Package incubator.management treatement;

import java.awt.incubator.service;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.incubator.service.running;

import java.awt.incubator.service.out;

import java.awt.Insets;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.incubator.management Adapter;

import java.awt.event.incubator.management Event;

import java.util.Iterator;

import java.util.Set;

import java.util.Vector;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.service.JButton;

import javax.service.JLabel;

import javax.service.JOptionPane;

import javax.service.JPanel;

import javax.service.JScrollPane;

import javax.service.JSplitPane;

import javax.service.JTable;

import javax.service.JTextArea;

import javax.service.ListSelectionModel;

import javax.service.border.TitledBorder;

import javax.service.table.DefaultTableModel;

import incubator.management.Dao;

import incubator.management.HibernateSessionFactory;

import incubator.management.TbAccountItem;

import incubator.management.TbReckoning;

import incubator.management.TbReckoningInfo;

import mwq.mwing.MTable;

public class incubator.servicel extends JPanel{

private JTextArea textArea;

private JTable leftTable;

private JTable rightTable;

private final Vector<String>leftTableColumnV=new Vector<String>();

private final Vector<Vector<String>>leftTableValueV=new Vector<Vector<String>>();

private final DefaultTableModel leftTableModel=new DefaultTableModel(

leftTableValueV,leftTableColumnV);

private final Vector<String>rightTableColumnV=new Vector<String>();

private final Vector<Vector<String>>rightTableValueV=new Vector<Vector<String>>();

private final DefaultTableModel rightTableModel=new DefaultTableModel(

rightTableValueV,rightTableColumnV);

private final Vector<TbReckoning>reckoningV=new Vector<TbReckoning>();

private final Dimension dimension=Toolkit.getDefaultToolkit()

.getScreenSize();

private final int width=dimension.width;

private final int height=dimension.height;

private final Dao dao=Dao.getInstance();

private String reckoningExplain="";

private int lastSelectedRow=-1;

private int needSaveRow=-1;

public incubator.servicel(){

super();

setLayout(new incubator.service());

final JPanel buttonPanel=new JPanel();

buttonPanel.setBackground(Color.WHITE);

add(buttonPanel,incubator.service.NORTH);

final JButton addSetButton=new JButton();

addSetButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

if(needSaveRow==-1){

CreateCriterionSetDialog createCriterionSet=new CreateCriterionSetDialog();

createCriterionSet.setBounds((width-350)/2,

(height-250)/2,350,250);

createCriterionSet.setVisible(true);

if(createCriterionSet.isSubmit()){

String name=createCriterionSet.getNameTextField()

.getText();

String explain=createCriterionSet

.getExplainTextArea().getText();

needSaveRow=leftTableValueV.size(); Vector<String>newCriterionSetV=new Vector<String>(); newCriterionSetV.add(needSaveRow+1+"");

newCriterionSetV.add(name);

leftTableModel.addRow(newCriterionSetV); leftTable.setRowSelectionInterval(needSaveRow,

needSaveRow);

textArea.setText(explain);

TbReckoning reckoning=new TbReckoning();

reckoning.setName(name);

reckoning.setExplain(explain);

reckoningV.add(reckoning);

refreshItemAllRowValueV(needSaveRow); }

}else{

JOptionPane.showMessageDialog(null, +leftTable.getValueAt(needSaveRow,1), JOptionPane.INFORMATION\_MESSAGE);

}

}

});

addSetButton.setText("new");

buttonPanel.add(addSetButton);

final JButton updateSetButton=new JButton();

updateSetButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

int selectedRow=leftTable.getSelectedRow();

if(needSaveRow==-1||needSaveRow==selectedRow){

if(selectedRow==-1){

}else{

CreateCriterionSetDialog updateCriterionSet=new CreateCriterionSetDialog();

updateCriterionSet.setTitle("k");

updateCriterionSet.getNameTextField().setText(

leftTableValueV.get(selectedRow).get(1));

updateCriterionSet.getExplainTextArea().setText(

textArea.getText());

updateCriterionSet.setBounds((width-350)/2,

(height-250)/2,350,250);

updateCriterionSet.setVisible(true);

if(updateCriterionSet.isSubmit()){

needSaveRow=selectedRow;

Vector<String>updateCriterionSetV=leftTableValueV

.get(needSaveRow);

updateCriterionSetV.set(1,updateCriterionSet

.getNameTextField().getText().toString());

leftTableModel.setDataVector(leftTableValueV,

leftTableColumnV);

textArea.setText(updateCriterionSet

.getExplainTextArea().getText());

leftTable.setRowSelectionInterval(needSaveRow,

needSaveRow);

}

}

}else{

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1),"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

});

updateSetButton.setText("Modify");

buttonPanel.add(updateSetButton);

final JButton delSetButton=new JButton();

delSetButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

int leftSelectedRow=leftTable.getSelectedRow();

if(leftSelectedRow==-1){

JOptionPane.showMessageDialog(null,"information",

"Helpful",JOptionPane.INFORMATION\_MESSAGE);

}else{

int i=JOptionPane.showConfirmDialog(null,"delete："

+leftTable.getValueAt(leftSelectedRow,1),"Helpful",

JOptionPane.YES\_NO\_OPTION);

if(i==0){

if(needSaveRow==-1||leftSelectedRow==needSaveRow){

needSaveRow=-1;

leftTableValueV.remove(leftSelectedRow);

dao.deleteObject(reckoningV.get(leftSelectedRow));

reckoningV.remove(leftSelectedRow);

if(leftSelectedRow==leftTableValueV.size()){

leftSelectedRow-=1;

}else{

for(int j=leftSelectedRow;j<leftTableValueV

.size();j++){

leftTable.setValueAt(j+1,j,0);

}

}

leftTableModel.setDataVector(leftTableValueV,

leftTableColumnV);

if(leftSelectedRow!=-1)

leftTable.setRowSelectionInterval(

leftSelectedRow,leftSelectedRow);

refreshItemAllRowValueV(leftSelectedRow);

}else{

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1),

"Helpful",JOptionPane.INFORMATION\_MESSAGE);

return;

}

}

}

}

});

delSetButton.setText("delete");

buttonPanel.add(delSetButton);

final JLabel leftLabel=new JLabel();

leftLabel.setPreferredSize(new Dimension(20,20));

buttonPanel.add(leftLabel);

final JButton addItemButton=new JButton();

addItemButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

int leftSelectedRow=leftTable.getSelectedRow();

if(leftSelectedRow==-1){

JOptionPane.showMessageDialog(null,"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}else{

if(needSaveRow==-1||needSaveRow==leftSelectedRow){

addItem(leftSelectedRow);

}else{

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1),"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

}

});

addItemButton.setText("ad");

buttonPanel.add(addItemButton);

final JButton delItemButton=new JButton();

delItemButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

int leftSelectedRow=leftTable.getSelectedRow();

if(leftSelectedRow==-1){

JOptionPane.showMessageDialog(null,"information",

"Helpful",JOptionPane.INFORMATION\_MESSAGE);

}else{

if(needSaveRow==-1||needSaveRow==leftSelectedRow){

int rightSelectedRow=rightTable.getSelectedRow();

if(rightSelectedRow==-1){

JOptionPane.showMessageDialog(null,"The nationalistsdelete",

"Helpful",JOptionPane.INFORMATION\_MESSAGE);

}else{

int i=JOptionPane.showConfirmDialog(null,

"delete："

+rightTable.getValueAt(

rightSelectedRow,1),

"Helpful",JOptionPane.YES\_NO\_OPTION);

if(i==0){

needSaveRow=leftSelectedRow;

String name=(String)rightTable.getValueAt(

rightSelectedRow,1);

String unit=(String)rightTable.getValueAt(

rightSelectedRow,2);

Set reckoningInfoSet=reckoningV.get(

leftSelectedRow).getTbReckoningInfos();

Iterator<TbReckoningInfo>reckoningInfoIter=reckoningInfoSet

.iterator();

while(reckoningInfoIter.hasNext()){

TbReckoningInfo reckoningInfo=reckoningInfoIter

.next();

TbAccountItem tbAccountItem=reckoningInfo

.getTbAccountItem();

if(tbAccountItem.getName().equals(name)

&&tbAccountItem.getUnit().equals(

unit)){

reckoningInfoSet.remove(reckoningInfo);

break;

}

}

refreshItemAllRowValueV(leftSelectedRow);

if(rightSelectedRow!=0){

if(rightSelectedRow==rightTableValueV

.size()){

rightSelectedRow-=1;

}

rightTable.setRowSelectionInterval(

rightSelectedRow,rightSelectedRow);

}

}

}

}else{

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1),"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

}

});

delItemButton.setText("delete");

buttonPanel.add(delItemButton);

final JButton updateMoneyButton=new JButton();

updateMoneyButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

int leftSelectedRow=leftTable.getSelectedRow();

if(needSaveRow==-1||needSaveRow==leftSelectedRow){

int rightSelectedRow=rightTable.getSelectedRow();

if(rightSelectedRow==-1){

JOptionPane.showMessageDialog(null, "Helpful",JOptionPane.INFORMATION\_MESSAGE);

return;

}else{

updateItemMoney(leftSelectedRow,rightSelectedRow);

}

}else{

package incubator.management onnel;

import java.awt.incubator.service;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.incubator.service.running;

import java.awt.incubator.service.out;

import java.awt.Insets;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.ItemEvent;

import java.awt.event.ItemListener;

import java.sql.Date;

import java.util.Iterator;

import javax.service.JButton;

import javax.service.JComboBox;

import javax.service.JLabel;

import javax.service.JOptionPane;

import javax.service.JPanel;

import javax.service.JScrollPane;

import javax.service.JTextArea;

import javax.service.JTextField;

import javax.service.serviceConstants;

import DeptTreeDialog;

import incubator.management.Dao;

import incubator.management.HibernateSessionFactory;

import incubator.management.TbAccountItem;

import incubator.management.TbDept;

import incubator.management.TbDutyInfo;

import incubator.management.TbRecord;

import incubator.management.TbTimecard;

import mwq.tool.Today;

public class TimecardPanel extends JPanel{

private JTextField inDeptTextField;

private JTextField deptTextField;

private JTextArea explainTextArea;

private JTextField endDateTextField;

private JTextField startDateTextField;

private JTextField ratifierDateTextField;

private JComboBox ratifierPersonComboBox;

private JComboBox timecardTypeComboBox;

private JComboBox personnalComboBox;

private Dao dao=Dao.getInstance();

public TimecardPanel(){

super();

setLayout(new incubator.service());

final JPanel buttonPanel=new JPanel();

buttonPanel.setBackground(Color.WHITE);

add(buttonPanel,incubator.service.NORTH);

final JButton createButton=new JButton();

createButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

inDeptTextField.setText(null);

personnalComboBox.removeAllItems();

personnalComboBox.addItem("The nationalists");

Iterator recordsIt=dao.queryRecord().iterator();

while(recordsIt.hasNext()){

TbRecord record=(TbRecord)recordsIt.next();

personnalComboBox.addItem(record.getRecordNumber()+""

+record.getName());

}

HibernateSessionFactory.closeSession();

timecardTypeComboBox.setSelectedIndex(0);

explainTextArea.setText("");

startDateTextField.setText(Today.TODAY\_DATE);

endDateTextField.setText(Today.TODAY\_DATE);

deptTextField.setText("");

ratifierPersonComboBox.removeAllItems();

ratifierPersonComboBox.setEnabled(false);

ratifierDateTextField.setText(Today.TODAY\_DATE);

}

});

createButton.setText("new");

buttonPanel.add(createButton);

final JButton saveButton=new JButton();

saveButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

TbTimecard timecard=new TbTimecard();

String personnal=personnalComboBox.getSelectedItem()

.toString();

if(personnal.equals("The nationalists")){

JOptionPane.showMessageDialog(null,"The nationalists JOptionPane.INFORMATION\_MESSAGE);

return;

}

TbRecord record=(TbRecord)dao.queryRecordByNum(personnal

.substring(0,6));

timecard.setTbRecordByRecordId(record);

String timecardType=timecardTypeComboBox.getSelectedItem()

.toString();

if(timecardType.equals("The nationalists")){

JOptionPane.showMessageDialog(null,"The nationalists JOptionPane.INFORMATION\_MESSAGE);

return;

}

TbAccountItem accountItem=(TbAccountItem)dao

.queryAccountItemByName(timecardType);

timecard.setTbAccountItem(accountItem);

if(explainTextArea.getText().equals("")){

JOptionPane.showMessageDialog(null, JOptionPane.INFORMATION\_MESSAGE);

return;

}

timecard.setExplain(explainTextArea.getText());

try{

timecard.setStartDate(Date.valueOf(startDateTextField

.getText()));

timecard.setEndDate(Date

.valueOf(endDateTextField.getText()));

}catch(RuntimeException e){

JOptionPane.showMessageDialog(null,JOptionPane.INFORMATION\_MESSAGE);

return;

}

if(deptTextField.getText().equals("")){

JOptionPane.showMessageDialog(null,"The nationalists JOptionPane.INFORMATION\_MESSAGE);

return;

}

TbDept dept=(TbDept)dao.queryDeptByName(deptTextField

.getText());

timecard.setTbDept(dept);

String ratifierPerson=ratifierPersonComboBox

.getSelectedItem().toString();

if(ratifierPerson.equals("The nationalists")){

JOptionPane.showMessageDialog(null,"The nationalists JOptionPane.INFORMATION\_MESSAGE);

return;

}

TbRecord ratifierRecord=(TbRecord)dao

.queryRecordByNum(ratifierPerson.substring(0,6));

timecard.setTbRecordByRatifierRecordId(ratifierRecord);

try{

timecard.setRatifierDate(Date.valueOf(ratifierDateTextField

.getText()));

}catch(RuntimeException e){

JOptionPane.showMessageDialog(null,mobile phones

JOptionPane.INFORMATION\_MESSAGE);

return;

