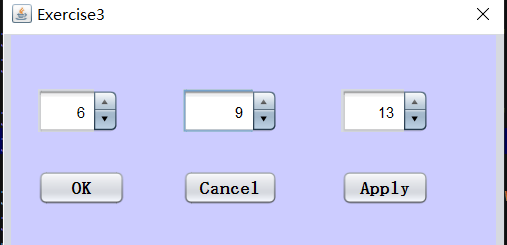
|  |
| --- |
| **用户界面设计与分析**  **实验报告**  项目名称 实验三  专业班级 软件1906  学 号 XJ20192501347  姓 名 解梦华  **实验成绩：**  **批阅教师：郁松**  年 月 日 |

### **1．需求分析**

* 实现一个时间显示窗口，用户可以对时间的修改，以及添加三个按钮，完成相应的事件。
* **实现界面如下：**



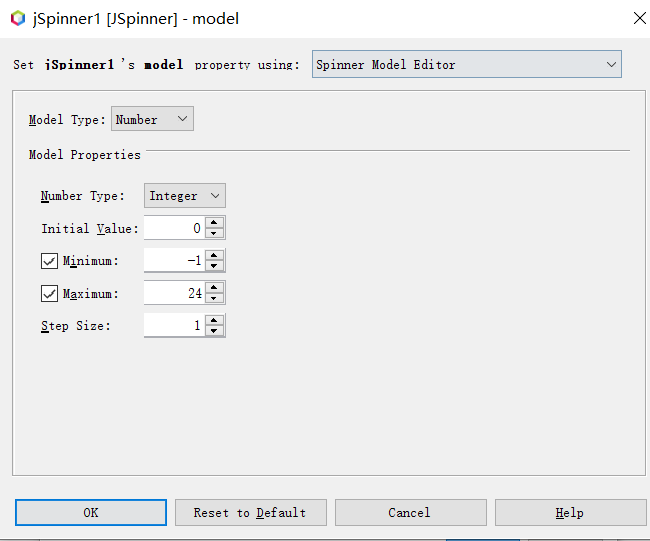
* **具体功能要求：**
* 首先应在时，分，秒显示的旁边放置按钮，一个用于增加值，另一个用作减少值。
* 另外对时，分，秒的数值范围进行限制，与我们日常生活的经验相同，时的数值应在0-23之间，分与秒的数值范围应在0-59之间。
* 另外布置的三个Button按钮分别实现的功能是“OK”按钮将时间设置成用户所设置的时间，并关闭窗口、“Cancel”按钮直接关闭窗口、“Apply”按钮则是应用当前时间，此次试验窗口可以视为实验四的组成部分，并实现实验四的逻辑功能。

### **概要设计**

* 首先用NetBeans实现界面的搭建，复刻原始界面。
* 程序运行后首先使用requestFocus（）函数将焦点聚焦到第一个jSpinner文本框。
* 设置jSpinner模型中的数据显示范围：
* 小时设置在0-23之间；
* 分钟设置在0-59之间；
* 秒钟设置在0-59之间。
* 给三个jSpinner组件添加stateChanged事件（状态改变事件），具体如下：
* 秒钟当前为59时，加一变为0，同时分钟加一（若为59的话与此实现逻辑相同），小时数若为23，实现逻辑相同。若秒钟数为0，减一变为59，分钟，小时变化逻辑与上相同；
* 分钟数若为59或0，小时数若为23或0实现逻辑与秒钟数实现逻辑相同。
* 引入Timer类，设置Timer计数器，不断获取其值，实现秒数的自加，使时间可以自动变化。
* 实现三个按钮事件。

### **详细设计**

* 首先新建一个jPanel面板，将三个jSpinner组件，以及三个jButton组件拖拽到面板中。
* 进入jSpinner的Model中设置相关数据范围，数值类型。（当然模型类型也可以用日期）。具体界面如下：



* 编写jSpinner的stataChanged事件，即当用户自定义增加与减少到临界值时时、分、秒的具体变化，这个地方编写代码时逻辑思路很容易出现错误，因为具体逻辑的实现可能与我们日常生活经验有些出入。

