**学生信息管理系统**

**系统说明报告**

**姓名：薛雄**

**学号:20163776**

# 1项目目的与意义

## 1.1项目背景说明

设计一个投票管理系统界面，输入备选人后，点击确认，选择框内出现备选人的姓名。点击取消按钮候选人输入框为空，重新输入候选人名单，选中候选人的选择框时，应该给候选人一票。点击确认时表示产生一张选票。点击刷新按钮回到程序的初始界面重新设置候选人。根据选择框的状态变化统计出候选人的最后得票数，并可按选取多少点击排序按钮对候选人进行排序。

## 1.2项目目的与意义

目的：

⑴ 复习、巩固 Java 语言的基础知识，进一步加深对 Java 语言的理解和掌握；

⑵ 大作业为学生提供了一个既动手又动脑，独立实践的机会，将课本上的

理论知识和实际有机的结合起来，锻炼学生的分析解决实际问题的能力。提高学

生适应实际，实践编程的能力；

⑶ 培养学生在项目开发中寻求合作的意识、创新意识及能力

意义：

熟悉java语言的基础知识，联系实际生活，进一步提高学生的软件开发技术，培养我们的分析、解决问题的能力，提高我们实践论文撰写能力。

# 2 软件开发环境与技术说明

## 2.1软件开发环境 操作系统: Microsoft Windows 10 程序语言:Java 开发工具: MyEclipse 10

## 2.2软件开发技术描述

# 开发软件用的是MyEclipse10，项目开发时用到了多文件的方法，采用重载类的形式实现，界面设计的话，采用了GUI图形界面设计。

# 3系统分析与设计

## 3.1项目需求分析说明

对于学生信息的处理，此系统可方便管理学生信息。

## 3.2系统结构设计图

**投票系统**

**结果排序**

**投票显示**

**投票结束**

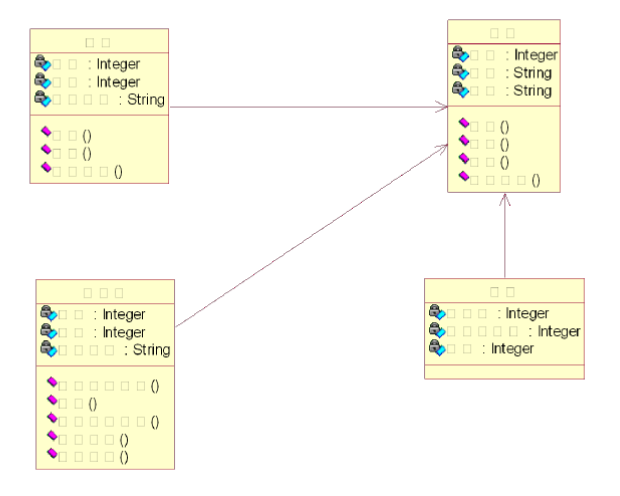
**投票选择**

**备选显示示**

**备选输入**

## 3.3系统用例图：

**3.4系统类图：**



## 3.5系统详细介绍：

设计一个投票管理系统界面，输入备选人后，点击确认，选择框内出现备选人的姓名。点击取消按钮候选人输入框为空，重新输入候选人名单，选中候选人的选择框时，应该给候选人一票。点击确认时表示产生一张选票。点击刷新按钮回到程序的初始界面重新设置候选人。根据选择框的状态变化统计出候选人的最后得票数，并可按选取多少点击排序按钮对候选人进行排序。

# 4系统源代码

## 4.1系统源代码文件说明

先制作主界面，然后再实现各个按钮的功能。

界面设计时，先定义好各个部件：标签Label、文本框TextFiled、按钮Button、复选框Checkbox、Scroll滚动条，我把整个界面分成很多个小面板，每一行部件都在一个小面板上面，最后将这些小面板一个个添加上去add（），像流式布局FlowLayout、方位布局BorderLayout、网格布局GridLayout主要是用到了这三种布局，中间那几块面板通过setBackground（）将其背景色改为绿色。经过不断修改测试运行，一个投票系统的主界面就做出来了。

