

```
1 package ui;
2 import java.awt.BorderLayout;
3 import java.awt.FlowLayout;
4 import java.awt.GridLayout;
5 import java.awt.event.ActionEvent;
6 import java.util.Set;
7 import javax.swing.AbstractAction;
8 import javax.swing.ComboBoxModel;
9 import javax.swing.DefaultComboBoxModel;
10 import javax.swing.JButton;
11 import javax.swing.JComboBox;
12 import javax.swing.JLabel;
13 import javax.swing.JList;
14 import javax.swing.JPanel;
15 import javax.swing.JScrollBar;
16 import javax.swing.ListModel;
17 import javax.swing.WindowConstants;
18 import javax.swing.SwingUtilities;
19 public class OperationSelectFrame extends javax.swing.JFrame {
20     private JPanel OperationSelectPanel;
21     private AbstractAction abstractAction_operationSelect;
22     private JButton Button_OperationSelectNext;
23     private JComboBox OperationSelectComboBox;
24     private JLabel OperationSelectLabel;
25     public static void main(String[] args) {
26         SwingUtilities.invokeLater(new Runnable() {
27             public void run() {
28                 OperationSelectFrame inst = new OperationSelectFrame();
29                 inst.setLocationRelativeTo(null);
30                 inst.setVisible(true);
31             }
32         });
33     }
34     public OperationSelectFrame() {
35         super();
36         initGUI();
37     }
38     public OperationSelectFrame(Set<String> opList) {
39         super();
40         String[] optionList= new String[opList.size()];
41         opList.toArray(optionList);
42         initGUI(optionList);
43     }
44     private void initGUI() {
45         initGUI(null);
46     }
47     private void initGUI(String[] optionList) {
48         try {
49             this.setTitle("操作选择");
50             this.setResizable(false);
```

```

51      GridLayout thisLayout = new GridLayout(1, 1);
52      getContentPane().setLayout(thisLayout);
53      setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
54      {
55          OperationSelectPanel = new JPanel();
56          getContentPane().add(OperationSelectPanel);
57          OperationSelectPanel.setBackground(new java.awt.Color(252,252,203));
58          OperationSelectPanel.setPreferredSize(new java.awt.Dimension(403, 275));
59          OperationSelectPanel.setLayout(null);
60          {
61              OperationSelectLabel = new JLabel();
62              OperationSelectPanel.add(OperationSelectLabel);
63              OperationSelectLabel.setText("\u8bf7\u9009\u62e9\u5f85\u6d4b\u7684\u64cd\u4f5c");
64              OperationSelectLabel.setBounds(18, 28, 122, 17);
65          }
66          {
67              ComboBoxModel OptionSelectComboBoxModel;
68              if (optionList == null) {
69                  OptionSelectComboBoxModel =
70                      new DefaultComboBoxModel(new String[] { "Item One",
71 "Item Two" });
72              } else {
73                  OptionSelectComboBoxModel =
74                      new DefaultComboBoxModel(optionList);
75              }
76              OperationSelectComboBox = new JComboBox();
77              OperationSelectPanel.add(OperationSelectComboBox);
78              OperationSelectComboBox.setModel(OptionSelectComboBoxModel);
79              OperationSelectComboBox.setBounds(20, 65, 199, 24);
80          }
81          {
82              Button_OperationSelectNext = new JButton();
83              OperationSelectPanel.add(Button_OperationSelectNext);
84              Button_OperationSelectNext.setText("\u4e0b\u4e00\u6b65");
85              Button_OperationSelectNext.setBounds(272, 198, 80, 24);
86              Button_OperationSelectNext.setAction(getAbstractAction_operationSelect());
87          }
88      }
89      pack();
90      setSize(412, 311);
91  } catch (Exception e) {
92      e.printStackTrace();
93  }
94  }
95  private AbstractAction getAbstractAction_operationSelect() {
96      if (abstractAction_operationSelect == null) {
97          abstractAction_operationSelect = new AbstractAction("\u4e0b\u4e00\u6b65", null)
98      {
99          public void actionPerformed(ActionEvent evt) {}
100      };

```

```
101         }
102         return abstractAction_operationSelect;
103     }
104 }
105 package ui;
106 import java.awt.BorderLayout;
107 import java.awt.Canvas;
108 import java.awt.Color;
109 import java.awt.Component;
110 import java.awt.FlowLayout;
111 import java.awt.Font;
112 import java.awt.GridBagConstraints;
113 import java.awt.GridBagLayout;
114 import java.awt.GridLayout;
115 import java.awt.Insets;
116 import java.awt.event.ActionEvent;
117 import java.awt.event.ActionListener;
118 import java.awt.event.KeyAdapter;
119 import java.awt.event.KeyEvent;
120 import java.util.Map;
121 import java.util.Queue;
122 import java.util.concurrent.TimeUnit;
123 import javax.swing.AbstractAction;
124 import javax.swing.BorderFactory;
125 import javax.swing.ButtonGroup;
126 import javax.swing.JButton;
127 import javax.swing.JDesktopPane;
128 import javax.swing.JInternalFrame;
129 import javax.swing.JLabel;
130 import javax.swing.JPanel;
131 import javax.swing.JRadioButton;
132 import javax.swing.JScrollPane;
133 import javax.swing.JSplitPane;
134 import javax.swing.JTextArea;
135 import javax.swing.JTextField;
136 import javax.swing.JTextPane;
137 import javax.swing.Timer;
138 import javax.swing.WindowConstants;
139 import javax.swing.SwingUtilities;
140 import javax.swing.border.EtchedBorder;
141 import javax.swing.border.TitledBorder;
142 import javax.swing.text.BadLocationException;
143 import javax.swing.text.Document;
144 import javax.swing.text.SimpleAttributeSet;
145 import javax.swing.text.StyleConstants;
146 import javax.swing.text.StyledDocument;
147 import cn.edu.ustb.webservicetester.model.OperationInfo;
148 import cn.edu.ustb.webservicetester.model.TestCase;
149 import cn.edu.ustb.webservicetester.model.TestSet;
150 import cn.edu.ustb.webservicetester.model.TestSetWithPartition;
```

```

151 import cn.edu.ustb.webservicetester.thread.TestCaseExecuter;
152 import cn.edu.ustb.webservicetester.thread.TestProcessWriter;
153 import cn.edu.ustb.webservicetester.util.ServiceInvoker;
154 public class ExecutionFrame extends javax.swing.JFrame {
155     private Color bgColor = new Color(252, 252, 203);
156     private JSplitPane jSplitPane1;
157     private JSplitPane jSplitPane2;
158     private JScrollPane jScrollPane1;
159     private JScrollPane jScrollPane2;
160     private JTextField jTextField1;
161     private ButtonGroup buttonGroup1;
162     private JButton jButton4;
163     private JButton jButton3;
164     private JButton jButton2;
165     private AbstractAction abstractAction1;
166     private JButton jButton1;
167     private JTextPane jTextPane1;
168     private StyledDocument doc;
169     private JPanel jPanel1;
170     private JPanel jPanel3;
171     private JPanel jPanel2;
172     private Timer timeAction;
173     private JLabel jLabel11;
174     private JTextField jTextField3;
175     private JLabel jLabel10;
176     private JLabel jLabel9;
177     private JTextField jTextField2;
178     private JRadioButton jRadioButton2;
179     private JRadioButton jRadioButton1;
180     private int testI = 0;
181     private boolean isStop = false;
182     private java.util.Map<String, String> testSetting;
183     private OperationInfo operationInfo;
184     private TestSet testSet = null;
185     private TestSetWithPartition testSetWP = null;
186     private java.util.Queue<String> pQueue = new
187 java.util.concurrent.ConcurrentLinkedQueue<String>();
188     private boolean testFinished = false;
189     private Thread executeThread = null;
190     private Thread showProcessThread = null;
191     private double epsilon = 0.0;
192     public static void main(String[] args) {
193         SwingUtilities.invokeLater(new Runnable() {
194             public void run() {
195                 ExecutionFrame inst = new ExecutionFrame();
196                 inst.setLocationRelativeTo(null);
197                 inst.setVisible(true);
198             }
199         });
200     }