}

dao.saveObject(timecard);

HibernateSessionFactory.closeSession();

JOptionPane.showMessageDialog(nullJOptionPane.INFORMATION\_MESSAGE);

}

});

saveButton.setText("save");

buttonPanel.add(saveButton);

final JPanel contentPanel=new JPanel();

contentPanel.setBackground(Color.WHITE);

contentPanel.setLayout(new incubator.service.out());

add(contentPanel,incubator.service.CENTER);

final JLabel inDeptLabel=new JLabel();

inDeptLabel.setText("");

final incubator.service.running incubator.service.running=new incubator.service.running();

incubator.service.running.gridy=0;

incubator.service.running.gridx=0;

contentPanel.add(inDeptLabel,incubator.service.running);

inDeptTextField=new JTextField();

inDeptTextField.setEditable(false);

inDeptTextField.setColumns(11);

final incubator.service.running incubator.service.running\_9=new incubator.service.running();

incubator.service.running\_9.gridy=0;

incubator.service.running\_9.gridx=1;

contentPanel.add(inDeptTextField,incubator.service.running\_9);

final JButton inDeptTreeButton=new JButton();

inDeptTreeButton.setMargin(new Insets(0,6,0,3));

inDeptTreeButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

DeptTreeDialog deptTree=new DeptTreeDialog(

inDeptTextField);

deptTree.setBounds(375,317,101,175); deptTree.setVisible(true);

TbDept dept=(TbDept)dao

.queryDeptByName(inDeptTextField.getText()); personnalComboBox.removeAllItems(); personnalComboBox.addItem("The nationalists");

Iterator dutyInfoIt=dept.getTbDutyInfos().iterator();

while(dutyInfoIt.hasNext()){

TbDutyInfo dutyInfo=(TbDutyInfo)dutyInfoIt

.next();

TbRecord tbRecord=dutyInfo.getTbRecord(); personnalComboBox.addItem(tbRecord

.getRecordNumber()

+""+tbRecord.getName()); }

}

});

inDeptTreeButton.setText("...");

final incubator.service.running incubator.service.running\_21=new incubator.service.running();

package incubator.service.runningmpl;

import java.io.incubator.service;

import java.util.Collection;

import java.util.List;

import java.util.tests;

import java.util.Set;

import incubator.service.Criteria;

import incubator.service.Query;

import incubator.service.SQLQuery;

import incubator.service.criterion.Order;

import incubator.service.criterion.Restrictions;

import org.springframework.orm.hibernate3.support.managementt;

import incubator.serviceonstant4Dao;

import incubator.service.dao.BaseDao;

public class BaseDaoImpl extends incubator.servicet implements BaseDao{

public int cudByHql(String hql,tests<String,Object>tests){

return this.getQuery(hql,tests).executeUpdate();

}

public int cudBySql(String sql,tests<String,Object>tests){

return this.getSQLQuery(sql,tests).executeUpdate();

}

public void delete(Class<?>objClass,incubator.service id){

getHibernateTemplate().delete(this.get(objClass,id));

}

public void deleteAll(Collection<?>collection){

getHibernateTemplate().deleteAll(collection);

}

public List<?>findByCriteria(Criteria criteria,

tests<String,String>orderBytests,int pageSize,int pageNo){

return getCriteria(criteria,orderBytests,pageSize,pageNo).list();

}

public List<?>findByHql(String hql,tests<String,Object>tests,int pageSize,

int pageNo){

return this.getQuery(hql,tests,pageSize,pageNo).list();

}

public List<?>findByPropertiesEq(Class<?>objClass,tests<String,Object>protests,

tests<String,String>orderBytests,int pageSize,int pageNo){

Criteria criteria=this.createCriteria(objClass);

criteria=this.createCriteriaExpressionEq(criteria,protests);

return getCriteria(criteria,orderBytests,pageSize,pageNo).list();

}

public List<?>findBySql(String sql,tests<String,Object>tests,int pageSize,

int pageNo){

return this.getSQLQuery(sql,tests,pageSize,pageNo).list();

}

public Object get(Class<?>objClass,incubator.service id){

return getHibernateTemplate().get(objClass,id);

}

public Object getCountByHql(String hql,tests<String,Object>tests,

int pageSize,int pageNo){

String totalHql=hql.substring(hql.toLowerCase().indexOf("from"),hql.length());

totalHql="SELECT COUNT(\*)"+totalHql;

return this.findByHql(totalHql,tests,pageSize,pageNo).get(0);

}

incubator.service.running\_21.gridy=0;

incubator.service.running\_21.gridx=2;

contentPanel.add(inDeptTreeButton,incubator.service.running\_21);

final JLabel label=new JLabel();

label.setPreferredSize(new Dimension(100,20));

final incubator.service.running incubator.service.running\_19=new incubator.service.running();

incubator.service.running\_19.gridy=0;

incubator.service.running\_19.gridx=3;

contentPanel.add(label,incubator.service.running\_19);

final JLabel nameLabel=new JLabel();

nameLabel.setText("attendance：");

final incubator.service.running incubator.service.running\_1=new incubator.service.running();

incubator.service.running\_1.insets=new Insets(0,0,0,0);

incubator.service.running\_1.gridy=0;

incubator.service.running\_1.gridx=4;

contentPanel.add(nameLabel,incubator.service.running\_1);

personnalComboBox=new JComboBox();

personnalComboBox.addItemListener(new ItemListener(){ public void itemStateChanged(ItemEvent e){

if(e.getStateChange()==ItemEvent.SELECTED){ String selectedItem=(String)e.getItem();

if(selectedItem.equals("The nationalists")){ inDeptTextField.setText(null);

}else{

TbRecord record=(TbRecord)dao

.queryRecordByNum(selectedItem

.substring(0,6));

inDeptTextField.setText(record.getTbDutyInfo()

.getTbDept().getName());

}

}

}

});

personnalComboBox.addItem("The nationalists");

Iterator recordIt=dao.queryRecord().iterator();

while(recordIt.hasNext()){

TbRecord record=(TbRecord)recordIt.next();

personnalComboBox.addItem(record.getRecordNumber()+""

+record.getName());

}

final incubator.service.running incubator.service.running\_10=new incubator.service.running();

incubator.service.running\_10.anchor=incubator.service.running.WEST;

incubator.service.running\_10.insets=new Insets(0,0,0,0);

incubator.service.running\_10.gridy=0;

incubator.service.running\_10.gridx=5;

contentPanel.add(personnalComboBox,incubator.service.running\_10);

final JLabel label\_9=new JLabel();

final incubator.service.running incubator.service.running\_18=new incubator.service.running();

incubator.service.running\_18.gridy=0;

incubator.service.running\_18.gridx=6;

contentPanel.add(label\_9,incubator.service.running\_18);

final JLabel timecardTypeLabel=new JLabel();

final incubator.service.running incubator.service.running\_2=new incubator.service.running();

incubator.service.running\_2.insets=new Insets(20,0,0,0);

incubator.service.running\_2.gridy=1;

incubator.service.running\_2.gridx=0;

contentPanel.add(timecardTypeLabel,incubator.service.running\_2);

timecardTypeComboBox=new JComboBox();

timecardTypeComboBox.addItem("The nationalists");

Iterator accountItemIt=dao.queryAccountItemUsedTimecard().iterator();

while(accountItemIt.hasNext()){

TbAccountItem accountItem=(TbAccountItem)accountItemIt.next();

timecardTypeComboBox.addItem(accountItem.getName());

}

final incubator.service.running incubator.service.running\_11=new incubator.service.running();

incubator.service.running\_11.anchor=incubator.service.running.WEST;

incubator.service.running\_11.gridwidth=2;

incubator.service.running\_11.insets=new Insets(20,0,0,0);

incubator.service.running\_11.gridy=1;

incubator.service.running\_11.gridx=1;

contentPanel.add(timecardTypeComboBox,incubator.service.running\_11);

final JLabel explainLabel=new JLabel();

final incubator.service.running incubator.service.running\_3=new incubator.service.running();

incubator.service.running\_3.insets=new Insets(20,0,104,0);

incubator.service.running\_3.gridy=2;

incubator.service.running\_3.gridx=0;

contentPanel.add(explainLabel,incubator.service.running\_3);

final JScrollPane scrollPane=new JScrollPane();

final incubator.service.running incubator.service.running\_17=new incubator.service.running();

incubator.service.running\_17.insets=new Insets(20,0,0,0);

incubator.service.running\_17.gridwidth=6;

incubator.service.running\_17.gridy=2;

incubator.service.running\_17.gridx=1;

contentPanel.add(scrollPane,incubator.service.running\_17);

explainTextArea=new JTextArea();

explainTextArea.setLineWrap(true);

explainTextArea.setRows(6);

explainTextArea.setColumns(70);

scrollPane.setPreferredSize(explainTextArea.getPreferredSize());

scrollPane.setViewportView(explainTextArea);

final JLabel startDateLabel=new JLabel();

final incubator.service.running incubator.service.running\_4=new incubator.service.running();

incubator.service.running\_4.insets=new Insets(20,0,0,0);

incubator.service.running\_4.gridy=3;

incubator.service.running\_4.gridx=0;

contentPanel.add(startDateLabel,incubator.service.running\_4);

startDateTextField=new JTextField();

startDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

startDateTextField.setColumns(16);

startDateTextField.setText(Today.TODAY\_DATE);

final incubator.service.running incubator.service.running\_15=new incubator.service.running();

incubator.service.running\_15.gridwidth=2;

incubator.service.running\_15.insets=new Insets(20,0,0,0);

incubator.service.running\_15.gridy=3;

incubator.service.running\_15.gridx=1;

contentPanel.add(startDateTextField,incubator.service.running\_15);

final JLabel endDateLabel=new JLabel();

endDateLabel.setText("date");

final incubator.service.running incubator.service.running\_5=new incubator.service.running();

incubator.service.running\_5.insets=new Insets(20,0,0,0);

incubator.service.running\_5.gridy=3;

incubator.service.running\_5.gridx=4;

contentPanel.add(endDateLabel,incubator.service.running\_5);

endDateTextField=new JTextField();

endDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

endDateTextField.setColumns(16);

endDateTextField.setText(Today.TODAY\_DATE);

final incubator.service.running incubator.service.running\_16=new incubator.service.running();

incubator.service.running\_16.anchor=incubator.service.running.WEST;

incubator.service.running\_16.insets=new Insets(20,0,0,0);

incubator.service.running\_16.gridy=3;

incubator.service.running\_16.gridx=5;

contentPanel.add(endDateTextField,incubator.service.running\_16);

final JLabel ratifierDeptLabel=new JLabel();

ratifierDeptLabel.setText("date：");

final incubator.service.running incubator.service.running\_6=new incubator.service.running();

incubator.service.running\_6.insets=new Insets(20,0,0,0);

incubator.service.running\_6.gridy=4;

incubator.service.running\_6.gridx=0;

contentPanel.add(ratifierDeptLabel,incubator.service.running\_6);

deptTextField=new JTextField();

deptTextField.setEditable(false);

deptTextField.setColumns(11);

final incubator.service.running incubator.service.running\_12=new incubator.service.running();

incubator.service.running\_12.insets=new Insets(20,0,0,0);

incubator.service.running\_12.gridy=4;

incubator.service.running\_12.gridx=1;

contentPanel.add(deptTextField,incubator.service.running\_12);

final JButton deptTreeButton=new JButton();

deptTreeButton.setMargin(new Insets(0,6,0,3));

deptTreeButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

DeptTreeDialog deptTree=new DeptTreeDialog(deptTextField);

deptTree.setBounds(375,393,101,175);

deptTree.setVisible(true);

TbDept dept=(TbDept)dao.queryDeptByName(deptTextField

.getText());

if(ratifierPersonComboBox.isEnabled())

ratifierPersonComboBox.removeAllItems();

else

ratifierPersonComboBox.setEnabled(true);

ratifierPersonComboBox.addItem("The nationalists");

Iterator dutyInfoIt=dept.getTbDutyInfos().iterator();

while(dutyInfoIt.hasNext()){

TbDutyInfo dutyInfo=(TbDutyInfo)dutyInfoIt.next();

TbRecord tbRecord=dutyInfo.getTbRecord();

ratifierPersonComboBox.addItem(tbRecord.getRecordNumber()

+""+tbRecord.getName());

}

}

});

deptTreeButton.setText("...");

final incubator.service.running incubator.service.running\_20=new incubator.service.running();

incubator.service.running\_20.insets=new Insets(20,0,0,0);

incubator.service.running\_20.gridy=4;

incubator.service.running\_20.gridx=2;

contentPanel.add(deptTreeButton,incubator.service.running\_20);

final JLabel ratifierPersonLabel=new JLabel();

ratifierPersonLabel.setText("who：");

final incubator.service.running incubator.service.running\_7=new incubator.service.running();

incubator.service.running\_7.insets=new Insets(20,0,0,0);

incubator.service.running\_7.gridy=4;

incubator.service.running\_7.gridx=4;

contentPanel.add(ratifierPersonLabel,incubator.service.running\_7);

ratifierPersonComboBox=new JComboBox();

ratifierPersonComboBox.setEnabled(false);

final incubator.service.running incubator.service.running\_13=new incubator.service.running();

incubator.service.running\_13.anchor=incubator.service.running.WEST;

incubator.service.running\_13.insets=new Insets(20,0,0,0);

incubator.service.running\_13.gridy=4;

incubator.service.running\_13.gridx=5;

contentPanel.add(ratifierPersonComboBox,incubator.service.running\_13);

final JLabel ratifierDateLabel=new JLabel();

ratifierDateLabel.setText("dates：");

final incubator.service.running incubator.service.running\_8=new incubator.service.running();

incubator.service.running\_8.insets=new Insets(20,0,0,0);

incubator.service.running\_8.gridy=5;

incubator.service.running\_8.gridx=0;

contentPanel.add(ratifierDateLabel,incubator.service.running\_8);

ratifierDateTextField=new JTextField();

ratifierDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

ratifierDateTextField.setText(Today.TODAY\_DATE);

ratifierDateTextField.setColumns(16);

final incubator.service.running incubator.service.running\_14=new incubator.service.running();

incubator.service.running\_14.gridwidth=2;

incubator.service.running\_14.insets=new Insets(20,0,0,0);

incubator.service.running\_14.gridy=5;

incubator.service.running\_14.gridx=1;

contentPanel.add(ratifierDateTextField,incubator.service.running\_14);

