```
package ui;
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.util.Set;
import javax.swing.AbstractAction;
import javax.swing.ComboBoxModel;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JScrollBar;
import javax.swing.ListModel;
import javax.swing.WindowConstants;
import javax.swing.SwingUtilities;
public class OperationSelectFrame extends javax.swing.JFrame {
	private JPanel OperationSelectPanel;
	private AbstractAction abstractAction_operationSelect;
	private JButton Button_OperationSelectNext;
	private JComboBox OperationSelectComboBox;
	private JLabel OperationSelectLabel;
	public static void main(String[] args) {
		SwingUtilities.invokeLater(new Runnable() {
			public void run() {
				OperationSelectFrame inst = new OperationSelectFrame();
				inst.setLocationRelativeTo(null);
				inst.setVisible(true);
			}
		});
	}
	public OperationSelectFrame() {
		super();
		initGUI();
	}
	public OperationSelectFrame(Set<String> opList) {
		super();
		String[] optionList= new String[opList.size()];
		opList.toArray(optionList);
		initGUI(optionList);
	}
	private void initGUI() {
		initGUI(null);
	}
	private void initGUI(String[] optionList) {
		try {
			this.setTitle("操作选择");
			this.setResizable(false);
```

```
            GridLayout thisLayout = new GridLayout(1, 1);
            getContentPane().setLayout(thisLayout);
            setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
            {
                OperationSelectPanel = new JPanel();
                getContentPane().add(OperationSelectPanel);
                OperationSelectPanel.setBackground(new java.awt.Color(252,252,203));
                OperationSelectPanel.setPreferredSize(new java.awt.Dimension(403, 275));
                OperationSelectPanel.setLayout(null);
                {
                    OperationSelectLabel = new JLabel();
                    OperationSelectPanel.add(OperationSelectLabel);
    OperationSelectLabel.setText("\u8bf7\u9009\u62e9\u5f85\u6d4b\u7684\u64cd\u4f5c");
                    OperationSelectLabel.setBounds(18, 28, 122, 17);
                }
                {
                    ComboBoxModel OptionSelectComboBoxModel;
                    if (optionList == null) {
                        OptionSelectComboBoxModel =
                                new DefaultComboBoxModel(new String[] {"Item One",
"Item Two"});
                    } else {
                        OptionSelectComboBoxModel =
                                new DefaultComboBoxModel(optionList);
                    }
                    OperationSelectComboBox = new JComboBox();
                    OperationSelectPanel.add(OperationSelectComboBox);
                    OperationSelectComboBox.setModel(OptionSelectComboBoxModel);
                    OperationSelectComboBox.setBounds(20, 65, 199, 24);
                }
                {
                    Button_OperationSelectNext = new JButton();
                    OperationSelectPanel.add(Button_OperationSelectNext);
                    Button_OperationSelectNext.setText("\u4e0b\u4e00\u6b65");
                    Button_OperationSelectNext.setBounds(272, 198, 80, 24);
                Button_OperationSelectNext.setAction(getAbstractAction_operationSelect());
                }
            }
            pack();
            setSize(412, 311);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    private AbstractAction getAbstractAction_operationSelect() {
        if(abstractAction_operationSelect == null) {
            abstractAction_operationSelect = new AbstractAction("\u4e0b\u4e00\u6b65", null)
{
                public void actionPerformed(ActionEvent evt) {}
            };
        }
        return abstractAction_operationSelect;
    }
}
package ui;
import java.awt.BorderLayout;
import java.awt.Canvas;
import java.awt.Color;
import java.awt.Component;
import java.awt.FlowLayout;
import java.awt.Font;
import java.awt.GridBagConstraints;
import java.awt.GridBagLayout;
import java.awt.GridLayout;
import java.awt.Insets;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.util.Map;
import java.util.Queue;
import java.util.concurrent.TimeUnit;
import javax.swing.AbstractAction;
import javax.swing.BorderFactory;
import javax.swing.ButtonGroup;
import javax.swing.JButton;
import javax.swing.JDesktopPane;
import javax.swing.JInternalFrame;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JScrollPane;
import javax.swing.JSplitPane;
import javax.swing.JTextArea;
import javax.swing.JTextField;
import javax.swing.JTextPane;
import javax.swing.Timer;
import javax.swing.WindowConstants;
import javax.swing.SwingUtilities;
import javax.swing.border.EtchedBorder;
import javax.swing.border.TitledBorder;
import javax.swing.text.BadLocationException;
import javax.swing.text.Document;
import javax.swing.text.SimpleAttributeSet;
import javax.swing.text.StyleConstants;
import javax.swing.text.StyledDocument;
import cn.edu.ustb.webservicetester.model.OperationInfo;
import cn.edu.ustb.webservicetester.model.TestCase;
import cn.edu.ustb.webservicetester.model.TestSet;
import cn.edu.ustb.webservicetester.model.TestSetWithPartition;
import cn.edu.ustb.webservicetester.thread.TestCaseExecuter;
import cn.edu.ustb.webservicetester.thread.TestProcessWriter;
import cn.edu.ustb.webservicetester.util.ServiceInvoker;
public class ExecutionFrame extends javax.swing.JFrame {
    private Color bgColor = new Color(252, 252, 203);
    private JSplitPane jSplitPane1;
    private JSplitPane jSplitPane2;
    private JScrollPane jScrollPane1;
    private JScrollPane jScrollPane2;
    private JTextField jTextField1;
    private ButtonGroup buttonGroup1;
    private JButton jButton4;
    private JButton jButton3;
    private JButton jButton2;
    private AbstractAction abstractAction1;
    private JButton jButton1;
    private JTextPane jTextPane1;
    private StyledDocument doc;
    private JPanel jPanel1;
    private JPanel jPanel3;
    private JPanel jPanel2;
    private Timer timeAction;
    private JLabel jLabel11;
    private JTextField jTextField3;
    private JLabel jLabel10;
    private JLabel jLabel9;
    private JTextField jTextField2;
    private JRadioButton jRadioButton2;
    private JRadioButton jRadioButton1;
    private int testI = 0;
    private boolean isStop = false;
    private java.util.Map<String, String> testSetting;
    private OperationInfo operationInfo;
    private TestSet testSet = null;
    private TestSetWithPartition testSetWP = null;
    private java.util.Queue<String> pQueue = new
java.util.concurrent.ConcurrentLinkedQueue<String>();
    private boolean testFinished = false;
    private Thread executeThread = null;
    private Thread showProcessThread = null;
    private double epsilon = 0.0;
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                ExecutionFrame inst = new ExecutionFrame();
                inst.setLocationRelativeTo(null);
                inst.setVisible(true);
            }
        });
    }
    public ExecutionFrame() {
        super();
        initGUI();
    }
    public ExecutionFrame(java.util.Map<String, String> tSetting, OperationInfo oi, TestSet ts) {
        super();
        testSetting = tSetting;
        operationInfo = oi;
        testSet = ts;
        initGUI(tSetting, oi);
    }
    public ExecutionFrame(java.util.Map<String, String> tSetting, OperationInfo oi, TestSetWithPartition tswp) {
        super();
        testSetting = tSetting;
        operationInfo = oi;
        testSetWP = tswp;
        System.out.println(testSetWP.numOfPartitions());
        System.out.println(testSetWP.numOfTestCases());