```

```
201     public ExecutionFrame() {
202         super();
203         initGUI();
204     }
205     public ExecutionFrame(java.util.Map<String, String> tSetting, OperationInfo oi, TestSet ts) {
206         super();
207         testSetting = tSetting;
208         operationInfo = oi;
209         testSet = ts;
210         initGUI(tSetting, oi);
211     }
212     public ExecutionFrame(java.util.Map<String, String> tSetting, OperationInfo oi,
213 TestSetWithPartition tswp) {
214         super();
215         testSetting = tSetting;
216         operationInfo = oi;
217         testSetWP = tswp;
218         System.out.println(testSetWP.numOfPartitions());
219         System.out.println(testSetWP.numOfTestCases());
220         initGUI(tSetting, oi);
221     }
222     private void initGUI(Map<String, String> tSetting, OperationInfo oi) {
223         try {
224             setTitle("测试执行");
225             getContentPane().add(getJSplitPane1());
226             setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
227             setResizable(false);
228             pack();
229             setSize(800, 600);
230         } catch (Exception e) {
231             e.printStackTrace();
232         }
233     }
234     private Component getJSplitPane1() {
235         jSplitPane1 = new JSplitPane(JSplitPane.VERTICAL_SPLIT);
236         jSplitPane1.setDividerLocation(440);
237         jSplitPane1.setDividerSize(5);
238         jSplitPane1.setEnabled(false);
239         jSplitPane1.add(getJSplitPane2(), JSplitPane.LEFT);
240         jSplitPane1.add(getJPanel3(), JSplitPane.RIGHT);
241         return jSplitPane1;
242     }
243     private Component getJSplitPane2() {
244         jSplitPane2 = new JSplitPane();
245         jSplitPane2.setDividerLocation(200);
246         jSplitPane2.setDividerSize(5);
247         jSplitPane2.setEnabled(false);
248         jSplitPane2.add(getJScrollPane1(), JSplitPane.LEFT);
249         jSplitPane2.add(getJScrollPane2(), JSplitPane.RIGHT);
250         return jSplitPane2;
```

```
251     }
252     private Component getJScrollPane2() {
253         if (jScrollPane2 == null) {
254             jScrollPane2 = new JScrollPane();
255             jScrollPane2.setBackground(new java.awt.Color(228,236,243));
256             jScrollPane2.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder
257 (EtchedBorder.LOWERED), "Testing Process", TitledBorder.TRAILING, TitledBorder.TOP));
258             jScrollPane2.setViewportView(getJPane2());
259         }
260         return jScrollPane2;
261     }
262     private Component getJPane2() {
263         jPanel2 = new JPanel();
264         GridLayout jPanel2Layout = new GridLayout(1, 1);
265         jPanel2Layout.setColumns(1);
266         jPanel2Layout.setHgap(5);
267         jPanel2Layout.setVgap(5);
268         jPanel2.setBackground(bgColor);
269         jPanel2.setLayout(jPanel2Layout);
270         jPanel2.add(getJTextPane1());
271         return jPanel2;
272     }
273     private Component getJTextPane1() {
274         jTextPane1 = new JTextPane();
275         jTextPane1.setBackground(new java.awt.Color(228,236,243));
276         doc = jTextPane1.getStyledDocument();
277         return jTextPane1;
278     }
279     private Component getJScrollPane1() {
280         jScrollPane1 = new JScrollPane();
281         jScrollPane1.setBackground(new java.awt.Color(228,236,243));
282         jScrollPane1.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder
283 (EtchedBorder.LOWERED), "Test Setting", TitledBorder.LEADING, TitledBorder.TOP));
284         jScrollPane1.setViewportView(getJPanel1());
285         return jScrollPane1;
286     }
287     private Component getJPanel1() {
288         jPanel1 = new JPanel();
289         GridBagLayout jPanel1Layout = new GridBagLayout();
290         jPanel1.setBackground(new java.awt.Color(228,236,243));
291         jPanel1Layout.rowWeights = new double[testSetting.size() * 2 + 1];
292         jPanel1Layout.rowHeights = new int[testSetting.size() * 2 + 1];
293         for (int i = 0; i <= testSetting.size() * 2; ++i) {
294             jPanel1Layout.rowWeights[i] = 0.1;
295             if (i == testSetting.size() * 2) {
296                 jPanel1Layout.rowHeights[i] = 7;
297             } else {
298                 if (i % 2 == 0) {
299                     jPanel1Layout.rowHeights[i] = 25;
300                 } else {
```

```

301         jPanel1Layout.rowHeights[i] = 30;
302     }
303 }
304 }
305 jPanel1Layout.columnWeights = new double[] {0.1, 0.1, 0.1, 0.1, 0.1};
306 jPanel1Layout.columnWidths = new int[] {7, 7, 7, 7, 7};
307 jPanel1.setLayout(jPanel1Layout);
308 JLabel jLabelTemp1, jLabelTemp2;
309 java.util.Iterator<String> keyIter = testSetting.keySet().iterator();
310 int settingIndex = 0;
311 while (keyIter.hasNext()) {
312     StringBuffer labelBuf = new StringBuffer(" · ");
313     String key = keyIter.next();
314     labelBuf.append(key);
315     String value = testSetting.get(key);
316     System.out.print(key);
317     System.out.print(":");
318     System.out.println(value);
319     jLabelTemp1 = new JLabel(labelBuf.toString());
320     jLabelTemp2 = new JLabel(value);
321     jPanel1.add(jLabelTemp1, new GridBagConstraints(0, settingIndex * 2, 4, 1, 0.0,
322 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
323     jPanel1.add(jLabelTemp2, new GridBagConstraints(1, settingIndex * 2 + 1, 3, 1,
324 0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0,
325 0));
326     ++settingIndex;
327 }
328 return jPanel1;
329 }
330 private Component getJPanel3() {
331     jPanel3 = new JPanel();
332     GridBagLayout jPanel3Layout = new GridBagLayout();
333     jPanel3Layout.rowWeights = new double[] {0.1, 0.1, 0.1, 0.1};
334     jPanel3Layout.rowHeights = new int[] {7, 7, 7, 7};
335     jPanel3Layout.columnWeights = new double[] {0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.0, 0.1, 0.0,
336 0.1, 0.0, 0.1};
337     jPanel3Layout.columnWidths = new int[] {7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7};
338     jPanel3.setLayout(jPanel3Layout);
339     jPanel3.setBackground(new java.awt.Color(228,236,243));
340     jPanel3.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(Etch
341 edBorder.LOWERED), "Test Controller", TitledBorder.CENTER, TitledBorder.TOP));
342     jPanel3.add(getJButton1(), new GridBagConstraints(7, 2, 1, 1, 0.0, 0.0,
343 GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
344     jPanel3.add(getJButton2(), new GridBagConstraints(9, 2, 1, 1, 0.0, 0.0,
345 GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
346     jPanel3.add(getJButton3(), new GridBagConstraints(11, 2, 1, 1, 0.0, 0.0,
347 GridBagConstraints.WEST, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
348     jPanel3.add(getJButton4(), new GridBagConstraints(5, 2, 1, 1, 0.0, 0.0,
349 GridBagConstraints.EAST, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
350     jPanel3.add(getJRadioButton1(), new GridBagConstraints(0, 0, 1, 1, 0.0, 0.0,

