# 项目说明

1. **需求分析**

在实验室局域网中，输入指定ip段，如192.168.1.2-192.168.1.200

得到：

1. 计算机ip，
2. 开放的端口，
3. mac地址。

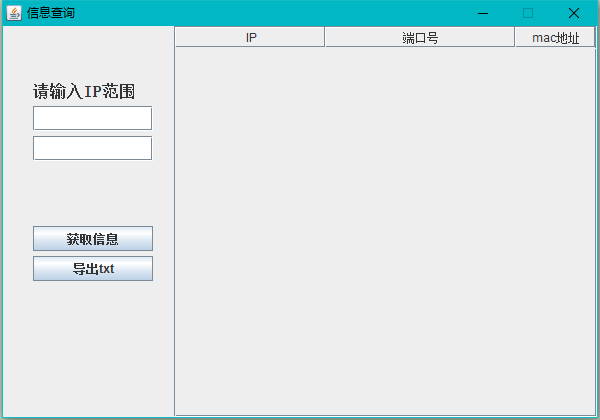
功能：

1. 可以在界面中显示，
2. 也可以导出为txt文档。

**要求：使用java语言，最好用myeclipse写。**

1. **页面说明**

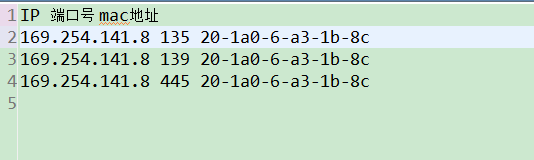
1.启动页面



1. 获取信息（ip地址，端口号、mac地址）



1. 导出txt



1. **功能具体说明**

**1、窗口属性设置类：PublicWindowSet**

**方法**：

（1）设置窗口大小：windowAttribute

//窗口大小

**public** **static** **void** windowAttribute(JFrame jfr, **int** wide, **int** high, String title) {

*windowIsCenter*(wide, high, jfr);

jfr.setTitle(title);

jfr.setSize(wide, high);

jfr.setVisible(**true**);

jfr.setResizable(**false**);

}

（2）设置窗口中心布局：windowIsCenter

//中心布局

**public** **static** **void** windowIsCenter(**int** windowWide, **int** windowHigh, JFrame frame) {

**int** screenWide = Toolkit.*getDefaultToolkit*().getScreenSize().width;

**int** screenHigh = Toolkit.*getDefaultToolkit*().getScreenSize().height;

**int** x = screenWide / 2 - windowWide / 2;

**int** y = screenHigh / 2 - windowHigh / 2;

frame.setLocation(x, y);

}

（3）设置窗提示窗口：promptPopUp

//提示面板

**public** **static** **void** promptPopUp(String content, String title, JFrame jfr) {

JOptionPane.*showMessageDialog*(jfr.getContentPane(), content, title, JOptionPane.***INFORMATION\_MESSAGE***);

}

**2、主窗口类：AdmintratorWindow**

属性：

//设置主窗口长和宽

**private** **static** **final** **int** ***WINDOWWIDE*** = 600;

**private** **static** **final** **int** ***WINDOWHIGH*** = 420;

//主窗口对象、右侧图表格式、图表对象

**private** JFrame frame = **new** JFrame();

**private** DefaultTableModel tableModel;

**private** JTable table = **null**;

//txt文件对象

**private** File file1 = **null**;

//标题控件

**private** JLabel jText = **null**;

//两个文本输入框（输入IP范围）

**private** JTextField jTextStart = **null**;

**private** JTextField jTextEnd = **null**;

//两个按钮控件（获取IP、端口、mac地址信息；输出txt文件）

**private** JButton getItemsBtn = **null**;

**private** JButton outputBtn = **null**;

//保存所有的信息条，保存单个信息条

**private** List<LinkedList<String>> strList = **new** LinkedList<LinkedList<String>>();

**private** LinkedList<String> infoList = **null**;

主要方法：

//主窗口构造方法

**public** AdmintratorWindow() {

**this**.liftInformation();//左侧交互面板

**this**.rigthInformation();//右侧图表显示面板

//设置主窗口名

PublicWindowSet.*windowAttribute*(**this**, ***WINDOWWIDE***, ***WINDOWHIGH***, "信息查询");