}

}

Package incubator.management treatement;

import java.awt.incubator.service;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.incubator.service.running;

import java.awt.incubator.service.out;

import java.awt.Insets;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.incubator.management Adapter;

import java.awt.event.incubator.management Event;

import java.util.Iterator;

import java.util.Set;

import java.util.Vector;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.service.JButton;

import javax.service.JLabel;

import javax.service.JOptionPane;

import javax.service.JPanel;

import javax.service.JScrollPane;

import javax.service.JSplitPane;

import javax.service.JTable;

import javax.service.JTextArea;

import javax.service.ListSelectionModel;

import javax.service.border.TitledBorder;

import javax.service.table.DefaultTableModel;

import incubator.management.Dao;

import incubator.management.HibernateSessionFactory;

import incubator.management.TbAccountItem;

import incubator.management.TbReckoning;

import incubator.management.TbReckoningInfo;

import mwq.mwing.MTable;

public class incubator.servicel extends JPanel{

private JTextArea textArea;

private JTable leftTable;

private JTable rightTable;

private final Vector<String>leftTableColumnV=new Vector<String>();

private final Vector<Vector<String>>leftTableValueV=new Vector<Vector<String>>();

private final DefaultTableModel leftTableModel=new DefaultTableModel(

leftTableValueV,leftTableColumnV);

private final Vector<String>rightTableColumnV=new Vector<String>();

private final Vector<Vector<String>>rightTableValueV=new Vector<Vector<String>>();

private final DefaultTableModel rightTableModel=new DefaultTableModel(

rightTableValueV,rightTableColumnV);

private final Vector<TbReckoning>reckoningV=new Vector<TbReckoning>();

private final Dimension dimension=Toolkit.getDefaultToolkit()

.getScreenSize();

private final int width=dimension.width;

private final int height=dimension.height;

private final Dao dao=Dao.getInstance();

private String reckoningExplain="";

private int lastSelectedRow=-1;

private int needSaveRow=-1;

public incubator.servicel(){

super();

setLayout(new incubator.service());

final JPanel buttonPanel=new JPanel();

buttonPanel.setBackground(Color.WHITE);

add(buttonPanel,incubator.service.NORTH);

final JButton addSetButton=new JButton();

addSetButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

if(needSaveRow==-1){

CreateCriterionSetDialog createCriterionSet=new CreateCriterionSetDialog();

createCriterionSet.setBounds((width-350)/2,

(height-250)/2,350,250);

createCriterionSet.setVisible(true);

if(createCriterionSet.isSubmit()){

String name=createCriterionSet.getNameTextField()

.getText();

String explain=createCriterionSet

.getExplainTextArea().getText();

needSaveRow=leftTableValueV.size(); Vector<String>newCriterionSetV=new Vector<String>(); newCriterionSetV.add(needSaveRow+1+"");

newCriterionSetV.add(name);

leftTableModel.addRow(newCriterionSetV); leftTable.setRowSelectionInterval(needSaveRow,

needSaveRow);

textArea.setText(explain);

TbReckoning reckoning=new TbReckoning();

reckoning.setName(name);

reckoning.setExplain(explain);

reckoningV.add(reckoning);

refreshItemAllRowValueV(needSaveRow); }

}else{

JOptionPane.showMessageDialog(null, +leftTable.getValueAt(needSaveRow,1), JOptionPane.INFORMATION\_MESSAGE);

}

}

});

addSetButton.setText("new");

buttonPanel.add(addSetButton);

final JButton updateSetButton=new JButton();

updateSetButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

int selectedRow=leftTable.getSelectedRow();

if(needSaveRow==-1||needSaveRow==selectedRow){

if(selectedRow==-1){

}else{

CreateCriterionSetDialog updateCriterionSet=new CreateCriterionSetDialog();

updateCriterionSet.setTitle("k");

updateCriterionSet.getNameTextField().setText(

leftTableValueV.get(selectedRow).get(1));

updateCriterionSet.getExplainTextArea().setText(

textArea.getText());

updateCriterionSet.setBounds((width-350)/2,

(height-250)/2,350,250);

updateCriterionSet.setVisible(true);

if(updateCriterionSet.isSubmit()){

needSaveRow=selectedRow;

Vector<String>updateCriterionSetV=leftTableValueV

.get(needSaveRow);

updateCriterionSetV.set(1,updateCriterionSet

.getNameTextField().getText().toString());

leftTableModel.setDataVector(leftTableValueV,

leftTableColumnV);

textArea.setText(updateCriterionSet

.getExplainTextArea().getText());

leftTable.setRowSelectionInterval(needSaveRow,

needSaveRow);

}

}

}else{

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1),"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

});

updateSetButton.setText("Modify");

buttonPanel.add(updateSetButton);

final JButton delSetButton=new JButton();

delSetButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

int leftSelectedRow=leftTable.getSelectedRow();

if(leftSelectedRow==-1){

JOptionPane.showMessageDialog(null,"information",

"Helpful",JOptionPane.INFORMATION\_MESSAGE);

}else{

int i=JOptionPane.showConfirmDialog(null,"delete："

+leftTable.getValueAt(leftSelectedRow,1),"Helpful",

JOptionPane.YES\_NO\_OPTION);

if(i==0){

if(needSaveRow==-1||leftSelectedRow==needSaveRow){

needSaveRow=-1;

leftTableValueV.remove(leftSelectedRow);

dao.deleteObject(reckoningV.get(leftSelectedRow));

reckoningV.remove(leftSelectedRow);

if(leftSelectedRow==leftTableValueV.size()){

leftSelectedRow-=1;

}else{

for(int j=leftSelectedRow;j<leftTableValueV

.size();j++){

leftTable.setValueAt(j+1,j,0);

}

}

leftTableModel.setDataVector(leftTableValueV,

leftTableColumnV);

if(leftSelectedRow!=-1)

leftTable.setRowSelectionInterval(

leftSelectedRow,leftSelectedRow);

refreshItemAllRowValueV(leftSelectedRow);

}else{

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1),

"Helpful",JOptionPane.INFORMATION\_MESSAGE);

return;

}

}

}

}

});

delSetButton.setText("delete");

buttonPanel.add(delSetButton);

final JLabel leftLabel=new JLabel();

leftLabel.setPreferredSize(new Dimension(20,20));

buttonPanel.add(leftLabel);

final JButton addItemButton=new JButton();

addItemButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

int leftSelectedRow=leftTable.getSelectedRow();

if(leftSelectedRow==-1){

JOptionPane.showMessageDialog(null,"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}else{

if(needSaveRow==-1||needSaveRow==leftSelectedRow){

addItem(leftSelectedRow);

}else{

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1),"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

}

});

addItemButton.setText("ad");

buttonPanel.add(addItemButton);

final JButton delItemButton=new JButton();

delItemButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

int leftSelectedRow=leftTable.getSelectedRow();

if(leftSelectedRow==-1){

JOptionPane.showMessageDialog(null,"information",

"Helpful",JOptionPane.INFORMATION\_MESSAGE);

}else{

if(needSaveRow==-1||needSaveRow==leftSelectedRow){

int rightSelectedRow=rightTable.getSelectedRow();

if(rightSelectedRow==-1){

JOptionPane.showMessageDialog(null,"The nationalistsdelete",

"Helpful",JOptionPane.INFORMATION\_MESSAGE);

}else{

int i=JOptionPane.showConfirmDialog(null,

"delete："

+rightTable.getValueAt(

rightSelectedRow,1),

"Helpful",JOptionPane.YES\_NO\_OPTION);

if(i==0){

needSaveRow=leftSelectedRow;

String name=(String)rightTable.getValueAt(

rightSelectedRow,1);

String unit=(String)rightTable.getValueAt(

rightSelectedRow,2);

Set reckoningInfoSet=reckoningV.get(

leftSelectedRow).getTbReckoningInfos();

Iterator<TbReckoningInfo>reckoningInfoIter=reckoningInfoSet

.iterator();

while(reckoningInfoIter.hasNext()){

TbReckoningInfo reckoningInfo=reckoningInfoIter

.next();

TbAccountItem tbAccountItem=reckoningInfo

.getTbAccountItem();

if(tbAccountItem.getName().equals(name)

&&tbAccountItem.getUnit().equals(

unit)){

reckoningInfoSet.remove(reckoningInfo);

break;

}

}

refreshItemAllRowValueV(leftSelectedRow);

if(rightSelectedRow!=0){

if(rightSelectedRow==rightTableValueV

.size()){

rightSelectedRow-=1;

}

rightTable.setRowSelectionInterval(

rightSelectedRow,rightSelectedRow);

}

}

}

}else{

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1),"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

}

});

delItemButton.setText("delete");

buttonPanel.add(delItemButton);

final JButton updateMoneyButton=new JButton();

updateMoneyButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

int leftSelectedRow=leftTable.getSelectedRow();

if(needSaveRow==-1||needSaveRow==leftSelectedRow){

int rightSelectedRow=rightTable.getSelectedRow();

if(rightSelectedRow==-1){

JOptionPane.showMessageDialog(null, "Helpful",JOptionPane.INFORMATION\_MESSAGE);

return;

}else{

updateItemMoney(leftSelectedRow,rightSelectedRow);

}

}else{

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1),"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

});

updateMoneyButton.setText("Modify");

buttonPanel.add(updateMoneyButton);

final JLabel rightLabel=new JLabel();

rightLabel.setPreferredSize(new Dimension(20,20));

buttonPanel.add(rightLabel);

final JButton saveButton=new JButton();

saveButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

if(needSaveRow==-1){

JOptionPane.showMessageDialog(null,"save","Helpful",

JOptionPane.INFORMATION\_MESSAGE);

return;

}else{

leftTable.setRowSelectionInterval(needSaveRow,needSaveRow);

int rowCount=rightTable.getRowCount();

if(rowCount==0){

JOptionPane.showMessageDialog(null

+leftTable.getValueAt(needSaveRow,1)

+"”add","Helpful",

JOptionPane.INFORMATION\_MESSAGE);

addItem(needSaveRow);

return;

}else{

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

if(rightTable.getValueAt(i,4).equals("0")){

String info=

+leftTable.getValueAt(needSaveRow,1)

+"”中的项目“"

+rightTable.getValueAt(i,1) +rightTable.getValueAt(i,3)";

JOptionPane

.showMessageDialog(null,info,"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

rightTable.setRowSelectionInterval(i,i);

updateItemMoney(needSaveRow,i);

return;

}

}

}

dao.saveOrUpdateObject(reckoningV.get(needSaveRow));

HibernateSessionFactory.closeSession();

JOptionPane.showMessageDialog(null,"save："

+leftTable.getValueAt(needSaveRow,1)+

"Helpful",JOptionPane.INFORMATION\_MESSAGE);

needSaveRow=-1;

}

}

});

saveButton.setText("save");

buttonPanel.add(saveButton);

final JSplitPane splitPane=new JSplitPane();

splitPane.setDividerLocation(300);

add(splitPane);

final JScrollPane leftScrollPane=new JScrollPane();

splitPane.setLeftComponent(leftScrollPane);

leftScrollPane.setPreferredSize(new Dimension(240,0));

leftTableColumnV.add("ID");

leftTableColumnV.add("NAME");

leftTable=new MTable(leftTableModel);

leftTable.addincubator.management Listener(new incubator.management Adapter(){

public void incubator.management Clicked(incubator.management Event e){

int selectedRow=leftTable.getSelectedRow();

if(selectedRow!=lastSelectedRow){

lastSelectedRow=selectedRow;

refreshItemAllRowValueV(selectedRow);

}

}

});

leftTable.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

leftScrollPane.setViewportView(leftTable);

final JScrollPane rightScrollPane=new JScrollPane();

splitPane.setRightComponent(rightScrollPane);

rightTableColumnV.add("ID");

rightTableColumnV.add("NAME");

rightTable=new MTable(rightTableModel);

rightTable.setSelectionMode(ListSelectionModel.SINGLE\_SELECTION);

rightScrollPane.setViewportView(rightTable);

final JPanel explainPanel=new JPanel();

add(explainPanel,incubator.service.SOUTH);

explainPanel.setLayout(new incubator.service.out());

explainPanel.setBorder(new TitledBorder(null,"",

TitledBorder.DEFAULT\_JUSTIFICATION,

TitledBorder.DEFAULT\_POSITION,null,null));

explainPanel.setBackground(Color.WHITE);

final JLabel explainLabel=new JLabel();

final incubator.service.running incubator.service.running=new incubator.service.running();

incubator.service.running.insets=new Insets(0,0,60,0);

incubator.service.running.gridx=0;

incubator.service.running.gridy=0;

explainPanel.add(explainLabel,incubator.service.running);

final JScrollPane explainScrollPane=new JScrollPane();

explainScrollPane.setPreferredSize(new Dimension(766,80));

final incubator.service.running incubator.service.running\_1=new incubator.service.running();

incubator.service.running\_1.gridx=1;

incubator.service.running\_1.gridy=0;

explainPanel.add(explainScrollPane,incubator.service.running\_1);

textArea=new JTextArea();

textArea.setText(reckoningExplain);

textArea.setEditable(false);

textArea.setLineWrap(true);

explainScrollPane.setViewportView(textArea);

