**题号： 提高题-2** 秒表计时器

1、设计任务与目标

**目标：**设计一个与手机秒表类似的计时器，以钟表图形的方式呈现。

1. 当用户用鼠标单击界面的“开始”按钮或图标时，秒表开始计时，表及数字都自动发生变化；当用户再次点击按钮，秒表暂停计时，按钮字样变为“继续”；当用户用鼠标点击“继续”按钮，秒表继续计时。
2. 界面上还有一个“复位”按钮，当未点击“开始”按钮，没有进入计时状态时，“复位”按钮为灰色不可选，当开始计时后，“复位”按钮变为可选状态，当用户点击复位按钮时，秒表计时清零。

**设计任务**：这题重点在于利用java Swing里面的组件构造出一个钟表的图形，并且其能够随着时间转动。主要考察了图形化窗口界面的设计（比如利用什么组件，如何构建钟表图形，采用什么布局，组件应该放在哪些位置比较合理），以及如何让静态的钟表转动起来（可以利用java中的Timer类或者线程技术）。

2、类及数据的设计

设计三个类：一个画布类（MyCanvas），一个是计时器类（MyTimer）一个是主类（Miaobiao）。

画布类主要用来构建钟表图形（包括钟表上的数字，指针，轮廓）。

画布类：继承了java中的Canvas类，重写了paint方法

paint方法作用：构造钟表图形

计时器类主要用来更新指针终点的坐标。

计时器类：继承了java中的TimerTask类，重写了run方法

run方法作用：更新指针终点的坐标

主类主要用来展示窗口，组合各个Swing组件。

主类：

主类中的变量： pai （用来计算钟表图形上的坐标）

jf （总的窗口）

jp\_one （第一个面板）

jp\_two （第二个面板）

jp\_three （第三个面板）

location\_x （指针终点的x坐标）

location\_y （指针终点的y坐标）

myCanvas （画布，用于显示钟表图形）

jb\_one （第一个按钮——开始按钮）

jb\_two （第二个按钮——暂停按钮）

jb\_three （第三个按钮——复位按钮）

jd （设置未开始点击暂停和复位的弹窗）

jta （设置弹窗的内容）

tf （单行文本域，显示当前时长）

ta （多行文本域，显示记录的次序和时长）

isStop （判断是否点击了暂停按钮）

t （记录时长）

haveStart （判断是否点击了开始按钮）

number （记录次序）

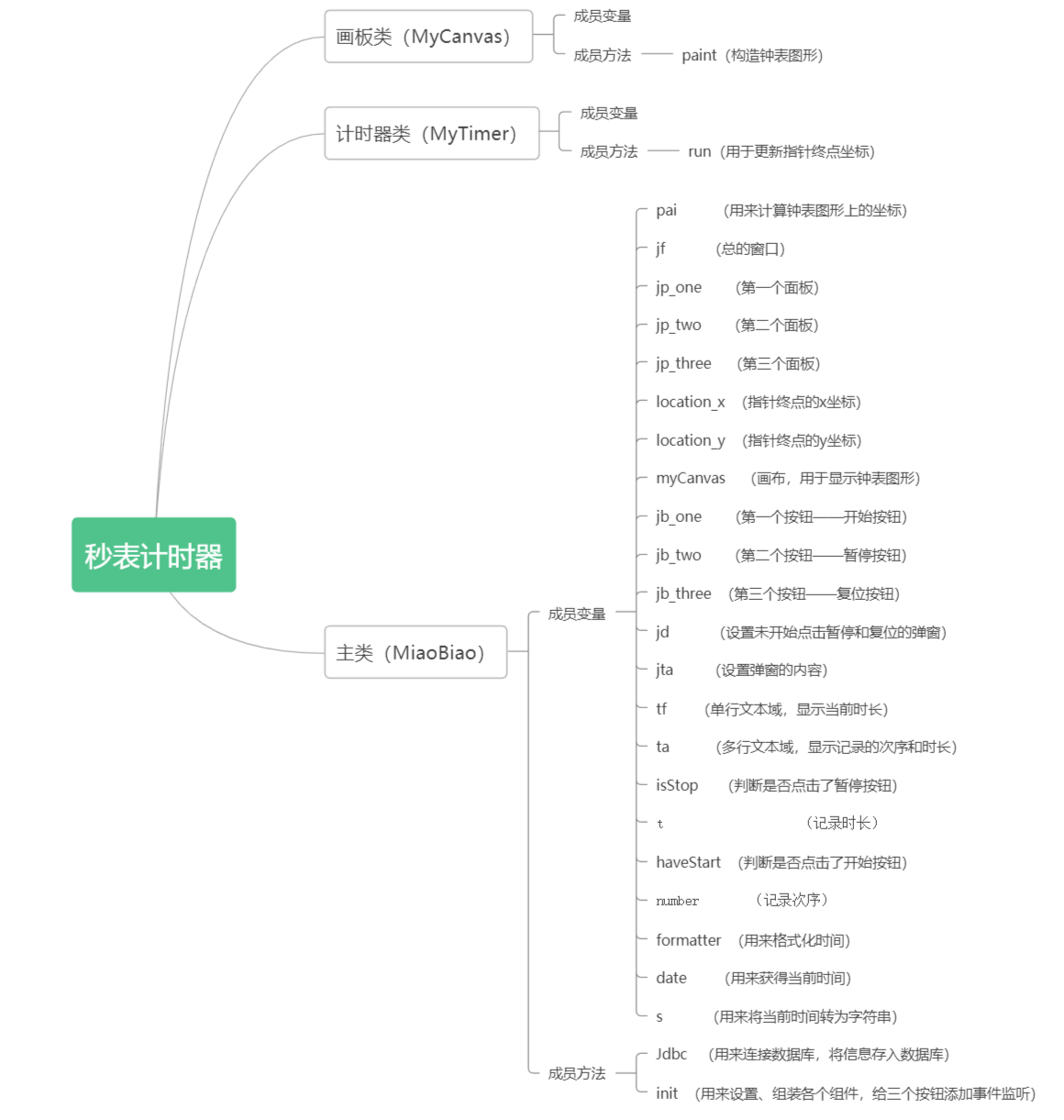
formatter （用来格式化时间）

date （用来获得当前时间）

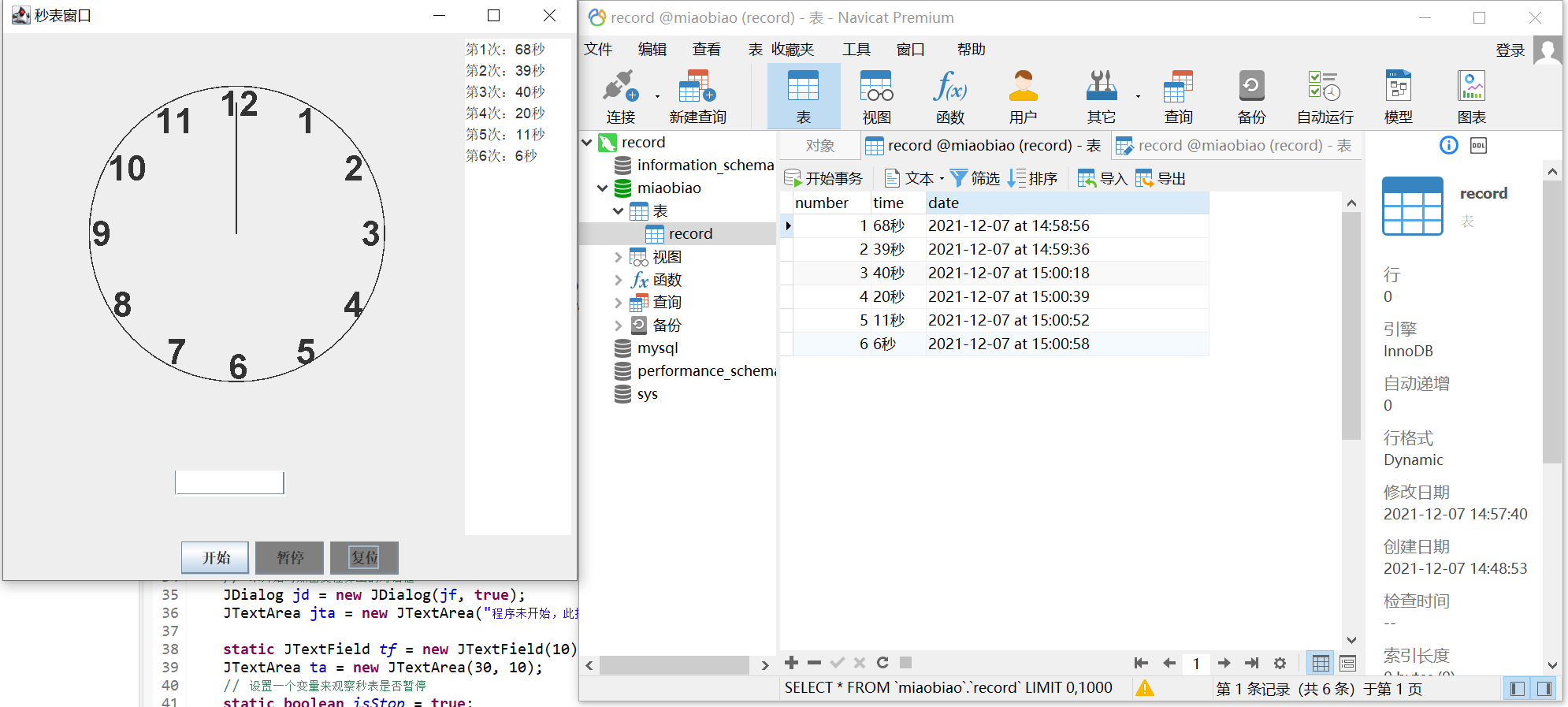
s （用来将当前时间转为字符串）

主类中的方法：Jdbc （用来连接数据库，将信息存入数据库）

init （用来设置、组装各个组件，给三个按钮添加事件监听）