        initGUI(tSetting, oi);
    }
    private void initGUI(Map<String, String> tSetting, OperationInfo oi) {
        try {
            setTitle("测试执行");
            getContentPane().add(getJSplitPane1());
            setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
            setResizable(false);
            pack();
            setSize(800, 600);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    private Component getJSplitPane1() {
        jSplitPane1 = new JSplitPane(JSplitPane.VERTICAL_SPLIT);
        jSplitPane1.setDividerLocation(440);
        jSplitPane1.setDividerSize(5);
        jSplitPane1.setEnabled(false);
        jSplitPane1.add(getJSplitPane2(), JSplitPane.LEFT);
        jSplitPane1.add(getJPanel3(), JSplitPane.RIGHT);
        return jSplitPane1;
    }
    private Component getJSplitPane2() {
        jSplitPane2 = new JSplitPane();
        jSplitPane2.setDividerLocation(200);
        jSplitPane2.setDividerSize(5);
        jSplitPane2.setEnabled(false);
        jSplitPane2.add(getJScrollPane1(), JSplitPane.LEFT);
        jSplitPane2.add(getJScrollPane2(), JSplitPane.RIGHT);
        return jSplitPane2;
    }
    private Component getJScrollPane2() {
        if (jScrollPane2 == null) {
            jScrollPane2 = new JScrollPane();
            jScrollPane2.setBackground(new java.awt.Color(228,236,243));
    jScrollPane2.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder
(EtchedBorder.LOWERED), "Testing Process", TitledBorder.TRAILING, TitledBorder.TOP));
            jScrollPane2.setViewportView(getJPane2());
        }
        return jScrollPane2;
    }
    private Component getJPane2() {
        jPanel2 = new JPanel();
        GridLayout jPanel2Layout = new GridLayout(1, 1);
        jPanel2Layout.setColumns(1);
        jPanel2Layout.setHgap(5);
        jPanel2Layout.setVgap(5);
        jPanel2.setBackground(bgColor);
        jPanel2.setLayout(jPanel2Layout);
        jPanel2.add(getJTextPane1());
        return jPanel2;
    }
    private Component getJTextPane1() {
        jTextPane1 = new JTextPane();
        jTextPane1.setBackground(new java.awt.Color(228,236,243));
        doc = jTextPane1.getStyledDocument();
        return jTextPane1;
    }
    private Component getJScrollPane1() {
        jScrollPane1 = new JScrollPane();
        jScrollPane1.setBackground(new java.awt.Color(228,236,243));
    jScrollPane1.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder
(EtchedBorder.LOWERED), "Test Setting", TitledBorder.LEADING, TitledBorder.TOP));
        jScrollPane1.setViewportView(getJPanel1());
        return jScrollPane1;
    }
    private Component getJPanel1() {
        jPanel1 = new JPanel();
        GridBagLayout jPanel1Layout = new GridBagLayout();
        jPanel1.setBackground(new java.awt.Color(228,236,243));
        jPanel1Layout.rowWeights = new double[testSetting.size() * 2 + 1];
        jPanel1Layout.rowHeights = new int[testSetting.size() * 2 + 1];
        for (int i = 0; i <= testSetting.size() * 2; ++i) {
            jPanel1Layout.rowWeights[i] = 0.1;
            if (i == testSetting.size() * 2) {
                jPanel1Layout.rowHeights[i] = 7;
            } else {
                if (i % 2 == 0) {
                    jPanel1Layout.rowHeights[i] = 25;
                } else {
                    jPanel1Layout.rowHeights[i] = 30;
                }
            }
        }
        jPanel1Layout.columnWeights = new double[] {0.1, 0.1, 0.1, 0.1, 0.1};
        jPanel1Layout.columnWidths = new int[] {7, 7, 7, 7, 7};
        jPanel1.setLayout(jPanel1Layout);
        JLabel jLabelTemp1, jLabelTemp2;
        java.util.Iterator<String> keyIter = testSetting.keySet().iterator();
        int settingIndex = 0;
        while (keyIter.hasNext()) {
            StringBuffer labelBuf = new StringBuffer(" · ");
            String key = keyIter.next();
            labelBuf.append(key);
            String value = testSetting.get(key);
            System.out.print(key);
            System.out.print(":");
            System.out.println(value);
            jLabelTemp1 = new JLabel(labelBuf.toString());
            jLabelTemp2 = new JLabel(value);
            jPanel1.add(jLabelTemp1, new GridBagConstraints(0, settingIndex * 2, 4, 1, 0.0,
0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
            jPanel1.add(jLabelTemp2, new GridBagConstraints(1, settingIndex * 2 + 1, 3, 1,
0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0,
0));
            ++settingIndex;
        }
        return jPanel1;
    }
    private Component getJPanel3() {
        jPanel3 = new JPanel();
        GridBagLayout jPanel3Layout = new GridBagLayout();
        jPanel3Layout.rowWeights = new double[] {0.1, 0.1, 0.1, 0.1};
        jPanel3Layout.rowHeights = new int[] {7, 7, 7, 7};
        jPanel3Layout.columnWeights = new double[] {0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.0, 0.1, 0.0,
0.1, 0.0, 0.1};
        jPanel3Layout.columnWidths = new int[] {7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7};
        jPanel3.setLayout(jPanel3Layout);
        jPanel3.setBackground(new java.awt.Color(228,236,243));
    jPanel3.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(Etch
edBorder.LOWERED), "Test Controller", TitledBorder.CENTER, TitledBorder.TOP));
        jPanel3.add(getJButton1(), new GridBagConstraints(7, 2, 1, 1, 0.0, 0.0,
GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJButton2(), new GridBagConstraints(9, 2, 1, 1, 0.0, 0.0,
GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJButton3(), new GridBagConstraints(11, 2, 1, 1, 0.0, 0.0,
GridBagConstraints.WEST, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJButton4(), new GridBagConstraints(5, 2, 1, 1, 0.0, 0.0,
GridBagConstraints.EAST, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJRadioButton1(), new GridBagConstraints(0, 0, 1, 1, 0.0, 0.0,
GridBagConstraints.EAST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJRadioButton2(), new GridBagConstraints(0, 1, 1, 1, 0.0, 0.0,
GridBagConstraints.EAST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJTextField1(), new GridBagConstraints(1, 0, 1, 1, 0.0, 0.0,
GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJTextField2(), new GridBagConstraints(1, 1, 1, 1, 0.0, 0.0,
GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJLabel9(), new GridBagConstraints(2, 0, 2, 1, 0.0, 0.0,
GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJLabel10(), new GridBagConstraints(2, 1, 2, 1, 0.0, 0.0,
GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJTextField3(), new GridBagConstraints(7, 0, 1, 1, 0.0, 0.0,
GridBagConstraints.WEST, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
        jPanel3.add(getJLabel11(), new GridBagConstraints(5, 0, 1, 1, 0.0, 0.0,
GridBagConstraints.EAST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
        getButtonGroup1();
        return jPanel3;
    }
    private Component getJButton1() {
        jButton1 = new JButton();
        jButton1.setText("Start");
        jButton1.setAction(getAbstractAction1());
        return jButton1;
    }
    private void initGUI() {
        try {
            setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
            {
                jSplitPane1 = new JSplitPane(JSplitPane.VERTICAL_SPLIT);
                getContentPane().add(jSplitPane1);
                setResizable(false);
                {
                    jSplitPane2 = new JSplitPane();
                    jSplitPane1.add(jSplitPane2, JSplitPane.LEFT);
                    {
                    jPanel2 = new JPanel();