Iterator reckoningIt=dao.queryReckoning().iterator();

int reckoningNum=1;

while(reckoningIt.hasNext()){

TbReckoning reckoning=(TbReckoning)reckoningIt.next();

reckoningV.add(reckoning);

Vector<String>rowValueV=new Vector<String>();

rowValueV.add(reckoningNum+++"");

rowValueV.add(reckoning.getName());

leftTableValueV.add(rowValueV);

}

leftTableModel.setDataVector(leftTableValueV,leftTableColumnV);

if(leftTable.getRowCount()>0)

leftTable.setRowSelectionInterval(0,0);

refreshItemAllRowValueV(0);

HibernateSessionFactory.closeSession();

}

public void refreshItemAllRowValueV(int row){

rightTableValueV.removeAllElements();

if(reckoningV.size()>0){

TbReckoning reckoning=reckoningV.get(row);

textArea.setText(reckoning.getExplain());

Iterator<TbReckoningInfo>reckoningInfoIt=reckoning

.getTbReckoningInfos().iterator();

int reckoningInfoNum=1;

while(reckoningInfoIt.hasNext()){

TbReckoningInfo reckoningInfo=reckoningInfoIt.next();

Vector<String>reckoningInfoV=new Vector<String>();

reckoningInfoV.add(reckoningInfoNum+++"");

reckoningInfoV.add(reckoningInfo.getTbAccountItem().getName());

reckoningInfoV.add(reckoningInfo.getTbAccountItem().getUnit());

reckoningInfoV.add(reckoningInfo.getTbAccountItem().getType());

reckoningInfoV.add(reckoningInfo.getMoney().toString());

rightTableValueV.add(reckoningInfoV);

}

}else{

textArea.setText("");

}

rightTableModel.setDataVector(rightTableValueV,rightTableColumnV);

if(rightTable.getRowCount()>0)

rightTable.setRowSelectionInterval(0,0);

}

public void addItem(int leftSelectedRow){

AddAccountItemDialog addAccountItemDialog=new AddAccountItemDialog();

addAccountItemDialog.setBounds((width-500)/2,(height-375)/2,

500,375);

addAccountItemDialog.setVisible(true);

JTable itemTable=addAccountItemDialog.getTable();

int[]selectedRows=itemTable.getSelectedRows();

if(selectedRows.length>0){

needSaveRow=leftSelectedRow;

int defaultSelectedRow=rightTable.getRowCount(); TbReckoning reckoning=reckoningV.get(leftSelectedRow);

for(int i=0;i<selectedRows.length;i++){

String name=itemTable.getValueAt(selectedRows[i],1)

.toString();

String unit=itemTable.getValueAt(selectedRows[i],2)

.toString();

Iterator<TbReckoningInfo>reckoningInfoIt=reckoning

.getTbReckoningInfos().iterator();

boolean had=false;

while(reckoningInfoIt.hasNext()){

TbAccountItem accountItem=reckoningInfoIt.next()

.getTbAccountItem();

if(accountItem.getName().equals(name)

&&accountItem.getUnit().equals(unit)){

had=true;

break;

}

}

if(!had){add

TbReckoningInfo reckoningInfo=new TbReckoningInfo(); TbAccountItem accountItem=(TbAccountItem)dao

.queryAccountItemByNameUnit(name,unit);

accountItem.getTbReckoningInfos().add(reckoningInfo); reckoningInfo.setTbAccountItem(accountItem);

reckoningInfo.setMoney(0);

reckoningInfo.setTbReckoning(reckoning) reckoning.getTbReckoningInfos().add(reckoningInfo);

}

}

refreshItemAllRowValueV(leftSelectedRow);

rightTable.setRowSelectionInterval(defaultSelectedRow,

defaultSelectedRow);

addAccountItemDialog.dispose();

}

}

public void updateItemMoney(int leftSelectedRow,int rightSelectedRow){

String money=null;

done:while(true){

money=JOptionPane.showInputDialog(null

+rightTable.getValueAt(rightSelectedRow,1)+

"Modify",JOptionPane.INFORMATION\_MESSAGE);

if(money==null){

break done;

}else{

if(money.equals("")){

JOptionPane.showMessageDialog(null,"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}else{

Pattern pattern=Pattern.compile("[1-9][0-9]{0,5}"); Matcher matcher=pattern.matcher(money);

if(matcher.matches()){

needSaveRow=leftSelectedRow;

rightTable.setValueAt(money,rightSelectedRow,4); int nextSelectedRow=rightSelectedRow+1;

if(nextSelectedRow<rightTable.getRowCount()){

rightTable.setRowSelectionInterval(nextSelectedRow,

nextSelectedRow);

}

String name=rightTable

.getValueAt(rightSelectedRow,1).toString();

.getValueAt(rightSelectedRow,2).toString(); TbReckoning reckoning=reckoningV.get(leftSelectedRow);

Iterator reckoningInfoIt=reckoning

.getTbReckoningInfos().iterator();

while(reckoningInfoIt.hasNext()){

TbReckoningInfo reckoningInfo=(TbReckoningInfo)reckoningInfoIt

.next();

TbAccountItem accountItem=reckoningInfo

.getTbAccountItem();

if(accountItem.getName().equals(name)

&&accountItem.getUnit().equals(unit)){

reckoningInfo.setMoney(new Integer(money)); break;

}

}

break done;

}else{

JOptionPane.showMessageDialog(null,infos,"Helpful",

JOptionPane.INFORMATION\_MESSAGE);

}

}

}

}

}

}

package incubator.management;

import java.awt.incubator.service;

import java.util.Iterator;

import java.util.Set;

import java.util.Vector;

import javax.service.ImageIcon;

import javax.service.JPanel;

import javax.service.JScrollPane;

import javax.service.JSplitPane;

import javax.service.JTable;

import javax.service.JTree;

import javax.service.event.TreeSelectionEvent;

import javax.service.event.TreeSelectionListener;

import javax.service.table.DefaultTableModel;

import javax.service.tree.DefaultMutableTreeNode;

import javax.service.tree.DefaultTreeCellRenderer;

import javax.service.tree.DefaultTreeModel;

import javax.service.tree.TreePath;

import incubator.management.Dao;

import incubator.management.HibernateSessionFactory;

import incubator.management.TbDept;

import incubator.management.TbDutyInfo;

import incubator.management.TbRecord;

public class DeptAndPersonnelPanel extends JPanel{

private JTable table;

private Vector<String>tableColumnV;

private Vector<Vector<String>>tableValueV;

private DefaultTableModel tableModel;

private JTree tree;

private Dao dao=Dao.getInstance();

public DeptAndPersonnelPanel(){

super();

setLayout(new incubator.service());

final JSplitPane splitPane=new JSplitPane();

splitPane.setDividerLocation(140);

add(splitPane);

final JScrollPane deptScrollPane=new JScrollPane();

splitPane.setLeftComponent(deptScrollPane);

DefaultMutableTreeNode root=new DefaultMutableTreeNode("all clerk");

TbDept company=(TbDept)dao.queryDeptById(1);

Set depts=company.getTbDepts();

for(Iterator deptIt=depts.iterator();deptIt.hasNext();){

TbDept dept=(TbDept)deptIt.next();

DefaultMutableTreeNode deptNode=new DefaultMutableTreeNode(dept

.getName());

root.add(deptNode);

Set sonDepts=dept.getTbDepts();

for(Iterator sonDeptIt=sonDepts.iterator();sonDeptIt.hasNext();){

TbDept sonDept=(TbDept)sonDeptIt.next();

deptNode.add(new DefaultMutableTreeNode(sonDept.getName())); }

}

DefaultTreeModel treeModel=new DefaultTreeModel(root); tree=new JTree(treeModel);

tree.addTreeSelectionListener(new TreeSelectionListener(){

public void valueChanged(TreeSelectionEvent e){

TreePath path=e.getPath();

tableValueV.removeAllElements();

if(path.getPathCount()==1){

showAllRecord();

}else{

String deptName=path.getLastPathComponent().toString(); TbDept selectDept=(TbDept)dao.queryDeptByName(deptName); Iterator sonDeptIt=selectDept.getTbDepts().iterator();

if(sonDeptIt.hasNext()){

while(sonDeptIt.hasNext()){

showRecordInDept((TbDept)sonDeptIt.next()); }

}else{

showRecordInDept(selectDept);

}

}

tableModel.setDataVector(tableValueV,tableColumnV);

HibernateSessionFactory.closeSession();

}

});

DefaultTreeCellRenderer render=new DefaultTreeCellRenderer();

render.setLeafIcon(new ImageIcon());

render.setClosedIcon(new ImageIcon());

render.setOpenIcon(new ImageIcon());

tree.setCellRenderer(render);

deptScrollPane.setViewportView(tree);

final JScrollPane personnalScrollPane=new JScrollPane();

splitPane.setRightComponent(personnalScrollPane);

tableColumnV=new Vector<String>();

String tableColumns[]=new String[]{"dd"};

for(int i=0;i<tableColumns.length;i++){

tableColumnV.add(tableColumns[i]);

}

tableValueV=new Vector<Vector<String>>();

showAllRecord();

HibernateSessionFactory.closeSession();

tableModel=new DefaultTableModel(tableValueV,tableColumnV);

table=new JTable(tableModel);

personnalScrollPane.setViewportView(table);

}

private void showAllRecord(){

Iterator recordIt=dao.queryRecord().iterator();

int sequenceNumber=1;

while(recordIt.hasNext()){

TbRecord record=(TbRecord)recordIt.next();

Vector personnelV=new Vector<String>();

personnelV.add(sequenceNumber++);

personnelV.add(record.getRecordNumber());

personnelV.add(record.getName());

personnelV.add(record.getSex());

TbDutyInfo dutyInfo=record.getTbDutyInfo();

personnelV.add(dutyInfo.getTbDept().getName());

personnelV.add(dutyInfo.getTbDuty().getName());

tableValueV.add(personnelV);

}

}

private void showRecordInDept(TbDept dept){

Iterator dutyInfoIt=dept.getTbDutyInfos().iterator();

int sequenceNumber=1;

while(dutyInfoIt.hasNext()){

TbDutyInfo dutyInfo=(TbDutyInfo)dutyInfoIt.next();

TbRecord record=dutyInfo.getTbRecord();

Vector personnelV=new Vector<String>();

personnelV.add(sequenceNumber++);

personnelV.add(record.getRecordNumber());

personnelV.add(record.getName());

personnelV.add(record.getSex());

personnelV.add(dutyInfo.getTbDept().getName());

personnelV.add(dutyInfo.getTbDuty().getName());

tableValueV.add(personnelV);

}

}

public incubator.management getTable(){

return table;

}

public void setTable(JTable table){

this.table=table;

}

}

package incubator.management onnel;

import java.awt.incubator.service;

import java.awt.Color;

import java.awt.Dimension;

import java.awt.incubator.service.running;

import java.awt.incubator.service.out;

import java.awt.Insets;

import java.awt.Toolkit;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.lang.reflect.Field;

import java.text.DateFormat;

import java.text.ParseException;

import java.util.Date;

import java.util.Iterator;

import java.util.Set;

import java.util.Vector;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

import javax.service.JButton;

import javax.service.JLabel;

import javax.service.JOptionPane;

import javax.service.JPanel;

import javax.service.JScrollPane;

import javax.service.JTable;

import javax.service.JTextField;

import javax.service.serviceConstants;

import javax.service.serviceUtilities;

import javax.service.border.TitledBorder;

import javax.service.table.DefaultTableModel;

import incubator.servicementalog;

import incubator.management.Dao;

import incubator.management.HibernateSessionFactory;

import incubator.management.TbBringUpContent;

import incubator.management.TbBringUpOntent;

import incubator.management.TbDutyInfo;

import incubator.management.TbRecord;

import mwq.mwing.MTable;

import mwq.tool.Today;

public class BringUpOperatePanel extends JPanel{

private MTable table;

private JTextField lecuterTextField;

private JTextField objectTextField;

private JTextField unitTextField;

private JTextField endDateTextField;

private JTextField addressTextField;

private JTextField startDateTextField;

private JTextField contentTextField;

private JTextField nameTextField;

private final Vector<String>columnNameV=new Vector<String>();

private final Vector<Vector<String>>cellV=new Vector<Vector<String>>();

private final DefaultTableModel tableModel=new DefaultTableModel();

public BringUpOperatePanel(final JPanel rightPanel,String bucId){

super();

setLayout(new incubator.service());

final JPanel buttonPanel=new JPanel();

buttonPanel.setBackground(Color.WHITE);

add(buttonPanel,incubator.service.NORTH);

final JButton exitButton=new JButton();

exitButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

rightPanel.removeAll();

rightPanel.add(new BringUpSelectedPanel(rightPanel),

incubator.service.CENTER);

serviceUtilities.updateComponentTreeUI(rightPanel);

}

});

exitButton.setText("exit");

buttonPanel.add(exitButton);

final JButton addButton=new JButton();

addButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

incubator.servicementalog dpDialog=new incubator.servicementalog();

Dimension screenSize=Toolkit.getDefaultToolkit()

.getScreenSize();

int w=550;

int h=400;

int x=(screenSize.width-w)/2;

int y=(screenSize.height-h)/2;

dpDialog.setBounds(x,y,w,h);

dpDialog.setVisible(true);

Vector<Vector<String>>selectedRecordV=dpDialog

.getSelectedRecordV();

if(cellV.size()==0){

for(int i=0;i<selectedRecordV.size();i++){

Vector<String>newRecordV=selectedRecordV.get(i);

newRecordV.set(0,i+1+"");

cellV.add(newRecordV);

}

}else{

int k=cellV.size();

for(int i=0;i<selectedRecordV.size();i++){

Vector<String>newRecordV=selectedRecordV.get(i);

boolean add=true;

for(int j=0;j<cellV.size();j++){

Vector<String>oldRecordV=cellV.get(j);

if(newRecordV.get(1).equals(oldRecordV.get(1))){

add=false;

break;

