或者图像地址。（二维码直接返回图像，一维码返回图片地址） 13](#_Toc22662)

[3.6 全局变量类global.java 15](#_Toc22538)

[3.7 流水号生成和图片地址生成类LshAndImgSrcControl.java 16](#_Toc20339)

[3.8 用户界面driver.java 17](#_Toc31753)

[3.9 图片转PDF类：ImgToPdf.java 19](#_Toc8136)

[4 系统测试 20](#_Toc21049)

[5 系统总结 21](#_Toc21326)

# 1 需求分析

设计一个小程序完成对Excel文件的读取，并将数据写在表单上转化成图片和pdf。

## 1.1主菜单

主菜单很简单三个选项， 先得选一，不然二不能执行。

## 1.2各项功能

### 1.2.1导出图片功能

当选择功能1是，导出图片到根目录下的export 文件夹里，如果正确导出无差错，那么会显示导出成功，还会给出其他提示信息。

### 1.2.2导出PDF功能

当选择功能2使，导出pdf文件到根目录下的export文件夹中，他需要有前置文件图片文件，所以如果功能1没有用，那么功能2是不能使用的。如果正确导出pdf文件，那么会给出导出成功的字样并将文件名输出。

### 1.2.3退出

当选择功能0时，程序会停止，并返回常数1.

# **2 系统设计**

## 2.1用户用例图

系统只有一个角色，那就是使用者，我们需要将Excel文件快递单通过图片处理程序将它转化为图片形式的快递单，和PDF形式的快递单。角色的用户用例如下图2-1所示：

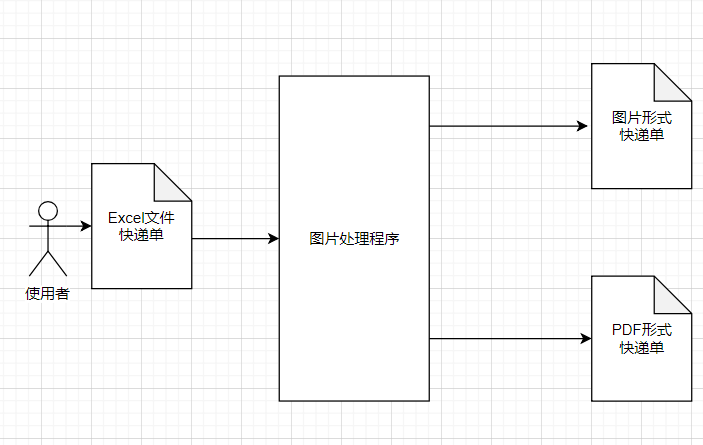


图2-1 用户用例图

## 2.2 ER图

只有一个实体类，他是Information，记载了从Excel读取来的数据

如图3-1

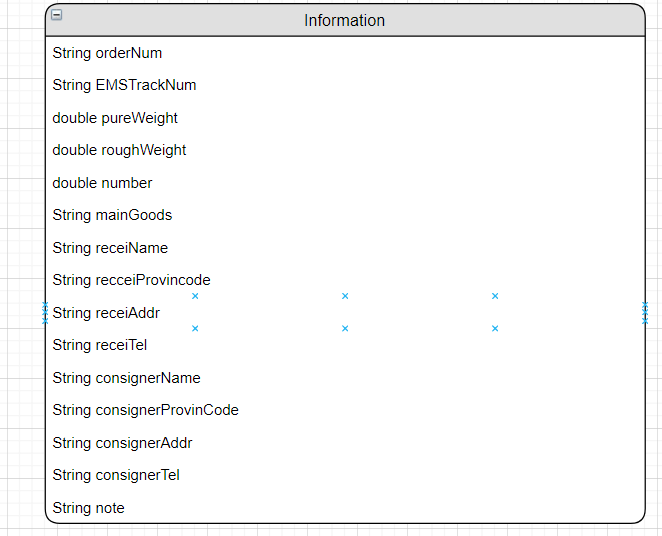


图3-1

## 2.3 UML类图（Class Diagram）

对于用户功能模块，共设计如下6个类。

* 数全局变量类global 记载了整个项目里的全局变量。方便修改。
* VO实体类Information：与Excel表格进行映射的类。主要由属性，setter, getter方法组成，VO类中的属性与表中的字段相对应，每一个VO类的对象都表示Excel中的每一条记录
* 图片形成类：该类做到了将Excel文件读取过来的信息整合到图片上去，并在相应的位置写出来。
* DA二维码一维码形成类：DAndQRCode，通过给的字符串形成二维码和一维码
* 业订单号文件名处理类：LshAndImgSrcControl ，处理文件名和处理订单号
* 图片转pdf类，该类中写了如何将图片转换成pdf