                        GridLayout jPanel2Layout = new GridLayout(1, 1);
                        jPanel2Layout.setColumns(1);
                        jPanel2Layout.setHgap(5);
                        jPanel2Layout.setVgap(5);
                        jSplitPane2.add(jPanel2, JSplitPane.RIGHT);
                        jPanel2.setBackground(bgColor);
    jPanel2.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(Etch
edBorder.LOWERED), "Testing Process", TitledBorder.TRAILING, TitledBorder.TOP));
                        jPanel2.setLayout(jPanel2Layout);
                        {
                            jTextPane1 = new JTextPane();
                            jPanel2.add(jTextPane1);
                            jTextPane1.setBackground(bgColor);
                            jTextPane1.setText("jTextPane1\n");
                            doc = jTextPane1.getStyledDocument();
                            javax.swing.text.SimpleAttributeSetattr=new
                            javax.swing.text.SimpleAttributeSet();
                            StyleConstants.setBold(attr, true);
                            StyleConstants.setForeground(attr, Color.RED);
                            doc.insertString(doc.getLength(), "jTextPane2\n", attr);
                            StyleConstants.setBold(attr, false);
                            StyleConstants.setForeground(attr, Color.BLACK);
                            doc.insertString(doc.getLength(), "jTextPane3\n", attr);
                            addTestProcess();
                        }
                    }
                    {
                        jScrollPane1 = new JScrollPane();
    jScrollPane1.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder
(EtchedBorder.LOWERED), "Test Setting", TitledBorder.LEADING, TitledBorder.TOP));
                        jScrollPane1.setBackground(bgColor);
                        jSplitPane2.add(jScrollPane1, JSplitPane.LEFT);
                        {
                            jPanel1 = new JPanel();
                            GridBagLayout jPanel1Layout = new GridBagLayout();
                            jScrollPane1.setViewportView(jPanel1);
                            jPanel1.setBackground(bgColor);
    jPanel1.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(Etch
edBorder.LOWERED), "Test Setting", TitledBorder.LEADING, TitledBorder.TOP));
                            jPanel1Layout.rowWeights = new double[] {0.1, 0.1, 0.1, 0.1,
0.1, 0.1, 0.1, 0.1, 0.1};
                            jPanel1Layout.rowHeights = new int[] {25, 30, 25, 30, 25, 30,
25, 30, 7};
                            jPanel1Layout.columnWeights = new double[] {0.1, 0.1, 0.1,
0.1};
                            jPanel1Layout.columnWidths = new int[] {7, 7, 7, 7};
                            jPanel1.setLayout(jPanel1Layout);
                            {
                                JLabel jLabel1 = new JLabel();
                                jPanel1.add(jLabel1, new GridBagConstraints(0, 0, 2, 1,
0.0, 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
                                jLabel1.setText(" · Web Service:    ");
                            }
                            {
                                JLabel jLabel2 = new JLabel();
                                jPanel1.add(jLabel2, new GridBagConstraints(1, 1, 2, 1,
0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0,
0));
    jLabel2.setText("localhost:8080/axis2/services/AddService");
                            }
                            {
                                JLabel jLabel3 = new JLabel();
                                jPanel1.add(jLabel3, new GridBagConstraints(0, 2, 2, 1,
0.0, 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
```

```
                    jLabel3.setText(" · Operation: ");
                }
                {
                    JLabel jLabel4 = new JLabel();
                    jPanel1.add(jLabel4, new GridBagConstraints(1, 3, 1, 1, 0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
                    jLabel4.setText("add");
                }
                {
                    JLabel jLabel5 = new JLabel();
                    jPanel1.add(jLabel5, new GridBagConstraints(0, 4, 2, 1, 0.0, 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
                    jLabel5.setText(" · Teseting Strategy:");
                }
                {
                    JLabel jLabel6 = new JLabel();
                    jPanel1.add(jLabel6, new GridBagConstraints(1, 5, 1, 1, 0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
                    jLabel6.setText("Random Testing");
                }
                {
                    JLabel jLabel7 = new JLabel();
                    jPanel1.add(jLabel7, new GridBagConstraints(0, 6, 1, 1, 0.0, 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
                    jLabel7.setText(" · Test Suite:");
                }
                {
                    JLabel jLabel8 = new JLabel();
                    jPanel1.add(jLabel8, new GridBagConstraints(1, 7, 2, 1, 0.0, 0.0, GridBagConstraints.NORTHWEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
                    jLabel8.setText("testcaseset4addservice$add.xml");
                }
            }
        }
        jSplitPane2.setDividerLocation(200);
        jSplitPane2.setDividerSize(5);
        jSplitPane2.setEnabled(false);
    }
    {
        jPanel3 = new JPanel();
        GridBagLayout jPanel3Layout = new GridBagLayout();
        jSplitPane1.add(jPanel3, JSplitPane.RIGHT);
        jPanel3.setBackground(bgColor);
  jPanel3.setBorder(BorderFactory.createTitledBorder(BorderFactory.createEtchedBorder(EtchedBorder.LOWERED), "Test Controller", TitledBorder.CENTER, TitledBorder.TOP));
        jPanel3Layout.rowWeights = new double[] {0.1, 0.1, 0.1, 0.1};
        jPanel3Layout.rowHeights = new int[] {7, 7, 7, 7};
```

```
                    jPanel3Layout.columnWeights = new double[] {0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1};
                    jPanel3Layout.columnWidths = new int[] {7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7};
                    jPanel3.setLayout(jPanel3Layout);
                    {
                        jButton1 = new JButton();
                        jButton1.setText("Start");
                        jButton1.setAction(getAbstractAction1());
                        jPanel3.add(jButton1, new GridBagConstraints(7, 2, 1, 1, 0.0, 0.0, GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
                        jPanel3.add(getJButton2(), new GridBagConstraints(9, 2, 1, 1, 0.0, 0.0, GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
                        jPanel3.add(getJButton3(), new GridBagConstraints(11, 2, 1, 1, 0.0, 0.0, GridBagConstraints.WEST, GridBagConstraints.NONE, new Insets(0, 0, 0, 0), 0, 0));
                        jPanel3.add(getJButton4(), new GridBagConstraints(5, 2, 1, 1, 0.0, 0.0, GridBagConstraints.CENTER, GridBagConstraints.HORIZONTAL, new Insets(0, 0, 0, 0), 0, 0));
                    }
                }
                jSplitPane1.setDividerLocation(450);
                jSplitPane1.setDividerSize(5);
                jSplitPane1.setEnabled(false);
            }
            pack();
            setSize(800, 600);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    private void addTestProcess() {
        timeAction = new Timer(2000, new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                SimpleAttributeSet attr = new SimpleAttributeSet();
                if (testI++ < 10) {
                    if (testI % 3 == 0) {