秒钟部分代码如下：

//秒钟状态改变事件  
secondSpinner.addChangeListener*(*new ChangeListener*() {* @Override  
 public void stateChanged*(*ChangeEvent e*) {* //获取到当前时分秒  
 int hour = Integer.*parseInt(*hourSpinner.getValue*()*.toString*())***;** int minute = Integer.*parseInt(*minuteSpinner.getValue*()*.toString*())***;** int second = Integer.*parseInt(*secondSpinner.getValue*()*.toString*())***;** //秒时分的变化退位  
 if *(*second == -**1***) {* if *(*minute == **0***) {* if *(*hour == **0***) {* hourSpinner.setValue*(***23***)***;** *}* else *{* hourSpinner.setValue*(*hour - **1***)***;** *}* minuteSpinner.setValue*(***59***)***;** *}* else *{* minuteSpinner.setValue*(*minute - **1***)***;** *}* secondSpinner.setValue*(***59***)***;** *}* //秒时分的变化进位  
 if *(*second == **60***) {* if *(*minute == **59***) {* if *(*hour == **23***) {* hourSpinner.setValue*(***0***)***;** *}* else *{* hourSpinner.setValue*(*hour + **1***)***;** *}* minuteSpinner.setValue*(***0***)***;** *}* else *{* minuteSpinner.setValue*(*minute + **1***)***;** *}* secondSpinner.setValue*(***0***)***;** *}  
 }  
})***;**

* 设置Timer定时器设置定时的重复操作，对秒钟进行Timer定时设计，间隔为1s，重复获取当前秒数，并每次赋值给jSpinner3,关键代码如下：

public MainFrameController*(){* initComponents*()***;** initListeners*()***;** Timer timer=new Timer*()***;** //定时器执行任务  
 timer.schedule*(*new TimerTask*() {* @Override  
 public void run*() {* int second=Integer.*parseInt(*secondSpinner.getValue*()*.toString*())***;** second++**;** secondSpinner.setValue*(*second*)***;** repaint*()***;** *}  
 }***,0,1000***)***;***}*

* 对“OK”，“Candel”按钮添加mouseClicked()函数，即关闭当前窗口。即调用System.exit()函数即可。

### **调试分析**

* 调试过程中，时、分、秒的数值变化开始出现了很多错误。比如当秒钟数值为59，分钟为59，小时数为23时，秒钟数值增加时分钟会正常变化，但小时数却没变化，具体的原因便是编写代码时没考虑到某些特殊的时间点。

