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11802**

**姓 名： 邹正**

**学 号： 1804241014**

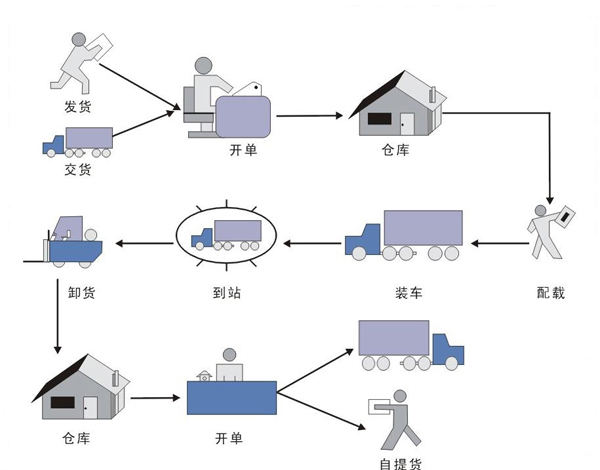
**指导老师： 聂刚**

**成 绩：**

**完成日期： 2020年6月20日**

# 1 需求分析

电子商务等相关产业的发展促进了物流快递行业的繁荣,提升物流快递企业的信息化水平能够为确保物流快递企业的业务处理能力，为企业的发展提供技术保障。

图为物流快递公司业务运作总体流程分析图。归纳总结物流快递的总体业务流程，其主要包括揽货、仓储、在途运输和配送四个环节。

**物流快递业务运作总体流程图**

**揽货环节：**

在客户有快递需求的情况下，通过电话、网络等方式向物流快递公司提出揽货需求。货物收揽可以由客户将货物送达物流公司网点，也可以由物流公司安排快递员上门收货。在货物成功收揽之后，需要开具相应的收货单，标明货物的种类、发件人、收件人等信息。

# **2 系统设计要求**



## 1.1Java读写excel文件:

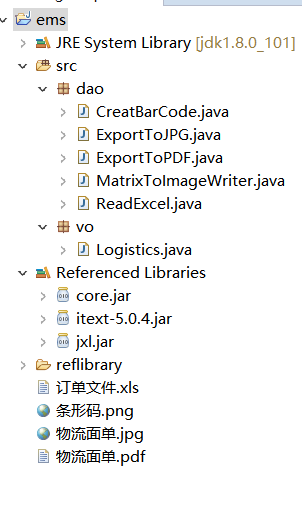
## 1.2Java条码二维码操作:

## 1.3Java 动态生成jpg图片:

# 1.4Java生成pdf:

# **3 系统实现**

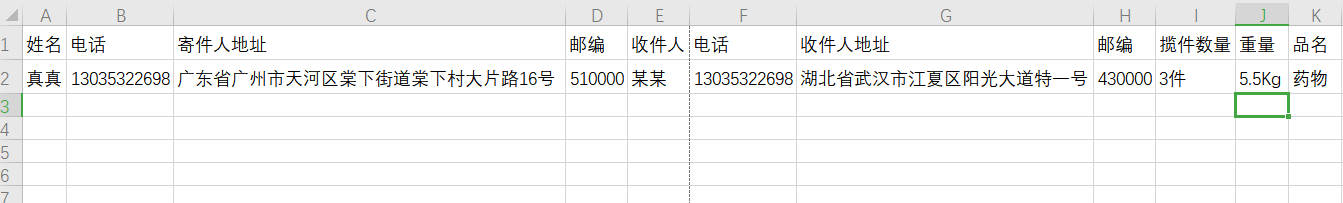
## 3.1 项目结构



## 3.2 配置文件

### 3.2.1 新建xls文件

输入测试样板



### 3.2.2 /ems/src/dao/CreatBarCode.java文件

package dao;

import java.io.File;

import com.google.zxing.BarcodeFormat;

import com.google.zxing.MultiFormatWriter;

import com.google.zxing.common.BitMatrix;

public class CreatBarCode {

public void encode(String imgPath) {

int codeWidth = 3 + // start guard

(7 \* 6) + // left bars

5 + // middle guard

(7 \* 6) + // right bars

3; // end guard

codeWidth = Math.max(codeWidth, 105);

try {

BitMatrix bitMatrix = new MultiFormatWriter().encode("8754963215841",

BarcodeFormat.EAN\_13, codeWidth, 50, null);