                        StyleConstants.setBold(attr, true);
                        StyleConstants.setForeground(attr, Color.RED);
                    } else {
                        StyleConstants.setBold(attr, false);
                        StyleConstants.setForeground(attr, Color.BLACK);
                    }
                    try {
                        doc.insertString(doc.getLength(), "动态添加第"+testI+"行\n", attr);
                    } catch (Exception e1) {
                        e1.printStackTrace();
                    }
                }
            }
        });
        timeAction.start();
    }
    private AbstractAction getAbstractAction1() {
        if(abstractAction1 == null) {
            abstractAction1 = new AbstractAction("Start", null) {
                int tem = -1;
                int tec = -1;
                public void actionPerformed(ActionEvent evt) {
                    if (jRadioButton1.isSelected()) {
                        tem = new Integer(jTextField1.getText());
                        tec = TestCaseExecuter.NUM_OF_CASES;
                    } else if (jRadioButton2.isSelected()){
                        tem = new Integer(jTextField2.getText());
                        tec = TestCaseExecuter.NUM_OF_FAULTS;
                    }
                    if (testSet != null) {
                        executeThread = new Thread(new TestCaseExecuter(testSetting.get("Web Service"), operationInfo, testSet, pQueue, tec, tem));
                    } else if (testSetWP != null) {
                        if (testSetting.get("Epsilon") == null) {
                            executeThread = new Thread(new TestCaseExecuter(testSetting.get("Web Service"), operationInfo, testSetWP, pQueue, tec, tem));
                        } else {
                            double e = Double.parseDouble(testSetting.get("Epsilon"));
                            executeThread = new Thread(new TestCaseExecuter(testSetting.get("Web Service"), operationInfo, testSetWP, e, pQueue, tec, tem));
                        }
                    } else {}
                    executeThread.start();
                    showProcessThread = new Thread(new TestProcessWriter(doc, pQueue));
                    showProcessThread.start();
                }
            };
        }
        return abstractAction1;
    }
    private JButton getJButton2() {
        if(jButton2 == null) {
            jButton2 = new JButton();
            jButton2.setText("Stop");
        }
        return jButton2;
    }
    private JButton getJButton3() {
        if(jButton3 == null) {
            jButton3 = new JButton();
            jButton3.setText("Download report");
        }
        return jButton3;
    }
    private JButton getJButton4() {
        if(jButton4 == null) {
            jButton4 = new JButton();
            jButton4.setText("<< Back");
        }
        return jButton4;
    }
    private ButtonGroup getButtonGroup1() {
        if(buttonGroup1 == null) {
            buttonGroup1 = new ButtonGroup();
            buttonGroup1.add(jRadioButton1);
            buttonGroup1.add(jRadioButton2);
        }
        return buttonGroup1;
    }
    private JRadioButton getJRadioButton1() {
        if(jRadioButton1 == null) {
            jRadioButton1 = new JRadioButton();
            jRadioButton1.setBackground(new java.awt.Color(228,236,243));
            jRadioButton1.setText("执行");
            jRadioButton1.setSelected(true);
        }
        return jRadioButton1;
    }
    private JRadioButton getJRadioButton2() {
        if(jRadioButton2 == null) {
            jRadioButton2 = new JRadioButton();
            jRadioButton2.setBackground(new java.awt.Color(228,236,243));
            jRadioButton2.setText("杀死");
        }
        return jRadioButton2;
    }
    private JTextField getJTextField1() {
        if(jTextField1 == null) {
            jTextField1 = new JTextField();
            jTextField1.setText("");
        }
        return jTextField1;
    }
    private JTextField getJTextField2() {
        if(jTextField2 == null) {
            jTextField2 = new JTextField();
            jTextField2.setText("");
        }
        return jTextField2;
    }
    private JLabel getJLabel9() {
            if(jLabel9 == null) {
                jLabel9 = new JLabel();
                jLabel9.setText("个测试用例后停止");
            }
            return jLabel9;
        }
        private JLabel getJLabel10() {
            if(jLabel10 == null) {
                jLabel10 = new JLabel();
                jLabel10.setText("个错误后停止");
            }
            return jLabel10;
        }
        private JTextField getJTextField3() {
            if(jTextField3 == null) {
                jTextField3 = new JTextField();
                jTextField3.setText("");
                jTextField3.addKeyListener(new KeyAdapter() {
                    public void keyPressed(KeyEvent evt) {
                        jTextField3KeyPressed(evt);
                    }
                });
            }
            return jTextField3;
        }
        private JLabel getJLabel11() {
            if(jLabel11 == null) {
                jLabel11 = new JLabel();
                jLabel11.setText("概率调整参数：");
            }
            return jLabel11;
        }
        private void jTextField3KeyPressed(KeyEvent evt) {
            System.out.println("jTextField3.keyPressed, event="+evt);
            if (evt.getKeyChar() == '\n') {
                System.out.println(jTextField3.getText());
                double e = Double.parseDouble(jTextField3.getText());
                testSetting.put("Epsilon", jTextField3.getText());
            }
        }
}
package ui;
import java.awt.BorderLayout;
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.util.Map;
import java.util.Set;
import javax.swing.*;
import cn.edu.ustb.mt4ws.tcg.WsdlOperationFormat;
import cn.edu.ustb.mt4ws.wsdl.parser.WsdlReader11;
import cn.edu.ustb.webservicetester.controllor.ParseWsdlControllor;
import cn.edu.ustb.webservicetester.util.WsdlParser;
public class MainFrame extends javax.swing.JFrame {
        private JPanel jPanel1;
        private JTextField jTextField_wsdl;
        private AbstractAction ResetWSDLAction;
        private AbstractAction ParseWSDLAction;
        private JButton jButton_wsdlreset;
        private JButton jButton_wsdlJIEXI;
        private JLabel jLabel_wsdl;
        public static void main(String[] args) {
            SwingUtilities.invokeLater(new Runnable() {
                public void run() {
                    MainFrame inst = new MainFrame();
                    inst.setLocationRelativeTo(null);
                    inst.setVisible(true);
                }
            });
        }
        public MainFrame() {
            super();
            initGUI();
        }
        private void initGUI() {
            try {
                this.setTitle("服务定位");
                this.setResizable(false);//关闭最大化
                GridLayout thisLayout = new GridLayout(1, 1);
                getContentPane().setLayout(thisLayout);
                setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
                {
                    jPanel1 = new JPanel();
                    getContentPane().add(jPanel1);
                    jPanel1.setBackground(new java.awt.Color(228,236,243));
                    jPanel1.setPreferredSize(new java.awt.Dimension(403, 275));
                    jPanel1.setLayout(null);
                    {
                        jTextField_wsdl = new JTextField();
                        jPanel1.add(jTextField_wsdl);