### **用户使用说明**

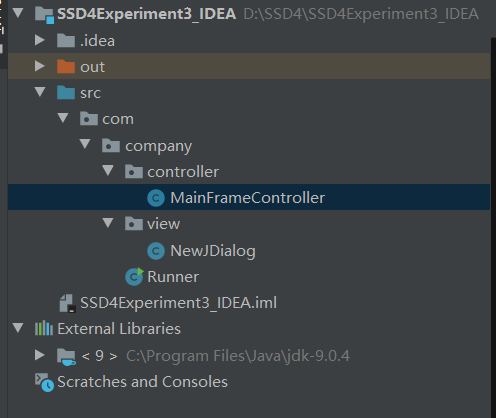


### **测试结果**

* 程序能够按照实现逻辑正常运行。

### **附录**

* **架构如下：**



* **实验源码：**

package com.company.controller**;**import com.company.view.NewJDialog**;**import javax.swing.\***;**import javax.swing.event.ChangeEvent**;**import javax.swing.event.ChangeListener**;**import java.util.Timer**;**import java.util.TimerTask**;**public class MainFrameController extends javax.swing.JFrame *{* public MainFrameController*(){* initComponents*()***;** initListeners*()***;** Timer timer=new Timer*()***;** //定时器执行任务  
 timer.schedule*(*new TimerTask*() {* @Override  
 public void run*() {* int second=Integer.*parseInt(*secondSpinner.getValue*()*.toString*())***;** second++**;** secondSpinner.setValue*(*second*)***;** repaint*()***;** *}  
 }***,0,1000***)***;** *}* public void showMainWindows*(){* windowFrame.setVisible*(*true*)***;** *}* public void initComponents*(){* windowFrame= new NewJDialog*()***;** hourSpinner=windowFrame.getjSpinner1*()***;** minuteSpinner=windowFrame.getjSpinner2*()***;** secondSpinner=windowFrame.getjSpinner3*()***;** okButton=windowFrame.getjButton1*()***;** cancelButton=windowFrame.getjButton2*()***;** applyButton=windowFrame.getjButton3*()***;** *}* public void initListeners*(){* //分钟状态改变事件  
 hourSpinner.addChangeListener*(*new ChangeListener*() {* @Override  
 public void stateChanged*(*ChangeEvent e*) {* //获取到当前时  
 int hour = Integer.*parseInt(*hourSpinner.getValue*()*.toString*())***;** //时的变化退位  
 if *(*hour== -**1***) {* hourSpinner.setValue*(***23***)***;** *}* //时的变化进位  
 if *(*hour == **24***) {* hourSpinner.setValue*(***0***)***;** *}  
 }  
 })***;** //分钟状态改变事件  
 minuteSpinner.addChangeListener*(*new ChangeListener*() {* @Override  
 public void stateChanged*(*ChangeEvent e*) {* //获取到当前时分  
 int hour = Integer.*parseInt(*hourSpinner.getValue*()*.toString*())***;** int minute = Integer.*parseInt(*minuteSpinner.getValue*()*.toString*())***;** //时分的变化退位  
 if *(*minute== -**1***) {* if *(*hour == **0***) {* hourSpinner.setValue*(***23***)***;** *}* else *{* hourSpinner.setValue*(*hour - **1***)***;** *}* minuteSpinner.setValue*(***59***)***;** *}* //时分的变化进位  
 if *(*minute == **60***) {* if *(*hour == **23***) {* hourSpinner.setValue*(***0***)***;** *}* else *{* hourSpinner.setValue*(*hour + **1***)***;** *}* minuteSpinner.setValue*(***0***)***;** *}  
 }  
 })***;** //秒钟状态改变事件  
 secondSpinner.addChangeListener*(*new ChangeListener*() {* @Override  
 public void stateChanged*(*ChangeEvent e*) {* //获取到当前时分秒  
 int hour = Integer.*parseInt(*hourSpinner.getValue*()*.toString*())***;** int minute = Integer.*parseInt(*minuteSpinner.getValue*()*.toString*())***;** int second = Integer.*parseInt(*secondSpinner.getValue*()*.toString*())***;** //秒时分的变化退位  
 if *(*second == -**1***) {* if *(*minute == **0***) {* if *(*hour == **0***) {* hourSpinner.setValue*(***23***)***;** *}* else *{* hourSpinner.setValue*(*hour - **1***)***;** *}* minuteSpinner.setValue*(***59***)***;** *}* else *{* minuteSpinner.setValue*(*minute - **1***)***;** *}* secondSpinner.setValue*(***59***)***;** *}* //秒时分的变化进位  
 if *(*second == **60***) {* if *(*minute == **59***) {* if *(*hour == **23***) {* hourSpinner.setValue*(***0***)***;** *}* else *{* hourSpinner.setValue*(*hour + **1***)***;** *}* minuteSpinner.setValue*(***0***)***;** *}* else *{* minuteSpinner.setValue*(*minute + **1***)***;** *}* secondSpinner.setValue*(***0***)***;** *}  
 }  
 })***;** *}* NewJDialog windowFrame**;** private javax.swing.JButton okButton**;** private javax.swing.JButton cancelButton**;** private javax.swing.JButton applyButton**;** private javax.swing.JSpinner hourSpinner**;** private javax.swing.JSpinner minuteSpinner**;** private javax.swing.JSpinner secondSpinner**;***}*

/\*  
 \* To change this license header, choose License Headers in Project Properties.  
 \* To change this template file, choose Tools | Templates  
 \* and open the template in the editor.  
 \*/  
package com.company.view**;**import javax.swing.\***;***/\*\*  
 \*  
 \** ***@author*** *25070  
 \*/*public class NewJDialog extends javax.swing.JDialog *{  
  
 /\*\*  
 \* Creates new form NewJDialog  
 \*/* public NewJDialog*() {* this.setTitle*(*"Exercise3"*)***;** initComponents*()***;** this.setLocationRelativeTo*(*null*)***;** *}  
  
  
  