数据：数据包括三个部分（记录的次序，记录时长，记录的时刻），在点击复位按钮时，记录的次序和记录时长会直接在窗口右侧的多行文本域显示，并且数据的三个部分会全部记录在数据库中（如下图）。



3、源程序

**package** Miaobiao;

**import** javax.swing.\*;

**import** java.awt.event.\*;

**import** java.sql.\*;

**import** java.text.SimpleDateFormat;

**import** java.awt.\*;

**import** java.util.\*;

**import** java.util.Date;

**import** java.util.Timer;

**public** **class** MiaoBiao {

//获得记录的时间

SimpleDateFormat formatter= **new** SimpleDateFormat("yyyy-MM-dd 'at' HH:mm:ss ");

Date date = **new** Date(System.*currentTimeMillis*());

String s;

// 创造各个组件

**final** **static** **double** ***pai*** = 3.14159;

JFrame jf = **new** JFrame("秒表窗口");

JPanel jp\_one = **new** JPanel();

JPanel jp\_two = **new** JPanel();

JPanel jp\_three = **new** JPanel();

**static** **int** *location\_x* = 0;

**static** **int** *location\_y* = 110;

**static** MyCanvas *myCanvas* = **new** MyCanvas();

JButton jb\_one = **new** JButton("开始");

JButton jb\_two = **new** JButton("暂停");

JButton jb\_three = **new** JButton("复位");

// 未开始时点击复位弹出的对话框

JDialog jd = **new** JDialog(jf, **true**);

JTextArea jta = **new** JTextArea("程序未开始，此按钮不可选！");

**static** JTextField *tf* = **new** JTextField(10);

JTextArea ta = **new** JTextArea(30, 10);