                        jTextField_wsdl.setBounds(20, 65, 347, 24);
        jTextField_wsdl.setText("http://localhost:8080/axis2/services/AddService?wsdl");
                    }
                    {
                        jLabel_wsdl = new JLabel();
                        jPanel1.add(jLabel_wsdl);
        jLabel_wsdl.setText("\u8bf7\u8f93\u5165\u5f85\u6d4b\u670d\u52a1\u5730\u5740");
                        jLabel_wsdl.setBounds(18, 28, 122, 17);
                        jButton_wsdlreset = new JButton();
                        jPanel1.add(jButton_wsdlreset);
                        jButton_wdlreset.setText("\u91cd    \u7f6e");
                        jButton_wsdlreset.setBounds(244, 157, 80, 30);
                        jButton_wsdlreset.setAction(getResetWSDLAction());
                    }
                    {
                        jButton_wsdlJIEXI = new JButton();
                        jPanel1.add(jButton_wsdlJIEXI);
                        jButton_wsdlJIEXI.setText("\u89e3    \u6790");
                        jButton_wsdlJIEXI.setBounds(60, 157, 80, 30);
                        jButton_wsdlJIEXI.setAction(getParseWSDLAction());
                    }
                    {
                        jButton_wsdlreset = new JButton();
                        jPanel1.add(jButton_wsdlreset);
                        jButton_wsdlreset.setText("\u91cd    \u7f6e");
                        jButton_wsdlreset.setBounds(244, 157, 80, 30);
                        jButton_wsdlreset.setAction(getResetWSDLAction());
                    }
                }
                pack();
                this.setSize(412, 311);
            } catch (Exception e) {
                e.printStackTrace();
            }
        }
        private AbstractAction getParseWSDLAction() {
            if(ParseWSDLAction == null) {
                ParseWSDLAction = new AbstractAction("\u89e3    \u6790", null) {
                    public void actionPerformed(ActionEvent evt) {
                        String wsdlurl = jTextField_wsdl.getText();
                        ParseWsdlControllor parseControllor = new ParseWsdlControllor();
                        parseControllor.parseWsdlControl(wsdlurl);JFrame);
    OperationSelectFrame(opList);
                    }
                };
            }
            return ParseWSDLAction;
        }
        private AbstractAction getResetWSDLAction() {
            if(ResetWSDLAction == null) {
                ResetWSDLAction = new AbstractAction("\u91cd    \u7f6e", null) {
                    public void actionPerformed(ActionEvent evt) {
                        jTextField_wsdl.setText("");
                    }
                };
            }
            return ResetWSDLAction;
        }
    }
    package ui;
```

```
import java.awt.FlowLayout;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import javax.swing.AbstractAction;
import javax.swing.ComboBoxModel;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JComboBox;
import javax.swing.JLabel;
import javax.swing.JList;
import javax.swing.JPanel;
import javax.swing.JScrollBar;
import javax.swing.JScrollPane;
import javax.swing.JSlider;
import javax.swing.JSpinner;
import javax.swing.JTextArea;
import javax.swing.ListModel;
import javax.swing.ListSelectionModel;
import javax.swing.SpinnerListModel;
import javax.swing.WindowConstants;
import javax.swing.SwingUtilities;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;
import cn.edu.ustb.webservicetester.model.OperationInfo;
import cn.edu.ustb.webservicetester.model.ParameterInfo;
public class OperationSelectionFrame2 extends javax.swing.JFrame {
    private JPanel jPanel1;
    private JScrollPane jScrollPane1;
    private JList parameterInfoJList;
    private JScrollPane jScrollPane2;
    private JTextArea jTextAreaParameterRestriction;
    private JLabel jLabelParameterRestriction;
    private JLabel jLabelParameterInformation;
    private AbstractAction operationSelectionAction;
    private JComboBox operationSelectionComboBox;
    private JLabel operationSelectionLabel;
    private java.util.List<OperationInfo> opInfoList;
    private java.util.List<ParameterInfo> pmInfoList;
    private ListSelectionListener lslistener;
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                OperationSelectionFrame2 inst = new OperationSelectionFrame2();
                inst.setLocationRelativeTo(null);
                inst.setVisible(true);
            }
        });
    }
    public OperationSelectionFrame2() {
        super();
        initGUI();
    }
    public OperationSelectionFrame2(String frame) {
        super();
        initGUI();
    }
    public OperationSelectionFrame2(java.util.List list) {
        super();
        lslistener = new ListSelectionListener() {
            public void valueChanged(ListSelectionEvent e) {
                    parameterSelectionChanged();
                }
            };
        initGUI(list);
    }
    private void initGUI() {
        initGUI(null);
    }
    private void initGUI(java.util.List<OperationInfo> oil) {
        this.opInfoList = oil;
        try {
            setTitle("操作选择");
            setResizable(false);
            GridLayout thisLayout = new GridLayout(1, 1);
            getContentPane().setLayout(thisLayout);
            setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
            {
                jPanel1 = new JPanel();
                getContentPane().add(jPanel1);
                jPanel1.setLayout(null);
                jPanel1.setBackground(new java.awt.Color(228,236,243));
                jPanel1.setPreferredSize(new java.awt.Dimension(506, 349));
                {
                    operationSelectionLabel = new JLabel();
                    jPanel1.add(operationSelectionLabel);
    operationSelectionLabel.setText("\u8bf7\u9009\u62e9\u5f85\u6d4b\u8bd5\u7684\u64cd\u4f5c");
                    operationSelectionLabel.setBounds(12, 36, 128, 17);
                }
                {
                    ComboBoxModel operationSelectionComboBoxModel;
                    if (opInfoList == null)
                        operationSelectionComboBoxModel =
                            new DefaultComboBoxModel(
                                    new String[] { "Item One", "Item Two" });
                    else {
                        String[] opNames = new String[opInfoList.size()];
                        for (int i = 0; i < opNames.length; ++i) {
                            opNames[i] = opInfoList.get(i).getOpName();
                        }
                        operationSelectionComboBoxModel = new
```

```
DefaultComboBoxModel(opNames);
                    }
                    operationSelectionComboBox = new JComboBox();
                    jPanel1.add(operationSelectionComboBox);
        operationSelectionComboBox.setModel(operationSelectionComboBoxModel);
                    operationSelectionComboBox.setBounds(140, 32, 162, 24);
                    operationSelectionComboBox.setAction(getOperationSelectionAction());
                }
                {
                    jScrollPane1 = new JScrollPane();
                    jPanel1.add(jScrollPane1);
                    jScrollPane1.setBounds(24, 101, 188, 132);
                    jPanel1 = new JPanel();
                    getContentPane().add(jPanel1);
                    jPanel1.setLayout(null);
                    jPanel1.setBackground(new java.awt.Color(228,236,243));
                    jPanel1.setPreferredSize(new java.awt.Dimension(506, 349));

                    {