接下来就是对各个功能的实现了，定义几个类变量来记录投票相关记录

如：

Checkbox candidate[]         //选择框数组，代表候选人

TextField personvote[]  //文本条数组，显示每个人的得票情况

int count[]                            //记录每个人的得票数

int totalvote=0,                             //总票数

int peoplenumber=0;                        //候选人个数

int count1=0,invalidatedTicket=0,abstention=0; //分别表示选的人数，废票数，弃权票数

对各个按钮进行注册监听addActionListener（）

在输入候选人名单时是输入一串名字，所以要从这一串名字中单个的提取出来并统计出候选人个数，在完成这项功能时要用到StringTokenizer类。

String g=canditate.getText();//获取输入的候选人

StringTokenizer st=new StringTokenizer(g);//字符串分析器

peoplenumble=st.countTokens();//统计候选人数 int i=0;

while(st.hasMoreTokens())

{   candidatelist[i]=st.nextToken();

 i++;}//获取语言符号（候选人名单）  点击确认按钮后，根据候选人个数添加单选按钮，并以候选人名字作为单选按钮的标签名。

  for(int j=0;j<10;j++)

candidate[j].setLabel(candidatelist[j]); //将候选人名单添加到复选框里 将多余的选框设置为不可见

for(int j=peoplenumble;j<10;j++)

candidate[j].setVisible(false)  点击“取消”的话输入候选人的文本框为空，此时可以重新输入，点击下面“确定”则统计选了多少人并通过这个判断是废票还是弃权，然后分别给候选人增加票数并输出相应的信息

；

for(int j=0;j<10;j++)

{

if(candidate[j].getState())

      count1++;

}//统计选了多少人  if(count1==0) abstention++;//一个都没选则选票作为弃权

if(count1>3) invalidatedTicket++;//选的人数超过3个则选票作废

if(count1<=3&&count1>0)

{    for(int j=0;j<peoplenumble;j++)

if(candidate[j].getState())

count[j]++;}//统计候选人所得票数

out.setText("已经统计了:"+totalvote+"张选票，其中弃权票:"+abstention+"作废票:"+invalidatedTicket);//输出统计结果

for(int j=0;j<peoplenumble;j++)

personvote[j].setText(""+candidatelist[j]+"得票 数:"+count[j]);//输出各个候选人得票数  点击“排序”则比较各个候选人的票数由高到低进行排序

for(int j=0;j<peoplenumble;j++)

for(int i=j+1;i<peoplenumble;i++)

if(count[j]<count[i])

{       m=count[j];count[j]=count[i];count[i]=m;

 n=candidatelist[j];

candidatelist[j]=candidatelist[i];

candidatelist[i]=n;        }//按得票数由多到少进行排序  第 6 页   “刷新”按钮通过清空所有文本框中的数据，所有变量变为初始化值；

“使用说明”通过弹出Frame窗体来实现，在该窗体中新建了几个文本框TextFeild和Scroll滚轮用来显示帮助信息。

“保存结果”也是通过弹出Franme窗体，窗体里面有个文本域TextArea，用来输出统计的结果，在文本域中，用户可以复制里面的内容然后保存在其他文件中。

## 4.2源代码

mport java.awt.\*;

import java.util.\*;

import java.applet.\*;

import java.awt.event.\*;

public class operation extends Applet implements ActionListener{

/\*\*

\*

\*/

private static final long serialVersionUID = -1875379925164868258L;

Label hint,result,notice,banquan;

TextField canditate; //输入候选人文本框

TextField out; //显示选举结果的文本框

Button confirm1,cancle,confirm2,refresh,sort;//分别表示确认、取消、确定、刷新、排序

Button help; //使用说明

Button save; //保存统计结果

Checkbox candidate[]=new Checkbox[10]; //选择框数组，代表候选人

TextField t1,t2,t3,t4,t5,t6,t7,t8,t9,t10;

TextField

personvote[]={t1,t2,t3,t4,t5,t6,t7,t8,t9,t10}; //文本条数组，显示每个人的得票情况

String candidatelist[]=new String[10]; //候选人名单

int count[]={0,0,0,0,0,0,0,0,0,0}; //记录每个人的得票数

int totalvote=0; //总票数

int peoplenumble=0; //候选人个数

int count1=0,invalidatedTicket=0,abstention=0; //分别表示选的人数，废票数，弃权票数

class Help extends Frame

{

/\*\*

\*

\*/

private static final long serialVersionUID = -3125967631941144662L;

//“使用说明”的弹出窗体

Panel p=new Panel();

TextField help[]=new TextField[6];

void help()

{

//super("使用说明");

p.setLayout(new GridLayout(6,1));

for(int i=0;i<6;i++)

help[i]=new TextField(10);

for(int i=0;i<6;i++)