MatrixToImageWriter

.writeToFile(bitMatrix, "png", new File(imgPath));

} catch (Exception e) {

e.printStackTrace();

}

}

}

### 

### 3.2.3 /ems/src/dao/ExportToJPG.java文件

package dao;

import java.awt.Color;

import java.awt.Font;

import java.awt.Graphics;

import java.awt.image.BufferedImage;

import java.io.BufferedOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import javax.imageio.ImageIO;

import com.itextpdf.text.DocumentException;

import com.sun.image.codec.jpeg.JPEGCodec;

import com.sun.image.codec.jpeg.JPEGImageEncoder;

import jxl.read.biff.BiffException;

import vo.Logistics;

public class ExportToJPG {

int width=700;//图片宽度

int height=300;//图片高度

// 创建JPG文件到指定路径下

public void createJpg(String path,BufferedImage image) {

try{

FileOutputStream fos = new FileOutputStream(path);

BufferedOutputStream bos = new BufferedOutputStream(fos);

JPEGImageEncoder encoder = JPEGCodec.createJPEGEncoder(bos);

encoder.encode(image);

bos.close();

fos.close();

}catch(FileNotFoundException fnfe){

System.out.println(fnfe);

}catch(IOException ioe){

System.out.println(ioe);

}

}

public void graphicsGeneration(Logistics log) throws FileNotFoundException, IOException

{

BufferedImage image1 = ImageIO.read(new FileInputStream("D:\\Java\\eclipse-workspace\\Eclipse\\ems\\条形码.png"));

int imageWidth = 1000;// 图片宽度

int imageHeight = 800;// 图片高度

BufferedImage image = new BufferedImage(imageWidth, imageHeight, BufferedImage.TYPE\_INT\_RGB);

Graphics g = image.getGraphics();

g.setColor(Color.white);

g.fillRect(0, 0, imageWidth, imageHeight); //填充背景颜色

g.setColor(Color.black);

//画出长为800，宽为400的矩形框

g.drawRect(100,100,800,400);

g.drawLine(100,200,900,200);

g.drawLine(100,300,900,300);

g.drawLine(100,400,900,400);

g.drawLine(500,300,500,400);

g.setFont(new Font("宋体", Font.BOLD, 60));

g.drawString("EMS物流", 180, 95);

g.setFont(new Font("宋体", Font.BOLD, 25));

g.drawString("订单号：", 480, 95);

g.drawString(log.getTaskNO(), 580, 95);

g.drawString("寄件人姓名：", 120, 130);

g.drawString(log.getSender(), 280, 130);

g.drawString(" 电话：", 380, 130);

g.drawString(log.getSenPhone(), 500, 130);

g.drawString("地址：", 120, 160);

g.drawString(log.getSenAdd(), 180, 160);

g.drawString("邮编：", 120, 190);

g.drawString(log.getSenPost(), 200, 190);

g.drawString("收件人姓名：", 120, 230);

g.drawString(log.getRecipient(),280, 230);

g.drawString(" 电话：", 380, 230);

g.drawString(log.getRecPhone(),500, 230);

g.drawString("地址：", 120, 260);

g.drawString(log.getRecAdd(), 180, 260);

g.drawString("邮编：", 120, 290);

g.drawString(log.getRecPoSt(), 200, 290);

g.setFont(new Font("宋体", Font.BOLD, 22));

g.drawString("收件人/代收件人：", 520, 330);

g.drawString("签收时间： 年 月 日", 520, 380);

g.drawString("内件品名：", 120, 330);

g.drawString(log.getProduct(), 230, 330);

g.drawString("数量：", 350, 330);

g.drawString(log.getRemarks(), 420, 330);

g.drawString("重量：", 140, 380);

g.drawString(log.getAmount(), 210, 380);

g.setFont(new Font("宋体", Font.BOLD, 40));

g.drawString("订单号：", 120, 470);

g.drawString(log.getTaskNO(), 270, 470);

g.drawImage(image1,570, 15, 350, 60, null);

createJpg(".//物流面单.jpg", image);