各类的结构及类之间的关系如图2-5所示：

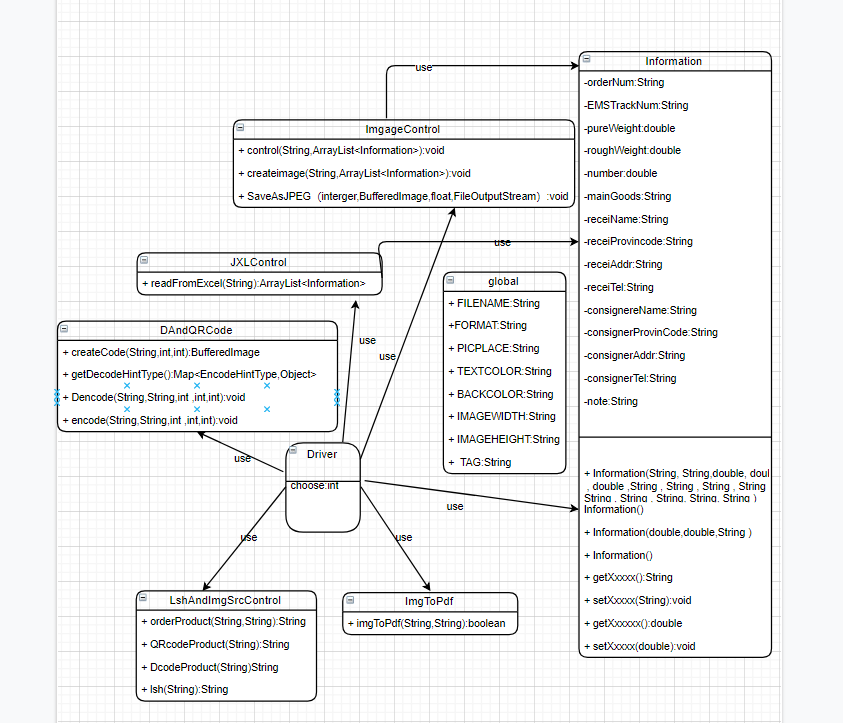
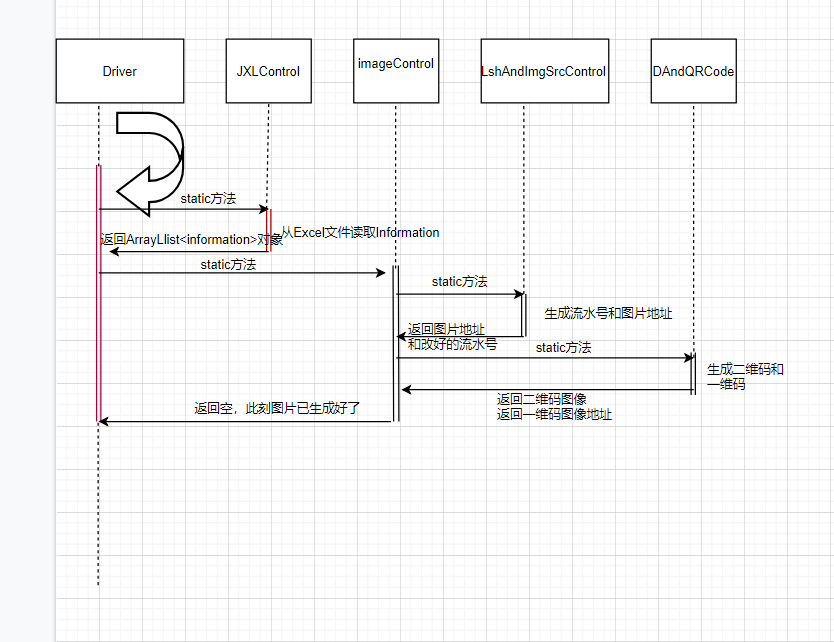