```

[illegible]


```

401         doc = jTextPane1.getStyledDocument();
402         javax.swing.text.SimpleAttributeSetattr=new
403         javax.swing.text.SimpleAttributeSet();
404         StyleConstants.setBold(attr, true);
405         StyleConstants.setForeground(attr, Color.RED);
406         doc.insertString(doc.getLength(), "jTextPane2\n", attr);
407         StyleConstants.setBold(attr, false);
408         StyleConstants.setForeground(attr, Color.BLACK);
409         doc.insertString(doc.getLength(), "jTextPane3\n", attr);
410         addTestProcess();
411     }
412 }
413 {
414     JScrollPane1 = new JScrollPane();
415     JScrollPane1.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder
416 (EtchedBorder.LOWERED), "Test Setting", TitledBorder.LEADING, TitledBorder.TOP));
417     JScrollPane1.setBackground(bgColor);
418     JSplitPane2.add(JScrollPane1, JSplitPane.LEFT);
419     {
420         JPanel1 = new JPanel();
421         GridBagLayout jPanel1Layout = new GridBagLayout();
422         JScrollPane1.setViewportView(JPanel1);
423         JPanel1.setBackground(bgColor);
424         JPanel1.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(Etch
425 edBorder.LOWERED), "Test Setting", TitledBorder.LEADING, TitledBorder.TOP));
426         jPanel1Layout.rowWeights = new double[] {0.1, 0.1, 0.1, 0.1,
427 0.1, 0.1, 0.1, 0.1, 0.1};
428         jPanel1Layout.rowHeights = new int[] {25, 30, 25, 30, 25, 30,
429 25, 30, 7};
430         jPanel1Layout.columnWeights = new double[] {0.1, 0.1, 0.1,
431 0.1};
432         jPanel1Layout.columnWidths = new int[] {7, 7, 7, 7};
433         JPanel1.setLayout(jPanel1Layout);
434         {
435             JLabel jLabel1 = new JLabel();
436             jPanel1.add(jLabel1, new GridBagConstraints(0, 0, 2, 1,
437 0.0, 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
438             jLabel1.setText(" · Web Service:  ");
439         }
440         {
441             JLabel jLabel2 = new JLabel();
442             jPanel1.add(jLabel2, new GridBagConstraints(1, 1, 2, 1,
443 0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0,
444 0));
445             jLabel2.setText("localhost:8080/axis2/services/AddService");
446         }
447         {
448             JLabel jLabel3 = new JLabel();
449             jPanel1.add(jLabel3, new GridBagConstraints(0, 2, 2, 1,
450 0.0, 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));

```

```

451         jLabel3.setText(" · Operation: ");
452     }
453     {
454         JLabel jLabel4 = new JLabel();
455         jPanel1.add(jLabel4, new GridBagConstraints(1, 3, 1, 1,
456 0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0,
457 0));
458         jLabel4.setText("add");
459     }
460     {
461         JLabel jLabel5 = new JLabel();
462         jPanel1.add(jLabel5, new GridBagConstraints(0, 4, 2, 1,
463 0.0, 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
464         jLabel5.setText(" · Teseting Strategy:");
465     }
466     {
467         JLabel jLabel6 = new JLabel();
468         jPanel1.add(jLabel6, new GridBagConstraints(1, 5, 1, 1,
469 0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0,
470 0));
471         jLabel6.setText("Random Testing");
472     }
473     {
474         JLabel jLabel7 = new JLabel();
475         jPanel1.add(jLabel7, new GridBagConstraints(0, 6, 1, 1,
476 0.0, 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
477         jLabel7.setText(" · Test Suite:");
478     }
479     {
480         JLabel jLabel8 = new JLabel();
481         jPanel1.add(jLabel8, new GridBagConstraints(1, 7, 2, 1,
482 0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0,
483 0));
484         jLabel8.setText("testcaseset4addservice$add.xml");
485     }
486     }
487 }
488 jSplitPane2.setDividerLocation(200);
489 jSplitPane2.setDividerSize(5);
490 jSplitPane2.setEnabled(false);
491 }
492 {
493     jPanel3 = new JPanel();
494     GridBagLayout jPanel3Layout = new GridBagLayout();
495     jSplitPane1.add(jPanel3, JSplitPane.RIGHT);
496     jPanel3.setBackground(bgColor);
497     jPanel3.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(Etch
498 edBorder.LOWERED), "Test Controller", TitledBorder.CENTER, TitledBorder.TOP));
499     jPanel3Layout.rowWeights = new double[] {0.1, 0.1, 0.1, 0.1};
500     jPanel3Layout.rowHeights = new int[] {7, 7, 7, 7};

```

```
501         jPanel3Layout.columnWeights = new double[] {0.1, 0.1, 0.1, 0.1, 0.1, 0.1,
502 0.1, 0.1, 0.1, 0.1, 0.1, 0.1};
503         jPanel3Layout.columnWidths = new int[] {7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
504 7};
505         jPanel3.setLayout(jPanel3Layout);
506         {
507             jButton1 = new JButton();
508             jButton1.setText("Start");
509             jButton1.setAction(getAbstractAction1());
510             jPanel3.add(jButton1, new GridBagConstraints(7, 2, 1, 1, 0.0, 0.0,
511 GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
512             jPanel3.add(getJButton2(), new GridBagConstraints(9, 2, 1, 1, 0.0,
513 0.0, GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0,
514 0));
515             jPanel3.add(getJButton3(), new GridBagConstraints(11, 2, 1, 1, 0.0,
516 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
517             jPanel3.add(getJButton4(), new GridBagConstraints(5, 2, 1, 1, 0.0,
518 0.0, GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0,
519 0));
520         }
521     }
522     jSplitPane1.setDividerLocation(450);
523     jSplitPane1.setDividerSize(5);
524     jSplitPane1.setEnabled(false);
525 }
526 pack();
527 setSize(800, 600);
528 } catch (Exception e) {
529     e.printStackTrace();
530 }
531 }
532 private void addTestProcess() {
533     timeAction = new Timer(2000, new ActionListener() {
534         @Override
535         public void actionPerformed(ActionEvent e) {
536             SimpleAttributeSet attr = new SimpleAttributeSet();
537             if (testI++ < 10) {
538                 if (testI % 3 == 0) {
539                     StyleConstants.setBold(attr, true);
540                     StyleConstants.setForeground(attr, Color.RED);
541                 } else {
542                     StyleConstants.setBold(attr, false);
543                     StyleConstants.setForeground(attr, Color.BLACK);
544                 }
545                 try {
546                     doc.insertString(doc.getLength(), "动态添加第"+testI+"行\n", attr);
547                 } catch (Exception e1) {
548                     e1.printStackTrace();
549                 }
550             }
551         }
552     });
553 }
```

```

551         }
552     });
553     timeAction.start();
554 }
555 private AbstractAction getAbstractAction1() {
556     if(abstractAction1 == null) {
557         abstractAction1 = new AbstractAction("Start", null) {
558             int tem = -1;
559             int tec = -1;
560             public void actionPerformed(ActionEvent evt) {
561                 if (jRadioButton1.isSelected()) {
562                     tem = new Integer(jTextField1.getText());
563                     tec = TestCaseExecuter.NUM_OF_CASES;
564                 } else if (jRadioButton2.isSelected()){
565                     tem = new Integer(jTextField2.getText());
566                     tec = TestCaseExecuter.NUM_OF_FAULTS;
567                 }
568                 if (testSet != null) {
569                     executeThread = new Thread(new
570 TestCaseExecuter(testSetting.get("Web Service"), operationInfo, testSet, pQueue, tec, tem));
571                 } else if (testSetWP != null) {
572                     if (testSetting.get("Epsilon") == null) {
573                         executeThread = new Thread(new
574 TestCaseExecuter(testSetting.get("Web Service"), operationInfo, testSetWP, pQueue, tec, tem));
575                     } else {
576                         double e = Double.parseDouble(testSetting.get("Epsilon"));
577                         executeThread = new Thread(new
578 TestCaseExecuter(testSetting.get("Web Service"), operationInfo, testSetWP, e, pQueue, tec, tem));
579                     }
580                 } else {}
581                 executeThread.start();
582                 showProcessThread = new Thread(new TestProcessWriter(doc,
583 pQueue));
584                 showProcessThread.start();
585             }
586         };
587     }
588     return abstractAction1;
589 }
590 private JButton getJButton2() {
591     if(jButton2 == null) {
592         jButton2 = new JButton();
593         jButton2.setText("Stop");
594     }
595     return jButton2;
596 }
597 private JButton getJButton3() {
598     if(jButton3 == null) {
599         jButton3 = new JButton();
600         jButton3.setText("Download report");