}

public static void main(String[] args) throws BiffException, IOException, DocumentException {

// TODO Auto-generated method stub

String imgPath = "./条形码.png";

File file=new File("./订单文件.xls");

String imagePath = "./物流面单.jpg";

String pdfPath = "./物流面单.pdf";

Logistics logistics=new ReadExcel().readFromExcel(file);

new CreatBarCode().encode(imgPath);

new ExportToJPG().graphicsGeneration(logistics);

new ExportToPDF().jpgToPdf(imagePath, pdfPath);

System.out.println("图片和PDF生成成功！");

}

}

## 

## 3.2.4/ems/src/dao/ExportToPDF.java

package dao;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import javax.imageio.ImageIO;

import com.itextpdf.text.Document;

import com.itextpdf.text.DocumentException;

import com.itextpdf.text.Image;

import com.itextpdf.text.Rectangle;

import com.itextpdf.text.pdf.PdfWriter;

public class ExportToPDF {

public void jpgToPdf(String imagePath,String pdfPath) throws IOException, DocumentException

{

BufferedImage img = ImageIO.read(new File(imagePath));

FileOutputStream fos = new FileOutputStream(pdfPath);

Document doc = new Document(null, 0, 0, 0, 0);

doc.setPageSize(new Rectangle(img.getWidth(), img.getHeight()));

Image image = Image.getInstance(imagePath);

PdfWriter.getInstance(doc, fos);

doc.open();

doc.add(image);

doc.close();

}

}

## 3.2.5 /ems/src/dao/MatrixToImageWriter.java

package dao;

import java.awt.image.BufferedImage;

import java.io.File;

import java.io.IOException;

import java.io.OutputStream;

import javax.imageio.ImageIO;

import java.util.Hashtable;

import com.google.zxing.common.BitMatrix;

import com.google.zxing.BarcodeFormat;

import com.google.zxing.EncodeHintType;

import com.google.zxing.MultiFormatWriter;

/\*\*

\* 二维码的生成需要借助MatrixToImageWriter类，该类是由Google提供的，可以将该类直接拷贝到源码中使用

\*/

public class MatrixToImageWriter {

private static final int BLACK = 0xFF000000;

private static final int WHITE = 0xFFFFFFFF;

private MatrixToImageWriter() {

}

public static BufferedImage toBufferedImage(BitMatrix matrix) {

int width = matrix.getWidth();

int height = matrix.getHeight();

BufferedImage image = new BufferedImage(width, height,

BufferedImage.TYPE\_INT\_RGB);

for (int x = 0; x < width; x++) {

for (int y = 0; y < height; y++) {

image.setRGB(x, y, matrix.get(x, y) ? BLACK : WHITE);

}

}

return image;

}

public static void writeToFile(BitMatrix matrix, String format, File file)

throws IOException {

BufferedImage image = toBufferedImage(matrix);

if (!ImageIO.write(image, format, file)) {

throw new IOException("Could not write an image of format "

+ format + " to " + file);

}

}

public static void writeToStream(BitMatrix matrix, String format,

OutputStream stream) throws IOException {

BufferedImage image = toBufferedImage(matrix);

if (!ImageIO.write(image, format, stream)) {

throw new IOException("Could not write an image of format " + format);

}

}

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

String text = "http://www.bilibili.com"; // 二维码内容

int width = 1000; // 二维码图片宽度

int height = 700; // 二维码图片高度

String format = "jpg";// 二维码的图片格式

Hashtable<EncodeHintType, String> hints = new Hashtable<EncodeHintType, String>();

hints.put(EncodeHintType.CHARACTER\_SET, "utf-8"); // 内容所使用字符集编码

BitMatrix bitMatrix = new MultiFormatWriter().encode(text,

BarcodeFormat.QR\_CODE, width, height, hints);

// 生成二维码

File outputFile = new File("d:" + File.separator + "new.jpg");

MatrixToImageWriter.writeToFile(bitMatrix, format, outputFile);