// 设置一个变量来观察秒表是否暂停

**static** **boolean** *isStop* = **true**;

// 设置一个变量记录时间

**static** **private** **int** *t* = 0;

// 设置一个变量是否开始

**static** **boolean** *haveStart* = **false**;

// 设置一个变量记录秒表运行次数

**static** **int** *number* = 0;

**private** **static** **class** MyTimer **extends** TimerTask {

@Override

**public** **void** run() {

**if** (*isStop*) {

} **else** {

//更新指针终点的x，y坐标，并且更新单行文本域的内容

*t*++;

*location\_x* = (**int**) (Math.*cos*((***pai*** / 2 - *t* \* ***pai*** / 30)) \* 110);

*location\_y* = (**int**) (Math.*sin*((***pai*** / 2 - *t* \* ***pai*** / 30)) \* 110);

*tf*.setText("" + *t* + "秒");

*myCanvas*.repaint();

}

// **TODO** Auto-generated method stub

}

}

/\* void Jdbc() throws SQLException

{

Driver driver =new com.mysql.jdbc.Driver();

String url="jdbc:mysql://localhost:3306/miaobiao";

Properties info =new Properties();

info.setProperty("user", "root");

info.setProperty("password", "root");

Connection conn=driver.connect(url, info);

String sql="insert into record(number,time,date) values(?,?,?)";

PreparedStatement ps=conn.prepareStatement(sql);

ps.setInt(1, number);

ps.setString(2, t+"秒");

ps.setString(3,s);

ps.execute();

ps.close();

conn.close();