                        ListModel parameterInfoListModel =
                                new DefaultComboBoxModel(
                                        new String[] {});
                        parameterInfoJList = new JList();
    parameterInfoJList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
                        jScrollPane1.setViewportView(parameterInfoJList);
                        parameterInfoJList.setModel(parameterInfoListModel);
                        parameterInfoJList.setPreferredSize(new    java.awt.Dimension(185,
129));
                        parameterInfoJList.addListSelectionListener(lslistener);
                    }
                }
                {
                    jScrollPane2 = new JScrollPane();
                    jPanel1.add(jScrollPane2);
                    jPanel1.add(getJLabelParameterInformation());
                    jPanel1.add(getJLabelParameterRestriction());
                    jScrollPane2.setBounds(274, 101, 183, 132);
                    jScrollPane2.setViewportView(getJTextAreaParameterRestriction());
                }
            }
            pack();
            this.setSize(502, 377);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    private AbstractAction getOperationSelectionAction() {
        if(operationSelectionAction == null) {
            operationSelectionAction = new AbstractAction("", null) {
                public void actionPerformed(ActionEvent evt) {
                    int opIndexSelected = operationSelectionComboBox.getSelectedIndex();
                    OperationInfo opInfo = opInfoList.get(opIndexSelected);
                    pmInfoList = opInfo.getParaList();
                    DefaultComboBoxModel dcbm =
(DefaultComboBoxModel)(parameterInfoJList.getModel());
                    dcbm.removeAllElements();
                    for (ParameterInfo pi : pmInfoList) {
                        dcbm.addElement(pi.toString());
                    }
                }
            };
        }
        return operationSelectionAction;
    }
    private JLabel getJLabelParameterInformation() {
        if(jLabelParameterInformation == null) {
            jLabelParameterInformation = new JLabel();
            jLabelParameterInformation.setText("\u8f93\u5165\u53c2\u6570");
            jLabelParameterInformation.setBounds(24, 73, 61, 17);
        }
        return jLabelParameterInformation;
    }
    private JLabel getJLabelParameterRestriction() {
        if(jLabelParameterRestriction == null) {
            jLabelParameterRestriction = new JLabel();
            jLabelParameterRestriction.setText("\u53c2\u6570\u8bf4\u660e");
            jLabelParameterRestriction.setBounds(274, 73, 64, 17);
        }
        return jLabelParameterRestriction;
    }
    private JTextArea getJTextAreaParameterRestriction() {
        if(jTextAreaParameterRestriction == null) {
            jLabelParameterRestriction = new JLabel();
            jLabelParameterRestriction.setText("\u53c2\u6570\u8bf4\u660e");
            jLabelParameterRestriction.setBounds(274, 73, 64, 17);
            jTextAreaParameterRestriction = new JTextArea();
            jTextAreaParameterRestriction.setEditable(false);
        }
        return jTextAreaParameterRestriction;
    }
    private void parameterSelectionChanged() {
        int parameterIndexSelected = parameterInfoJList.getSelectedIndex();
        if (parameterIndexSelected >= 0) {
            String rstrction = pmInfoList.get(parameterIndexSelected).getRestriction();
            jTextAreaParameterRestriction.setText(rstrction);
        }
    }
}
package ui;
import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ItemEvent;
import java.awt.event.ItemListener;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import javax.swing.AbstractAction;
import javax.swing.ButtonGroup;
import javax.swing.ComboBoxModel;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JLabel;
import javax.swing.JPanel;
import javax.swing.JRadioButton;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.WindowConstants;
import javax.swing.SwingUtilities;
import javax.swing.table.DefaultTableModel;
import cn.edu.ustb.webservicetester.model.ParameterIntervalScenario;
import cn.edu.ustb.webservicetester.model.ParameterInfo;
import cn.edu.ustb.webservicetester.model.ParameterPartitionScenario;
public class PartitionSettingFrame extends javax.swing.JFrame implements ItemListener{
    private java.util.List<ParameterInfo> paraInfos;
    private java.util.List<JComboBox> paraTypeList;
    private java.util.List<JCheckBox> paraList;
    private JLabel partitionBasisJLabel;
    private JLabel partitionSettingJLabel;
    private JPanel jPanel1;
    private JScrollPane jScrollPane1;
    private java.util.Map<JCheckBox, JComboBox> paraTypePair;
    private JTable paraPartitionTable;
    private DefaultTableModel paraPartitionTableModel;
    private java.util.Map<String, DefaultTableModel> paraPartitionScenarioTableModels;
    private JButton preStepButton;
    private JButton nextStepButton;
    public static final int DISTANCE1 = 29;
    public static final int DISTANCE2 = 30;
    public static final int CHECK_BOX_X = 26;
    public static final int CHECK_BOX_Y = 51;
    public static final int CHECK_BOX_WIDTH = 105;
    public static final int CHECK_BOX_HEIGHT = 21;
    private AbstractAction nextStepAction;
    public static final int COMBO_BOX_X = 155;
    public static final int COMBO_BOX_Y = 49;
    public static final int COMBO_BOX_WIDTH = 118;
    public static final int COMBO_BOX_HEIGHT = 24;
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                PartitionSettingFrame inst = new PartitionSettingFrame();
                inst.setLocationRelativeTo(null);
                inst.setVisible(true);
            }
        });
    }
    public PartitionSettingFrame() {
        super();
        initGUI();
    }
    public PartitionSettingFrame(java.util.List<ParameterInfo> paraInfoList) {
        super();
        initGUI(paraInfoList);
    }
    private void initGUI() {
        initGUI(null);
    }
    private void initGUI(java.util.List<ParameterInfo> paraInfoList) {
        if (paraInfoList == null) {
            paraInfos = new java.util.LinkedList<ParameterInfo>();
            for (int i = 0; i < 4; ++i) {
                ParameterInfo tpi = new ParameterInfo();
                tpi.setName("parameter " + (i + 1));
                tpi.setRestriction("");
                tpi.setClassInfo("Object");
                paraInfos.add(tpi);
            }
        }
        else {
            paraInfos = paraInfoList;
        }
        try {
            ParameterInfo tpi = new ParameterInfo();
            tpi.setName("parameter " + (i + 1));
            tpi.setRestriction("");
            tpi.setClassInfo("Object");
            paraInfos.add(tpi);
            GridLayout thisLayout = new GridLayout(1, 1);
            thisLayout.setHgap(5);
            thisLayout.setVgap(5);
            thisLayout.setColumns(1);
            getContentPane().setLayout(thisLayout);
            setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
            this.setTitle("\u5206\u533a\u8bbe\u5b9a");
            setResizable(false);
            {
                jPanel1 = new JPanel();
                getContentPane().add(jPanel1);
                jPanel1.setLayout(null);
                jPanel1.setBackground(new java.awt.Color(252,252,203));
                {
                    partitionBasisJLabel = new JLabel();
                    jPanel1.add(partitionBasisJLabel);
                    partitionBasisJLabel.setText("1.
