package MyMicrowave;

import java.awt.\*;

import java.awt.event.\*;

import javax.swing.\*;

class time\_progressbar extends Thread{

JProgressBar mw\_progressbar;//进度条

int time;

int second = 0;

JTextArea mw\_info;

//只为模拟加工完成 自动关机

JButton start\_button;

boolean easy\_start;

int []mode\_num;

JTextField time\_field;

int temperature;

JTextField fire\_field;

//只为模拟自动关机

public time\_progressbar(JProgressBar mw\_progressbar,int time,JTextArea mw\_info, JButton start\_button, boolean easy\_start, int []mode\_num, JTextField time\_field, int temperature, JTextField fire\_field) {

this.mw\_progressbar = mw\_progressbar;

this.time = time;

this.mw\_info = mw\_info;

this.start\_button = start\_button;

this.easy\_start = easy\_start;

this.mode\_num = mode\_num;

this.time\_field = time\_field;

this.temperature = temperature;

this.fire\_field = fire\_field;

}

//加工完成 自动关机

public void turn\_off(){

start\_button.setText("开机");

easy\_start = true;

for(int i = 0;i<3;i++)

mode\_num[i] = 0;

time = 10;

time\_field.setText(new String().valueOf(time));

temperature = 100;

fire\_field.setText(new String().valueOf(temperature));

}

public void run(){

// System.out.println("测试 \_关键指示: "+stop\_start);

while(true){

//System.out.println("测试 \_关键指示: "+stop\_start);

mw\_progressbar.setStringPainted(true);

mw\_progressbar.setString("剩余时间: "+time+" min "+second+" s");

mw\_progressbar.setIndeterminate(true);//进度条左右移动

if(time == 0 && second == 0){

mw\_progressbar.setIndeterminate(false);

mw\_progressbar.setString("加工完成!");

mw\_info.setText("");

mw\_info.append("\n\n加工完成!快拿出去吃!");

turn\_off();

break;

}

if(second == 0){

time--;

second = 60;

}

second--;

try {

Thread.sleep(1000);

}

catch (InterruptedException e) {

e.printStackTrace();

}

}//while

System.out.println("线程结束");

}//run

}

class work{

JButton [] mw\_button = new JButton[8];//功能按钮集

JButton [] num\_button = new JButton[10];//数字按钮集

String [] mode\_string = {"热牛奶","烤肉串","蒸米饭","烤面包","烤红薯","烤鱼","爆米花","烤排骨"};

String [] num\_string = {"1","2","3","4","5","6","7","8","9","0"};

JFrame mw\_windows;

JPanel mw\_panel;

JPanel button\_panel;//放置按钮的容器

JPanel pic\_panel;//设置图片的容器

JLabel pic\_label;//图片标签

JButton start\_button;

//处理时间 和火力大小

JPanel field;

JTextField time\_field;

JLabel time\_label;

JTextField fire\_field;

JLabel fire\_label;

JPanel zero\_delete;//盛放0和删除的容器

JButton delete;//删除按钮

JTextArea mw\_info;//状态信息

int temperature = 100;//温度100°C

int time = 10;//时间10min

int flag = 0;//线程数组使用

int[] mode\_num = {0,0,0};//选择的功能 不到达指定的数量 无法开始工作 1为工作先决条件 选择物品 模拟放入的食物

boolean easy\_start = true;//开始/停止

boolean easy\_field =false;//时间 温度文本框

JPanel num\_panel;//放置数字按钮的容器

JProgressBar mw\_progressbar;//进度条

int j = 0;

int \_j = 0;

time\_progressbar[] work\_thread;

String temp\_mode;//用来记录选取的食物 在状态信息面板中显示出啦

time\_progressbar temp;

public work(JFrame mw\_windows,JPanel mw\_panel) {

//状态信息面板 模拟显示屏

mw\_info = new JTextArea();

mw\_info.setBackground(Color.pink);

mw\_info.setBounds(740, 70, 200, 100);

mw\_info.setEditable(false);

mw\_info.append("欢迎使用!");

mw\_panel.add(mw\_info);

this.mw\_windows = mw\_windows;

this.mw\_panel = mw\_panel;

time\_label = new JLabel("时间(min):");

fire\_label = new JLabel("温度(℃):");

time\_field = new JTextField();//时间文本框

time\_field.setFocusable(true);

time\_field.requestFocus();

time\_field.addMouseListener(new MouseListener() {

public void mouseReleased(MouseEvent e) {

}

public void mousePressed(MouseEvent e) {

}

public void mouseExited(MouseEvent e) {

}

public void mouseEntered(MouseEvent e) {

}

public void mouseClicked(MouseEvent e) {

time\_field.setFocusable(true);

time\_field.requestFocus();

easy\_field = true;

