java课程设计报告

浙江工业大学 基础课程大型实验

实验学期：2019/2020(1)

实验题目：幸运观众号码抽取器

学生姓名：陈昊天

学生学号：201806061201

学生班级：计算机科学学院软件工程1802班

提交日期：2019年12月14日

实验题目和要求

问题描述

开发一个幸运观众手机号码抽取的图形用户界面程序，要求可设定获奖主题、获奖等级和人数，可录入观众姓名、手机号存入文本文件中（每行放一个观众姓名和手机号）。启动开始后能随机抽取存在文本文件中的若干个观众手机号码，显示时隐藏最后两位号码，并将获奖主题、获奖等级和获奖名单、手机号、获奖时间等保存在获奖的文本文件中，并可查询历史抽奖情况。

实验环境

在Jetbrains Intellij IDEA下进行开发，操作系统为Windows10 64位操作系统。

实验进度说明

2019年11月29日 完成代码编写

2019年12月12日 编写实验报告

系统功能组成

系统有界面设计、事件逻辑、存储处理逻辑三个部分组成。其中，界面设计和事件逻辑主要集中于MainFrame，存储处理逻辑主要集中于Cache,History等类。

界面设计





布局采用了响应式的布局，将整个界面分成左边功能选择栏和右面内容显示栏两页。大多数空间的排布使用了拓展性最强的GridBagLayout，从而设计出丰富好看的页面。菜单的部分使用了CardLayout实现功能菜单的切换。

按照区块，又将界面分成抽奖设置和记录，人员管理，中奖详情显示三个部分。

**抽奖设置**

抽奖可设置的内容为标题，一等奖、二等奖和三等奖的个数。三个奖的总和必须大于零，每一项均为一个非负整数。并且通过事件触发的机制实现实时对输入内容的判断，如果输入非法，则会自动将抽奖按钮设置为不可点击。



点击抽奖按钮后，则会自动添加一条记录，并立即显示抽奖结果。

**人员管理**

人员管理包括导入和导出名单的功能，通过设计特殊的PeopleTableModel来实现动态添加和删除人员名单的功能。最后一行可以添加，添加后会自动添加空行。若输入非法，则自动将输入替换为空字符串，过滤错误的输入。同时会自动清除中间的空行。

**中奖详情**

中奖详情使用了表格的方式进行显示，可以自动将最后两个字符替换为\*\*。

存储处理逻辑

此软件对应的缓存类为Cache，提供了软件所需要的所有缓存。同时基于事件机制加入了实时保存设置信息功能。能够在下一次打开时自动恢复离开前的状态。同时Cache提供了三个表格之间交互的一种逻辑。

系统技术要点

本软件的整个实现方面主要有以下几个难点。

### 如何优雅的构造界面

写程序时找了很多资料，最后确认主要使用GridBagLayout作为主要的布局方式，因为其有功能性强，支持响应式的布局。在构造界面的过程中，使用了类似于分支构造的方法将一个主布局不断划为为两个子布局。在MainFrame中，以G开头的函数均为构造界面的函数。

GridBagLayout虽然以类似于表格形式进行布局，但是还是和表格的布局有很大的区别的，其最重要的特性可以设置某一个元素的**约束**，简而言之，就是以什么样的形式呈现布局。

在布局中有两步关键的操作，一个是将元素加入Panel以及设置约束，这两步的实例代码为。

root.setLayout(layout); *//设置布局方式*

root.add(leftPair.getKey()); *//在根元素中添加子元素*

layout.setConstraints(leftPair.getKey(),leftPair.getValue()); *//设置布局中一个元素的约束*

还有一步的代码为创建约束，下面为示例代码。

GridBagConstraints constraints = new GridBagConstraints(); *//创建一个GirdBagConstraints*

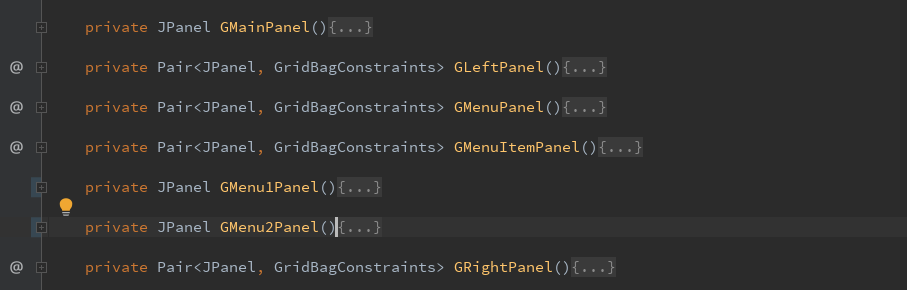
constraints.fill = GirdBagConstraints.BOTH; *//元素的填充方式，一般设置成BOTH，表示横向和纵向都自动填充剩余空间。*

constraints.gridwidth = 1; *//设置占据表格的格子数，如果该值为0，表示属于改行的尾部元素。*

constrainst.weightx = 0; *//窗体缩放时，元素宽度的调整系数。0表示固定宽度。*

constrainst.weighty = 1; *//窗体缩放是，元素高度的调整稀疏。*

在加上使用递归的方式构造界面，是可以打造出一个比较整齐的界面的。



并将需要注册事件的控件作为字段，使用registryEvent函数统一处理函数，并使用lambda实现简化方法的实现操作。

### 对JTable控件的深入学习与掌握

一开始认为JTable就是一个单纯显示表格状信息的一种控件，经过深入了解其背后的机制之后发现其还有很多有趣的功能。

第一个是JTable控件本身是与表头是分离的，所以在一开始编写的时候，发现并没有表头这个选项。经过查找相关资料之后解决了这个问题。关键代码如下。

JPanel tablePanel = new JPanel(new BorderLayout());