```

```
601     }
602     return jButton3;
603 }
604 private JButton getJButton4() {
605     if(jButton4 == null) {
606         jButton4 = new JButton();
607         jButton4.setText("<< Back");
608     }
609     return jButton4;
610 }
611 private ButtonGroup getButtonGroup1() {
612     if(buttonGroup1 == null) {
613         buttonGroup1 = new ButtonGroup();
614         buttonGroup1.add(jRadioButton1);
615         buttonGroup1.add(jRadioButton2);
616     }
617     return buttonGroup1;
618 }
619 private JRadioButton getJRadioButton1() {
620     if(jRadioButton1 == null) {
621         jRadioButton1 = new JRadioButton();
622         jRadioButton1.setBackground(new java.awt.Color(228,236,243));
623         jRadioButton1.setText("执行");
624         jRadioButton1.setSelected(true);
625     }
626     return jRadioButton1;
627 }
628 private JRadioButton getJRadioButton2() {
629     if(jRadioButton2 == null) {
630         jRadioButton2 = new JRadioButton();
631         jRadioButton2.setBackground(new java.awt.Color(228,236,243));
632         jRadioButton2.setText("杀死");
633     }
634     return jRadioButton2;
635 }
636 private JTextField getJTextField1() {
637     if(jTextField1 == null) {
638         jTextField1 = new JTextField();
639         jTextField1.setText("");
640     }
641     return jTextField1;
642 }
643 private JTextField getJTextField2() {
644     if(jTextField2 == null) {
645         jTextField2 = new JTextField();
646         jTextField2.setText("");
647     }
648     return jTextField2;
649 }
650 private JLabel getJLabel9() {
```

```
651         if(jLabel9 == null) {
652             jLabel9 = new JLabel();
653             jLabel9.setText("个测试用例后停止");
654         }
655         return jLabel9;
656     }
657     private JLabel getJLabel10() {
658         if(jLabel10 == null) {
659             jLabel10 = new JLabel();
660             jLabel10.setText("个错误后停止");
661         }
662         return jLabel10;
663     }
664     private JTextField getJTextField3() {
665         if(jTextField3 == null) {
666             jTextField3 = new JTextField();
667             jTextField3.setText("");
668             jTextField3.addKeyListener(new KeyAdapter() {
669                 public void keyPressed(KeyEvent evt) {
670                     jTextField3KeyPressed(evt);
671                 }
672             });
673         }
674         return jTextField3;
675     }
676     private JLabel getJLabel11() {
677         if(jLabel11 == null) {
678             jLabel11 = new JLabel();
679             jLabel11.setText("概率调整参数: ");
680         }
681         return jLabel11;
682     }
683     private void jTextField3KeyPressed(KeyEvent evt) {
684         System.out.println("jTextField3.keyPressed, event="+evt);
685         if (evt.getKeyChar() == '\n') {
686             System.out.println(jTextField3.getText());
687             double e = Double.parseDouble(jTextField3.getText());
688             testSetting.put("Epsilon", jTextField3.getText());
689         }
690     }
691 }
692 package ui;
693 import java.awt.BorderLayout;
694 import java.awt.FlowLayout;
695 import java.awt.GridLayout;
696 import java.awt.event.ActionEvent;
697 import java.util.Map;
698 import java.util.Set;
699 import javax.swing.*;
700 import cn.edu.ustb.mt4ws.tcg.WsdlOperationFormat;
```

```
701 import cn.edu.ustb.mt4ws.wsdl.parser.WsdlReader11;
702 import cn.edu.ustb.webservicetester.controller.ParseWsdlController;
703 import cn.edu.ustb.webservicetester.util.WsdlParser;
704 public class MainFrame extends javax.swing.JFrame {
705     private JPanel jPanel1;
706     private JTextField jTextField_wsdl;
707     private AbstractAction ResetWSDLAction;
708     private AbstractAction ParseWSDLAction;
709     private JButton jButton_wsdlreset;
710     private JButton jButton_wsdlJIEXI;
711     private JLabel jLabel_wsdl;
712     public static void main(String[] args) {
713         SwingUtilities.invokeLater(new Runnable() {
714             public void run() {
715                 MainFrame inst = new MainFrame();
716                 inst.setLocationRelativeTo(null);
717                 inst.setVisible(true);
718             }
719         });
720     }
721     public MainFrame() {
722         super();
723         initGUI();
724     }
725     private void initGUI() {
726         try {
727             this.setTitle("服务定位");
728             this.setResizable(false); // 关闭最大化
729             GridLayout thisLayout = new GridLayout(1, 1);
730             getContentPane().setLayout(thisLayout);
731             setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
732             {
733                 jPanel1 = new JPanel();
734                 getContentPane().add(jPanel1);
735                 jPanel1.setBackground(new java.awt.Color(228, 236, 243));
736                 jPanel1.setPreferredSize(new java.awt.Dimension(403, 275));
737                 jPanel1.setLayout(null);
738                 {
739                     jTextField_wsdl = new JTextField();
740                     jPanel1.add(jTextField_wsdl);
741                     jTextField_wsdl.setBounds(20, 65, 347, 24);
742                     jTextField_wsdl.setText("http://localhost:8080/axis2/services/AddService?wsdl");
743                 }
744                 {
745                     jLabel_wsdl = new JLabel();
746                     jPanel1.add(jLabel_wsdl);
747                     jLabel_wsdl.setText("\u8bf7\u8f93\u5165\u5f85\u6d4b\u670d\u52a1\u5730\u5740");
748                     jLabel_wsdl.setBounds(18, 28, 122, 17);
749                     jButton_wsdlreset = new JButton();
750                     jPanel1.add(jButton_wsdlreset);
```

```
751         jButton_wsdlreset.setText("\u91cd  \u7f6e");
752         jButton_wsdlreset.setBounds(244, 157, 80, 30);
753         jButton_wsdlreset.setAction(getResetWSDLAction());
754     }
755     {
756         jButton_wsdlJIEXI = new JButton();
757         jPanel1.add(jButton_wsdlJIEXI);
758         jButton_wsdlJIEXI.setText("\u89e3  \u6790");
759         jButton_wsdlJIEXI.setBounds(60, 157, 80, 30);
760         jButton_wsdlJIEXI.setAction(getParseWSDLAction());
761     }
762     {
763         jButton_wsdlreset = new JButton();
764         jPanel1.add(jButton_wsdlreset);
765         jButton_wsdlreset.setText("\u91cd  \u7f6e");
766         jButton_wsdlreset.setBounds(244, 157, 80, 30);
767         jButton_wsdlreset.setAction(getResetWSDLAction());
768     }
769     }
770     pack();
771     this.setSize(412, 311);
772 } catch (Exception e) {
773     e.printStackTrace();
774 }
775 }
776 private AbstractAction getParseWSDLAction() {
777     if(ParseWSDLAction == null) {
778         ParseWSDLAction = new AbstractAction("\u89e3  \u6790", null) {
779             public void actionPerformed(ActionEvent evt) {
780                 String wsdlurl = jTextField_wsdl.getText();
781                 ParseWsdlControllor parseControllor = new ParseWsdlControllor();
782                 parseControllor.parseWsdlControl(wsdlurl);JFrame);
783             OperationSelectFrame(opList);
784             }
785         };
786     }
787     return ParseWSDLAction;
788 }
789 private AbstractAction getResetWSDLAction() {
790     if(ResetWSDLAction == null) {
791         ResetWSDLAction = new AbstractAction("\u91cd  \u7f6e", null) {
792             public void actionPerformed(ActionEvent evt) {
793                 jTextField_wsdl.setText("");
794             }
795         };
796     }
797     return ResetWSDLAction;
798 }
799 }
800 package ui;
```



```
801 import java.awt.FlowLayout;
802 import java.awt.GridLayout;
803 import java.awt.event.ActionEvent;
804 import javax.swing.AbstractAction;
805 import javax.swing.ComboBoxModel;
806 import javax.swing.DefaultComboBoxModel;
807 import javax.swing.JComboBox;
808 import javax.swing.JLabel;
809 import javax.swing.JList;
810 import javax.swing.JPanel;
811 import javax.swing.JScrollBar;
812 import javax.swing.JScrollPane;
813 import javax.swing.JSlider;
814 import javax.swing.JSpinner;
815 import javax.swing.JTextArea;
816 import javax.swing.ListModel;
817 import javax.swing.ListSelectionModel;
818 import javax.swing.SpinnerListModel;
819 import javax.swing.WindowConstants;
820 import javax.swing.SwingUtilities;
821 import javax.swing.event.ListSelectionEvent;
822 import javax.swing.event.ListSelectionListener;
823 import cn.edu.ustb.webservicetester.model.OperationInfo;
824 import cn.edu.ustb.webservicetester.model.ParameterInfo;
825 public class OperationSelectionFrame2 extends javax.swing.JFrame {
826     private JPanel jPanel1;
827     private JScrollPane jScrollPane1;
828     private JList parameterInfoJList;
829     private JScrollPane jScrollPane2;
830     private JTextArea jTextAreaParameterRestriction;
831     private JLabel jLabelParameterRestriction;
832     private JLabel jLabelParameterInformation;
833     private AbstractAction operationSelectionAction;
834     private JComboBox operationSelectionComboBox;
835     private JLabel operationSelectionLabel;
836     private java.util.List<OperationInfo> opInfoList;
837     private java.util.List<ParameterInfo> pmInfoList;
838     private ListSelectionListener lslistener;
839     public static void main(String[] args) {
840         SwingUtilities.invokeLater(new Runnable() {
841             public void run() {
842                 OperationSelectionFrame2 inst = new OperationSelectionFrame2();
843                 inst.setLocationRelativeTo(null);
844                 inst.setVisible(true);
845             }
846         });
847     }
848     public OperationSelectionFrame2() {
849         super();
850         initGUI();
```

```

851     }
852     public OperationSelectionFrame2(String frame) {
853         super();
854         initGUI();
855     }
856     public OperationSelectionFrame2(java.util.List list) {
857         super();
858         lslister = new ListSelectionListener() {
859             public void valueChanged(ListSelectionEvent e) {
860                 parameterSelectionChanged();
861             }
862         };
863         initGUI(list);
864     }
865     private void initGUI() {
866         initGUI(null);
867     }
868     private void initGUI(java.util.List<OperationInfo> oil) {
869         this.opInfoList = oil;
870         try {
871             setTitle("操作选择");
872             setResizable(false);
873             GridLayout thisLayout = new GridLayout(1, 1);
874             getContentPane().setLayout(thisLayout);
875             setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
876             {
877                 JPanel1 = new JPanel();
878                 getContentPane().add(JPanel1);
879                 JPanel1.setLayout(null);
880                 JPanel1.setBackground(new java.awt.Color(228,236,243));
881                 JPanel1.setPreferredSize(new java.awt.Dimension(506, 349));
882                 {
883                     operationSelectionLabel = new JLabel();
884                     JPanel1.add(operationSelectionLabel);
885                     operationSelectionLabel.setText("\u8bf7\u9009\u62e9\u5f85\u6d4b\u8bd5\u7684\u64cd\u4f5
886 c");
887                     operationSelectionLabel.setBounds(12, 36, 128, 17);
888                 }
889                 {
890                     ComboBoxModel operationSelectionComboBoxModel;
891                     if (opInfoList == null)
892                         operationSelectionComboBoxModel =
893                             new DefaultComboBoxModel(
894                                 new String[] { "Item One", "Item Two" });
895                     else {
896                         String[] opNames = new String[opInfoList.size()];
897                         for (int i = 0; i < opNames.length; ++i) {
898                             opNames[i] = opInfoList.get(i).getOpName();
899                         }
900                         operationSelectionComboBoxModel = new