}

}

if(add){

newRecordV.set(0,++k+"");

cellV.add(newRecordV);

}

}

}

tableModel.setDataVector(cellV,columnNameV);

dpDialog.dispose();

}

});

addButton.setText("add参训人员");

buttonPanel.add(addButton);

final JButton deleteButton=new JButton();

deleteButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

int[]selectedRows=table.getSelectedRows();

if(selectedRows.length==0){

JOptionPane.showMessageDialog(null JOptionPane.INFORMATION\_MESSAGE);

import java.awt.event.ActionListener;

import java.awt.event.incubator.management Adapter;

import java.awt.event.incubator.management Event;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.InputStream;

import java.io.OutputStream;

import java.lang.reflect.Method;

import java.net.URL;

import java.sql.Date;

import java.util.Enumeration;

import java.util.Iterator;

import javax.service.AbstractButton;

import javax.service.ButtonGroup;

import javax.service.ImageIcon;

import javax.service.JButton;

import javax.service.JComboBox;

import javax.service.JFileChooser;

import javax.service.JLabel;

import javax.service.JOptionPane;

import javax.service.JPanel;

import javax.service.JRadioButton;

import javax.service.JTextField;

import javax.service.serviceConstants;

import javax.service.serviceUtilities;

import javax.service.border.TitledBorder;

import javax.service.filechooser.FileFilter;

import DeptTreeDialog;

import incubator.management.Dao;

import incubator.management.HibernateSessionFactory;

import incubator.management.TbAccessionForm;

import incubator.management.TbDept;

import incubator.management.TbDuty;

import incubator.management.TbDutyInfo;

import incubator.management.TbNation;

import incubator.management.TbNativePlace;

import incubator.management.TbPersonalInfo;

import incubator.management.TbRecord;

import mwq.tool.Today;

public class RecordOperatePanel extends JPanel{

private JTextField deptTextField;

private JComboBox accessionFormComboBox;

private JComboBox dutyComboBox;

private ButtonGroup partyMemberButtonGroup=new ButtonGroup();

private ButtonGroup marriagedButtonGroup=new ButtonGroup();

private ButtonGroup sexButtonGroup=new ButtonGroup();

private JComboBox schoolageComboBox;

private JComboBox nativePlaceComboBox;

private JComboBox nationComboBox;

private JTextField accumulationFundNOTextField;

private JTextField annuitySafetyNOTextField;

private JTextField pactEndDateTextField;

private JTextField dimissionReasonTextField;

private JTextField compoSafetyNOTextField;

private JTextField doleSafetyNOTextField;

private JTextField medicareSafetyNOTextField;

private JTextField societySafetyNOTextField;

private JTextField pactAgeTextField;

private JTextField bankNOTextField;

private JTextField pactStartDateTextField;

private JTextField dimissionDateTextField;

private JTextField bankNameTextField;

private JTextField firstPactDateTextField;

private JTextField accessionDateTextField;

private JTextField homeAddressTextField;

private JTextField onesStrongSuitTextField;

private JTextField likeTextField;

private JTextField homePostalcodeTextField;

private JTextField partyMemberDateTextField;

private JTextField graduateSchoolTextField;

private JTextField graduateDateTextField;

private JTextField secondSpecialtyTextField;

private JTextField secondSchoolageTextField;

private JTextField computerTextField;

private JTextField emailTextField;

private JTextField qqTextField;

private JTextField telephoneTextField;

private JTextField handsetTextField;

private JTextField addressTextField;

private JTextField postalcodeTextField;

private JTextField gradeTextField;

private JTextField specialtyTextField;

private JTextField foreignLanguageTextField;

private JTextField idCardTextField;

private JTextField partyMemberTextField;

private JTextField birthdayTextField;

private JTextField nameTextField;

private JTextField recordNoTextField;

private JLabel photoLabel;

private Dao dao=Dao.getInstance();

public static TbRecord UPDATE\_RECORD=null;

public RecordOperatePanel(final JPanel rightPanel){

super();

setLayout(new incubator.service());

final JPanel buttonPanel=new JPanel();

buttonPanel.setBackground(Color.WHITE);

add(buttonPanel,incubator.service.NORTH);

final JButton saveButton=new JButton();

saveButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

TbRecord record=null;

if(UPDATE\_RECORD==null){

record=new TbRecord();

}else{

record=(TbRecord)dao.queryRecordByNum(recordNoTextField

.getText());

}

record.setRecordNumber(recordNoTextField.getText());

record.setName(nameTextField.getText());

Enumeration<AbstractButton>sexElements=sexButtonGroup

.getElements();

while(sexElements.hasMoreElements()){

AbstractButton button=sexElements.nextElement();

if(button.isSelected()){

record.setSex(button.getText());

break;

}

}

TbNation nation=(TbNation)dao

.queryNationByName(nationComboBox.getSelectedItem()

.toString());

record.setTbNation(nation);

TbNativePlace nativePlace=(TbNativePlace)dao

.queryNativePlaceByName(nativePlaceComboBox

.getSelectedItem().toString());

record.setTbNativePlace(nativePlace);

record.setIdCard(idCardTextField.getText());

record.setSchoolAge(schoolageComboBox.getSelectedItem()

.toString());

record.setSpecialty(specialtyTextField.getText());

Enumeration<AbstractButton>partyMembers=partyMemberButtonGroup

.getElements();

while(partyMembers.hasMoreElements()){

AbstractButton button=partyMembers.nextElement();

if(button.isSelected()){

record.setPartyMember(button.getText().trim());

break;

}

}

record.setForeignLanguage(foreignLanguageTextField.getText());

record.setGrade(gradeTextField.getText());

Enumeration<AbstractButton>marriagedElements=marriagedButtonGroup

.getElements();

while(marriagedElements.hasMoreElements()){

AbstractButton button=marriagedElements.nextElement();

if(button.isSelected()){

record.setMarriaged(button.getText());

break;

}

}

record.setPostalcode(postalcodeTextField.getText());

record.setAddress(addressTextField.getText());

Method[]methods=record.getClass().getMethods();

String recordMethodNames[]={"name","sex","tbNation",

"tbNativePlace;","idCard","schoolAge","specialty",

"partyMember","foreignLanguage","grade",

"marriaged","postalcode","address"};

for(int i=0;i<methods.length;i++){

String methodName=methods[i].getName();

if(methodName.startsWith("get")){

for(int m=0;m<recordMethodNames.length;m++){

if(methodName.substring(3).toLowerCase().equals(

recordMethodNames[m].toLowerCase())){

Object object=null;

try{

object=methods[i].invoke(record,null);

}catch(Exception e){

e.printStackTrace();

}

if(object==null){

System.out.println(methods[i].getName()

+"="+object);

JOptionPane.showMessageDialog(null,

JOptionPane.WARNING\_MESSAGE);

return;

}

}

}

}

}

String myBirthday=birthdayTextField.getText();

if(myBirthday.equals("YYYY-MM-DD")){

JOptionPane.showMessageDialog(null JOptionPane.WARNING\_MESSAGE);

return;

}else{

try{

Date birthday=Date.valueOf(myBirthday);

record.setBirthday(birthday);

}catch(RuntimeException e){

return;

}

}

if(photoLabel.getIcon()!=null){

File selectPhoto=new File(photoLabel.getIcon().toString());/

URL url=this.getClass().getResource("/personnel\_photo/"); StringBuffer uriBuffer=new StringBuffer(url.toString());

String selectPhotoName=selectPhoto.getName();

int i=selectPhotoName.lastIndexOf(".");

uriBuffer.append(recordNoTextField.getText());

uriBuffer.append(selectPhotoName.substring(i));

try{

File photo=new File(new URL(uriBuffer.toString())

.toURI());

record.setPhoto(photo.getName());

if(!photo.exists()){

photo.createNewFile();

}

InputStream inStream=new FileInputStream(selectPhoto); OutputStream outStream=new FileOutputStream(photo); int readBytes=0;

byte[]buffer=new byte[10240];

while((readBytes=inStream.read(buffer,0,10240))!=-1){ outStream.write(buffer,0,readBytes); }

outStream.close();

inStream.close();

}catch(Exception e){

e.printStackTrace();

}

}

TbPersonalInfo personalInfo=null;

if(UPDATE\_RECORD==null)

personalInfo=new TbPersonalInfo();

else

personalInfo=record.getTbPersonalInfo();

personalInfo.setHandset(handsetTextField.getText());

personalInfo.setTelephone(telephoneTextField.getText());

personalInfo.setQq(qqTextField.getText());

personalInfo.setEMail(emailTextField.getText());

personalInfo.setSecondSchoolAge(secondSchoolageTextField

.getText());

personalInfo.setSecondSpecialty(secondSpecialtyTextField

.getText());

String myGraduateDate=graduateDateTextField.getText();

if(!myGraduateDate.equals("YYYY-MM-DD")

&&myGraduateDate.length()!=0){

try{

Date date=Date.valueOf(myGraduateDate);

personalInfo.setGraduateDate(date);

}catch(RuntimeException e){

JOptionPane.showMessageDialog(null,"erro",

JOptionPane.WARNING\_MESSAGE);

return;

}

}

personalInfo.setGraduateSchool(graduateSchoolTextField

.getText());

personalInfo.setComputerGrade(computerTextField.getText());

personalInfo.setLikes(likeTextField.getText());

String myPartyMemberDate=partyMemberDateTextField.getText();

if(!myPartyMemberDate.equals("YYYY-MM-DD")

&&myPartyMemberDate.length()!=0){

try{

Date date=Date.valueOf(myPartyMemberDate);

personalInfo.setPartyMemberDate(date);

}catch(RuntimeException e){

JOptionPane.showMessageDialog(JOptionPane.WARNING\_MESSAGE);

return;

}

}

personalInfo.setOnesStrongSuit(onesStrongSuitTextField

.getText());

personalInfo.setPostalcode(homePostalcodeTextField.getText());

personalInfo.setAddress(homeAddressTextField.getText());

TbDutyInfo dutyInfo=null;

if(UPDATE\_RECORD==null)

dutyInfo=new TbDutyInfo();

else

dutyInfo=record.getTbDutyInfo();

TbDept dept=(TbDept)dao.queryDeptByName(deptTextField

.getText());

dutyInfo.setTbDept(dept);

TbDuty duty=(TbDuty)dao.queryDutyByName(dutyComboBox

.getSelectedItem().toString());

dutyInfo.setTbDuty(duty);

TbAccessionForm accessionForm=(TbAccessionForm)dao

.queryAccessionFormByName(accessionFormComboBox

.getSelectedItem().toString());

dutyInfo.setTbAccessionForm(accessionForm);

String myDimissionDate=dimissionDateTextField.getText();

if(!myDimissionDate.equals("YYYY-MM-DD")

&&myDimissionDate.length()!=0){

try{

Date date=Date.valueOf(myDimissionDate);

dutyInfo.setDimissionDate(date);

}catch(RuntimeException e){

JOptionPane.showMessageDialog(JOptionPane.WARNING\_MESSAGE);

return;

}

}

dutyInfo.setDimissionReason(dimissionReasonTextField.getText()); String myPactStartDate=pactStartDateTextField.getText();

if(!myPactStartDate.equals("YYYY-MM-DD")

&&myPactStartDate.length()!=0){

try{

Date date=Date.valueOf(myPactStartDate);

dutyInfo.setPactStartDate(date);

if(firstPactDateTextField.getText().length()==0)

dutyInfo.setFirstPactDate(date);

}catch(RuntimeException e){

JOptionPane.showMessageDialog(null,

JOptionPane.WARNING\_MESSAGE);

return;

}

}

String myPactEndDate=pactEndDateTextField.getText();

if(!myPactEndDate.equals("YYYY-MM-DD")

&&myPactEndDate.length()!=0){

try{

Date date=Date.valueOf(myPactEndDate);

dutyInfo.setPactEndDate(date);

}catch(RuntimeException e){

JOptionPane.showMessageDialog(null,

JOptionPane.WARNING\_MESSAGE);

return;

}

}

dutyInfo.setBankName(bankNameTextField.getText());

dutyInfo.setBankNo(bankNOTextField.getText());

dutyInfo.setAnnuitySafetyNo(annuitySafetyNOTextField.getText());

dutyInfo.setMedicareSafetyNo(medicareSafetyNOTextField

.getText());

dutyInfo.setCompoSafetyNo(compoSafetyNOTextField.getText());

dutyInfo.setDoleSafetyNo(doleSafetyNOTextField.getText());

dutyInfo.setSocietySafetyNo(societySafetyNOTextField.getText());

dutyInfo.setAccumulationFundNo(accumulationFundNOTextField

.getText());

methods=dutyInfo.getClass().getMethods();

String dutyInfoMethodNames[]={"tbDept","tbDuty",

"tbAccessionForm"};

for(int i=0;i<methods.length;i++){

String methodName=methods[i].getName();

if(methodName.startsWith("get")){

for(int m=0;m<dutyInfoMethodNames.length;m++){

if(methodName.substring(3).toLowerCase().equals(

dutyInfoMethodNames[m].toLowerCase())){

Object object=null;

try{

object=methods[i].invoke(dutyInfo,null);

System.out.println("method="+methods[i]);

System.out.println("object="+object);

}catch(Exception e){

e.printStackTrace();

}

if(object==null){

JOptionPane.showMessageDialog(null,

new String[]{

JOptionPane.WARNING\_MESSAGE);

return;

}

}

}

}

}

String myAccessionDate=accessionDateTextField.getText();

if(myAccessionDate.equals("YYYY-MM-DD")){

JOptionPane.showMessageDialog( JOptionPane.WARNING\_MESSAGE);

return;

}else{

try{

Date date=Date.valueOf(myAccessionDate);

dutyInfo.setAccessionDate(date);

}catch(RuntimeException e)

{

JOptionPane.showMessageDialog(JOptionPane.WARNING\_MESSAGE);

return;

