

# 目录

海运集装箱管理需求说明书

项目开发内容:

要求:

总体设计

运行设计

运行控制

源代码

main 包

Start程序运行入口

Gui 程序主界面

fil包

Data 提单数据

Parse解析提单

Count 统计

NewFile 新建提单

net包

Server服务器

Client客户端

# 海运集装箱管理需求说明书

## 项目开发内容:

=====

读取预配报文文件（参见 06062313、06062314），将报文重新格式化，并进行统计分析，然后将结果写入文件。  
文件体格式如下：

船名：航次：提单号：目的港#  
尺寸：箱型：数量：经营人#  
尺寸：箱型：数量：经营人#

TITANIC:603E:SNL        018576:SIGAPORE#  
20:GP:4:HUB#  
40:RF:5:SNL#

### 单船多票文件结构:

YUNFENG:603E:SNL	0185762:SIGAPORE#
20:GP:4:HUB#	
40:RF:5:SNL#	
YUNFENG:603E:JL	4547357:PUSAN#
20:GP:4:HUB#	
40:RT:5:JL#	
HANY ANG:603E:HUB	8576276:SIGAPORE#
20:RH:4:HUB#	
40:RF:5:SNL#	
YUNFENG:603E:SNL	4195325:CNSHA#
20:RF:4:HUB#	
40:GP:5:JL#	

### 提单类型的数据结构:

英文名称	中文名称
shipName	船名
voyage	航次
blno	提单号
destination	目的港
cnttype	箱型

cntsize	集装箱尺寸
cntqnt	箱量
cntoperator	箱经营人
remark	备注

要求：

1、从文件中读取数据，进行解析，并展示。形式如下：

船名	航次	提单号	目的港	尺寸	箱型	数量	经营人	备注
YUNFENG	603E	SNL 0185762	SIGAPORE	油罐箱 框架箱	保温集装箱	5	远洋海运公司	SOC
YUNFENG	603E	SNL 0185762	SIGAPORE	冷冻箱 冷高箱 油...	通用集装箱	4	大洋海运集团	SOC
YUNFENG	603E	SNL 0185762	SIGAPORE	油罐箱 框架箱	保温集装箱	5	远洋海运公司	SOC
YUNFENG	603E	SNL 0185762	SIGAPORE	冷冻箱 冷高箱 油...	GF	4	大洋海运集团	SOC
YUNFENG	603E	SNL 0185762	SIGAPORE	油罐箱 框架箱	保温集装箱	5	远洋海运公司	SOC
YUNFENG	603E	SNL 0185762	SIGAPORE	冷冻箱 冷高箱 油...	通用集装箱	4	大洋海运集团	SOC
YUNFENG	603E	SNL 0185762	SIGAPORE	冷冻箱 冷高箱 油...	干货高箱	4	大洋海运集团	SOC
YUNFENG	603E	SNL 0185762	SIGAPORE	冷冻箱 冷高箱 油...	通用集装箱	4	大洋海运集团	SOC

显示在 JTable中 。每一条数据，要显示箱子的信息和箱子所在船的信息。

2、对解析后数据要提供保存功能，保存到数据库中（现阶段利用文件替代数据，将数据保存在文件中）。

3、完成新建提单的功能。

4、完成统计功能。对解析后的内容进行统计。

统计如下功能：

- 1). 总共有多少个航次
- 3). 总共有多少票业务（按提单号统计）
- 6). 所有业务的箱型尺寸汇总（每个尺寸的箱子的个数）
- 7). 所有业务中有多少个箱子
- 8). 计算总共有多少个 TEU(twentyfoot equivalent un统计出尺寸是 20 的箱子的个数。
- 9) 统计每个经营人的数量。

5、将统计结果存入文件，同时显示在界面上。

统计信息				
航次	业务	尺寸(单位:英尺)	箱子	TEU
船名:YUNFENG 数...	业务数量:1	尺寸:20 数量:5 尺寸:40 数量:3	箱子总数:35	TEU数量:5
<input type="button" value="导出文件"/> <input type="button" value="关闭统计"/>				