JScrollPane scrollPane = new JScrollPane(tablePanel);

tableHistory = new JTable(cache.histroyTableModel);

tablePanel.add(tableHistory.getTableHeader(), BorderLayout.NORTH);

tablePanel.add(tableHistory, BorderLayout.CENTER);

当然Table也是有几个关键的个性化设置的

table.setFont(new Font(Font.MONOSPACED, Font.PLAIN, 16)); *//设置字体*

table.setRowHeight(24); *//设置行高*

table.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION); *//设置只能选中单行*

table.getTableHeader().setReorderingAllowed(false); *//设置表格的列不能调整次序*

最核心的部分在于表格呈现数据的部分。有使用Object[][]数组来作为表格数据的，但是这种方法显示静态数据是比较不错的，但是如果要让表格的功能更加强大，我们可以实现AbstractTableModel，以扩展表格的功能。

下面是Abstract的关键成员函数

public class MyTableModel extends AbstractTableModel{

@Override

public boolean isCellEditable(int rowIndex, int columnIndex); *//设置每一个表格元素的编辑状态。*

@Override

public String getColumnName(int columnIndex); *//获取某一列的标题。*

@Override

public int getRowCount(); *//获取行的个数。*

@Override

public int getColumnCount(); *//获取列的个数。*

@Override

public Object getValueAt(int rowIndex, int columnIndex); *//访问表格某行某列的元素。*

@Override

public void setValueAt(Object aValue, int rowIndex, int columnIndex); *//设置表格某一行的元素。*

}

### 如何优雅的实现软件实时保存信息，能够及时保持软件的状态

软件的实时状态均存储在Cache类中，当发生更改时，自动触发保存文件的操作，从而做到能够实时保存信息的功能。在刚打开软件的时候，会首先从文件恢复软件的状态。

### 如何优雅的实现人员的管理

最后使用重写AbstractTableModel来解决这个问题，方便有快捷。其关键的函数在于setValueAt()，可以实现动态管理名单的功能。

@Override

public void setValueAt(Object aValue, int rowIndex, int columnIndex) {

String value = (String)aValue;

if(columnIndex == 1){ *//手机号判断，如判断为非法手机号，自动置为空*

if(!Checker.isPhoneNumber(value))

value = "";

} else {

if(!Checker.isName(value))

value = "";

}

String[] element = data.get(rowIndex);

element[columnIndex] = value;

if(rowIndex == data.size() - 1){ *//为最后一项*

if(!isElementEmpty(element)){ *//不为空*

data.setElementAt(element, rowIndex);

data.add(new String[]{"",""});

fireTableCellUpdated(rowIndex, columnIndex);

fireTableRowsInserted(rowIndex + 1, rowIndex + 1);

}

} else {

if (isElementEmpty(element)) {

data.remove(rowIndex);

fireTableRowsDeleted(rowIndex, rowIndex);

} else {

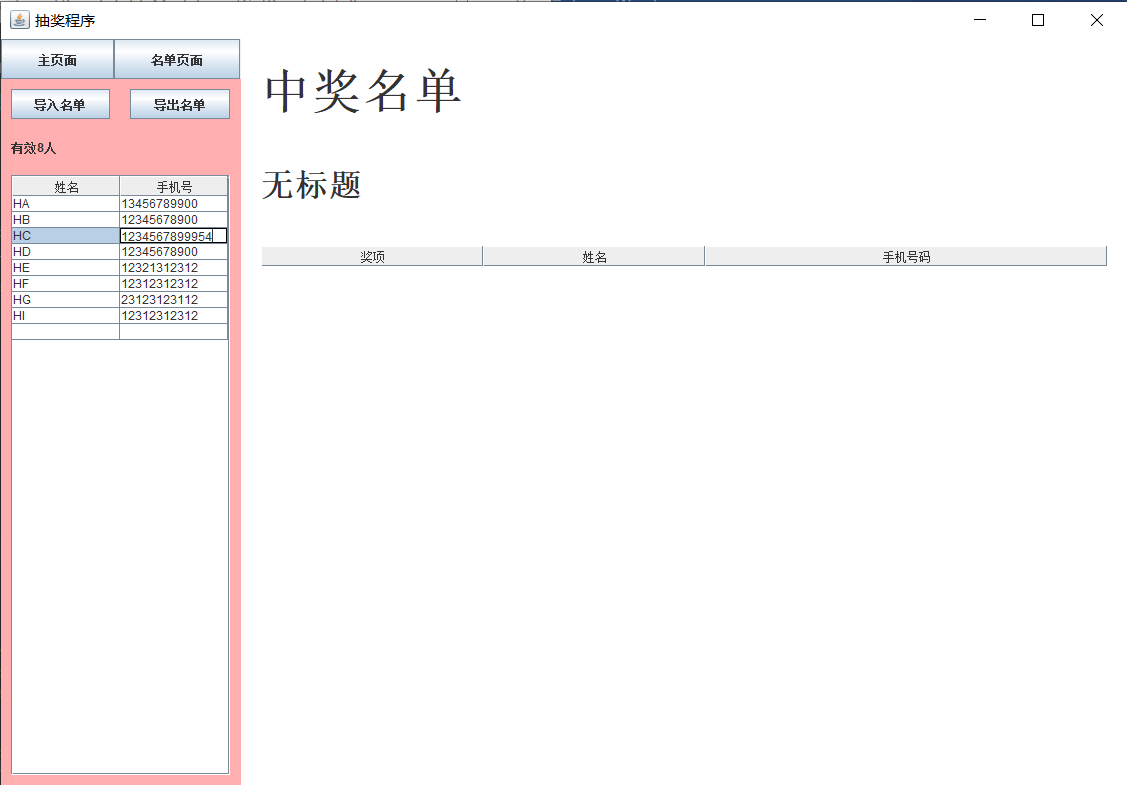
data.setElementAt(element, rowIndex);