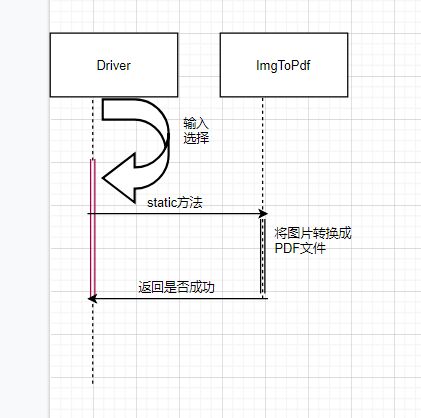
图2-5 用户功能类图

## 2.4 UML时序图（Sequence Diagram）

### 2.4.1 生成图片



**2.4.2生成PDF文件**



## 2.5 UML活动图（Activity Diagram）

### 2.5.1 生成图片

输入选择，应该为1输入选择1 ，他会开始生成图片，如果是其他选择，那么他会提示输入错误。

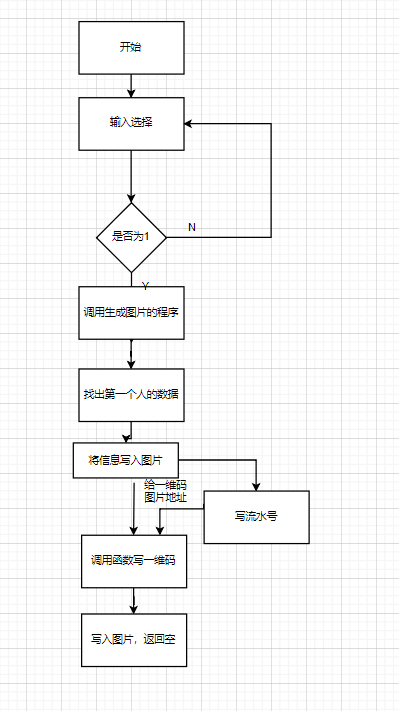


图2-2 生成图片流程图

2.5.2生成PDF文件

如果没有先生成图片那么是不能选择2的，它是将图片变成PDF，如果没有图片那么就变不了，会出错。

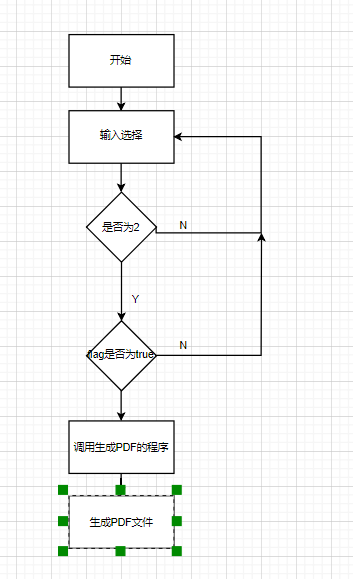
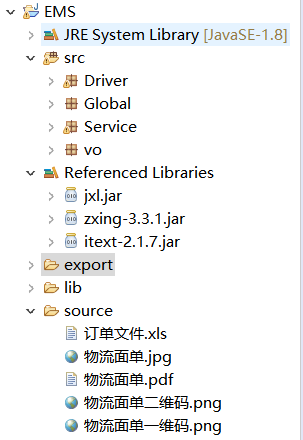


图2-3 生成PDF流程图

**3 系统实现**

## 3.1 项目结构



## 3.3 VO类Information.java

实体类代码，是Excel到java对象的映射。

**package** vo;

**public** **class** Information {

**private** String orderNum; //订单顺序号

**private** String EMSTrackNum; //EMS物流单号

**private** **double** pureWeight; //净重

**private** **double** roughWeight; //毛重

**private** **double** number; //件数

**private** String mainGoods; //主要商品

**private** String receiName; //接收人姓名

**private** String receiProvinCode; //收件人省市区代码

**private** String receiAddr; //收件人地址

**private** String receiTel; //收件人电话

**private** String consignerName; //发货人名称

**private** String consignerProvinCode; //发货人省市区代码

**private** String consignerAddr; //发货人地址

**private** String consignerTel; //发货人电话

**private** String note; //备注

//生成get，set方法

**public** String getOrderNum() {

**return** orderNum;

}

**public** **void** setOrderNum(String orderNum) {

**this**.orderNum = orderNum;

}

**public** String getEMSTrackNum() {

**return** EMSTrackNum;

}

**public** **void** setEMSTrackNum(String eMSTrackNum) {

EMSTrackNum = eMSTrackNum;