```

```

901 DefaultComboBoxModel(opNames);
902     }
903     operationSelectionComboBox = new JComboBox();
904     jPanel1.add(operationSelectionComboBox);
905     operationSelectionComboBox.setModel(operationSelectionComboBoxModel);
906     operationSelectionComboBox.setBounds(140, 32, 162, 24);
907     operationSelectionComboBox.setAction(getOperationSelectionAction());
908     }
909     {
910         jScrollPane1 = new JScrollPane();
911         jPanel1.add(jScrollPane1);
912         jScrollPane1.setBounds(24, 101, 188, 132);
913         jPanel1 = new JPanel();
914         getContentPane().add(jPanel1);
915         jPanel1.setLayout(null);
916         jPanel1.setBackground(new java.awt.Color(228,236,243));
917         jPanel1.setPreferredSize(new java.awt.Dimension(506, 349));
918
919         {
920             ListModel parameterInfoListModel =
921                 new DefaultComboBoxModel(
922                     new String[] {});
923             parameterInfoJList = new JList();
924             parameterInfoJList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
925             jScrollPane1.setViewportViewView(parameterInfoJList);
926             parameterInfoJList.setModel(parameterInfoListModel);
927             parameterInfoJList.setPreferredSize(new java.awt.Dimension(185,
928 129));
929             parameterInfoJList.addListSelectionListener(lslister);
930         }
931     }
932     {
933         jScrollPane2 = new JScrollPane();
934         jPanel1.add(jScrollPane2);
935         jPanel1.add(getJLabelParameterInformation());
936         jPanel1.add(getJLabelParameterRestriction());
937         jScrollPane2.setBounds(274, 101, 183, 132);
938         jScrollPane2.setViewportViewView(getJTextAreaParameterRestriction());
939     }
940 }
941 pack();
942 this.setSize(502, 377);
943 } catch (Exception e) {
944     e.printStackTrace();
945 }
946 }
947 private AbstractAction getOperationSelectionAction() {
948     if(operationSelectionAction == null) {
949         operationSelectionAction = new AbstractAction("", null) {
950             public void actionPerformed(ActionEvent evt) {

```

```

951         int opIndexSelected = operationSelectionComboBox.getSelectedIndex();
952         OperationInfo opInfo = opInfoList.get(opIndexSelected);
953         pmInfoList = opInfo.getParaList();
954         DefaultComboBoxModel dcbm =
955 (DefaultComboBoxModel)(parameterInfoJList.getModel());
956         dcbm.removeAllElements();
957         for (ParameterInfo pi : pmInfoList) {
958             dcbm.addElement(pi.toString());
959         }
960     }
961 };
962 }
963 return operationSelectionAction;
964 }
965 private JLabel getJLabelParameterInformation() {
966     if(jLabelParameterInformation == null) {
967         jLabelParameterInformation = new JLabel();
968         jLabelParameterInformation.setText("\u8f93\u5165\u53c2\u6570");
969         jLabelParameterInformation.setBounds(24, 73, 61, 17);
970     }
971     return jLabelParameterInformation;
972 }
973 private JLabel getJLabelParameterRestriction() {
974     if(jLabelParameterRestriction == null) {
975         jLabelParameterRestriction = new JLabel();
976         jLabelParameterRestriction.setText("\u53c2\u6570\u8bf4\u660e");
977         jLabelParameterRestriction.setBounds(274, 73, 64, 17);
978     }
979     return jLabelParameterRestriction;
980 }
981 private JTextArea getJTextAreaParameterRestriction() {
982     if(jTextAreaParameterRestriction == null) {
983         jLabelParameterRestriction = new JLabel();
984         jLabelParameterRestriction.setText("\u53c2\u6570\u8bf4\u660e");
985         jLabelParameterRestriction.setBounds(274, 73, 64, 17);
986         jTextAreaParameterRestriction = new JTextArea();
987         jTextAreaParameterRestriction.setEditable(false);
988     }
989     return jTextAreaParameterRestriction;
990 }
991 private void parameterSelectionChanged() {
992     int parameterIndexSelected = parameterInfoJList.getSelectedIndex();
993     if (parameterIndexSelected >= 0) {
994         String rstrction = pmInfoList.get(parameterIndexSelected).getRestriction();
995         jTextAreaParameterRestriction.setText(rstrction);
996     }
997 }
998 }
999 package ui;
1000 import java.awt.BorderLayout;