        \u9009\u62e9\u5206\u533a\u4f9d\u636e\u53c2\u6570");
                    partitionBasisJLabel.setBounds(26, 22, 129, 17);
                }
                {
                    paraList = new java.util.ArrayList<JCheckBox>(paraInfos.size());
                    paraTypeList = new java.util.ArrayList<JComboBox>(paraInfos.size());
                    paraPartitionScenarioTableModels = new java.util.HashMap<String,
        DefaultTableModel>(paraInfos.size());
                    paraTypePair = new java.util.HashMap<JCheckBox, JComboBox>();
                    for (int i = 0; i < paraInfos.size(); ++i) {
                        JCheckBox tempCheckBox = new JCheckBox();
                        jPanel1.add(tempCheckBox);
                        tempCheckBox.setText(paraInfos.get(i).getName());
                        tempCheckBox.setBounds(CHECK_BOX_X, CHECK_BOX_Y +
        DISTANCE2 * i, CHECK_BOX_WIDTH, CHECK_BOX_HEIGHT);
                        tempCheckBox.addItemListener(this);
                        tempCheckBox.setBackground(new java.awt.Color(252, 252, 203));
                        paraList.add(tempCheckBox);
                        JComboBox tempComboBox = new JComboBox();
                        jPanel1.add(tempComboBox);
                        tempComboBox.setModel(getComboBoxModel());
                        tempComboBox.setBounds(COMBO_BOX_X, COMBO_BOX_Y
        + DISTANCE2 * i, COMBO_BOX_WIDTH, COMBO_BOX_HEIGHT);
                        tempComboBox.setVisible(false);
                        paraTypeList.add(tempComboBox);
                        paraTypePair.put(tempCheckBox, tempComboBox);
                        DefaultTableModel tdtm = new DefaultTableModel(null, new
        String[]{"", "参数 " + tempCheckBox.getText() + " 的分区区间", "权重"});
                        paraPartitionScenarioTableModels.put(tempCheckBox.getText(),
        tdtm);
                    }
                }
                {
                    partitionSettingJLabel = new JLabel();
                    jPanel1.add(partitionSettingJLabel);
                    partitionSettingJLabel.setText("2. 设定各参数分区");
                    partitionSettingJLabel.setBounds(26, 22 + DISTANCE1 * 2 +
        DISTANCE2 * (paraInfos.size() - 1), 129, 17);
                }
                {
                    jScrollPane1 = new JScrollPane();
                    jPanel1.add(jScrollPane1);
                    jScrollPane1.setBounds(26, 22 + DISTANCE1 * 3 + DISTANCE2 *
(paraInfos.size() - 1), 500 - 26 * 2, 444 - (22 + DISTANCE1 * 3 + DISTANCE2 * (paraInfos.size()
- 1)) - 30 - DISTANCE1 * 3);
                    jScrollPane1.setViewportView(getParaPartitionTable());
                }
                {
                    preStepButton = new JButton();
                    jPanel1.add(preStepButton);
                    preStepButton.setText("上一步");
                    preStepButton.setBounds(60, 444 - 30 - DISTANCE1 * 2, 80, 30);
                }
                {
                    tempCheckBox.setBackground(new java.awt.Color(252, 252, 203));
                    paraList.add(tempCheckBox);
                    JComboBox tempComboBox = new JComboBox();
                    Panel1.add(tempComboBox);
                    tempComboBox.setModel(getComboBoxModel());
                    nextStepButton = new JButton();
                    jPanel1.add(nextStepButton);
                    nextStepButton.setText("下一步");
                    nextStepButton.setBounds(500 - 60 - 80, 444 - 30 -DISTANCE1 * 2, 80,
30);
                    nextStepButton.setAction(getNextStepAction());
                }
            }
            pack();
            setSize(500, 444);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    private Component getParaPartitionTable() {
        if (paraPartitionTable == null) {
            String[] name = {"", "参数　的分区区间", "权重"};
            paraPartitionTableModel = new DefaultTableModel(null, name);
            paraPartitionTableModel.addRow(new String[]{"1", "", ""});
            paraPartitionTable = new JTable(paraPartitionTableModel) {
                public boolean isCellEditable(int row, int column) {
                    if (column >= 1)
                        return true;
                    else return false;
                }
            };
            paraPartitionTable.addKeyListener(new KeyAdapter(){
                public void keyReleased(KeyEvent e) {
                    if (e.getKeyChar() == '\n') {
                        if (paraPartitionTable.getSelectedColumn() >=
paraPartitionTable.getColumnCount() - 1
                                && paraPartitionTable.getSelectedRow() >=
paraPartitionTable.getRowCount() - 1
                                &&
paraPartitionTable.getValueAt(paraPartitionTable.getSelectedRow(),
paraPartitionTable.getSelectedColumn()).toString().length() > 0) {
    System.out.println(paraPartitionTable.getValueAt(paraPartitionTable.getSelectedRow(),
paraPartitionTable.getSelectedColumn()).toString().length());
                        paraPartitionTable.editCellAt(paraPartitionTable.getSelectedRow(),
paraPartitionTable.getSelectedColumn());
                        ((DefaultTableModel)paraPartitionTable.getModel()).addRow(new
String[]{new Integer(paraPartitionTable.getRowCount() + 1).toString(), "", ""});
                    paraPartitionTable.editCellAt(paraPartitionTable.getRowCount() - 1, 0);
                        }
                    }
                }
            });
        }
        return paraPartitionTable;
    }
    public void itemStateChanged(ItemEvent e) {
        Object source = e.getItemSelectable();
        JCheckBox changedCheckBox = (JCheckBox)source;
        if (changedCheckBox.isSelected()) {
            paraTypePair.get(changedCheckBox).setVisible(true);
= paraPartitionSenarioTableModels.get(changedCheckBox.getText());
    paraPartitionTable.setModel(paraPartitionScenarioTableModels.get(changedCheckBox.getTe
xt()));
            if
(paraPartitionScenarioTableModels.get(changedCheckBox.getText()).getRowCount() <= 0)
        paraPartitionScenarioTableModels.get(changedCheckBox.getText()).addRow(new
String[]{"1", "", ""});
        } else {
            paraTypePair.get(changedCheckBox).setVisible(false);
        }
    }
    private ComboBoxModel getComboBoxModel() {
        ComboBoxModel jComboBoxModel =
                new DefaultComboBoxModel(
                        new String[] { "连续型", "枚举型" });
        return jComboBoxModel;
    }
    private AbstractAction getNextStepAction() {
        if(nextStepAction == null) {
            nextStepAction = new AbstractAction("\u4e0b\u4e00\u6b65", null) {
                public void actionPerformed(ActionEvent evt) {
                    java.util.List<ParameterPartitionScenario> partitionSetting = new
java.util.ArrayList<ParameterPartitionScenario>();
                    ParameterPartitionScenario tempScenario;
                    for (JCheckBox jcb : paraList) {
                        if (jcb.isSelected()) {
                            tempScenario = new ParameterPartitionScenario();
                            for (ParameterInfo pi : paraInfos) {
                                if (pi.getName().equals(jcb.getText())) {
                                    tempScenario.setParaInfo(pi);
                                }
                            }
                            if ("连续的".equals(paraTypePair.get(jcb).getSelectedItem())) {
                                for (ParameterInfo pi :paraInfos) {
                                    if (pi.getName().equals(jcb.getText())) {
                                        pi.setType(ParameterInfo.CONTINUOUS);
                                    }
                                }
                            } else {
                                for (ParameterInfo pi :paraInfos) {
                                    if (pi.getName().equals(jcb.getText())) {
                                        pi.setType(ParameterInfo.SCATTERED);
                                    }
                                }
                            }
                            DefaultTableModel dtm =
paraPartitionScenarioTableModels.get(jcb.getText());
                            for (int i = 0; i < dtm.getRowCount(); ++i) {
                                ParameterIntervalScenario paraInter = new
ParameterIntervalScenario();
                                paraInter.setRange(dtm.getValueAt(i, 1).toString());
                                paraInter.setWeight(Integer.valueOf(dtm.getValueAt(i,
2).toString()));
                                tempScenario.add(paraInter);
                            }
                            partitionSetting.add(tempScenario);
                        } else {
                            ;
                        }
                    }
                }
            };
        }
        return nextStepAction;
    }
}
package ui;
import java.awt.BorderLayout;
import java.awt.Component;