}

}

## 3.2.6 /ems/src/dao/ReadExcel.java

package dao;

import java.io.File;

import java.io.IOException;

import java.text.SimpleDateFormat;

import jxl.Sheet;

import jxl.Workbook;

import jxl.read.biff.BiffException;

import vo.Logistics;

public class ReadExcel {

public Logistics readFromExcel(File file) throws BiffException, IOException

{

Logistics log=new Logistics();

//1:创建workbook

Workbook workbook=Workbook.getWorkbook(file);

//2:获取第一个工作表sheet

Sheet sheet=workbook.getSheet(0);

log.setSender(sheet.getCell(0,1).getContents());

log.setSenPhone(sheet.getCell(1,1).getContents());

log.setSenAdd(sheet.getCell(2,1).getContents());

log.setSenPost(sheet.getCell(3,1).getContents());

log.setRecipient(sheet.getCell(4,1).getContents());

log.setRecPhone(sheet.getCell(5,1).getContents());

log.setRecAdd(sheet.getCell(6,1).getContents());

log.setRecPoSt(sheet.getCell(7,1).getContents());

log.setRemarks(sheet.getCell(8,1).getContents());

log.setAmount(sheet.getCell(9,1).getContents());

log.setProduct(sheet.getCell(10,1).getContents());

int i=(int)(Math.random()\*900)+100;

SimpleDateFormat df = new SimpleDateFormat("yyyy MMdd hhmmss SSS");

String data=df.format(new java.util.Date())+i;

log.setTaskNO(data);

//最后一步：关闭资源

workbook.close();

return log;

}

}

## 

## 3.2.7 /ems/src/vo/Logistics.java

package vo;

public class Logistics {

String taskNO;

String sender;

String recipient;

String senPhone;

String recPhone;

String senAdd;

String recAdd;

String senPost;

String recPoSt;

String remarks;

String amount;

String product;

public Logistics(String taskNo,String sender, String recipient, String senPhone, String recPhone, String senAdd, String recAdd,

String senPost, String recPoSt, String remarks, String amount,String product) {

super();

this.taskNO=taskNo;

this.sender = sender;

this.recipient = recipient;

this.senPhone = senPhone;

this.recPhone = recPhone;

this.senAdd = senAdd;

this.recAdd = recAdd;

this.senPost = senPost;

this.recPoSt = recPoSt;

this.remarks = remarks;

this.amount = amount;

this.product = product;

}

public Logistics() {

super();

}

public String getTaskNO() {

return taskNO;

}

public void setTaskNO(String taskNO) {

this.taskNO = taskNO;

}

public String getSender() {

return sender;

}

public void setSender(String sender) {

this.sender = sender;

}

public String getRecipient() {

return recipient;

}

public void setRecipient(String recipient) {

this.recipient = recipient;

}

public String getSenPhone() {

return senPhone;

}

public void setSenPhone(String senPhone) {

this.senPhone = senPhone;

}

public String getRecPhone() {

return recPhone;

}

public void setRecPhone(String recPhone) {

this.recPhone = recPhone;

}

public String getSenAdd() {

return senAdd;

}

public void setSenAdd(String senAdd) {

this.senAdd = senAdd;

}

public String getRecAdd() {

return recAdd;

}

public void setRecAdd(String recAdd) {

this.recAdd = recAdd;

}

public String getSenPost() {

return senPost;

}

public void setSenPost(String senPost) {

this.senPost = senPost;

}

public String getRecPoSt() {

return recPoSt;

}

public void setRecPoSt(String recPoSt) {

this.recPoSt = recPoSt;

}

public String getRemarks() {

return remarks;

}

public void setRemarks(String remarks) {

this.remarks = remarks;

}

public String getAmount() {

return amount;

}

public void setAmount(String amount) {

this.amount = amount;

}

public String getProduct() {

return product;

}

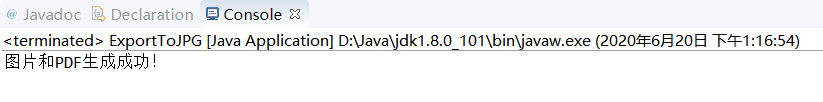
public void setProduct(String product) {

this.product = product;

}

}

# **4 系统测试**



# **5 总结**

生成.jpg文件的数值把控有些麻烦，需要多次调试，错的多了会比较烦，在网上查找一些例子很有帮助。