{

p.add(help[i]);

}

ScrollPane scroll=new ScrollPane();

scroll.add(p);

add(scroll);

help[0].setText("请写入十个候选人");

help[1].setText("1:在文本框中输入候选人名单，点击“确认”以完成候选人的设置，点击“取消”可以重新设置候选人。");

help[2].setText("2:对候选人进行投票，点击下面的“确定”以确认选票。(注意：每点一次确定将产生一张选票！)");

help[3].setText("3:确定选票后，会自动统计结果，点击“排序”可以对候选人所得的票数由高到低进行排序。");

help[4].setText("4:点击“刷新”可以重新设置候选人，并开始新的一轮投票");

help[5].setText("5:在任何时候可以点击“使用说明”来查看帮助，点击“保存结果”，可以将统计以文本的形式显示出来。");

setSize(600,200);

setVisible(true);

addWindowListener(new closeWin());

}

class closeWin extends WindowAdapter

{

public void windowClosing(WindowEvent e)

{

Window w=e.getWindow();

w.dispose();

}

}

}

class Save extends Frame

{ //“保存结果”的弹出窗体

/\*\*

\*

\*/

private static final long serialVersionUID = -27189968758766359L;

TextArea save;

Save()

{

super("统计结果");

save=new TextArea(11,1);

add(save);

save.setText(out.getText()+'\n'+personvote[0].getText()+'\n'+personvote[1].getText()+'\n'+personvote[2].getText()+'\n'+personvote[3].getText()+'\n'+personvote[4].getText()+'\n'+personvote[5].getText()+'\n'

+personvote[6].getText()+'\n'+personvote[7].getText()+'\n'+personvote[8].getText()+'\n'+personvote[9].getText()+'\n');

setSize(300,300);

setVisible(true);

addWindowListener(new closeWin());

}

class closeWin extends WindowAdapter

{

public void windowClosing(WindowEvent e)

{

Window w=e.getWindow();

w.dispose();

}

}

}

public void init()

{

hint=new Label("首先输入候选人的名字（人数不超过10，名字之间用空格分隔）:");

notice=new Label("用下面的选择框统计选票(最多选3人):");

result=new Label("选举结果:");

banquan=new Label("版权所属：薛雄");

canditate=new TextField(50);

confirm1=new Button("确认");

cancle=new Button("取消");

confirm2=new Button("确定");

refresh=new Button("刷新");

sort=new Button("排序");

confirm2.setEnabled(false);

refresh.setEnabled(false);

sort.setEnabled(false);

help=new Button("请写入十个候选人");

save=new Button("保存结果");

save.setEnabled(false);

out=new TextField(50);

for(int i=0;i<10;i++)

personvote[i]=new TextField(80);

Panel p=new Panel();

Panel p1=new Panel();

Panel p2=new Panel();

Panel p3=new Panel();

Panel p4=new Panel();

Panel p5=new Panel();

Panel p6=new Panel();

Panel p7=new Panel();

Panel pa=new Panel();

Panel pb=new Panel();

Panel pc=new Panel();

setLayout(new BorderLayout());

pa.setLayout(new GridLayout(7,1));

pb.setLayout(new BorderLayout());

p4.setLayout(new GridLayout(1,5));

p5.setLayout(new GridLayout(1,5));

p1.add(hint);

p2.add(canditate);

p2.add(help);

p3.add(confirm1);

p3.add(cancle);

p3.add(notice);

p4.setBackground(Color.green);

p5.setBackground(Color.green);

p6.setBackground(Color.green);

for(int i=0;i<5;i++)

{ //创建候选人选项

candidate[i]=new Checkbox(candidatelist[i]);

p4.add(candidate[i]);

}

for(int i=5;i<10;i++)

{ //创建候选人选项

candidate[i]=new Checkbox(candidatelist[i]);

p5.add(candidate[i]);

}

for(int j=0;j<10;j++)

{

candidate[j].setEnabled(false);

}

p6.add(confirm2);

p6.add(refresh);

p6.add(sort);

p7.add(result);

p7.add(out);

p7.add(save);

pa.add(p1);

pa.add(p2);

pa.add(p3);

pa.add(p4);

pa.add(p5);

pa.add(p6);

pa.add(p7);

p.setLayout(new GridLayout(10,1));

for(int i=0;i<10;i++)