}

}

if(UPDATE\_RECORD==null){

personalInfo.setTbRecord(record);

record.setTbPersonalInfo(personalInfo);

dutyInfo.setTbRecord(record);

record.setTbDutyInfo(dutyInfo);

dao.saveObject(record);

UPDATE\_RECORD=record;

}else{

dao.updateObject(record);

}

HibernateSessionFactory.closeSession();

JOptionPane.showMessageDialog(null, JOptionPane.INFORMATION\_MESSAGE);

}

});

saveButton.setText("save");

buttonPanel.add(saveButton);

final JButton exitButton=new JButton();

exitButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent arg0){

rightPanel.removeAll();

rightPanel.add(new RecordSelectedPanel(rightPanel));

serviceUtilities.updateComponentTreeUI(rightPanel);

}

});

exitButton.setText("exit");

buttonPanel.add(exitButton);

final JPanel contentPanel=new JPanel();

contentPanel.setBackground(Color.WHITE);

contentPanel.setBorder(new TitledBorder(null,"",

TitledBorder.DEFAULT\_JUSTIFICATION,

TitledBorder.DEFAULT\_POSITION,null,null));

contentPanel.setLayout(new incubator.service());

add(contentPanel,incubator.service.CENTER);

final JPanel recordNumPanel=new JPanel();

recordNumPanel.setBackground(Color.WHITE);

final FlowLayout flowLayout=new FlowLayout();

flowLayout.setAlignment(FlowLayout.LEFT);

recordNumPanel.setLayout(flowLayout);

contentPanel.add(recordNumPanel,incubator.service.NORTH);

final JLabel recordNoLabel=new JLabel();

recordNoLabel.setText("nub：");

recordNumPanel.add(recordNoLabel);

recordNoTextField=new JTextField();

recordNoTextField.setHorizontalAlignment(serviceConstants.CENTER);

recordNoTextField.setPreferredSize(new Dimension(125,20));

recordNoTextField.setEditable(false);

if(UPDATE\_RECORD==null){

String record=(String)dao.queryRecordOfMaxRecordNum();

if(record==null){

recordNoTextField.setText("T00001");

}else{

String recordNum=(Integer.valueOf(record.substring(1))+1)

+"";

StringBuffer id=new StringBuffer(6);

id.append(recordNum);

for(int i=0;i<5-recordNum.length();i++){

id.insert(0,'0');

}

id.insert(0,'T');

recordNoTextField.setText(id.toString());

}

}else{

recordNoTextField.setText(UPDATE\_RECORD.getRecordNumber());

}

recordNumPanel.add(recordNoTextField);

final JPanel infoPanel=new JPanel();

infoPanel.setLayout(new incubator.service());

infoPanel.setBackground(Color.WHITE);

contentPanel.add(infoPanel,incubator.service.CENTER);

final JPanel recordInfoPanel=new JPanel();

recordInfoPanel.setLayout(new incubator.service.out());

recordInfoPanel.setBorder(new TitledBorder(null,

TitledBorder.DEFAULT\_JUSTIFICATION,

TitledBorder.DEFAULT\_POSITION,null,null));

recordInfoPanel.setBackground(Color.WHITE);

infoPanel.add(recordInfoPanel,incubator.service.NORTH);

final JLabel nameLabel=new JLabel();

nameLabel.setText("");

final incubator.service.running incubator.service.running=new incubator.service.running();

incubator.service.running.insets=new Insets(5,0,10,0);

incubator.service.running.gridy=0;

incubator.service.running.gridx=0;

recordInfoPanel.add(nameLabel,incubator.service.running);

nameTextField=new JTextField();

nameTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

nameTextField.setText(UPDATE\_RECORD.getName());

final incubator.service.running incubator.service.running\_1=new incubator.service.running();

incubator.service.running\_1.insets=new Insets(5,0,10,20);

incubator.service.running\_1.gridy=0;

incubator.service.running\_1.gridx=1;

recordInfoPanel.add(nameTextField,incubator.service.running\_1);

final JLabel sexLabel=new JLabel();

sexLabel.setText();

final incubator.service.running incubator.service.running\_11=new incubator.service.running();

incubator.service.running\_11.insets=new Insets(5,0,10,0);

incubator.service.running\_11.gridy=0;

incubator.service.running\_11.gridx=2;

recordInfoPanel.add(sexLabel,incubator.service.running\_11);

final JRadioButton manRadioButton=new JRadioButton();

sexButtonGroup.add(manRadioButton);

manRadioButton.setBackground(Color.WHITE);

manRadioButton.setText("boy");

final incubator.service.running incubator.service.running\_3=new incubator.service.running();

incubator.service.running\_3.insets=new Insets(0,16,10,0);

incubator.service.running\_3.gridy=0;

incubator.service.running\_3.gridx=3;

recordInfoPanel.add(manRadioButton,incubator.service.running\_3);

final JRadioButton womanRadioButton=new JRadioButton();

sexButtonGroup.add(womanRadioButton);

womanRadioButton.setBackground(Color.WHITE);

womanRadioButton.setText("girl");

final incubator.service.running incubator.service.running\_95=new incubator.service.running();

incubator.service.running\_95.insets=new Insets(0,0,10,0);

incubator.service.running\_95.gridy=0;

incubator.service.running\_95.gridx=4;

recordInfoPanel.add(womanRadioButton,incubator.service.running\_95);

if(UPDATE\_RECORD!=null){

if(UPDATE\_RECORD.getSex().equals("boy"))

manRadioButton.setSelected(true);

else

womanRadioButton.setSelected(true);

}

final JLabel birthdayLabel=new JLabel();

birthdayLabel.setText("birthday");

final incubator.service.running incubator.service.running\_6=new incubator.service.running();

incubator.service.running\_6.insets=new Insets(5,0,10,0);

incubator.service.running\_6.gridy=0;

incubator.service.running\_6.gridx=5;

recordInfoPanel.add(birthdayLabel,incubator.service.running\_6);

birthdayTextField=new JTextField();

birthdayTextField.setHorizontalAlignment(serviceConstants.CENTER);

birthdayTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD==null){

birthdayTextField.setText("YYYY-MM-DD");

}else{

String date=UPDATE\_RECORD.getBirthday().toString();

date=date.substring(0,10);

birthdayTextField.setText(date);

}

final incubator.service.running incubator.service.running\_7=new incubator.service.running();

incubator.service.running\_7.gridwidth=2;

incubator.service.running\_7.insets=new Insets(5,0,10,0);

incubator.service.running\_7.gridy=0;

incubator.service.running\_7.gridx=6;

recordInfoPanel.add(birthdayTextField,incubator.service.running\_7);

final JLabel label\_2=new JLabel();

label\_2.setText("");

final incubator.service.running incubator.service.running\_93=new incubator.service.running();

incubator.service.running\_93.insets=new Insets(0,20,0,0);

incubator.service.running\_93.gridy=0;

incubator.service.running\_93.gridx=8;

recordInfoPanel.add(label\_2,incubator.service.running\_93);

photoLabel=new JLabel();

photoLabel.setHorizontalAlignment(serviceConstants.CENTER); photoLabel.setBorder(new TitledBorder(null,"",

TitledBorder.DEFAULT\_JUSTIFICATION,

TitledBorder.DEFAULT\_POSITION,null,null));

photoLabel.setPreferredSize(new Dimension(120,140));

if(UPDATE\_RECORD==null||UPDATE\_RECORD.getPhoto()==null){

URL url=this.getClass().getResource("/personnel\_photo/");

String photo=url.toString().substring(5)

+UPDATE\_RECORD.getPhoto();

photoLabel.setIcon(new ImageIcon(photo));

}

photoLabel.addincubator.management Listener(new incubator.management Adapter(){

public void incubator.management Clicked(incubator.management Event e){

if(e.getClickCount()==2){

JFileChooser fileChooser=new JFileChooser(); fileChooser.setFileFilter(new FileFilter(){

public String getDescription(){ return"img（.jpg;.gif）";

}

public boolean accept(File file){ if(file.isDirectory())

return true;

String fileName=file.getName()

.toLowerCase();

if(fileName.endsWith(".jpg")

||fileName

.endsWith(".gif")) return true;

return false;

}

});

int i=fileChooser.showOpenDialog(getParent()); if(i==fileChooser.APPROVE\_OPTION){

File file=fileChooser.getSelectedFile(); if(file!=null){

ImageIcon icon=new ImageIcon(file

.getAbsolutePath());

photoLabel.setText(null);

photoLabel.setIcon(icon);

}

}

}

}

});

final incubator.service.running incubator.service.running\_8=new incubator.service.running();

incubator.service.running\_8.insets=new Insets(0,0,0,0);

incubator.service.running\_8.gridheight=5;

incubator.service.running\_8.gridy=0;

incubator.service.running\_8.gridx=9;

recordInfoPanel.add(photoLabel,incubator.service.running\_8);

final JLabel label\_3=new JLabel();

label\_3.setText("");

final incubator.service.running incubator.service.running\_94=new incubator.service.running();

incubator.service.running\_94.gridy=0;

incubator.service.running\_94.gridx=10;

recordInfoPanel.add(label\_3,incubator.service.running\_94);

final JLabel nationLabel=new JLabel();

nationLabel.setText("The nationalists：");

final incubator.service.running incubator.service.running\_4=new incubator.service.running();

incubator.service.running\_4.insets=new Insets(0,0,10,0);

incubator.service.running\_4.gridy=1;

incubator.service.running\_4.gridx=0;

recordInfoPanel.add(nationLabel,incubator.service.running\_4);

nationComboBox=new JComboBox();

nationComboBox.addItem("The nationalists");

Iterator nationIt=dao.queryNation().iterator();

while(nationIt.hasNext()){

TbNation nation=(TbNation)nationIt.next();

nationComboBox.addItem(nation.getName());

}

if(UPDATE\_RECORD!=null){

String myNation=UPDATE\_RECORD.getTbNation().getName();

nationComboBox.setSelectedItem(myNation);

}

final incubator.service.running incubator.service.running\_5=new incubator.service.running();

incubator.service.running\_5.anchor=incubator.service.running.WEST;

incubator.service.running\_5.insets=new Insets(0,0,10,0);

incubator.service.running\_5.gridy=1;

incubator.service.running\_5.gridx=1;

recordInfoPanel.add(nationComboBox,incubator.service.running\_5);

final JLabel nativePlaceLabel=new JLabel();

nativePlaceLabel.setText("An injection：");

final incubator.service.running incubator.service.running\_2=new incubator.service.running();

incubator.service.running\_2.insets=new Insets(0,0,10,0);

incubator.service.running\_2.gridy=1;

incubator.service.running\_2.gridx=2;

recordInfoPanel.add(nativePlaceLabel,incubator.service.running\_2);

nativePlaceComboBox=new JComboBox();

nativePlaceComboBox.addItem("The nationalists");

Iterator nativePlaceIt=dao.queryNativePlace().iterator();

while(nativePlaceIt.hasNext()){

TbNativePlace nativePlace=(TbNativePlace)nativePlaceIt.next();

nativePlaceComboBox.addItem(nativePlace.getName());

}

if(UPDATE\_RECORD!=null){

String myNativePlace=UPDATE\_RECORD.getTbNativePlace().getName();

nativePlaceComboBox.setSelectedItem(myNativePlace);

}

final incubator.service.running incubator.service.running\_12=new incubator.service.running();

incubator.service.running\_12.anchor=incubator.service.running.WEST;

incubator.service.running\_12.gridwidth=2;

incubator.service.running\_12.insets=new Insets(0,0,10,0);

incubator.service.running\_12.gridy=1;

incubator.service.running\_12.gridx=3;

recordInfoPanel.add(nativePlaceComboBox,incubator.service.running\_12);

final JLabel idCardLabel=new JLabel();

idCardLabel.setText("ID");

final incubator.service.running incubator.service.running\_15=new incubator.service.running();

incubator.service.running\_15.insets=new Insets(0,0,10,0);

incubator.service.running\_15.gridy=1;

incubator.service.running\_15.gridx=5;

recordInfoPanel.add(idCardLabel,incubator.service.running\_15);

idCardTextField=new JTextField();

idCardTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

idCardTextField.setText(UPDATE\_RECORD.getIdCard());

final incubator.service.running incubator.service.running\_16=new incubator.service.running();

incubator.service.running\_16.gridwidth=2;

incubator.service.running\_16.insets=new Insets(0,0,10,0);

incubator.service.running\_16.gridy=1;

incubator.service.running\_16.gridx=6;

recordInfoPanel.add(idCardTextField,incubator.service.running\_16);

final JLabel schoolageLabel=new JLabel();

schoolageLabel.setText("Learn：");

final incubator.service.running incubator.service.running\_9=new incubator.service.running();

incubator.service.running\_9.insets=new Insets(0,0,10,0);

incubator.service.running\_9.gridy=2;

incubator.service.running\_9.gridx=0;

recordInfoPanel.add(schoolageLabel,incubator.service.running\_9);

schoolageComboBox=new JComboBox();

schoolageComboBox.addItem("The nationalists");

schoolageComboBox.addItem("High school");

schoolageComboBox.addItem("college");

schoolageComboBox.addItem("Undergraduat");

if(UPDATE\_RECORD!=null)

schoolageComboBox.setSelectedItem(UPDATE\_RECORD.getSchoolAge());

final incubator.service.running incubator.service.running\_10=new incubator.service.running();

incubator.service.running\_10.anchor=incubator.service.running.WEST;

incubator.service.running\_10.insets=new Insets(0,0,10,0);

incubator.service.running\_10.gridy=2;

incubator.service.running\_10.gridx=1;

recordInfoPanel.add(schoolageComboBox,incubator.service.running\_10);

final JLabel specialtyLabel=new JLabel();

specialtyLabel.setText();

final incubator.service.running incubator.service.running\_19=new incubator.service.running();

incubator.service.running\_19.insets=new Insets(0,0,10,0);

incubator.service.running\_19.gridy=2;