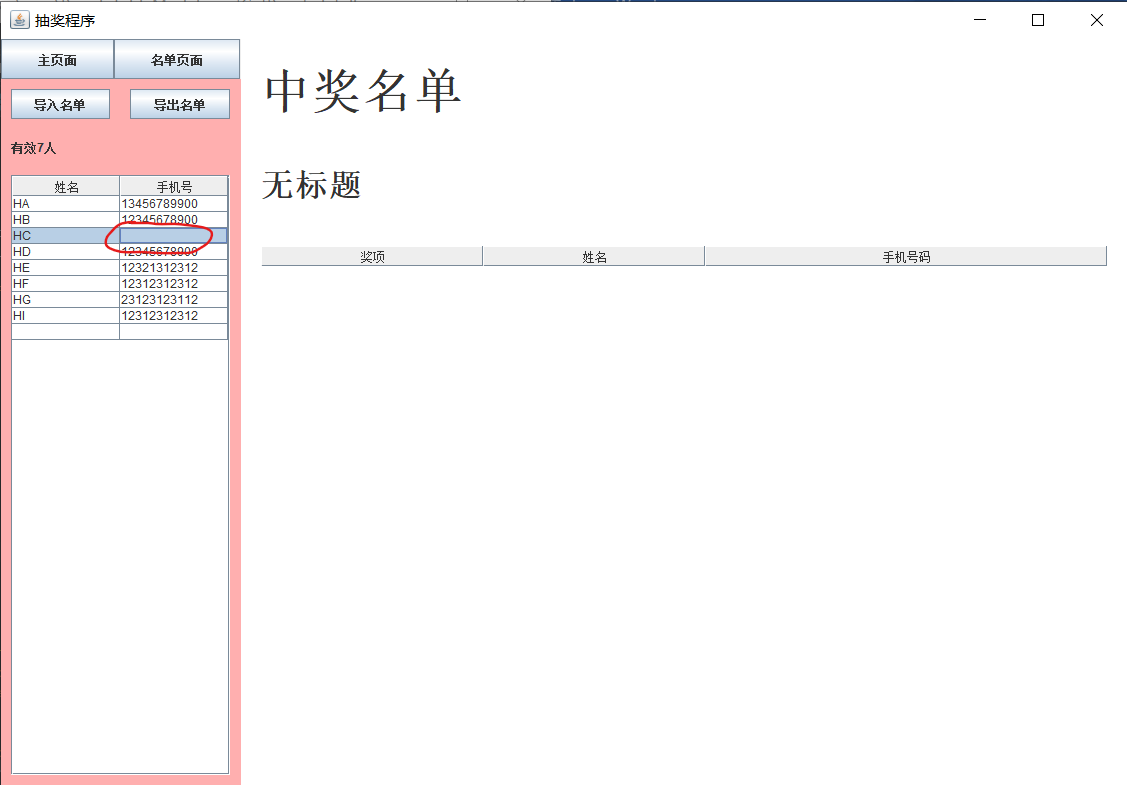
fireTableCellUpdated(rowIndex, columnIndex);

}

}

}





如上面两幅图所示，在输入错误的数据后，其会自动将错误数据重置为空。另外，添加数据，移除数据也可以通过类似的原理得到自动的调整，限于篇幅的原因，再次不做介绍。

心得总结

swing设计的界面可以很好的做到响应式的功能，但是布局并没有配有较好的实时预览工具和图形化搭建工具。单纯依靠代码来搭建布局的效率比较低。

同时，通过此课程设计功能主要复习了文件缓存、文件IO操作，表格的使用方法、事件监听器的实现方法、各种常见布局的使用方法。

所以，通过本次实验，还是收获到很多的知识的。