}

**public** **double** getPureWeight() {

**return** pureWeight;

}

**public** **void** setPureWeight(**double** pureWeight) {

**this**.pureWeight = pureWeight;

}

**public** **double** getRoughWeight() {

**return** roughWeight;

}

**public** **void** setRoughWeight(**double** roughWeight) {

**this**.roughWeight = roughWeight;

}

**public** **double** getNumber() {

**return** number;

}

**public** **void** setNumber(**double** number) {

**this**.number = number;

}

**public** String getMainGoods() {

**return** mainGoods;

}

**public** **void** setMainGoods(String mainGoods) {

**this**.mainGoods = mainGoods;

}

**public** String getReceiName() {

**return** receiName;

}

**public** **void** setReceiName(String receiName) {

**this**.receiName = receiName;

}

**public** String getReceiProvinCode() {

**return** receiProvinCode;

}

**public** **void** setReceiProvinCode(String receiProvinCode) {

**this**.receiProvinCode = receiProvinCode;

}

**public** String getReceiAddr() {

**return** receiAddr;

}

**public** **void** setReceiAddr(String receiAddr) {

**this**.receiAddr = receiAddr;

}

**public** String getReceiTel() {

**return** receiTel;

}

**public** **void** setReceiTel(String receiTel) {

**this**.receiTel = receiTel;

}

**public** String getConsignerName() {

**return** consignerName;

}

**public** **void** setConsignerName(String consignerName) {

**this**.consignerName = consignerName;

}

**public** String getConsignerProvinCode() {

**return** consignerProvinCode;

}

**public** **void** setConsignerProvinCode(String consignerProvinCode) {

**this**.consignerProvinCode = consignerProvinCode;

}

**public** String getConsignerAddr() {

**return** consignerAddr;

}

**public** **void** setConsignerAddr(String consignerAddr) {

**this**.consignerAddr = consignerAddr;

}

**public** String getConsignerTel() {

**return** consignerTel;

}

**public** **void** setConsignerTel(String consignerTel) {

**this**.consignerTel = consignerTel;

}

**public** String getNote() {

**return** note;

}

**public** **void** setNote(String note) {

**this**.note = note;

}

//生成构造方法

**public** Information(String orderNum, String eMSTrackNum, **double** pureWeight, **double** roughWeight, **double** number,

String mainGoods, String receiName, String receiProvinCode, String receiAddr, String receiTel,

String consignerName, String consignerProvinCode, String consignerAddr, String consignerTel, String note) {

**super**();

**this**.orderNum = orderNum;

EMSTrackNum = eMSTrackNum;

**this**.pureWeight = pureWeight;

**this**.roughWeight = roughWeight;

**this**.number = number;

**this**.mainGoods = mainGoods;

**this**.receiName = receiName;

**this**.receiProvinCode = receiProvinCode;

**this**.receiAddr = receiAddr;

**this**.receiTel = receiTel;

**this**.consignerName = consignerName;

**this**.consignerProvinCode = consignerProvinCode;

**this**.consignerAddr = consignerAddr;

**this**.consignerTel = consignerTel;

**this**.note = note;

}

**public** Information() {

**super**();

}

**public** Information(**double** roughWeight,**double** number,String mainGoods) {

**this**.roughWeight=roughWeight;

**this**.number=number;

**this**.mainGoods=mainGoods;

}

//生成toString 方法

@Override

**public** String toString() {

**return** "Information [orderNum=" + orderNum + ", EMSTrackNum=" + EMSTrackNum + ", pureWeight=" + pureWeight

+ ", roughWeight=" + roughWeight + ", number=" + number + ", mainGoods=" + mainGoods + ", receiName="

+ receiName + ", receiProvinCode=" + receiProvinCode + ", receiAddr=" + receiAddr + ", receiTel="

+ receiTel + ", consignerName=" + consignerName + ", consignerProvinCode=" + consignerProvinCode

+ ", consignerAddr=" + consignerAddr + ", consignerTel=" + consignerTel + ", note=" + note + "]\n";

}

}

## 3.4 Excel处理类JXLControl.java

JXLControl 里只有一个函数，他可以读入Excel文件的信息并存储在java实体对象中。

**package** Service;

**import** java.io.File;

**import** java.util.ArrayList;

**import** jxl.Cell;

**import** jxl.Sheet;

**import** jxl.Workbook;

**import** vo.Information;

**public** **class** jxlControl {

**public** **static** ArrayList<Information> readFromExcel(String fileName){

ArrayList<Information> list =**new** ArrayList<Information>();

**try** {

File file = **new** File(fileName);

Workbook workbook = Workbook.*getWorkbook*(file);

Sheet sheet = workbook.getSheet(0);

String array[]=**new** String [15];

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

**for**(**int** col = 0; col < sheet.getColumns();col++) {

Cell cell = sheet.getCell(col,row);

array[col]=cell.getContents();

}

**double** pureWeight=Double.*parseDouble*(array[2]);

**double** number = Double.*parseDouble*(array[4]);

Information info;

**if**(array[0].equals("")) {

String mainGoods=array[5];

info = **new** Information(pureWeight,number,array[5]);