incubator.service.running\_19.gridx=2;

recordInfoPanel.add(specialtyLabel,incubator.service.running\_19);

specialtyTextField=new JTextField();

specialtyTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

specialtyTextField.setText(UPDATE\_RECORD.getSpecialty());

final incubator.service.running incubator.service.running\_20=new incubator.service.running();

incubator.service.running\_20.gridwidth=2;

incubator.service.running\_20.insets=new Insets(0,0,10,20);

incubator.service.running\_20.gridy=2;

incubator.service.running\_20.gridx=3;

recordInfoPanel.add(specialtyTextField,incubator.service.running\_20);

final JLabel partyMemberLabel=new JLabel();

final incubator.service.running incubator.service.running\_13=new incubator.service.running();

incubator.service.running\_13.insets=new Insets(0,0,10,0);

incubator.service.running\_13.gridy=2;

incubator.service.running\_13.gridx=5;

recordInfoPanel.add(partyMemberLabel,incubator.service.running\_13);

final JRadioButton isPartyMemberRadioButton=new JRadioButton();

partyMemberButtonGroup.add(isPartyMemberRadioButton);

isPartyMemberRadioButton.setBackground(Color.WHITE);

isPartyMemberRadioButton.setText("yes");

final incubator.service.running incubator.service.running\_222=new incubator.service.running();

incubator.service.running\_222.insets=new Insets(0,10,10,0);

incubator.service.running\_222.gridy=2;

incubator.service.running\_222.gridx=6;

recordInfoPanel.add(isPartyMemberRadioButton,incubator.service.running\_222);

final JRadioButton notPartyMemberRadioButton=new JRadioButton();

partyMemberButtonGroup.add(notPartyMemberRadioButton);

notPartyMemberRadioButton.setBackground(Color.WHITE);

notPartyMemberRadioButton.setSelected(true);

notPartyMemberRadioButton.setText("no");

final incubator.service.running incubator.service.running\_961=new incubator.service.running();

incubator.service.running\_961.insets=new Insets(0,0,10,0);

incubator.service.running\_961.gridy=2;

incubator.service.running\_961.gridx=7;

recordInfoPanel.add(notPartyMemberRadioButton,incubator.service.running\_961);

if(UPDATE\_RECORD!=null){

if(UPDATE\_RECORD.getPartyMember().equals("yes"))

isPartyMemberRadioButton.setSelected(true);

else

notPartyMemberRadioButton.setSelected(true);

}

final JLabel foreignLanguageLabel=new JLabel();

foreignLanguageLabel.setText("Foreign languages：");

final incubator.service.running incubator.service.running\_17=new incubator.service.running();

incubator.service.running\_17.insets=new Insets(0,0,10,0);

incubator.service.running\_17.gridy=3;

incubator.service.running\_17.gridx=0;

recordInfoPanel.add(foreignLanguageLabel,incubator.service.running\_17);

foreignLanguageTextField=new JTextField();

foreignLanguageTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

foreignLanguageTextField

.setText(UPDATE\_RECORD.getForeignLanguage());

final incubator.service.running incubator.service.running\_18=new incubator.service.running();

incubator.service.running\_18.insets=new Insets(0,0,10,20);

incubator.service.running\_18.gridy=3;

incubator.service.running\_18.gridx=1;

recordInfoPanel.add(foreignLanguageTextField,incubator.service.running\_18);

final JLabel gradeLabel=new JLabel();

final incubator.service.running incubator.service.running\_23=new incubator.service.running();

incubator.service.running\_23.insets=new Insets(0,0,10,0);

incubator.service.running\_23.gridy=3;

incubator.service.running\_23.gridx=2;

recordInfoPanel.add(gradeLabel,incubator.service.running\_23);

gradeTextField=new JTextField();

gradeTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

gradeTextField.setText(UPDATE\_RECORD.getGrade());

final incubator.service.running incubator.service.running\_24=new incubator.service.running();

incubator.service.running\_24.gridwidth=2;

incubator.service.running\_24.insets=new Insets(0,0,10,20);

incubator.service.running\_24.gridy=3;

incubator.service.running\_24.gridx=3;

recordInfoPanel.add(gradeTextField,incubator.service.running\_24);

final JLabel marriagedLabel=new JLabel();

final incubator.service.running incubator.service.running\_21=new incubator.service.running();

incubator.service.running\_21.insets=new Insets(0,0,10,0);

incubator.service.running\_21.gridy=3;

incubator.service.running\_21.gridx=5;

recordInfoPanel.add(marriagedLabel,incubator.service.running\_21);

final JRadioButton notMarriagedRadioButton=new JRadioButton();

marriagedButtonGroup.add(notMarriagedRadioButton);

notMarriagedRadioButton.setBackground(Color.WHITE);

notMarriagedRadioButton.setText("");

final incubator.service.running incubator.service.running\_22=new incubator.service.running();

incubator.service.running\_22.insets=new Insets(0,10,10,0);

incubator.service.running\_22.gridy=3;

incubator.service.running\_22.gridx=6;

recordInfoPanel.add(notMarriagedRadioButton,incubator.service.running\_22);

final JRadioButton hasMarriagedRadioButton=new JRadioButton();

marriagedButtonGroup.add(hasMarriagedRadioButton);

hasMarriagedRadioButton.setBackground(Color.WHITE);

hasMarriagedRadioButton.setText("");

final incubator.service.running incubator.service.running\_96=new incubator.service.running();

incubator.service.running\_96.insets=new Insets(0,0,10,0);

incubator.service.running\_96.gridy=3;

incubator.service.running\_96.gridx=7;

recordInfoPanel.add(hasMarriagedRadioButton,incubator.service.running\_96);

if(UPDATE\_RECORD!=null){

if(UPDATE\_RECORD.getMarriaged().equals(""))

notMarriagedRadioButton.setSelected(true);

else

hasMarriagedRadioButton.setSelected(true);

}

final JLabel postalcodeLabel=new JLabel();

final incubator.service.running incubator.service.running\_25=new incubator.service.running();

incubator.service.running\_25.insets=new Insets(0,0,5,0);

incubator.service.running\_25.gridy=4;

incubator.service.running\_25.gridx=0;

recordInfoPanel.add(postalcodeLabel,incubator.service.running\_25);

postalcodeTextField=new JTextField();

postalcodeTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

postalcodeTextField.setText(UPDATE\_RECORD.getPostalcode());

final incubator.service.running incubator.service.running\_26=new incubator.service.running();

incubator.service.running\_26.insets=new Insets(0,0,5,20);

incubator.service.running\_26.gridy=4;

incubator.service.running\_26.gridx=1;

recordInfoPanel.add(postalcodeTextField,incubator.service.running\_26);

final JLabel addressLabel=new JLabel();

addressLabel.setText("address：");

final incubator.service.running incubator.service.running\_27=new incubator.service.running();

incubator.service.running\_27.insets=new Insets(0,0,5,0);

incubator.service.running\_27.gridy=4;

incubator.service.running\_27.gridx=2;

recordInfoPanel.add(addressLabel,incubator.service.running\_27);

addressTextField=new JTextField();

addressTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

addressTextField.setText(UPDATE\_RECORD.getAddress());

final incubator.service.running incubator.service.running\_28=new incubator.service.running();

incubator.service.running\_28.insets=new Insets(0,0,5,0);

incubator.service.running\_28.ipadx=205;

incubator.service.running\_28.gridwidth=5;

incubator.service.running\_28.gridy=4;

incubator.service.running\_28.gridx=3;

recordInfoPanel.add(addressTextField,incubator.service.running\_28);

final JPanel personnelInfoLanel=new JPanel();

personnelInfoLanel.setLayout(new incubator.service.out());

personnelInfoLanel.setBorder(new TitledBorder(null, TitledBorder.DEFAULT\_JUSTIFICATION,

TitledBorder.DEFAULT\_POSITION,null,null));

personnelInfoLanel.setBackground(Color.WHITE);

infoPanel.add(personnelInfoLanel,incubator.service.SOUTH);

TbPersonalInfo personalInfo=null;

if(UPDATE\_RECORD!=null)

personalInfo=UPDATE\_RECORD.getTbPersonalInfo();

final JLabel handsetLabel=new JLabel();

handsetLabel.setText("mobile phone：");

final incubator.service.running incubator.service.running\_29=new incubator.service.running();

incubator.service.running\_29.insets=new Insets(5,0,10,0);

incubator.service.running\_29.gridy=0;

incubator.service.running\_29.gridx=0;

personnelInfoLanel.add(handsetLabel,incubator.service.running\_29);

handsetTextField=new JTextField();

handsetTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

handsetTextField.setText(personalInfo.getHandset());

final incubator.service.running incubator.service.running\_30=new incubator.service.running();

incubator.service.running\_30.insets=new Insets(5,0,10,20);

incubator.service.running\_30.gridy=0;

incubator.service.running\_30.gridx=1;

personnelInfoLanel.add(handsetTextField,incubator.service.running\_30);

final JLabel telephoneLabel=new JLabel();

telephoneLabel.setText("mobile phones");

final incubator.service.running incubator.service.running\_31=new incubator.service.running();

incubator.service.running\_31.insets=new Insets(5,0,10,0);

incubator.service.running\_31.gridy=0;

incubator.service.running\_31.gridx=2;

personnelInfoLanel.add(telephoneLabel,incubator.service.running\_31);

telephoneTextField=new JTextField();

telephoneTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

telephoneTextField.setText(personalInfo.getTelephone());

final incubator.service.running incubator.service.running\_32=new incubator.service.running();

incubator.service.running\_32.insets=new Insets(5,0,10,20);

incubator.service.running\_32.gridy=0;

incubator.service.running\_32.gridx=3;

personnelInfoLanel.add(telephoneTextField,incubator.service.running\_32);

final JLabel qqLabel=new JLabel();

qqLabel.setText("QQ：");

final incubator.service.running incubator.service.running\_33=new incubator.service.running();

incubator.service.running\_33.insets=new Insets(5,0,10,0);

incubator.service.running\_33.gridy=0;

incubator.service.running\_33.gridx=4;

personnelInfoLanel.add(qqLabel,incubator.service.running\_33);

qqTextField=new JTextField();

qqTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

qqTextField.setText(personalInfo.getQq());

final incubator.service.running incubator.service.running\_34=new incubator.service.running();

incubator.service.running\_34.insets=new Insets(5,0,10,20);

incubator.service.running\_34.gridy=0;

incubator.service.running\_34.gridx=5;

personnelInfoLanel.add(qqTextField,incubator.service.running\_34);

final JLabel emailLabel=new JLabel();

emailLabel.setText("E\_mail：");

final incubator.service.running incubator.service.running\_35=new incubator.service.running();

incubator.service.running\_35.insets=new Insets(5,0,10,0);

incubator.service.running\_35.gridy=0;

incubator.service.running\_35.gridx=6;

personnelInfoLanel.add(emailLabel,incubator.service.running\_35);

emailTextField=new JTextField();

emailTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

emailTextField.setText(personalInfo.getEMail());

final incubator.service.running incubator.service.running\_36=new incubator.service.running();

incubator.service.running\_36.insets=new Insets(5,0,10,0);

incubator.service.running\_36.gridy=0;

incubator.service.running\_36.gridx=7;

personnelInfoLanel.add(emailTextField,incubator.service.running\_36);

final JLabel secondSchoolageLabel=new JLabel();

final incubator.service.running incubator.service.running\_37=new incubator.service.running();

incubator.service.running\_37.insets=new Insets(0,0,10,0);

incubator.service.running\_37.gridy=1;

incubator.service.running\_37.gridx=0;

personnelInfoLanel.add(secondSchoolageLabel,incubator.service.running\_37);

secondSchoolageTextField=new JTextField();

secondSchoolageTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

secondSchoolageTextField.setText(personalInfo.getSecondSchoolAge());

final incubator.service.running incubator.service.running\_42=new incubator.service.running();

incubator.service.running\_42.insets=new Insets(0,0,10,20);

incubator.service.running\_42.gridy=1;

incubator.service.running\_42.gridx=1;

personnelInfoLanel.add(secondSchoolageTextField,incubator.service.running\_42);

final JLabel secondSpecialtyLabel=new JLabel();

final incubator.service.running incubator.service.running\_38=new incubator.service.running();

incubator.service.running\_38.insets=new Insets(0,0,10,0);

incubator.service.running\_38.gridy=1;

incubator.service.running\_38.gridx=2;

personnelInfoLanel.add(secondSpecialtyLabel,incubator.service.running\_38);

secondSpecialtyTextField=new JTextField();

secondSpecialtyTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

secondSpecialtyTextField.setText(personalInfo.getSecondSpecialty());

final incubator.service.running incubator.service.running\_43=new incubator.service.running();

incubator.service.running\_43.insets=new Insets(0,0,10,20);

incubator.service.running\_43.gridy=1;

incubator.service.running\_43.gridx=3;

personnelInfoLanel.add(secondSpecialtyTextField,incubator.service.running\_43);

final JLabel graduateDateLabel=new JLabel();

final incubator.service.running incubator.service.running\_39=new incubator.service.running();

incubator.service.running\_39.insets=new Insets(0,0,10,0);

incubator.service.running\_39.gridy=1;

incubator.service.running\_39.gridx=4;

personnelInfoLanel.add(graduateDateLabel,incubator.service.running\_39);

graduateDateTextField=new JTextField();

graduateDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

graduateDateTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD==null||personalInfo.getGraduateDate()==null){

graduateDateTextField.setText("YYYY-MM-DD");

}else{

String date=personalInfo.getGraduateDate().toString();

date=date.substring(0,10);

graduateDateTextField.setText(date);