{

p.add(personvote[i]);

}

ScrollPane scroll=new ScrollPane();

scroll.add(p);

pc.add(banquan);

pb.add("Center",scroll);

pb.add("South",pc);

add("Center",pa);

add("South",pb);

confirm1.addActionListener(this);

cancle.addActionListener(this);

confirm2.addActionListener(this);

refresh.addActionListener(this);

sort.addActionListener(this);

help.addActionListener(this);

save.addActionListener(this);

} //面板的布局

public void actionPerformed(ActionEvent e)

{ //注册监听

String s=e.getActionCommand();

if(s.equals("确认"))

{

confirm1.setEnabled(false);

save.setEnabled(true);

confirm2.setEnabled(true);

refresh.setEnabled(true);

sort.setEnabled(true);help.

setEnabled(true);

String g=canditate.getText(); //获取输入的候选人

StringTokenizer st=new StringTokenizer(g); //字符串分析器

peoplenumble=st.countTokens(); //统计候选人数

int i=0;

while(st.hasMoreTokens())

{

candidatelist[i]=st.nextToken();

i++;} //获取语言符号（候选人名单）

for(int j=0;j<10;j++)

candidate[j].setLabel(candidatelist[j]);// 将候选人名单添加到复 选框里

for(int j=0;j<peoplenumble;j++)

candidate[j].setEnabled(true);

for(int j=peoplenumble;j<10;j++)

candidate[j].setVisible(false);//多余的选框设置为不可见

}

if(s.equals("取消"))

{ //重新设置候选人，进行重新投票

confirm1.setEnabled(true);

canditate.setText("");

}

if(s.equals("确定"))

{

totalvote++;

count1=0;

sort.setEnabled(true);

for(int j=0;j<10;j++)

{

if(candidate[j].getState())

count1++;

} //统计选了多少人

if(count1==0) abstention++; //一个都没选则选票作为弃权

if(count1>3)

invalidatedTicket++; //选的人数超过3个则选票作废

if(count1<=3&&count1>0)

{

for(int j=0;j<peoplenumble;j++)

if(candidate[j].getState())

count[j]++;

}//统计候选人所得票数

for(int i=0;i<10;i++)

candidate[i].setState(false);

for(int j=0;j<10;j++)

{

candidate[j].setState(false);

} //清空选框中的勾

out.setText("已经统计了:"+totalvote+"张选票，其中弃权票:"+abstention+"作废票:"+invalidatedTicket); //输出统计结果

for(int j=0;j<peoplenumble;j++)

personvote[j].setText(""+candidatelist[j]+"得票数:"+count[j]);//输出各个候选人得票数

if(s.equals("刷新"))

{

confirm1.setEnabled(true);

confirm2.setEnabled(false);

refresh.setEnabled(false);

sort.setEnabled(false);

save.setEnabled(false);

totalvote=0;

peoplenumble=0;

count1=0;

invalidatedTicket=0;

abstention=0;

canditate.setText("");

out.setText("");

for(int j=0;j<10;j++)

{

candidate[j].setState(false);

}

for(int j=peoplenumble;j<10;j++)

candidate[j].setVisible(true);

for(int j=0;j<10;j++)

candidatelist[j]="";

for(int j=0;j<10;j++)

count[j]=0;

for(int j=0;j<10;j++)

candidate[j].setLabel(candidatelist[j]);

for(int j=0;j<10;j++)

personvote[j].setText("");

}

if(s.equals("排序"))

{

sort.setEnabled(false);

int m;

String n;

for(int j=0;j<peoplenumble;j++)

for(int i=j+1;i<peoplenumble;i++)

if(count[j]<count[i])

{

m=count[j];

count[j]=count[i];

count[i]=m;

n=candidatelist[j];

candidatelist[j]=candidatelist[i];

candidatelist[i]=n;

} //按得票数由多到少进行排序

for(int j=0;j<peoplenumble;j++)

personvote[j].setText(""+candidatelist[j]+"得票数:"+count[j]); //输出排序后各候选人的票数

}

if(s.equals("使用说明"))

{

Help qwe=new Help();

qwe.help();

}

if(s.equals("保存结果"))

{

new Save();

}

}

}

}

# 5系统使用说明书

1. 打开程序
2. 输入候选人的名单
3. 点击确定，进行选票
4. 勾选人名，按确认并刷新
5. 排序，下方会出现结果

例图：