}

**else** {

**double** roughWeight=Double.*parseDouble*(array[3]);

info =**new** Information(array[0],array[1],pureWeight,

roughWeight,number,array[5],array[6],array[7],

array[8],array[9],array[10],array[11],array[12],

array[13],array[14]);

}

list.add(info);

}

}

**catch**(Exception e) {

e.printStackTrace();

}

**return** list;

}

}

## 3.5 二维码生成类DAndQRCode.java

## 实现根据信息生成二维码，一维码的图像或者图像地址。（二维码直接返回图像，一维码返回图片地址）

**package** Service;

**import** java.awt.image.BufferedImage;

**import** java.io.File;

**import** java.io.FileNotFoundException;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** java.util.HashMap;

**import** java.util.Map;

**import** com.google.zxing.BarcodeFormat;

**import** com.google.zxing.EncodeHintType;

**import** com.google.zxing.MultiFormatWriter;

**import** com.google.zxing.WriterException;

**import** com.google.zxing.client.j2se.MatrixToImageWriter;

**import** com.google.zxing.common.BitMatrix;

**import** com.google.zxing.qrcode.QRCodeWriter;

**import** com.google.zxing.qrcode.decoder.ErrorCorrectionLevel;

**public** **class** DAndQRCode {

// 格式

**public** **static** BufferedImage createCode(String qrUrl,**int** h,**int** w) {

MultiFormatWriter multiFormatWriter = **null**;

BitMatrix bm = **null**;

Map<EncodeHintType, Object> hints = *getDecodeHintType*();

BufferedImage image=**new** BufferedImage(w,h,BufferedImage.***TYPE\_INT\_RGB***);

**try** {

multiFormatWriter = **new** MultiFormatWriter();

// 参数顺序分别为：编码内容，编码类型，生成图片宽度，生成图片高度，设置参数

bm = multiFormatWriter.encode(qrUrl, BarcodeFormat.***QR\_CODE***, w, h, hints);

// 开始利用二维码数据创建Bitmap图片，分别设为黑（0xFFFFFFFF）白（0xFF000000）两色

**for** (**int** x = 0; x < w; x++) {

**for** (**int** y = 0; y < h; y++) {

image.setRGB(x, y, bm.get(x, y) ? Global.global.***TEXTCOLOR*** : Global.global.***BACKCOLOR***);

}

}

} **catch** (Exception e) {

e.printStackTrace();

}

**return** image;

}

**private** **static** Map<EncodeHintType, Object> getDecodeHintType() {

// 用于设置QR二维码参数

Map<EncodeHintType, Object> hints = **new** HashMap<EncodeHintType, Object>();

// 设置QR二维码的纠错级别（H为最高级别）具体级别信息

hints.put(EncodeHintType.***ERROR\_CORRECTION***, ErrorCorrectionLevel.***H***);

// 设置编码方式

hints.put(EncodeHintType.***CHARACTER\_SET***, "utf-8");

hints.put(EncodeHintType.***MARGIN***, 0);

hints.put(EncodeHintType.~~MAX\_SIZE~~, 350);

hints.put(EncodeHintType.~~MIN\_SIZE~~, 100);

**return** hints;

}

/\*

\* contents 内容

\* dest 地址

\* width 长度

\* height 高度

\* offset 偏移量

\*/

**public** **static** **void** Dencode(String contents,String dest,**int** width,**int** height,**int** offset){

**try**{

contents=**new** String(contents.getBytes("UTF-8"),"ISO-8859-1");

BitMatrix matrix=**new** MultiFormatWriter().encode(contents,BarcodeFormat.***CODE\_128***,width-offset, height);

MatrixToImageWriter.*writeToStream*(matrix,Global.global.***QRFORMAT***,**new** FileOutputStream(**new** File(dest)));

}**catch**(Exception e) {

e.printStackTrace();

}

}

**public** **static** **void** encode(String contents,String dest,**int** width,**int** height){

**try**{

contents=**new** String(contents.getBytes("UTF-8"),"ISO-8859-1");

QRCodeWriter writer=**new** QRCodeWriter();

BitMatrix matrix=writer.encode(contents, BarcodeFormat.***QR\_CODE***, width, height);

MatrixToImageWriter.*writeToStream*(matrix,Global.global.***QRFORMAT***, **new** FileOutputStream(**new** File(dest)));

}**catch**(Exception e) {

e.printStackTrace();