 /\*\*  
 \* This method is called from within the constructor to initialize the form.  
 \* WARNING: Do NOT modify this code. The content of this method is always  
 \* regenerated by the Form Editor.  
 \*/* @SuppressWarnings*(*"unchecked"*)* // <editor-fold defaultstate="collapsed" desc="Generated Code">  
 private void initComponents*() {* jPanel1 = new javax.swing.JPanel*()***;** jSpinner1 = new javax.swing.JSpinner*()***;** jSpinner2 = new javax.swing.JSpinner*()***;** jSpinner3 = new javax.swing.JSpinner*()***;** jButton1 = new javax.swing.JButton*()***;** jButton2 = new javax.swing.JButton*()***;** jButton3 = new javax.swing.JButton*()***;** setDefaultCloseOperation*(*javax.swing.WindowConstants.*DISPOSE\_ON\_CLOSE)***;** jPanel1.setBackground*(*new java.awt.Color*(***204, 204, 255***))***;** jSpinner1.setModel*(*new javax.swing.SpinnerNumberModel*(***0,** -**1, 24, 1***))***;** jSpinner2.setModel*(*new javax.swing.SpinnerNumberModel*(***0,** -**1, 60, 1***))***;** jSpinner3.setModel*(*new javax.swing.SpinnerNumberModel*(***0,** -**1, 60, 1***))***;** jButton1.setFont*(*new java.awt.Font*(*"宋体"**, 1, 14***))***;** // NOI18N  
 jButton1.setText*(*"OK"*)***;** jButton2.setFont*(*new java.awt.Font*(*"宋体"**, 1, 14***))***;** // NOI18N  
 jButton2.setText*(*"Cancel"*)***;** jButton3.setFont*(*new java.awt.Font*(*"宋体"**, 1, 14***))***;** // NOI18N  
 jButton3.setText*(*"Apply"*)***;** javax.swing.GroupLayout jPanel1Layout = new javax.swing.GroupLayout*(*jPanel1*)***;** jPanel1.setLayout*(*jPanel1Layout*)***;** jPanel1Layout.setHorizontalGroup*(* jPanel1Layout.createParallelGroup*(*javax.swing.GroupLayout.Alignment.*LEADING)* .addGroup*(*jPanel1Layout.createSequentialGroup*()* .addGap*(***21, 21, 21***)* .addGroup*(*jPanel1Layout.createParallelGroup*(*javax.swing.GroupLayout.Alignment.*LEADING)* .addComponent*(*jSpinner1**,** javax.swing.GroupLayout.*PREFERRED\_SIZE***, 65,** javax.swing.GroupLayout.*PREFERRED\_SIZE)* .addComponent*(*jButton1**,** javax.swing.GroupLayout.*PREFERRED\_SIZE***, 70,** javax.swing.GroupLayout.*PREFERRED\_SIZE))* .addGap*(***46, 46, 46***)* .addGroup*(*jPanel1Layout.createParallelGroup*(*javax.swing.GroupLayout.Alignment.*LEADING***,** false*)* .addComponent*(*jButton2**,** javax.swing.GroupLayout.*DEFAULT\_SIZE***, 70,** Short.*MAX\_VALUE)* .addComponent*(*jSpinner2*))* .addPreferredGap*(*javax.swing.LayoutStyle.ComponentPlacement.*RELATED***, 49,** Short.*MAX\_VALUE)* .addGroup*(*jPanel1Layout.createParallelGroup*(*javax.swing.GroupLayout.Alignment.*LEADING***,** false*)* .addComponent*(*jButton3**,** javax.swing.GroupLayout.*DEFAULT\_SIZE***, 70,** Short.*MAX\_VALUE)* .addComponent*(*jSpinner3*))* .addGap*(***54, 54, 54***))  
 )***;** jPanel1Layout.setVerticalGroup*(* jPanel1Layout.createParallelGroup*(*javax.swing.GroupLayout.Alignment.*LEADING)* .addGroup*(*jPanel1Layout.createSequentialGroup*()* .addGap*(***43, 43, 43***)* .addGroup*(*jPanel1Layout.createParallelGroup*(*javax.swing.GroupLayout.Alignment.*BASELINE)* .addComponent*(*jSpinner1**,** javax.swing.GroupLayout.*PREFERRED\_SIZE***, 35,** javax.swing.GroupLayout.*PREFERRED\_SIZE)* .addComponent*(*jSpinner2**,** javax.swing.GroupLayout.*PREFERRED\_SIZE***, 35,** javax.swing.GroupLayout.*PREFERRED\_SIZE)* .addComponent*(*jSpinner3**,** javax.swing.GroupLayout.*PREFERRED\_SIZE***, 35,** javax.swing.GroupLayout.*PREFERRED\_SIZE))* .addGap*(***30, 30, 30***)* .addGroup*(*jPanel1Layout.createParallelGroup*(*javax.swing.GroupLayout.Alignment.*BASELINE)* .addComponent*(*jButton1*)* .addComponent*(*jButton2*)* .addComponent*(*jButton3*))* .addContainerGap*(***93,** Short.*MAX\_VALUE))  
 )***;** javax.swing.GroupLayout layout = new javax.swing.GroupLayout*(*getContentPane*())***;** getContentPane*()*.setLayout*(*layout*)***;** layout.setHorizontalGroup*(* layout.createParallelGroup*(*javax.swing.GroupLayout.Alignment.*LEADING)* .addGroup*(*javax.swing.GroupLayout.Alignment.*TRAILING***,** layout.createSequentialGroup*()* .addContainerGap*()* .addComponent*(*jPanel1**,** javax.swing.GroupLayout.*DEFAULT\_SIZE***,** javax.swing.GroupLayout.*DEFAULT\_SIZE***,** Short.*MAX\_VALUE)* .addContainerGap*())  
 )***;** layout.setVerticalGroup*(* layout.createParallelGroup*(*javax.swing.GroupLayout.Alignment.*LEADING)* .addGroup*(*layout.createSequentialGroup*()* .addGap*(***1, 1, 1***)* .addComponent*(*jPanel1**,** javax.swing.GroupLayout.*PREFERRED\_SIZE***,** javax.swing.GroupLayout.*DEFAULT\_SIZE***,** javax.swing.GroupLayout.*PREFERRED\_SIZE)* .addContainerGap*(*javax.swing.GroupLayout.*DEFAULT\_SIZE***,** Short.*MAX\_VALUE))  
 )***;** pack*()***;** *}*// </editor-fold>  
  