6、Socket 数据传递，要在局域网中的不同计算机之间测试通过。（要求完成聊天和传送文件）

7、如果提单头是 SNL，HUB 要在记录的最后一行（备注）加上 SOC

8、实现翻译，根据下表，翻译对应的箱型，和尺寸。

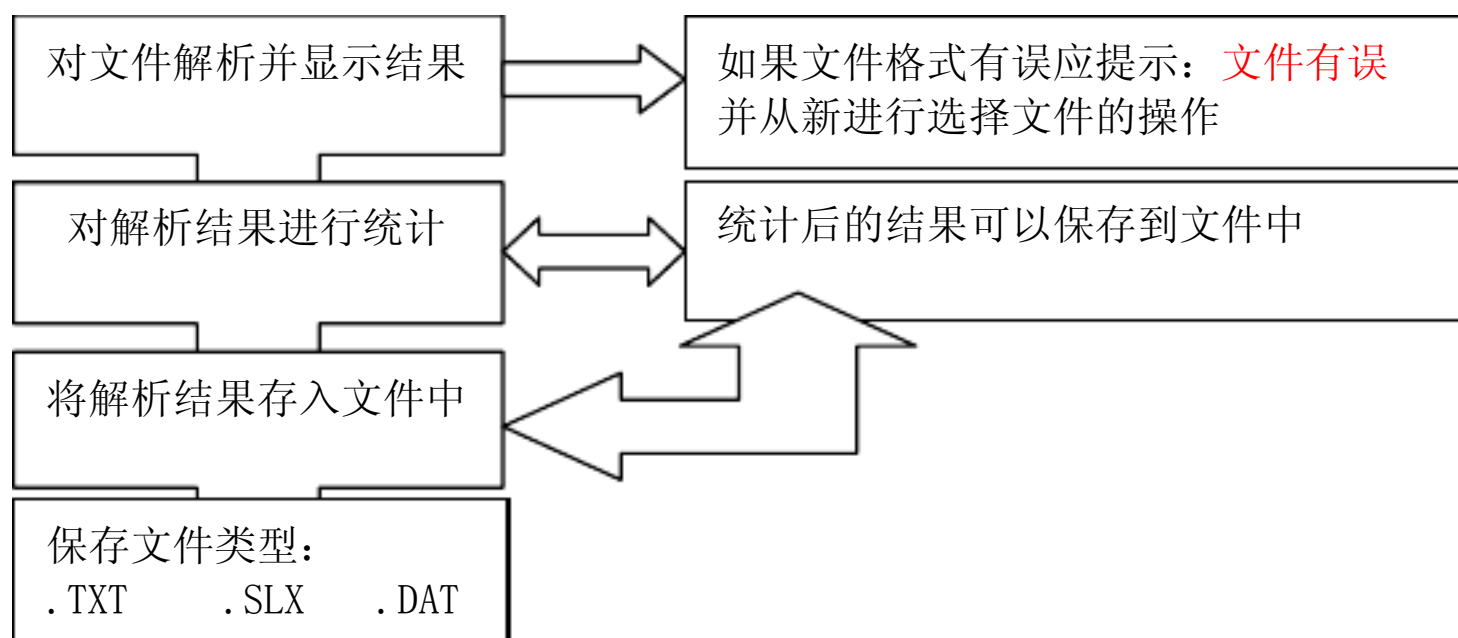
代码/原代号	箱型	箱型群组代码	主要特征	箱型代码	原代号
G/0	通用集装箱 (无通风装置)	GP		G0-G9	00-11
V/1	通风式通用集装箱	VH		V1-V9	13-19
B/2	干散货集装箱	BU, BK		B0-B9	20-24
S/2	以货物种类命名的集装箱	SN	汽车集装箱, 活鱼集装箱	S0-S8	25-29
R/3	保温集装箱	RE, RF, RT, RS, RH	制冷/加热	R0-R9	31-39
H/4	保温集装箱	HR, HI	外置式挂装制冷/加热装	H0-H9	40-49
U/5	敞顶式集装箱	UT	-端或两端开口	U0-U6	50-56
P/6	平台(和台架式)集装箱	PL	平台集装箱	P0-P9	60-69
T/7	罐式集装箱	TN	最低试验压力	T0-T9	70-79
A	空/陆/水联运集装箱	AS		A0	90