}

}

}

## 3.6 全局变量类global.java

这里有整个项目用到的全局变量。

**package** Global;

**public** **class** global {

**public** **static** **final** String ***FILENAME*** = "./source/订单文件.xls";

**public** **static** **final** String ***FORMAT***=".jpg";

**public** **static** **final** String ***QRFORMAT***="PNG";

**public** **static** **final** String ***PICPLACE*** = "./export/";

**public** **final** **static** **int** ***TEXTCOLOR*** = 0xFFFFFFFF;

**public** **final** **static** **int** ***BACKCOLOR*** = 0xFF00000;

**public** **final** **static** **int** ***IMAGEWIDTH*** = 955;

**public** **final** **static** **int** ***IMAGEHEIGHT*** = 1400;

**public** **final** **static** String ***TAG***="pic";

}

## 3.7 流水号生成和图片地址生成类LshAndImgSrcControl.java

有多个函数，生成一维码需要的地址来自于这里，由订单号新建的流水号也出自这里。

**package** Service;

**import** java.text.DateFormat;

**import** java.text.SimpleDateFormat;

**import** java.util.Date;

**public** **class** LshAndImgSrcControl {

**public** **static** String orderProduct(String orderNumber,String format) {

String src;

String head = Global.global.***PICPLACE***;

src = head+Global.global.***TAG***+orderNumber+format;

**return** src;

}

**public** **static** String QRcodeProduct(String orderNumber) {

String src;

String head = Global.global.***PICPLACE***;

src=head+"物流单号"+orderNumber+"一维码.png";

**return** src;

}

**public** **static** String DcodeProduct(String orderNumber) {

String src;

String head = Global.global.***PICPLACE***;

src=head+"物流单号"+orderNumber+"一维码.png";

**return** src;

}

**public** **static** String lsh (String lsh) {

Date data = **new** Date();

DateFormat format = **new** SimpleDateFormat("yyyyMMdd");

String time = format.format(data);

lsh=time+lsh;

**return** lsh;

}

**public** String getData(String a) {

Date date = **new** Date();

**if** (a == "cal") {

DateFormat format2 = **new** SimpleDateFormat("yyyyMMdd");

String time2 = format2.format(date);

**return** time2;

} **else** **if** (a == "time") {

DateFormat format = **new** SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

String time = format.format(date);

**return** time;

} **else**

**return** **null**;

}

}

## 3.8 用户界面driver.java

根据提示输入选择，达到不同的效果，0是退出，1是生成图片，2是生成PDF，如果是第二次尝试还有之前的图片，可以删除他们才进行尝试

**package** Driver;

**import** java.io.IOException;

**import** java.util.ArrayList;

**import** java.util.Scanner;

**import** Service.ImgToPdf;

**import** Service.DAndQRCode;

**import** Service.imageControl;

**import** Service.jxlControl;

**import** vo.Information;

**public** **class** driver {

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

**int** choose=1;

**boolean** flag=**false**;

String format=Global.global.***FORMAT***;

String fileName =Global.global.***FILENAME***;

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

ArrayList<Information>list = jxlControl.*readFromExcel*(fileName);

**while**(choose!=0) {

System.***out***.println("1.导出图片");

System.***out***.println("2.导出pdf");

System.***out***.println("0.退出");

System.***out***.println("输入你的选择");

choose = scan.nextInt();

**switch**(choose) {

**case** 1:

imageControl.*control*(format, list);

System.***out***.println("图片生成完成，在export中查看，只有前三张是结果，后面只是一维码");

flag=**true**;

**break**;

**case** 2:

**if**(flag==**false**)

System.***out***.println("你需要先执行命令1");

**else** {

**for**(**int** i=1;i<4;i++) {

**try** {

String imgSrc=Global.global.***PICPLACE***+Global.global.***TAG***+String.*valueOf*(i)+Global.global.***FORMAT***;

String pdfSrc = Global.global.***PICPLACE***+"pdf"+String.*valueOf*(i)+".pdf";

ImgToPdf.*imgToPdf*(imgSrc,pdfSrc);

System.***out***.println(pdfSrc.substring(9)+"导出完成");

} **catch** (IOException e) {

// **TODO** Auto-generated catch block

e.printStackTrace();

}

}

}

**break**;

**case** 0:

System.*exit*(1);

**break**;

**default** :

System.***out***.println("输入有误，没有这个选择");

}

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

}

}

}

## 3.9 图片转PDF类：ImgToPdf.java

描述该文件作用，并对里面的关键代码进行解释

**package** Service;

**import** java.io.File;

**import** java.io.FileOutputStream;

**import** java.io.IOException;

**import** com.lowagie.text.Document;

**import** com.lowagie.text.DocumentException;

**import** com.lowagie.text.Image;

**import** com.lowagie.text.PageSize;

**import** com.lowagie.text.pdf.PdfWriter;

**public** **class** ImgToPdf {

**public** **static** **boolean** imgToPdf(String imgFilePath, String pdfFilePath)**throws** IOException {

File file=**new** File(imgFilePath);