}

// 左边窗口交互栏

**public** **void** liftInformation() {

// 1).添加一个文本标签

jText = **new** JLabel("请输入IP范围");

jText.setFont(**new** Font("宋体", Font.***BOLD***, 16));

jText.setBounds(30, 40, 230, 50);

**this**.add(jText);

// 2).两个文本输入框

jTextStart = **new** JTextField();

jTextEnd = **new** JTextField();

jTextStart.setFont(**new** Font("宋体", Font.***BOLD***, 16));

jTextEnd.setFont(**new** Font("宋体", Font.***BOLD***, 16));

jTextStart.setBounds(30, 80, 120, 25);

jTextEnd.setBounds(30, 110, 120, 25);

**this**.add(jTextStart);

**this**.add(jTextEnd);

// 3).两个操作按钮

getItemsBtn = **new** JButton("获取信息");

outputBtn = **new** JButton("导出txt");

getItemsBtn.setBounds(30, 200, 120, 25);

outputBtn.setBounds(30, 230, 120, 25);

**this**.add(getItemsBtn);

**this**.add(outputBtn);

//添加监听响应

**this**.getItemsBtn.addActionListener(**this**);

**this**.outputBtn.addActionListener(**this**);

// 4).设置左右分隔标签

JLabel awayLabel = **new** JLabel();

awayLabel.setBounds(200, 50, 10, 450);

**this**.add(awayLabel);

}

// 右部列表信息

**public** **void** rigthInformation() {

// 设置JTabel的默认类型（数据来源、图表标题）

**this**.tableModel = **new** DefaultTableModel(**new** infoList().Info,

**new** infoList().titles);

// 添加JLabel组件

**this**.table = **new** JTable(**this**.tableModel);

//设置列宽度

table.getColumnModel().getColumn(0).setPreferredWidth(150);

table.getColumnModel().getColumn(1).setPreferredWidth(80);

table.getColumnModel().getColumn(2).setPreferredWidth(190);

// 禁止JLabel组件随着窗口的大小而改变

table.setAutoResizeMode(JTable.***AUTO\_RESIZE\_OFF***);

// JTabel组件表格的显示尺寸

table.setPreferredScrollableViewportSize(**new** Dimension(420, 0));

// 使列表不可以整列的进行移动

table.getTableHeader().setReorderingAllowed(**false**);

// 使列表框不可被编辑

table.~~enable~~(**false**);

// 给JTabel组件添加滑动条

JScrollPane scr = **new** JScrollPane(**this**.table);

// 将带滚动条的Tabel组件添加入面板的东部区域

**this**.add(scr, BorderLayout.***EAST***);

}

//图表数据信息

**class** infoList **extends** AbstractTableModel {

//标题

**public** String[] titles = { "IP", "端口号", "mac地址" };

//数据

**public** Object[][] Info = {}; //

**public** **int** getRowCount() {

**return** 0;

}

**public** **int** getColumnCount() {

**return** 0;

}

@Override

**public** Object getValueAt(**int** arg0, **int** arg1) {

**return** **null**;

}

}

//监听按钮响应方法

**public** **void** actionPerformed(ActionEvent e) {

// 1).“获取信息”按钮响应处理

**if** (e.getSource() == **this**.getItemsBtn) {

//获取文本起始IP地址

String start = jTextStart.getText();

String end = jTextEnd.getText();

getinfo(start, end, **this**);

}

// 2)."输出"响应按钮

**if** (e.getSource() == **this**.outputBtn) {

**try**

{

file1 = **new** File("信息.txt");

OutputStream outstream = **new** FileOutputStream(file1);

outstream.write("IP 端口号 mac地址".getBytes());

//按行输出信息（信息间隔为空格）

**for** (**int** rowNum = 0; rowNum < strList.size(); rowNum++) {

String StrLine = strList.get(rowNum).get(0)+" "