尺寸	箱型	对应类型	95 码
12 英尺	干货箱	GP	12G1
	干货高箱	GH (HC, HQ)	15G1
20 英尺	干货箱	GP	22G1
	干货高箱	GH (HC, HQ)	25G1
	挂衣	HT	22V1
	开顶箱	OT	22U1
	冷冻箱	RF	22R1
	冷高箱	RH	25R1
	油罐箱	TK	22T1

40 英尺	框架箱	FR	22P1
	干货箱	GP	42G1
	干货高箱	GH (, HQ)	45G1
	挂衣箱	HT	42V1
	开顶箱	OT	42U1
	冷冻箱	RF	42R1
	冷高箱	RH	45R1
	油罐箱	TK	42T1
45 英尺	框架箱	FR	42P1
	干货箱	GP	L2G`
	干货高箱	GH (HC, HQ)	L5G1
	挂衣箱	HT	L2V1
	开顶箱	OT	L2U1
	冷冻箱	RF	L2R1
	冷高箱	RH	L5R1
	油罐箱	TK	L2T1
	框架箱	FR	L2P1

9、不得在程序中出现业务数据或者业务文件名称等硬编码（配置文件可以），把所有需要硬编码的内容写入文件

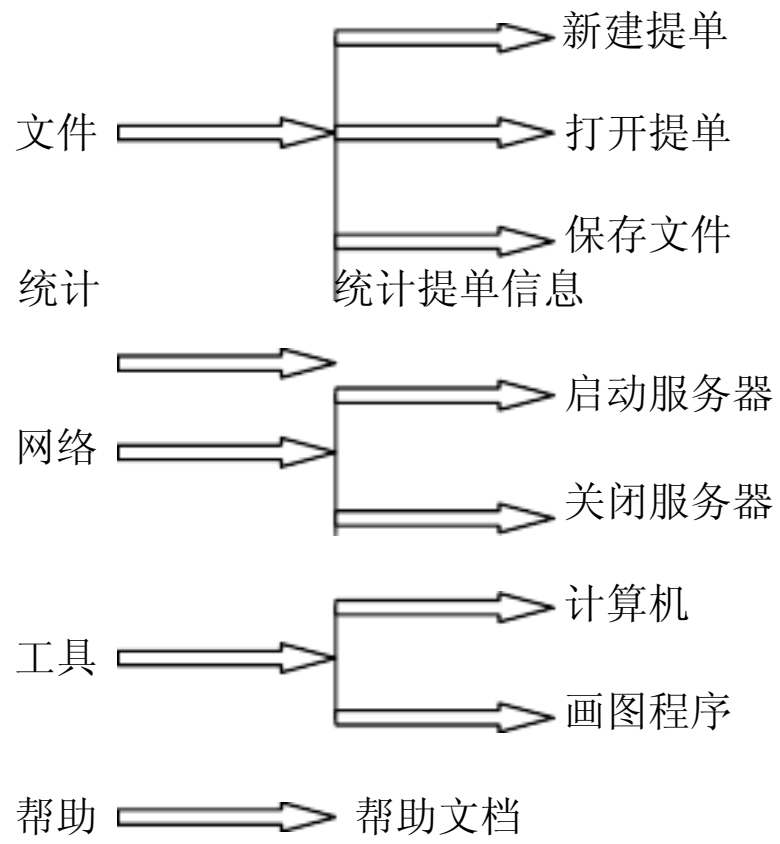
## 总体设计



## 运行设计

## 运行控制

双击该软件即可直接运行，点击界面当中的菜单即可进行相应的操作。  
菜单选项共分为：文件，统计，网络，工具，帮助（下图是表示菜单中的子菜单结构）



## 源代码

### main 包

#### Start程序运行入口

```
package main;

public class Start {
    public static void main(String[] args) {
        Gui g = new Gui();
        g.setVisible( true );
    }
}
```