```

```
1001 import java.awt.Component;
1002 import java.awt.GridLayout;
1003 import java.awt.event.ActionEvent;
1004 import java.awt.event.ItemEvent;
1005 import java.awt.event.ItemListener;
1006 import java.awt.event.KeyAdapter;
1007 import java.awt.event.KeyEvent;
1008 import javax.swing.AbstractAction;
1009 import javax.swing.ButtonGroup;
1010 import javax.swing.ComboBoxModel;
1011 import javax.swing.DefaultComboBoxModel;
1012 import javax.swing.JButton;
1013 import javax.swing.JCheckBox;
1014 import javax.swing.JComboBox;
1015 import javax.swing.JLabel;
1016 import javax.swing.JPanel;
1017 import javax.swing.JRadioButton;
1018 import javax.swing.JScrollPane;
1019 import javax.swing.JTable;
1020 import javax.swing.WindowConstants;
1021 import javax.swing.SwingUtilities;
1022 import javax.swing.table.DefaultTableModel;
1023 import cn.edu.ustb.webservicetester.model.ParameterIntervalScenario;
1024 import cn.edu.ustb.webservicetester.model.ParameterInfo;
1025 import cn.edu.ustb.webservicetester.model.ParameterPartitionScenario;
1026 public class PartitionSettingFrame extends javax.swing.JFrame implements ItemListener{
1027     private java.util.List<ParameterInfo> paraInfos;
1028     private java.util.List<JComboBox> paraTypeList;
1029     private java.util.List<JCheckBox> paraList;
1030     private JLabel partitionBasisJLabel;
1031     private JLabel partitionSettingJLabel;
1032     private JPanel jPanel1;
1033     private JScrollPane jScrollPane1;
1034     private java.util.Map<JCheckBox, JComboBox> paraTypePair;
1035     private JTable paraPartitionTable;
1036     private DefaultTableModel paraPartitionTableModel;
1037     private java.util.Map<String, DefaultTableModel> paraPartitionScenarioTableModels;
1038     private JButton preStepButton;
1039     private JButton nextStepButton;
1040     public static final int DISTANCE1 = 29;
1041     public static final int DISTANCE2 = 30;
1042     public static final int CHECK_BOX_X = 26;
1043     public static final int CHECK_BOX_Y = 51;
1044     public static final int CHECK_BOX_WIDTH = 105;
1045     public static final int CHECK_BOX_HEIGHT = 21;
1046     private AbstractAction nextStepAction;
1047     public static final int COMBO_BOX_X = 155;
1048     public static final int COMBO_BOX_Y = 49;
1049     public static final int COMBO_BOX_WIDTH = 118;
1050     public static final int COMBO_BOX_HEIGHT = 24;
```

```
1051     public static void main(String[] args) {
1052         SwingUtilities.invokeLater(new Runnable() {
1053             public void run() {
1054                 PartitionSettingFrame inst = new PartitionSettingFrame();
1055                 inst.setLocationRelativeTo(null);
1056                 inst.setVisible(true);
1057             }
1058         });
1059     }
1060     public PartitionSettingFrame() {
1061         super();
1062         initGUI();
1063     }
1064     public PartitionSettingFrame(java.util.List<ParameterInfo> paraInfoList) {
1065         super();
1066         initGUI(paraInfoList);
1067     }
1068     private void initGUI() {
1069         initGUI(null);
1070     }
1071     private void initGUI(java.util.List<ParameterInfo> paraInfoList) {
1072         if (paraInfoList == null) {
1073             paraInfos = new java.util.LinkedList<ParameterInfo>();
1074             for (int i = 0; i < 4; ++i) {
1075                 ParameterInfo tpi = new ParameterInfo();
1076                 tpi.setName("parameter " + (i + 1));
1077                 tpi.setRestriction("");
1078                 tpi.setClassInfo("Object");
1079                 paraInfos.add(tpi);
1080             }
1081         }
1082         else {
1083             paraInfos = paraInfoList;
1084         }
1085         try {
1086             ParameterInfo tpi = new ParameterInfo();
1087             tpi.setName("parameter " + (i + 1));
1088             tpi.setRestriction("");
1089             tpi.setClassInfo("Object");
1090             paraInfos.add(tpi);
1091             GridLayout thisLayout = new GridLayout(1, 1);
1092             thisLayout.setHgap(5);
1093             thisLayout.setVgap(5);
1094             thisLayout.setColumns(1);
1095             getContentPane().setLayout(thisLayout);
1096             setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
1097             this.setTitle("\u5206\u533a\u8bbe\u5b9a");
1098             setResizable(false);
1099         }
1100         JPanel1 = new JPanel();
```

```

1101         getContentPane().add(jPanel1);
1102         jPanel1.setLayout(null);
1103         jPanel1.setBackground(new java.awt.Color(252,252,203));
1104         {
1105             partitionBasisJLabel = new JLabel();
1106             jPanel1.add(partitionBasisJLabel);
1107             partitionBasisJLabel.setText("1.
1108 \u9009\u62e9\u5206\u533a\u4f9d\u636e\u53c2\u6570");
1109             partitionBasisJLabel.setBounds(26, 22, 129, 17);
1110         }
1111         {
1112             paraList = new java.util.ArrayList<JCheckBox>(paraInfos.size());
1113             paraTypeList = new java.util.ArrayList<JComboBox>(paraInfos.size());
1114             paraPartitionScenarioTableModels = new java.util.HashMap<String,
1115 DefaultTableModel>(paraInfos.size());
1116             paraTypePair = new java.util.HashMap<JCheckBox, JComboBox>();
1117             for (int i = 0; i < paraInfos.size(); ++i) {
1118                 JCheckBox tempCheckBox = new JCheckBox();
1119                 jPanel1.add(tempCheckBox);
1120                 tempCheckBox.setText(paraInfos.get(i).getName());
1121                 tempCheckBox.setBounds(CHECK_BOX_X, CHECK_BOX_Y +
1122 DISTANCE2 * i, CHECK_BOX_WIDTH, CHECK_BOX_HEIGHT);
1123                 tempCheckBox.addItemListener(this);
1124                 tempCheckBox.setBackground(new java.awt.Color(252, 252, 203));
1125                 paraList.add(tempCheckBox);
1126                 JComboBox tempComboBox = new JComboBox();
1127                 jPanel1.add(tempComboBox);
1128                 tempComboBox.setModel(getComboBoxModel());
1129                 tempComboBox.setBounds(COMBO_BOX_X, COMBO_BOX_Y
1130 + DISTANCE2 * i, COMBO_BOX_WIDTH, COMBO_BOX_HEIGHT);
1131                 tempComboBox.setVisible(false);
1132                 paraTypeList.add(tempComboBox);
1133                 paraTypePair.put(tempCheckBox, tempComboBox);
1134                 DefaultTableModel tdtm = new DefaultTableModel(null, new
1135 String[]{"", "参数 " + tempCheckBox.getText() + " 的分区区间", "权重"});
1136                 paraPartitionScenarioTableModels.put(tempCheckBox.getText(),
1137 tdtm);
1138             }
1139         }
1140         {
1141             partitionSettingJLabel = new JLabel();
1142             jPanel1.add(partitionSettingJLabel);
1143             partitionSettingJLabel.setText("2. 设定各参数分区");
1144             partitionSettingJLabel.setBounds(26, 22 + DISTANCE1 * 2 +
1145 DISTANCE2 * (paraInfos.size() - 1), 129, 17);
1146         }
1147         {
1148             jScrollPane1 = new JScrollPane();
1149             jPanel1.add(jScrollPane1);
1150             jScrollPane1.setBounds(26, 22 + DISTANCE1 * 3 + DISTANCE2 *