+strList.get(rowNum).get(1)+" "

+strList.get(rowNum).get(2);

outstream.write(StrLine.getBytes());

}

//刷新、关闭输出流

outstream.flush();

outstream.close();

//输出成功提示框

PublicWindowSet.*promptPopUp*("存储文件成功!!!", "提示", frame);

} **catch** (IOException ioe) {

ioe.printStackTrace();

}

}

}

//刷新图表信息

**private** **void** flushInfo() {

//清空图表历史信息

**while** (**this**.tableModel.getRowCount() > 0) {

**this**.tableModel.removeRow(**this**.tableModel.getRowCount() - 1);

}

//暂时存储单条信息

List<String> list = **new** LinkedList<String>();

//将strList信息显示到表格中

//strList = new LinkedList<LinkedList<String>>();

**if** (strList != **null**) {

**for** (**int** i = 0; i < strList.size(); i++) {

list = strList.get(i);

**this**.tableModel.addRow(**new** Object[] { list.get(0), list.get(1),

list.get(2) });

}

}

}

//“获取信息”方法

**private** **void** getinfo(String start, String end,

AdmintratorWindow admintratorWindow) {

List<String> ipList = **new** LinkedList<String>();

**int** k = 0;

k = start.lastIndexOf(".");

String index = start.substring(0, k + 1);

String lift = start.substring(k + 1, start.length());

String right = end.substring(k + 1, end.length());

**int** first = Integer.*valueOf*(lift);//端口起始

**int** last = Integer.*valueOf*(right);//端口结尾

//将ip范围添加到ipList中

**for** (**int** i = first; i <= last; i++) { // 对所有局域网Ip

String iip = index + i;

ipList.add(iip);

}

//获取信息（ip、端口、mac地址）

//依次遍历IP号-->遍历端口号（以一个端口为单位）

**for** (**int** j = 0; j < ipList.size(); j++) {

List<String> portList = **new** LinkedList<String>();

portList = getPorts(ipList.get(j));

**for** (**int** n = 0; n < portList.size(); n++) {

infoList = **new** LinkedList<String>();

// 1).添加IP号

infoList.add(ipList.get(j));

// 2).添加端口号

infoList.add(String.*valueOf*(portList.get(n)));

// 3).添加mac地址

**try** {

String mac = *getMacAddress*(ipList.get(j));

infoList.add(mac);

} **catch** (Exception e) {

infoList.add("null");

}

//将整条信息添加到strList中

strList.add(infoList);

}

}

//刷新加载显示到右侧图表中

flushInfo();

}

//获取端口号

**private** List<String> getPorts(String ipStr) {

List<String> result = **new** LinkedList<String>();

//遍历所有端口号0~65535

**for** (**int** port = 0; port <= 65535; port++) {

Socket connect = **new** Socket();

**try** {

connect.connect(**new** InetSocketAddress(ipStr, port), 100);

result.add(String.*valueOf*(port));

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

} **catch** (IOException e) {

}

}

**return** result;

}

//获取mac地址

**public** **static** String getMacAddress(String ip) **throws** SocketException,

UnknownHostException {

//获取网络接口

NetworkInterface ne = NetworkInterface.*getByInetAddress*(InetAddress

.*getByName*(ip));

//获取mac地址，并转换为16进制（以-间隔）

**byte**[] mac = ne.getHardwareAddress();

StringBuilder stringBuilder = **new** StringBuilder("");

**if** (mac == **null** || mac.length <= 0) {

**return** **null**;

}

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

**int** v = mac[i] & 0xFF;

String hv = Integer.*toHexString*(v);

**if** (hv.length() < 2) {

stringBuilder.append(0);

}

**if**(i == 0) {

stringBuilder.append(hv);

} **else** {

stringBuilder.append("-").append(hv);

}

}

**return** stringBuilder.toString();

}

获取ip、端口号、mac地址等信息：getinfo(String start, String end,

AdmintratorWindow admintratorWindow)

获取没占用的端口号：getPorts(String ipStr)

获取mac地址：getMacAddress(String ip)