#### Gui 程序主界面

```
package main;

import file.*;

import java.awt.Container;
import java.awt.event.ActionEvent;
```

```

import java.awt.event.ActionListener;

import java.io.*;

import javax.swing.*;

import net.*;

@SuppressWarnings("serial")
public class Gui extends JFrame implements ActionListener{
    JMenu file;
    JMenuItem newfile;
    JMenuItem openfile;
    JMenuItem savefile;
    JMenu count;
    JMenuItem countinfo;
    JMenu net;
    JMenuItem chat;
    JMenuItem server;
    JMenuItem client;
    JMenu tools;
    JMenuItem computer;
    JMenuItem draw;
    JMenu game;
    JMenuItem saolei;
    JMenu help;
    JMenuItem version;
    public Gui() {
        @SuppressWarnings("unused")
        Container con=this.getContentPane();
        this.setTitle("海运管理系统 @Zhao Tengyuan");
        this.setBounds(200, 200, 500, 500);
        JMenuBar bar = new JMenuBar();
        this.setJMenuBar(bar);
        file =new JMenu("文件");
        bar.add(file);
        newfile =new JMenuItem("新建");
        openfile = new JMenuItem("打开");
        savefile = new JMenuItem("保存");
        file.add(newfile);
        file.add(openfile);
        file.add(savefile);
        newfile.addActionListener(this);
        openfile.addActionListener(this);
        savefile.addActionListener(this);
        count = new JMenu("统计");
        bar.add(count);
        countinfo = new JMenuItem("统计信息");
        count.add(countinfo);
        countinfo.addActionListener(this);
        net = new JMenu("网络");
        bar.add(net);
        chat = new JMenuItem("聊天");
        server = new JMenuItem("服务器");
        client = new JMenuItem("客户端");
    }
}

```

```

net.add(chat);
net.add(server);
net.add(client);
chat.addActionListener(this);
server.addActionListener(this);
client.addActionListener(this);
tools = new JMenu("工具");
bar.add(tools);
computer = new JMenuItem("计算器");
draw = new JMenuItem("画图");
tools.add(computer);
tools.add(draw);
computer.addActionListener(this);
draw.addActionListener(this);
game = new JMenu("游戏");
bar.add(game);
saolei = new JMenuItem("扫雷");
game.add(saolei);
saolei.addActionListener(this);
help = new JMenu("帮助");
bar.add(help);
version = new JMenuItem("帮助信息");
help.add(version);
version.addActionListener(this);
}
JFrame jframe;用于主窗体
JTable jtable;用于显示打开内容
JScrollPane jScrollPane;用于添中 JTable
String[] headerSt;用于存储表头
Object[][] objSt;用于存储表体

Parse ps = new Parse();
public void actionPerformed(ActionEvent e) {
    if(e.getSource().equals(newfile)) {
        new NewFile();
    }
    if(e.getSource().equals(openfile)) {
        JFileChooser jfc = new JFileChooser("E:\\");
        int i = jfc.showOpenDialog(null);
        if (i == JFileChooser.APPROVE_OPTION) {
            File file = jfc.getSelectedFile();
            boolean isnot = true;
            String regex1 = "[a-zA-Z]+:[0-9a-zA-Z]+:[a-zA-Z\\s0-9]+:[a-zA-Z]+#";
            String regex2 = "[0-9]+:[a-zA-Z]+:[0-9]+:[a-zA-Z]+#";
            try {
                BufferedReader buf = new BufferedReader(new FileReader(file));
                //判断这个文件是否符合要求
                if (!buf.ready())判断是否为空
                    JOptionPane.showMessageDialog(null,"打开的文件格式错误!");
                isnot = false;
                return;
            } else {
                do {
                    String str = buf.readLine();
                    //判断读入的每一句是否为指定格式