 */\*\*  
 \** ***@param*** *args the command line arguments  
 \*/* // Variables declaration - do not modify  
 private javax.swing.JButton jButton1**;** private javax.swing.JButton jButton2**;** private javax.swing.JButton jButton3**;** private javax.swing.JPanel jPanel1**;** private javax.swing.JSpinner jSpinner1**;** private javax.swing.JSpinner jSpinner2**;** private javax.swing.JSpinner jSpinner3**;** // End of variables declaration  
 public JButton getjButton1*() {* return jButton1**;** *}* public JButton getjButton2*() {* return jButton2**;** *}* public JButton getjButton3*() {* return jButton3**;** *}* public JPanel getjPanel1*() {* return jPanel1**;** *}* public JSpinner getjSpinner1*() {* return jSpinner1**;** *}* public JSpinner getjSpinner2*() {* return jSpinner2**;** *}* public JSpinner getjSpinner3*() {* return jSpinner3**;** *}  
  
  
}*

package com.company**;**import com.company.controller.MainFrameController**;**import com.company.view.NewJDialog**;**public class Runner *{* public static void main*(*String args*[]) {* /\* Set the Nimbus look and feel \*/  
 //<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">  
 /\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.  
 \* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html  
 \*/  
 try *{* for *(*javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.*getInstalledLookAndFeels()) {* if *(*"Nimbus".equals*(*info.getName*())) {* javax.swing.UIManager.*setLookAndFeel(*info.getClassName*())***;** break**;** *}  
 }  
 }* catch *(*ClassNotFoundException ex*) {* java.util.logging.Logger.*getLogger(*NewJDialog.class.getName*())*.log*(*java.util.logging.Level.*SEVERE***,** null**,** ex*)***;** *}* catch *(*InstantiationException ex*) {* java.util.logging.Logger.*getLogger(*NewJDialog.class.getName*())*.log*(*java.util.logging.Level.*SEVERE***,** null**,** ex*)***;** *}* catch *(*IllegalAccessException ex*) {* java.util.logging.Logger.*getLogger(*NewJDialog.class.getName*())*.log*(*java.util.logging.Level.*SEVERE***,** null**,** ex*)***;** *}* catch *(*javax.swing.UnsupportedLookAndFeelException ex*) {* java.util.logging.Logger.*getLogger(*NewJDialog.class.getName*())*.log*(*java.util.logging.Level.*SEVERE***,** null**,** ex*)***;** *}* //</editor-fold>  
  
 /\* Create and display the dialog \*/  
 java.awt.EventQueue.*invokeLater(*new Runnable*() {* public void run*() {* MainFrameController mainFrameController = new MainFrameController*()***;** mainFrameController.showMainWindows*()***;** *}  
 })***;** *}  
  
}*