```

```

1151 (paraInfos.size() - 1), 500 - 26 * 2, 444 - (22 + DISTANCE1 * 3 + DISTANCE2 * (paraInfos.size()
1152 - 1)) - 30 - DISTANCE1 * 3);
1153     jScrollPane1.setViewportViewView(getParaPartitionTable());
1154     }
1155     {
1156         preStepButton = new JButton();
1157         jPanel1.add(preStepButton);
1158         preStepButton.setText("上一步");
1159         preStepButton.setBounds(60, 444 - 30 - DISTANCE1 * 2, 80, 30);
1160     }
1161     {
1162         tempCheckBox.setBackground(new java.awt.Color(252, 252, 203));
1163         paraList.add(tempCheckBox);
1164         JComboBox tempComboBox = new JComboBox();
1165         Panel1.add(tempComboBox);
1166         tempComboBox.setModel(getComboBoxModel());
1167         nextStepButton = new JButton();
1168         jPanel1.add(nextStepButton);
1169         nextStepButton.setText("下一步");
1170         nextStepButton.setBounds(500 - 60 - 80, 444 - 30 - DISTANCE1 * 2, 80,
1171 30);
1172         nextStepButton.setAction(getNextStepAction());
1173     }
1174     }
1175     pack();
1176     setSize(500, 444);
1177 } catch (Exception e) {
1178     e.printStackTrace();
1179 }
1180 }
1181 private Component getParaPartitionTable() {
1182     if (paraPartitionTable == null) {
1183         String[] name = {"", "参数 的分区区间", "权重"};
1184         paraPartitionTableModel = new DefaultTableModel(null, name);
1185         paraPartitionTableModel.addRow(new String[]{"1", "", ""});
1186         paraPartitionTable = new JTable(paraPartitionTableModel) {
1187             public boolean isCellEditable(int row, int column) {
1188                 if (column >= 1)
1189                     return true;
1190                 else return false;
1191             }
1192         };
1193         paraPartitionTable.addKeyListener(new KeyAdapter(){
1194             public void keyReleased(KeyEvent e) {
1195                 if (e.getKeyChar() == '\n') {
1196                     if (paraPartitionTable.getSelectedColumn() >=
1197 paraPartitionTable.getColumnCount() - 1
1198 && paraPartitionTable.getSelectedRow() >=
1199 paraPartitionTable.getRowCount() - 1
1200 &&

```



```

1201 paraPartitionTable.getValueAt(paraPartitionTable.getSelectedRow(),
1202 paraPartitionTable.getSelectedColumn()).toString().length() > 0) {
1203     System.out.println(paraPartitionTable.getValueAt(paraPartitionTable.getSelectedRow(),
1204 paraPartitionTable.getSelectedColumn()).toString().length());
1205         paraPartitionTable.editCellAt(paraPartitionTable.getSelectedRow(),
1206 paraPartitionTable.getSelectedColumn());
1207         ((DefaultTableModel)paraPartitionTable.getModel()).addRow(new
1208 String[]{ new Integer(paraPartitionTable.getRowCount() + 1).toString(), "", ""});
1209         paraPartitionTable.editCellAt(paraPartitionTable.getRowCount() - 1, 0);
1210     }
1211 }
1212 }
1213 });
1214 }
1215 return paraPartitionTable;
1216 }
1217 public void itemStateChanged(ItemEvent e) {
1218     Object source = e.getItemSelectable();
1219     JCheckBox changedCheckBox = (JCheckBox)source;
1220     if (changedCheckBox.isSelected()) {
1221         paraTypePair.get(changedCheckBox).setVisible(true);
1222 = paraPartitionSenarioTableModels.get(changedCheckBox.getText());
1223         paraPartitionTable.setModel(paraPartitionScenarioTableModels.get(changedCheckBox.getTe
1224 xt()));
1225         if
1226 (paraPartitionScenarioTableModels.get(changedCheckBox.getText()).getRowCount() <= 0)
1227         paraPartitionScenarioTableModels.get(changedCheckBox.getText()).addRow(new
1228 String[]{"1", "", ""});
1229     } else {
1230         paraTypePair.get(changedCheckBox).setVisible(false);
1231     }
1232 }
1233 private ComboBoxModel getComboBoxModel() {
1234     ComboBoxModel jComboBoxModel =
1235         new DefaultComboBoxModel(
1236             new String[] { "连续型", "枚举型" });
1237     return jComboBoxModel;
1238 }
1239 private AbstractAction getNextStepAction() {
1240     if(nextStepAction == null) {
1241         nextStepAction = new AbstractAction("\u4e0b\u4e00\u6b65", null) {
1242             public void actionPerformed(ActionEvent evt) {
1243                 java.util.List<ParameterPartitionScenario> partitionSetting = new
1244 java.util.ArrayList<ParameterPartitionScenario>();
1245                 ParameterPartitionScenario tempScenario;
1246                 for (JCheckBox jcb : paraList) {
1247                     if (jcb.isSelected()) {
1248                         tempScenario = new ParameterPartitionScenario();
1249                         for (ParameterInfo pi : paraInfos) {
1250                             if (pi.getName().equals(jcb.getText())) {