```



```

        if ((!str.matches(regex1)) && (!str.matches(regex2))) {
            JOptionPane.showMessageDialog(null, "打开的文件格式错误!");
            isnot = false;
            return;
        }
    } while (buf.ready());
}
buf.close();
// 如果文件符合要求, 则调用解析程序分解文件内容
if (isnot) {
    BufferedReader bufReader = new BufferedReader(new FileReader(file));
    while (bufReader.ready()) {
        String str = bufReader.readLine();
        // 每读一句, 调用一次解析方法
        ps.reader_Parse(str);
    }
    bufReader.close();
}
} catch (Exception e1) {
    e1.printStackTrace();
}
}
objStr = ps.getObjStr; // 取得显示内容
headerStr = ps.getHeader; // 取得表头
jtable = new JTable(objStr, headerStr);

jtable.updateUI; // 当第二次打开文件时实现文件的更新
jtable.clearSelection();
jtable.setAutoResizeMode(JTable.AUTO_RESIZE_ALL_COLUMNS);
jtable.setRowHeight(30); // 设置表格的高度
jtable.getColumnModel().getColumn(0).setPreferredWidth(100); // 设置表格宽度
JScrollPane scrollpane = new JScrollPane(jtable);
this.add(scrollpane);
this.setVisible(true);
}
if (e.getSource().equals(savefile)) {
    JFileChooser jfc = new JFileChooser("E:\\");
    int jin = jfc.showSaveDialog(null);
    File file = jfc.getSelectedFile();
    if (jin == JFileChooser.APPROVE_OPTION) {
        try {
            // 保存 jtable 中的内容
            PrintStream p = new PrintStream(file);
            for (int i = 0; i < ps.getList().size(); i++) {
                p.println(ps.getList().get(i));
            }
        } catch (Exception e1) {
            JOptionPane.showMessageDialog(null, "无法保存!");
        }
    }
}
if (e.getSource().equals(countinfo)) {
    try {
        Count count = new Count();
        count.cFrame();
    }
}

```

```

        } catch (Exception e1) {
            JOptionPane.showMessageDialog(null, "无法进行统计");
        }
    }
    if (e.getSource().equals(chat)) {
        MyTCP mtcp = new MyTCP();
        mtcp.setVisible(true);
    }
    if (e.getSource().equals(server)) {
        Server s = new Server();
        s.serv();
    }
    if (e.getSource().equals(client)) {
        Client c = new Client();
        c.clientFrame();
    }
    if (e.getSource().equals(computer)) {
        try {
            Runtime.getRuntime().exec("calc");
        } catch (IOException e1) {
            JOptionPane.showMessageDialog(null, "No Found The Order!");
        }
    }
    if (e.getSource().equals(draw)) {
        try {
            Runtime.getRuntime().exec("mspaint");
        } catch (IOException e1) {
            JOptionPane.showMessageDialog(null, "No Found The Order!");
        }
    }
    if (e.getSource().equals(saolei)) {
        try {
            Runtime.getRuntime().exec("winmine");
        } catch (IOException e1) {
            JOptionPane.showMessageDialog(null, "No Found The Order!");
        }
    }
    if (e.getSource().equals(version)) {
        JOptionPane.showMessageDialog(null, "Version:          V          1.1\nAuthor:          Zhao
Tengyuan\nEmail:zhty2007@126.com");
    }
}
}
}

```

## file包

## Data 提单数据

```

package file;

// 海运提单数据项目
public class Data {

```

```
private String shipname ; // 船名
private String voyage ; // 航次
private String blno ; // 提单号
private String destination ; // 目的港
private String cnnttype ; // 箱型
private String cntsize ; // 集装箱尺寸
private int cntqnt ; // 箱量
private String cntoperator ; // 箱经营人
private String remark ; // 备注

public String getShipname() {
    return shipname ;
}
public void setShipname(String shipname) {
    this.shipname = shipname;
}
public String getVoyage() {
    return voyage ;
}
public void setVoyage(String voyage) {
    this.voyage = voyage;
}
public String getBlno() {
    return blno ;
}
public void setBlno(String blno) {
    this.blno = blno;
}
public String getDestination() {
    return destination ;
}
public void setDestination(String destination) {
    this.destination = destination;
}
public String getCnnttype() {
    return cnnttype ;
}
public void setCnnttype(String cnnttype) {
    this.cnnttype = cnnttype;
}
public String getCntsize() {
    return cntsize ;
}
public void setCntsize(String cntsize) {
    this.cntsize = cntsize;
}
public int getCntqnt() {
    return cntqnt ;
}
public void setCntqnt( int cntqnt) {
    this.cntqnt = cntqnt;
}
public String getCntoperator() {
    return cntoperator ;
}
public void setCntoperator(String cntoperator) {
    this.cntoperator = cntoperator;
}
public String getRemark() {
```

```

        return remark ;
    }
    public void setRemark(String remark) {
        this.remark = remark;
    }
    public String toString() {
        return "船名: " + shipname + " 航次: " + voyage + " 提单号: " + blno
            + " 目的港: " + destination + " 尺寸: " + cntsize + " 箱型: "
            + cnnttype + " 箱量: " + cntqnt + " 经营人: " + cntoperator
            + " 备注: " + remark ;
    }
}