**if**(file.exists()){

Document document = **new** Document();

FileOutputStream fos = **null**;

**try** {

fos = **new** FileOutputStream(pdfFilePath);

PdfWriter.*getInstance*(document, fos);

// 添加PDF文档的某些信息，比如作者，主题等等

document.addAuthor("学生");

document.addSubject("pdf制作");

// 设置文档的大小

document.setPageSize(PageSize.***A4***);

// 打开文档

document.open();

// 写入一段文字

//document.add(new Paragraph("JUST TEST ..."));

// 读取一个图片

Image image = Image.*getInstance*(imgFilePath);

**float** imageHeight=image.getScaledHeight();

**float** imageWidth=image.getScaledWidth();

**int** i=0;

**while**(imageHeight>500||imageWidth>500){

image.scalePercent(100-i);

i++;

imageHeight=image.getScaledHeight();

imageWidth=image.getScaledWidth();

}

image.setAlignment(Image.***ALIGN\_CENTER***);

// //设置图片的绝对位置

// image.setAbsolutePosition(0, 0);

// image.scaleAbsolute(500, 400);

// 插入一个图片

document.add(image);

} **catch** (DocumentException de) {

System.***out***.println(de.getMessage());

} **catch** (IOException ioe) {

System.***out***.println(ioe.getMessage());

}

document.close();

fos.flush();

fos.close();

**return** **true**;

}**else**{

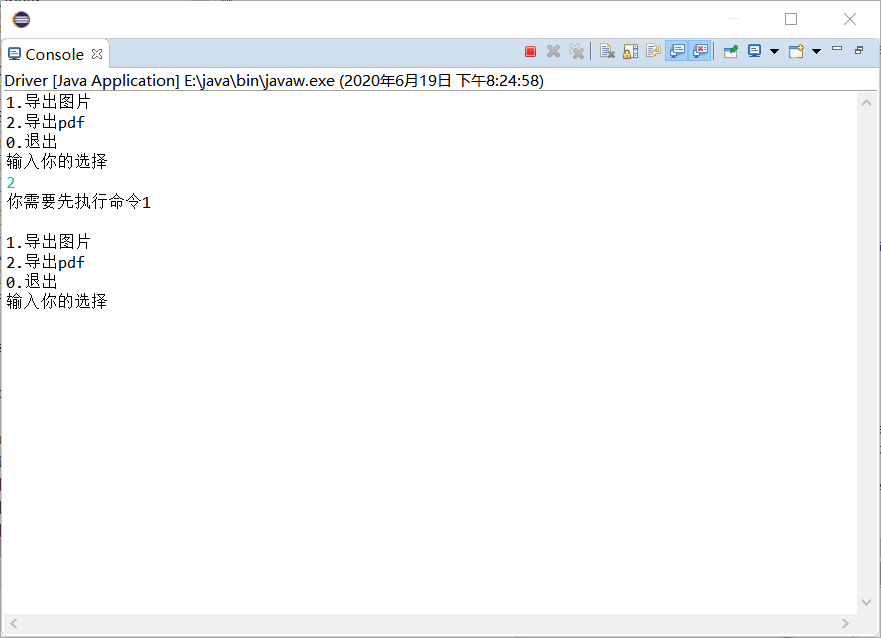
**return** **false**;