```

```
1251         tempScenario.setParaInfo(pi);
1252     }
1253 }
1254 if ("连续的".equals(paraTypePair.get(jcb).getSelectedItem())) {
1255     for (ParameterInfo pi : paraInfos) {
1256         if (pi.getName().equals(jcb.getText())) {
1257             pi.setType(ParameterInfo.CONTINUOUS);
1258         }
1259     }
1260 } else {
1261     for (ParameterInfo pi : paraInfos) {
1262         if (pi.getName().equals(jcb.getText())) {
1263             pi.setType(ParameterInfo.SCATTERED);
1264         }
1265     }
1266 }
1267 DefaultTableModel dtm =
1268 paraPartitionScenarioTableModels.get(jcb.getText());
1269 for (int i = 0; i < dtm.getRowCount(); ++i) {
1270     ParameterIntervalScenario paraInter = new
1271 ParameterIntervalScenario();
1272     paraInter.setRange(dtm.getValueAt(i, 1).toString());
1273     paraInter.setWeight(Integer.valueOf(dtm.getValueAt(i,
1274 2).toString()));
1275     tempScenario.add(paraInter);
1276 }
1277 partitionSetting.add(tempScenario);
1278 } else {
1279     ;
1280 }
1281 }
1282 }
1283 };
1284 }
1285 return nextStepAction;
1286 }
1287 }
1288 package ui;
1289 import java.awt.BorderLayout;
1290 import java.awt.Component;
1291 import java.awt.GridLayout;
1292 import java.awt.event.ActionEvent;
1293 import java.awt.event.KeyAdapter;
1294 import java.awt.event.KeyEvent;
1295 import java.util.Map;
1296 import javax.swing.AbstractAction;
1297 import javax.swing.ComboBoxModel;
1298 import javax.swing.DefaultCellEditor;
1299 import javax.swing.DefaultComboBoxModel;
1300 import javax.swing.JButton;
```

```
1301 import javax.swing.JComboBox;
1302 import javax.swing.JPanel;
1303 import javax.swing.JScrollPane;
1304 import javax.swing.JTable;
1305 import javax.swing.WindowConstants;
1306 import javax.swing.table.DefaultTableModel;
1307 import javax.swing.table.TableColumn;
1308 import javax.swing.table.TableColumnModel;
1309 import javax.swing.table.TableModel;
1310 import javax.swing.SwingUtilities;
1311 import cn.edu.ustb.webservicetester.model.OperationInfo;
1312 import cn.edu.ustb.webservicetester.model.ParameterInfo;
1313 import cn.edu.ustb.webservicetester.model.TestSet;
1314 import cn.edu.ustb.webservicetester.model.TestSetWithPartition;
1315 import cn.edu.ustb.webservicetester.util.Partitioner;
1316 public class PartitionSettingFrame2 extends javax.swing.JFrame {
1317     private JPanel jPanel1;
1318     private JScrollPane jScrollPane1;
1319     private AbstractAction abstractAction1;
1320     private JComboBox jComboBox1;
1321     private JTable jTable1;
1322     private TableModel jTable1Model;
1323     private AbstractAction abstractAction2;
1324     private JScrollPane jScrollPane2;
1325     private JTable jTable2;
1326     private TableModel jTable2Model;
1327     private JButton jButtonPre;
1328     private JButton jButtonNext;
1329     private AbstractAction removePartitionAction;
1330     private JButton jButtonRemovePartition;
1331     private AbstractAction abstractActionAddPartition;
1332     private JButton jButtonAddPartition;
1333     private java.util.List<ParameterInfo> paraInfoList;
1334     private OperationInfo operaInfo;
1335     private Map<String, String> testSetting;
1336     private TestSet testSet;
1337     private boolean addColumn = true;
1338     public static void main(String[] args) {
1339         SwingUtilities.invokeLater(new Runnable() {
1340             public void run() {
1341                 PartitionSettingFrame2 inst = new PartitionSettingFrame2();
1342                 inst.setLocationRelativeTo(null);
1343                 inst.setVisible(true);
1344             }
1345         });
1346     }
1347     public PartitionSettingFrame2() {
1348         super();
1349         initGUI();
1350     }
```

```
1351     public PartitionSettingFrame2(Map<String, String> testSettingMap, OperationInfo oi,
1352 TestSet ts) {
1353         super();
1354         operaInfo = oi;
1355         paraInfoList = operaInfo.getParaList();
1356         testSetting = testSettingMap;
1357         testSet = ts;
1358         initGUI();
1359     }
1360     private void initGUI() {
1361         try {
1362             setTitle("分区设定");
1363             setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
1364             this.setResizable(false);
1365             GridLayout thisLayout = new GridLayout(1, 1);
1366             getContentPane().setLayout(thisLayout);
1367             getContentPane().setBackground(new java.awt.Color(228,236,243));
1368             {
1369                 JPanel1 = new JPanel();
1370                 getContentPane().add(jPanel1);
1371                 jPanel1.setLayout(null);
1372                 jPanel1.setBackground(new java.awt.Color(228,236,243));
1373                 {
1374                     JScrollPane1 = new JScrollPane();
1375                     jPanel1.add(jScrollPane1);
1376                     jScrollPane1.setBounds(12, 10, 470, 200);
1377                     jScrollPane1.setViewportView(getJTable1());
1378                 }
1379                 {
1380                     ComboBoxModel jComboBox1Model = new DefaultComboBoxModel(
1381                         new String[] { "平均分布", "比例分布", "人工设定" });
1382                     jComboBox1 = new JComboBox();
1383                     jPanel1.add(jComboBox1);
1384                     jPanel1.add(getJButtonAddPartition());
1385                     jPanel1.add(getJButtonRemovePartition());
1386                     jPanel1.add(getJButtonNext());
1387                     jPanel1.add(getJButtonPre());
1388                     jPanel1.add(getJScrollPane2());
1389                     jComboBox1.setModel(jComboBox1Model);
1390                     jComboBox1.setBounds(12, 222, 105, 24);
1391                     jComboBox1.setAction(getAbstractAction1());
1392                 }
1393             }
1394             pack();
1395             setSize(500, 444);
1396         } catch (Exception e) {
1397             e.printStackTrace();
1398         }
1399     }
1400     private Component getJTable1() {
```

```

1401         if (operaInfo == null) {
1402             jTable1Model = new DefaultTableModel(new String[][] { { "1", "国际航班",
1403                 "经济舱", "学生", "43.3", "20030.0" } }, new String[] { "编号",
1404                 "航班类型", "座舱等级", "乘客", "行李重量", "经济舱票价" });
1405         } else {
1406             String[] columnNames = new String[paraInfoList.size() + 1];
1407             for (int i = 0; i <= paraInfoList.size(); ++i) {
1408                 if (i == 0) {
1409                     columnNames[i] = "id";
1410                 } else {
1411                     columnNames[i] = paraInfoList.get(i - 1).getName();
1412                 }
1413             }
1414             jTable1Model = new DefaultTableModel(new String[][] { }, columnNames);
1415         }
1416         jTable1 = new JTable(jTable1Model);
1417         jTable1.addKeyListener(new KeyAdapter() {
1418             public void keyReleased(KeyEvent e) {
1419                 int cidx = 5;
1420                 if (e.getKeyChar() == '\n') {
1421                     System.out.println(jTable1.getColumnCount());
1422                     System.out.println(jTable1.getSelectedColumn());
1423                     if (jTable1.getSelectedColumn() >= cidx
1424                         && jTable1.getSelectedRow() >= jTable1
1425                             .getRowCount() - 1
1426                         && jTable1.getValueAt(jTable1.getSelectedRow(),
1427                             cidx) != null
1428                         && jTable1
1429                             .getValueAt(jTable1.getSelectedRow(), cidx)
1430                             .toString().length() > 0) {
1431                         jTable1.editCellAt(jTable1.getSelectedRow(),
1432                             jTable1.getSelectedColumn());
1433                         ((DefaultTableModel) jTable1Model)
1434                             .addRow(new String[] { new Integer(jTable1
1435                                 .getRowCount() + 1).toString() });
1436                     }
1437                     jTable1.editCellAt(jTable1.getSelectedRow(),
1438                         jTable1.getSelectedColumn());
1439                 }
1440             }
1441         });
1442         return jTable1;
1443     }
1444     private AbstractAction getAbstractAction1() {
1445         if (abstractAction1 == null) {
1446             abstractAction1 = new AbstractAction("abstractAction1", null) {
1447                 public void actionPerformed(ActionEvent evt) {
1448                     if (jComboBox1.getSelectedItem().equals("人工设定")) {
1449                         if (addColumn) {
1450                             addColumn = false;

```

```

1451                ((DefaultTableModel) jTable1Model)
1452                    .addColumn("初始概率");
1453            }
1454        }
1455        if (jComboBox1.getSelectedItem().equals(" 平 均 分 布 ") ||
1456 jComboBox1.getSelectedItem().equals("比例分布")) {
1457            if (!addColumn) {
1458                addColumn = true;
1459                DefaultTableModel dtm = (DefaultTableModel) jTable1Model;
1460                int columnCount = dtm.getColumnCount() - 1;
1461                if (columnCount >= 0) {
1462                    TableColumnModel tcm = jTable1.getColumnModel();
1463                    TableColumn tc = tcm.getColumn(columnCount);
1464                    dtm.setColumnCount(columnCount);
1465                    tcm.removeColumn(tc);
1466                    dtm.setColumnCount(columnCount);
1467                }
1468            }
1469        }
1470        System.out.println(jTable1.getColumnCount());
1471    }
1472    };
1473    }
1474    return abstractAction1;
1475    }
1476    private JButton getJButtonAddPartition() {
1477        if (jButtonAddPartition == null) {
1478            jButtonAddPartition = new JButton();
1479            jButtonAddPartition.setText("\u6dfb\u52a0\u5206\u533a");
1480            jButtonAddPartition.setBounds(229, 220, 100, 30);
1481            jButtonAddPartition.setAction(getAbstractActionAddPartition());
1482        }
1483        return jButtonAddPartition;
1484    }
1485    private AbstractAction getAbstractActionAddPartition() {
1486        if (abstractActionAddPartition == null) {
1487            abstractActionAddPartition = new AbstractAction("添加分区", null) {
1488                public void actionPerformed(ActionEvent evt) {
1489                    ((DefaultTableModel) jTable1Model)
1490                        .addRow(new String[] { new Integer(jTable1
1491                            .getRowCount() + 1).toString() });
1492                }
1493            };
1494        }
1495        return abstractActionAddPartition;
1496    }
1497    private JButton getJButtonRemovePartition() {
1498        if (jButtonRemovePartition == null) {
1499            jButtonRemovePartition = new JButton();
1500            jButtonRemovePartition.setText("\u5220\u9664\u5206\u533a");

```