```

## Parse 解析提单

```

package file;

import java.util.*;

// 代码解析
public class Parse {
    String shipname ;
    String voyage ;
    String blno ;
    String destination ;
    int sizeCount = 0; // 尺寸数量
    boolean isnot = false ; // 标记以前list 是否有值
    Data hData = new Data();
    private final static List<Data> list = new Vector<Data>();
    // 返回表头
    public String[] getHeader() {
        isnot = true ;
        String[] header = { "船名", "航次", "提单号", "目的港", "集装箱尺寸", "箱型", "箱量",
            "箱经营人", "备注" }; // 设置显示表头
        return header;
    }
    // 解析程序入口
    public void reader_Parse(String str) {
        if (isnot) {
            list.clear();
            isnot = false ;
        }
        String regex1 = "[a-zA-Z]+:[0-9a-zA-Z]+:[a-zA-Z\\s0-9]+:[a-zA-Z]+#" ;
        if (str.matches(regex1)) { // 判断此句是否为提单头
            this.headerParse(str);
        } else { // 去解析提单内容
            this.bodyParse(str);
        }
    }
    // 拆分提单头, 把相应内容存于数组中
    public void headerParse(String str) {
        String str1 = str.replace("#", "");
        String[] str2 = str1.split(":");
        shipname = str2[0];
        voyage = str2[1];
    }
}

```

```

    blno = str2[2];
    destination = str2[3];
}
// 拆分提单内容
public void bodyParse(String str) {
    Data hData = new Data();
    String str1 = str.replace("#", "");
    String[] str2 = str1.split(":");
    String perator = str2[3]; // 判断经营人是否为SNL 或是HUB
    String remark = "";
    if (perator.equals("SNL") || perator.equals("HUB")) {
        remark = "SOC";
    }
    hData.setCntsize(this.parseUP(str2[0])); // 调用解释方法取出对应的值
    hData.setCnnttype(this.parseUP(str2[1]));
    hData.setCntqnt(Integer.valueOf(str2[2]));
    hData.setCntoperator(this.parseUP(str2[3]));
    hData.setRemark(remark);
    hData.setShipname(shipname);
    hData.setVoyage(voyage);
    hData.setBlno(blno);
    hData.setDestination(destination);
    hData.getShipname();
    hData.getVoyage();
    hData.getBlno();
    hData.getDestination();
    hData.getCnnttype();
    hData.getCntsize();
    hData.getCntqnt();
    hData.getCntoperator();
    hData.getRemark();
    list.add(hData); // 将得到的每一条提单存入集合中
}
// 从配置文件中解释出对应的文字内容
public String parseUP(String str) {
    ResourceBundle re = ResourceBundle.getBundle("config.Haiyun");
    Set<String> set = re.keySet(); // 将配置文件键值的Set 视图
    Iterator<String> iter = set.iterator();
    while (iter.hasNext()) { // 在配置文件中查找键值对变的值
        if (iter.next().equals(str)) {
            str = re.getString(str);
        }
    }
    return str;
}
// 返回表单内容
public Object[][] getObjStr() {
    int length = list.size();
    Object[][] obj = new Object[length][9];
    // 从集合中取出数据转存到二维数组中
    Iterator<Data> iter = list.iterator();
    int i = 0;
    while (iter.hasNext()) {
        hData = iter.next();
        obj[i][0] = hData.getShipname();
        obj[i][1] = hData.getVoyage();
        obj[i][2] = hData.getBlno();
        obj[i][3] = hData.getDestination();
        obj[i][4] = hData.getCntsize();
    }
}

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/47623012100011010>