}

});

fire\_field = new JTextField();//温度文本框

fire\_field.addMouseListener(new MouseListener() {

public void mouseReleased(MouseEvent e) {

}

public void mousePressed(MouseEvent e) {

}

public void mouseExited(MouseEvent e) {

}

public void mouseEntered(MouseEvent e) {

}

public void mouseClicked(MouseEvent e) {

fire\_field.setFocusable(true);

fire\_field.requestFocus();

easy\_field = false;

}

});

//添加时间、温度

field = new JPanel(new GridLayout(2,2,15,5));

field.setBackground(new Color(100, 180, 180));

field.add(time\_label);

field.add(time\_field);

field.add(fire\_label);

field.add(fire\_field);

field.setBounds(560, 100, 145, 50);

mw\_panel.add(field);

mw\_progressbar = new JProgressBar(1,100);

mw\_progressbar.setBackground(Color.YELLOW);

mw\_progressbar.setBounds(560, 10, 400, 30);

mw\_panel.add(mw\_progressbar);

start\_button = new JButton("开机");

start\_button.setFocusPainted(false);//好看的按钮(没有边框)

start\_button.setBackground(Color.GRAY);//按钮颜色

start\_button.setBounds(880, 210, 80, 75);

//开始按钮的相关时间监听

start\_button.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

time\_progressbar work\_thread = new time\_progressbar(mw\_progressbar, time,mw\_info,start\_button,easy\_start,mode\_num,time\_field,temperature,fire\_field);

temp = work\_thread;

if(easy\_start){

//好多事件 启动后 好多的功能

for(int i = 0;i<2;i++)

mode\_num[i] = 1;

System.out.println(mode\_num[0]+""+mode\_num[1]+""+mode\_num[2]);

if(mode\_num[0] == 1 && mode\_num[1] == 1 && mode\_num[2] == 1){

mw\_info.setText("");//清空文本域

mw\_info.append(new String("\n"+"总时长: "+time+" min"+"\n"));

mw\_info.append(new String("温度: "+temperature+" ℃"+"\n"));

mw\_info.append("食物: "+temp\_mode);

start\_button.setText("关机");

easy\_start = false;

start(work\_thread);

}

}

else{

//好多时间 停止所有已经启动的事件

start\_button.setText("开机");

easy\_start = true;

stop(work\_thread);

}

}

});

mw\_panel.add(start\_button);

pic\_panel = new JPanel();

pic\_label = new JLabel(new ImageIcon("G:\\Java\\Java项目\\MyMicrowave\\2.png"));//初始化图片

num\_panel = new JPanel();

num\_panel.setLayout(new GridLayout(3, 3, 10, 10));

num\_panel.setBackground(new Color(100, 180, 180));

//缺省温度和时间,显示在文本框中

time\_field.setText(new String().valueOf(time));

fire\_field.setText(new String().valueOf(temperature));

//数字键盘的构建

for(int i = 0;i<9;i++){

\_j = i;

num\_button[i] = new JButton(num\_string[i]);

num\_button[i].setFocusPainted(false);//好看的按钮(没有边框)

num\_button[i].setBackground(Color.GRAY);//按钮颜色

num\_panel.add(num\_button[i]);

num\_button[i].addActionListener(new ActionListener() {

int k = \_j;//防止不同步

public void actionPerformed(ActionEvent e) {

if(mode\_num[0]==0)//不满足三个条件 不能开始 为0说明没有在工作 可以赋予工作

if(easy\_field){

time = Integer.parseInt(num\_string[k])+time\*10;

System.out.println("时间文本框:"+time);//测试

time\_field.setText(new String().valueOf(time));

// mode\_num[0] = 1;

//mw\_progressbar.setString(new String().valueOf(time));

}

if(mode\_num[1]==0)//不满足三个条件 不能开始 为0说明没有在工作 可以赋予工作

if(!easy\_field){

temperature =Integer.parseInt(num\_string[k])+temperature\*10;

System.out.println("温度文本框:"+temperature);//测试

fire\_field.setText(new String().valueOf(temperature));

// mode\_num[1] = 1;

}

}

});

}

//放置数字按钮的容器

zero\_delete = new JPanel(new GridLayout(2,1,0,10));

zero\_delete.setBounds(820, 210, 50, 150);

zero\_delete.setBackground(new Color(100, 180, 180));