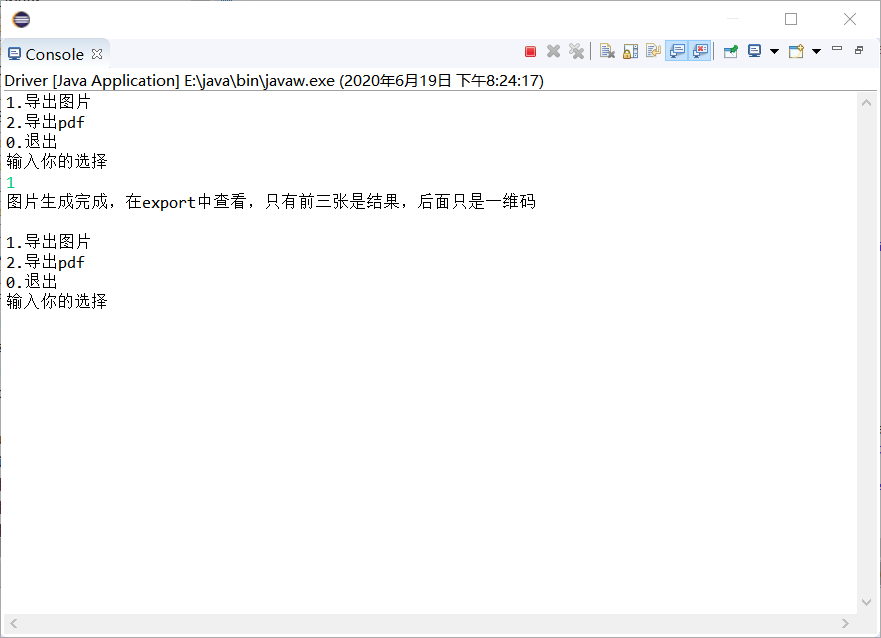
}

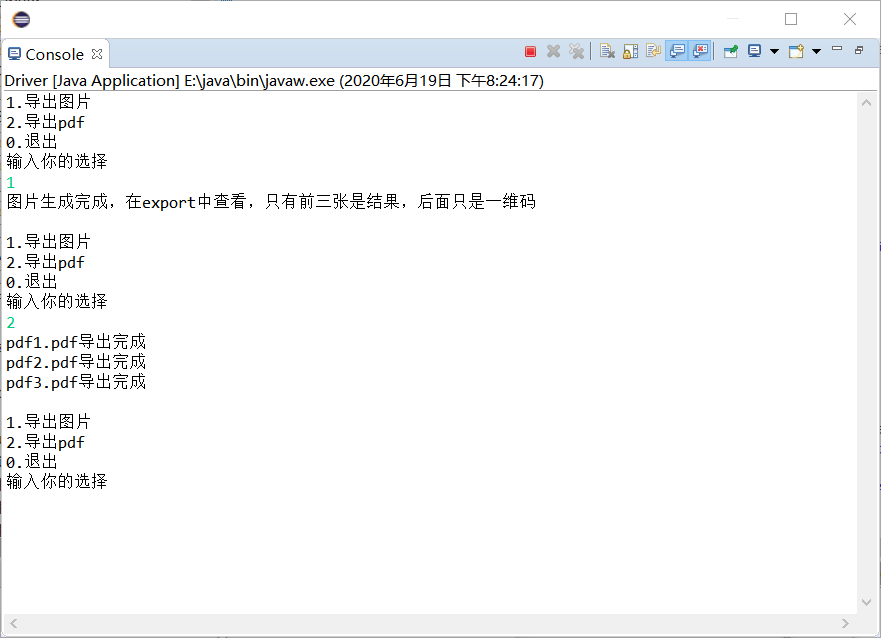
}

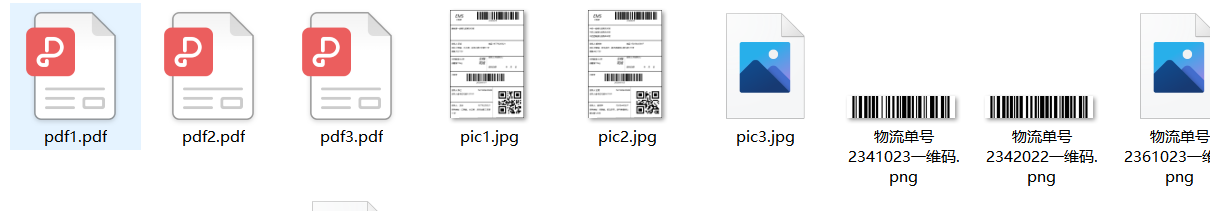
}

# **4 系统测试**









# **5 系统总结**

一开始有很多不会，感觉很难，特别是刚开始那会，看文档也看不懂，很想放弃，但是坚持写，慢慢的摸索它们的功能，一开始以为的浪费时间还是起了作用，满满的能看懂了，也能慢慢的写出来，整个程序不难，但是特别繁琐，弄了比较久。

还有就是我的**程序的不足，**他似乎最后没有释放图片资源，然后我退出去发现那个文件删不了，被占用了，把eclipse退了才能删除，不过可以查看，退了之后文件也是好的没有问题。本次**课收获很多**，特别是自己做的二维码能扫的时候特别高兴，整个学习过程也比较明白。**课程的建议**的话，我觉得还是要多督促我们认真完成作业。知识都讲的很好**。**

**以后发展方向**可能是物联网方面吧，我们的专业也要热爱，但是多这一门课的知识也很有意义，学到最后很多东西是可以触类旁通的，学科交叉，殊途同归。

**自评成绩75**