}\*/

// 组装函数

**public** **void** init() {

// 设置弹窗内容

jd.setBounds(300, 300, 300, 200);

jd.setTitle("错误！");

jd.add(jta);

// jd.setTe

// 给复位按钮设置成灰色

jb\_two.setBackground(Color.***gray***);

jb\_three.setBackground(Color.***gray***);

// 给按钮添加事件监听

jb\_one.addActionListener(**new** ActionListener() {

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

**if** (jb\_two.getActionCommand().equals("暂停")) {

*isStop* = **false**;

*haveStart* = **true**;

jb\_two.setBackground(**new** Color(236, 242, 248));

jb\_three.setBackground(**new** Color(236, 242, 248));

}

}

});

jb\_two.addActionListener(**new** ActionListener() {

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

// 判断这个按钮当前的状态

**if** (!*haveStart* && e.getActionCommand().equals("暂停")) {

jd.setVisible(**true**);

} **else** **if** (*haveStart* && e.getActionCommand().equals("暂停")) {

*isStop* = **true**;

jb\_two.setText("继续");

} **else** {

*isStop* = **false**;

jb\_two.setText("暂停");

}

}

});

jb\_three.addActionListener(**new** ActionListener() {

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

**if** (*haveStart*) {

// 更新成员变量的属性值

*isStop* = **true**;

*haveStart* = **false**;

*location\_x* = 0;

*location\_y* = 110;

jb\_two.setText("暂停");

// 在右边的文本域中更新内容

ta.append("第" + (++*number* )+ "次：" + *t* + "秒" + "\n");

//写入数据库

/\* try {

date = new Date(System.currentTimeMillis());

s=formatter.format(date);

Jdbc();

} catch (SQLException e1) {

// **TODO** Auto-generated catch block

e1.printStackTrace();

}\*/

*t* = 0;

*tf*.setText("");

// 给两个按钮回复颜色

jb\_three.setBackground(Color.***gray***);

jb\_two.setBackground(Color.***gray***);

*myCanvas*.repaint();

} **else** {

jd.setVisible(**true**);

}

}

});

// 给三个面板添加组件

*myCanvas*.setSize(**new** Dimension(360, 360));

//myCanvas.setBackground(Color.blue);

jp\_one.add(*myCanvas*);

jp\_one.add(*tf*);

jp\_two.add(jb\_one);

jp\_two.add(jb\_two);

jp\_two.add(jb\_three);

jp\_three.add(ta);

// 给窗口添加三个面板

jf.add(jp\_one, BorderLayout.***CENTER***);

jf.add(jp\_two, BorderLayout.***SOUTH***);

jf.add(jp\_three, BorderLayout.***EAST***);

// 设置窗口位置，大小，可见性，添加窗口关闭监听

jf.setBounds(200, 200, 500, 500);

jf.setVisible(**true**);

//jf.addWindowListener(new MyListener());

jf.setDefaultCloseOperation(1);

Timer timer = **new** Timer();

timer.schedule(**new** MyTimer(), 0, 1000);

}

**public** **static** **void** main(String[] args) {

**new** MiaoBiao().init();

}

// 创建画布

**private** **static** **class** MyCanvas **extends** Canvas {

@Override

**public** **void** paint(Graphics g) {

// 绘制钟表图

g.drawOval(60, 40, 250, 250);

g.setFont((**new** Font("Times", Font.***BOLD***, 30)));

g.drawString("12", 170, 65);

g.drawString("3", 290, 175);

g.drawString("6", 178, 288);

g.drawString("9", 62, 175);

g.drawString("1", 235, 80);

g.drawString("2", 275, 120);

g.drawString("4", 275, 235);

g.drawString("5", 235, 275);

g.drawString("7", 126, 275);

g.drawString("8", 80, 235);

g.drawString("10", 75, 120);

g.drawString("11", 115, 80);

// 绘制指针,利用计时器更新 a b 的值使指针不停的转动

**int** a = 185 + *location\_x*;

**int** b = 165 - *location\_y*;

g.drawLine(185, 165, a, b);