//数字0和删除统一处理

num\_button[9] = new JButton(num\_string[9]);

//添加烦人0的事件处理

num\_button[9].addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent arg0) {

if(mode\_num[0]==0)

if(easy\_field){

time = Integer.parseInt(num\_string[9])+time\*10;

System.out.println("时间文本框:"+time);//测试

time\_field.setText(new String().valueOf(time));

}

if(mode\_num[1]==0)

if(!easy\_field){

temperature =Integer.parseInt(num\_string[9])+temperature\*10;

System.out.println("温度文本框:"+temperature);//测试

fire\_field.setText(new String().valueOf(temperature));

}

}

});

//数字删除按钮

delete = new JButton("←");

delete.setBackground(Color.gray);

delete.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

if(mode\_num[0]==0)//工作过程中 不能修改时间

if(easy\_field){

time = time/10;

System.out.println("时间文本框:"+time);//测试

time\_field.setText(new String().valueOf(time));

}

if(mode\_num[1]==0)//工作过程中 不能修改温度

if(!easy\_field){

temperature =temperature/10;

System.out.println("温度文本框:"+temperature);//测试

fire\_field.setText(new String().valueOf(temperature));

}

}

});

num\_button[9].setFocusPainted(false);//好看的按钮(没有边框)

num\_button[9].setBackground(Color.GRAY);//按钮颜色

zero\_delete.add(num\_button[9]);

zero\_delete.add(delete);

mw\_panel.add(zero\_delete);

//功能图片按钮

button\_panel = new JPanel();

button\_panel.setLayout(new GridLayout(2, 4, 10, 10));

button\_panel.setBackground(new Color(100, 180, 180));

for( int i = 0;i<8;i++){

j = i;//传递 无法在监听器中使用i变量

mw\_button[i] = new JButton(mode\_string[i]);

mw\_button[i].setFocusPainted(false);//好看的按钮(没有边框)

mw\_button[i].setBackground(Color.GRAY);//按钮颜色

mw\_button[i].addActionListener(new ActionListener() {

int k = j;//直接使用j 会发生不同步现象

public void actionPerformed(ActionEvent arg0) {

if(mode\_num[2]==0){

pic\_label.setIcon(new ImageIcon(new String("G:\\Java\\Java项目\\MyMicrowave\\图片\\"+k+".jpg")));

mode\_num[2] = 1;

temp\_mode = mode\_string[k];

}

}

});

button\_panel.add(mw\_button[i]);

}

//图片区

pic\_panel.setBackground(new Color(100, 180, 180));//保险小缝隙的颜色残留,全部和主界面颜色统一

pic\_panel.add(pic\_label);

pic\_panel.setBounds(25, 20, 510, 530);

mw\_panel.add(pic\_panel);

//键盘区

num\_panel.setBounds(560, 210, 250, 150);

mw\_panel.add(num\_panel);

//功能区

button\_panel.setBounds(560, 380, 400, 150);

mw\_panel.add(button\_panel);

mw\_windows.add(mw\_panel);

}//构造方法到此结束

//正式开始工作了

public void start(time\_progressbar work\_thread){

//System.out.println("测试语句");

work\_thread.start();//多线程实现倒计时

}

//停止工作

public void stop(time\_progressbar work\_thread){

for(int i = 0;i<3;i++)

mode\_num[i] = 0;

time = 10;

time\_field.setText(new String().valueOf(time));

temperature = 100;

fire\_field.setText(new String().valueOf(temperature));

}

}

public class My {

public static void main(String [] args){

JFrame mw\_windows = new JFrame("MyMicrowave");

JPanel mw\_panel = new JPanel();

mw\_panel.setLayout(null);

mw\_panel.setBackground(new Color(100, 180, 180));//淡蓝色

ImageIcon mw\_icon = new ImageIcon("G:\\Java\\Java项目\\MyMicrowave\\1.png");

mw\_windows.setIconImage(mw\_icon.getImage());//窗口图标

mw\_windows.setSize(1000,600);//尺寸

mw\_windows.setLocationRelativeTo(null);//屏幕居中显示

mw\_windows.setResizable(false);//不让用户改变界面尺寸

mw\_windows.addWindowListener(new WindowAdapter() {

public void windowClosing(WindowEvent e){

System.exit(0);

}

}) ;

work sc = new work(mw\_windows, mw\_panel);

mw\_windows.show();

}

}