import java.awt.GridLayout;
import java.awt.event.ActionEvent;
import java.awt.event.KeyAdapter;
import java.awt.event.KeyEvent;
import java.util.Map;
import javax.swing.AbstractAction;
import javax.swing.ComboBoxModel;
import javax.swing.DefaultCellEditor;
import javax.swing.DefaultComboBoxModel;
import javax.swing.JButton;
import javax.swing.JComboBox;
import javax.swing.JPanel;
import javax.swing.JScrollPane;
import javax.swing.JTable;
import javax.swing.WindowConstants;
import javax.swing.table.DefaultTableModel;
import javax.swing.table.TableColumn;
import javax.swing.table.TableColumnModel;
import javax.swing.table.TableModel;
import javax.swing.SwingUtilities;
import cn.edu.ustb.webservicetester.model.OperationInfo;
import cn.edu.ustb.webservicetester.model.ParameterInfo;
import cn.edu.ustb.webservicetester.model.TestSet;
import cn.edu.ustb.webservicetester.model.TestSetWithPartition;
import cn.edu.ustb.webservicetester.util.Partitioner;
public class PartitionSettingFrame2 extends javax.swing.JFrame {
    private JPanel jPanel1;
    private JScrollPane jScrollPane1;
    private AbstractAction abstractAction1;
    private JComboBox jComboBox1;
    private JTable jTable1;
    private TableModel jTable1Model;
    private AbstractAction abstractAction2;
    private JScrollPane jScrollPane2;
    private JTable jTable2;
    private TableModel jTable2Model;
    private JButton jButtonPre;
    private JButton jButtonNext;
    private AbstractAction removePartitionAction;
    private JButton jButtonRemovePartition;
    private AbstractAction abstractActionAddPartition;
    private JButton jButtonAddPartition;
    private java.util.List<ParameterInfo> paraInfoList;
    private OperationInfo operaInfo;
    private Map<String, String> testSetting;
    private TestSet testSet;
    private boolean addColumn = true;
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                PartitionSettingFrame2 inst = new PartitionSettingFrame2();
                inst.setLocationRelativeTo(null);
                inst.setVisible(true);
            }
        });
    }
    public PartitionSettingFrame2() {
        super();
        initGUI();
    }
    public PartitionSettingFrame2(Map<String, String> testSettingMap, OperationInfo oi,
TestSet ts) {
        super();
        operaInfo = oi;
        paraInfoList = operaInfo.getParaList();
        testSetting = testSettingMap;
        testSet = ts;
        initGUI();
    }
    private void initGUI() {
        try {
            setTitle("分区设定");
            setDefaultCloseOperation(WindowConstants.DISPOSE_ON_CLOSE);
            this.setResizable(false);
            GridLayout thisLayout = new GridLayout(1, 1);
            getContentPane().setLayout(thisLayout);
            getContentPane().setBackground(new java.awt.Color(228,236,243));
            {
                jPanel1 = new JPanel();
                getContentPane().add(jPanel1);
                jPanel1.setLayout(null);
                jPanel1.setBackground(new java.awt.Color(228,236,243));
                {
                    jScrollPane1 = new JScrollPane();
                    jPanel1.add(jScrollPane1);
                    jScrollPane1.setBounds(12, 10, 470, 200);
                    jScrollPane1.setViewportView(getJTable1());
                }
                {
                    ComboBoxModel jComboBox1Model = new DefaultComboBoxModel(
                            new String[] { "平均分布", "比例分布", "人工设定" });
                    jComboBox1 = new JComboBox();
                    jPanel1.add(jComboBox1);
                    jPanel1.add(getJButtonAddPartition());
                    jPanel1.add(getJButtonRemovePartition());
                    jPanel1.add(getJButtonNext());
                    jPanel1.add(getJButtonPre());
                    jPanel1.add(getJScrollPane2());
                    jComboBox1.setModel(jComboBox1Model);
                    jComboBox1.setBounds(12, 222, 105, 24);
                    jComboBox1.setAction(getAbstractAction1());
                }
            }
            pack();
            setSize(500, 444);
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
    private Component getJTable1() {
        if (operaInfo == null) {
            jTable1Model = new DefaultTableModel(new String[][] { { "1", "国际航班",
                    "经济舱", "学生", "43.3", "20030.0" } }, new String[] { "编号",
                    "航班类型", "座舱等级", "乘客", "行李重量", "经济舱票价" });
        } else {
            String[] columnNames = new String[paraInfoList.size() + 1];
            for (int i = 0; i <= paraInfoList.size(); ++i) {
                if (i == 0) {
                    columnNames[i] = "id";
                } else {
                    columnNames[i] = paraInfoList.get(i - 1).getName();
                }
            }
            jTable1Model = new DefaultTableModel(new String[][] {}, columnNames);
        }
        jTable1 = new JTable(jTable1Model);
        jTable1.addKeyListener(new KeyAdapter() {
            public void keyReleased(KeyEvent e) {
                int cidx = 5;
                if (e.getKeyChar() == '\n') {
                    System.out.println(jTable1.getColumnCount());
                    System.out.println(jTable1.getSelectedColumn());
                    if (jTable1.getSelectedColumn() >= cidx
                            && jTable1.getSelectedRow() >= jTable1
                                    .getRowCount() - 1
                            && jTable1.getValueAt(jTable1.getSelectedRow(),
                                    cidx) != null
                            && jTable1
                                    .getValueAt(jTable1.getSelectedRow(), cidx)
                                    .toString().length() > 0) {
                        jTable1.editCellAt(jTable1.getSelectedRow(),
                                jTable1.getSelectedColumn());
                        ((DefaultTableModel) jTable1Model)
                                .addRow(new String[] { new Integer(jTable1
                                        .getRowCount() + 1).toString() });
                    }
                    jTable1.editCellAt(jTable1.getSelectedRow(),
                            jTable1.getSelectedColumn());
                }
            }
        });
        return jTable1;
    }
    private AbstractAction getAbstractAction1() {
        if (abstractAction1 == null) {
            abstractAction1 = new AbstractAction("abstractAction1", null) {
                public void actionPerformed(ActionEvent evt) {
                    if (jComboBox1.getSelectedItem().equals("人工设定")) {
                        if (addColumn) {
                            addColumn = false;
                            ((DefaultTableModel) jTable1Model)
                                    .addColumn("初始概率");
                        }
                    }
                    if (jComboBox1.getSelectedItem().equals("平均分布") ||
jComboBox1.getSelectedItem().equals("比例分布")) {
                        if (!addColumn) {
                            addColumn = true;
                            DefaultTableModel dtm = (DefaultTableModel) jTable1Model;
                            int columnCount = dtm.getColumnCount() - 1;
                            if (columnCount >= 0) {
                                TableColumnModel tcm = jTable1.getColumnModel();
                                TableColumn tc = tcm.getColumn(columnCount);
                                dtm.setColumnCount(columnCount);
                                tcm.removeColumn(tc);
                                dtm.setColumnCount(columnCount);
                            }
                        }
                    }
                    System.out.println(jTable1.getColumnCount());
                }
            };
        }
        return abstractAction1;
    }
    private JButton getJButtonAddPartition() {
        if (jButtonAddPartition == null) {
            jButtonAddPartition = new JButton();
            jButtonAddPartition.setText("\u6dfb\u52a0\u5206\u533a");
            jButtonAddPartition.setBounds(229, 220, 100, 30);
            jButtonAddPartition.setAction(getAbstractActionAddPartition());
        }
        return jButtonAddPartition;
    }
    private AbstractAction getAbstractActionAddPartition() {
        if (abstractActionAddPartition == null) {
            abstractActionAddPartition = new AbstractAction("添加分区", null) {
                public void actionPerformed(ActionEvent evt) {
                    ((DefaultTableModel) jTable1Model)
                            .addRow(new String[] { new Integer(jTable1
                                    .getRowCount() + 1).toString() });
                }
            };
        }
        return abstractActionAddPartition;
    }
    private JButton getJButtonRemovePartition() {
        if(jButtonRemovePartition == null) {
            jButtonRemovePartition = new JButton();
            jButtonRemovePartition.setText("\u5220\u9664\u5206\u533a");
```