}

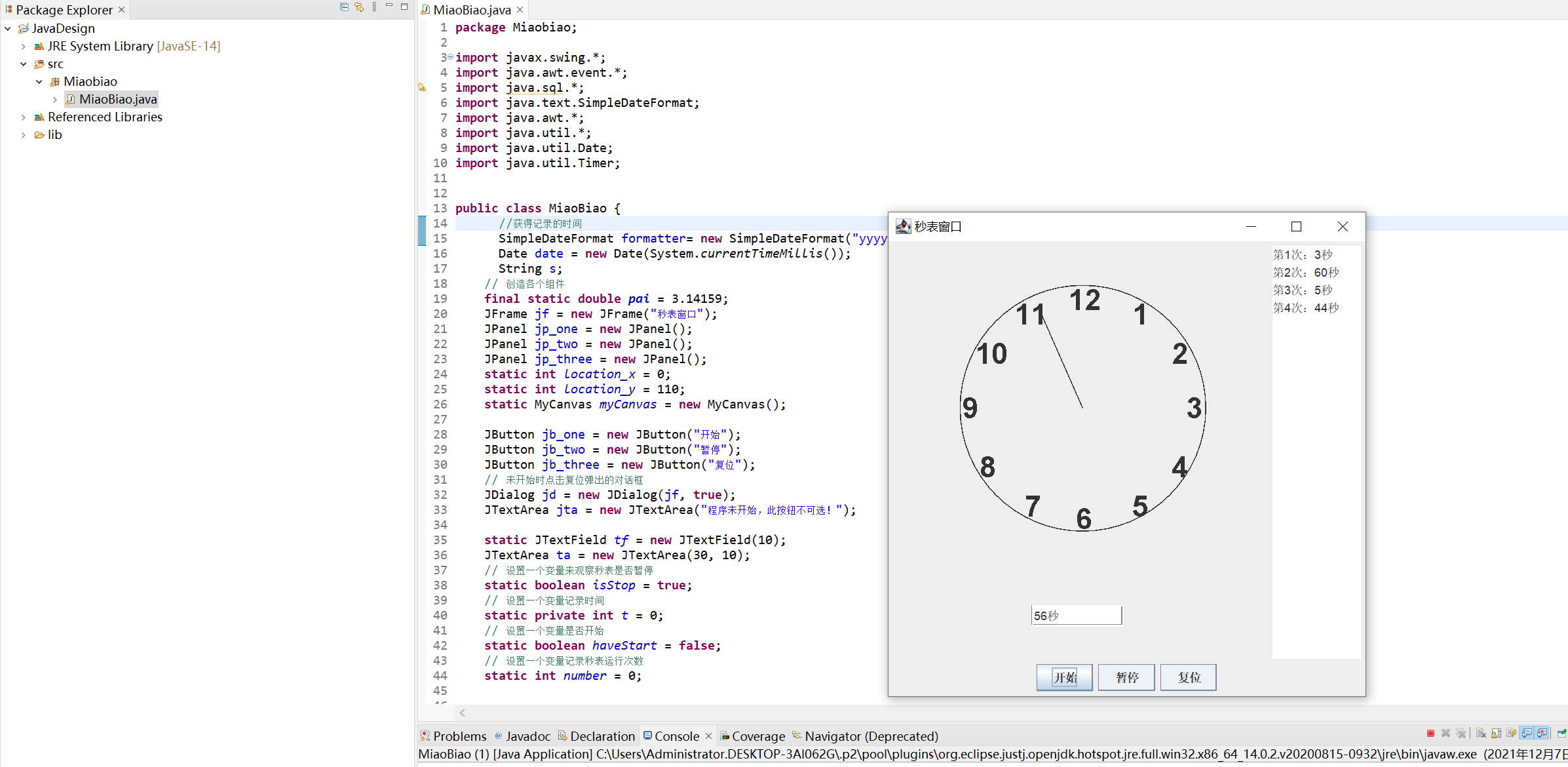
}

}

**注：为了让代码能够成功运行，有关数据库部分的代码被注释了。（注释掉的部分：第67-83行，第133-140行）**

4、程序运行、测试与分析

**程序运行**：程序能够成功运行



**程序测试：**

1

**程序分析：**程序能够符合题目的所有要求，并且结合实际考虑，在题目要求之外添加了如下几个功能：

1. 在未点击开始按钮时，暂停按钮也设置成灰色不可点击的状态
2. 在未点击开始前，点击暂停或者复位按钮的话，会有弹窗提醒，告诉使用者“程序未开始，此按钮不可选！”
3. 在窗口右侧添加了多行文本域，可以展示在一次使用中的多次记录的次序和时长
4. 使用了JDBC技术连接了java程序和mysql数据库，可以在点击复位按钮时自动把每次记录的次序，时长，记录的时间存入数据库中，便于数据的保存和查询。

程序的不足之处：

1. 因为采用了Java Swing中的组件来构造钟表，所以钟表图形简陋，窗口上的单行文本域和多行文本域也显得有些呆滞