}

final incubator.service.running incubator.service.running\_44=new incubator.service.running();

incubator.service.running\_44.insets=new Insets(0,0,10,20);

incubator.service.running\_44.gridy=1;

incubator.service.running\_44.gridx=5;

personnelInfoLanel.add(graduateDateTextField,incubator.service.running\_44);

final JLabel graduateSchoolLabel=new JLabel();

final incubator.service.running incubator.service.running\_40=new incubator.service.running();

incubator.service.running\_40.insets=new Insets(0,0,10,0);

incubator.service.running\_40.gridy=1;

incubator.service.running\_40.gridx=6;

personnelInfoLanel.add(graduateSchoolLabel,incubator.service.running\_40);

graduateSchoolTextField=new JTextField();

graduateSchoolTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

graduateSchoolTextField.setText(personalInfo.getGraduateSchool());

final incubator.service.running incubator.service.running\_45=new incubator.service.running();

incubator.service.running\_45.insets=new Insets(0,0,10,0);

incubator.service.running\_45.gridy=1;

incubator.service.running\_45.gridx=7;

personnelInfoLanel.add(graduateSchoolTextField,incubator.service.running\_45);

final JLabel computerLabel=new JLabel();

final incubator.service.running incubator.service.running\_46=new incubator.service.running();

incubator.service.running\_46.insets=new Insets(0,0,10,0);

incubator.service.running\_46.gridy=2;

incubator.service.running\_46.gridx=0;

personnelInfoLanel.add(computerLabel,incubator.service.running\_46);

computerTextField=new JTextField();

computerTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

computerTextField.setText(personalInfo.getComputerGrade());

final incubator.service.running incubator.service.running\_41=new incubator.service.running();

incubator.service.running\_41.insets=new Insets(0,0,10,20);

incubator.service.running\_41.gridy=2;

incubator.service.running\_41.gridx=1;

personnelInfoLanel.add(computerTextField,incubator.service.running\_41);

final JLabel likeLabel=new JLabel();

final incubator.service.running incubator.service.running\_47=new incubator.service.running();

incubator.service.running\_47.insets=new Insets(0,0,10,0);

incubator.service.running\_47.gridy=2;

incubator.service.running\_47.gridx=2;

personnelInfoLanel.add(likeLabel,incubator.service.running\_47);

likeTextField=new JTextField();

likeTextField.setPreferredSize(new Dimension(195,20));

if(UPDATE\_RECORD!=null)

likeTextField.setText(personalInfo.getLikes());

final incubator.service.running incubator.service.running\_54=new incubator.service.running();

incubator.service.running\_54.insets=new Insets(0,0,10,0);

incubator.service.running\_54.ipadx=340;

incubator.service.running\_54.gridwidth=5;

incubator.service.running\_54.gridy=2;

incubator.service.running\_54.gridx=3;

personnelInfoLanel.add(likeTextField,incubator.service.running\_54);

final JLabel partyMemberDateLabel=new JLabel();

final incubator.service.running incubator.service.running\_50=new incubator.service.running();

incubator.service.running\_50.insets=new Insets(0,0,10,0);

incubator.service.running\_50.gridy=3;

incubator.service.running\_50.gridx=0;

personnelInfoLanel.add(partyMemberDateLabel,incubator.service.running\_50);

partyMemberDateTextField=new JTextField();

partyMemberDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

partyMemberDateTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD==null||personalInfo.getPartyMemberDate()==null){

partyMemberDateTextField.setText("YYYY-MM-DD");

}else{

String date=personalInfo.getPartyMemberDate().toString();

date=date.substring(0,10);

partyMemberDateTextField.setText(date);

}

final incubator.service.running incubator.service.running\_52=new incubator.service.running();

incubator.service.running\_52.insets=new Insets(0,0,10,20);

incubator.service.running\_52.gridy=3;

incubator.service.running\_52.gridx=1;

personnelInfoLanel.add(partyMemberDateTextField,incubator.service.running\_52);

final JLabel onesStrongSuitLabel=new JLabel();

final incubator.service.running incubator.service.running\_48=new incubator.service.running();

incubator.service.running\_48.insets=new Insets(0,0,10,0);

incubator.service.running\_48.gridy=3;

incubator.service.running\_48.gridx=2;

personnelInfoLanel.add(onesStrongSuitLabel,incubator.service.running\_48);

onesStrongSuitTextField=new JTextField();

onesStrongSuitTextField.setPreferredSize(new Dimension(195,20));

if(UPDATE\_RECORD!=null)

onesStrongSuitTextField.setText(personalInfo.getOnesStrongSuit());

final incubator.service.running incubator.service.running\_55=new incubator.service.running();

incubator.service.running\_55.insets=new Insets(0,0,10,0);

incubator.service.running\_55.ipadx=340;

incubator.service.running\_55.gridwidth=5;

incubator.service.running\_55.gridy=3;

incubator.service.running\_55.gridx=3;

personnelInfoLanel.add(onesStrongSuitTextField,incubator.service.running\_55);

final JLabel homePostalcodeLabel=new JLabel();

final incubator.service.running incubator.service.running\_51=new incubator.service.running();

incubator.service.running\_51.insets=new Insets(0,0,5,0);

incubator.service.running\_51.gridy=4;

incubator.service.running\_51.gridx=0;

personnelInfoLanel.add(homePostalcodeLabel,incubator.service.running\_51);

homePostalcodeTextField=new JTextField();

homePostalcodeTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

homePostalcodeTextField.setText(personalInfo.getPostalcode());

final incubator.service.running incubator.service.running\_53=new incubator.service.running();

incubator.service.running\_53.insets=new Insets(0,0,5,20);

incubator.service.running\_53.gridy=4;

incubator.service.running\_53.gridx=1;

personnelInfoLanel.add(homePostalcodeTextField,incubator.service.running\_53);

final JLabel homeAddressLabel=new JLabel();

final incubator.service.running incubator.service.running\_49=new incubator.service.running();

incubator.service.running\_49.insets=new Insets(0,0,5,0);

incubator.service.running\_49.gridy=4;

incubator.service.running\_49.gridx=2;

personnelInfoLanel.add(homeAddressLabel,incubator.service.running\_49);

homeAddressTextField=new JTextField();

homeAddressTextField.setPreferredSize(new Dimension(195,20));

if(UPDATE\_RECORD!=null)

homeAddressTextField.setText(personalInfo.getAddress());

final incubator.service.running incubator.service.running\_56=new incubator.service.running();

incubator.service.running\_56.insets=new Insets(0,0,5,0);

incubator.service.running\_56.ipadx=340;

incubator.service.running\_56.gridwidth=5;

incubator.service.running\_56.gridy=4;

incubator.service.running\_56.gridx=3;

personnelInfoLanel.add(homeAddressTextField,incubator.service.running\_56);

final JPanel dutyInfoPanel=new JPanel();

dutyInfoPanel.setLayout(new incubator.service.out());

dutyInfoPanel.setBorder(new TitledBorder(null

TitledBorder.DEFAULT\_JUSTIFICATION,

TitledBorder.DEFAULT\_POSITION,null,null));

dutyInfoPanel.setBackground(Color.WHITE);

infoPanel.add(dutyInfoPanel);

TbDutyInfo dutyInfo=null;

if(UPDATE\_RECORD!=null)

dutyInfo=UPDATE\_RECORD.getTbDutyInfo();

final JLabel deptLabel=new JLabel();

final incubator.service.running incubator.service.running\_57=new incubator.service.running();

incubator.service.running\_57.insets=new Insets(5,0,10,0);

incubator.service.running\_57.gridy=0;

incubator.service.running\_57.gridx=0;

dutyInfoPanel.add(deptLabel,incubator.service.running\_57);

deptTextField=new JTextField();

deptTextField.setEditable(false);

deptTextField.setColumns(14);

if(UPDATE\_RECORD!=null)

deptTextField.setText(dutyInfo.getTbDept().getName());

final incubator.service.running incubator.service.running\_58=new incubator.service.running();

incubator.service.running\_58.gridy=0;

incubator.service.running\_58.gridx=1;

dutyInfoPanel.add(deptTextField,incubator.service.running\_58);

final JButton deptTreeButton=new JButton();

deptTreeButton.setMargin(new Insets(0,6,0,3));

deptTreeButton.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e){

DeptTreeDialog deptTree=new DeptTreeDialog(deptTextField);

deptTree.setBounds(252,437,125,175);

deptTree.setVisible(true);

}

});

deptTreeButton.setText("...");

final incubator.service.running incubator.service.running\_97=new incubator.service.running();

incubator.service.running\_97.insets=new Insets(0,0,0,20);

incubator.service.running\_97.gridy=0;

incubator.service.running\_97.gridx=2;

dutyInfoPanel.add(deptTreeButton,incubator.service.running\_97);

final JLabel dutyLabel=new JLabel();

final incubator.service.running incubator.service.running\_59=new incubator.service.running();

incubator.service.running\_59.insets=new Insets(5,0,10,0);

incubator.service.running\_59.gridy=0;

incubator.service.running\_59.gridx=3;

dutyInfoPanel.add(dutyLabel,incubator.service.running\_59);

dutyComboBox=new JComboBox();

dutyComboBox.addItem("The nationalists");

Iterator dutyIt=dao.queryDuty().iterator();

while(dutyIt.hasNext()){

TbDuty duty=(TbDuty)dutyIt.next();

dutyComboBox.addItem(duty.getName());

}

if(UPDATE\_RECORD!=null)

dutyComboBox.setSelectedItem(dutyInfo.getTbDuty().getName());

final incubator.service.running incubator.service.running\_60=new incubator.service.running();

incubator.service.running\_60.insets=new Insets(0,0,5,0);

incubator.service.running\_60.anchor=incubator.service.running.WEST;

incubator.service.running\_60.gridy=0;

incubator.service.running\_60.gridx=4;

dutyInfoPanel.add(dutyComboBox,incubator.service.running\_60);

final JLabel accessionDateLabel=new JLabel();

final incubator.service.running incubator.service.running\_61=new incubator.service.running();

incubator.service.running\_61.insets=new Insets(0,0,10,0);

incubator.service.running\_61.gridy=1;

incubator.service.running\_61.gridx=0;

dutyInfoPanel.add(accessionDateLabel,incubator.service.running\_61);

accessionDateTextField=new JTextField();

accessionDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

accessionDateTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD==null){

accessionDateTextField.setText("YYYY-MM-DD");

}else{

String date=dutyInfo.getAccessionDate().toString();

date=date.substring(0,10);

accessionDateTextField.setText(date);

}

final incubator.service.running incubator.service.running\_70=new incubator.service.running();

incubator.service.running\_70.gridwidth=2;

incubator.service.running\_70.insets=new Insets(0,0,10,20);

incubator.service.running\_70.gridy=1;

incubator.service.running\_70.gridx=1;

dutyInfoPanel.add(accessionDateTextField,incubator.service.running\_70);

final JLabel accessionFormLabel=new JLabel();

final incubator.service.running incubator.service.running\_65=new incubator.service.running();

incubator.service.running\_65.insets=new Insets(0,0,10,0);

incubator.service.running\_65.gridy=1;

incubator.service.running\_65.gridx=3;

dutyInfoPanel.add(accessionFormLabel,incubator.service.running\_65);

accessionFormComboBox=new JComboBox();

accessionFormComboBox.addItem("The nationalists");

Iterator accessionFormIt=dao.queryAccessionForm().iterator();

while(accessionFormIt.hasNext()){

TbAccessionForm accessionForm=(TbAccessionForm)accessionFormIt

.next();

accessionFormComboBox.addItem(accessionForm.getName());

}

if(UPDATE\_RECORD!=null)

accessionFormComboBox.setSelectedItem(dutyInfo.getTbAccessionForm()

.getName());

HibernateSessionFactory.closeSession();

final incubator.service.running incubator.service.running\_76=new incubator.service.running();

incubator.service.running\_76.anchor=incubator.service.running.WEST;

incubator.service.running\_76.insets=new Insets(0,0,5,0);

incubator.service.running\_76.gridy=1;

incubator.service.running\_76.gridx=4;

dutyInfoPanel.add(accessionFormComboBox,incubator.service.running\_76);

final JLabel dimissionDateLabel=new JLabel();

final incubator.service.running incubator.service.running\_69=new incubator.service.running();

incubator.service.running\_69.insets=new Insets(0,0,10,0);

incubator.service.running\_69.gridy=1;

incubator.service.running\_69.gridx=5;

dutyInfoPanel.add(dimissionDateLabel,incubator.service.running\_69);

dimissionDateTextField=new JTextField();

dimissionDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

dimissionDateTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD==null||dutyInfo.getDimissionDate()==null){

dimissionDateTextField.setText("YYYY-MM-DD");

}else{

String date=dutyInfo.getDimissionDate().toString();

date=date.substring(0,10);

dimissionDateTextField.setText(date);

}

final incubator.service.running incubator.service.running\_73=new incubator.service.running();

incubator.service.running\_73.insets=new Insets(0,0,10,20);

incubator.service.running\_73.gridy=1;

incubator.service.running\_73.gridx=6;

dutyInfoPanel.add(dimissionDateTextField,incubator.service.running\_73);

final JLabel dimissionReasonLabel=new JLabel();

final incubator.service.running incubator.service.running\_85=new incubator.service.running();

incubator.service.running\_85.insets=new Insets(0,0,10,0);

incubator.service.running\_85.gridy=1;

incubator.service.running\_85.gridx=7;

dutyInfoPanel.add(dimissionReasonLabel,incubator.service.running\_85);

dimissionReasonTextField=new JTextField();

dimissionReasonTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

dimissionReasonTextField.setText(dutyInfo.getDimissionReason());

final incubator.service.running incubator.service.running\_86=new incubator.service.running();

incubator.service.running\_86.insets=new Insets(0,0,10,0);

incubator.service.running\_86.gridy=1;

incubator.service.running\_86.gridx=8;

dutyInfoPanel.add(dimissionReasonTextField,incubator.service.running\_86);

final JLabel pactStartDateLabel=new JLabel();

final incubator.service.running incubator.service.running\_80=new incubator.service.running();

incubator.service.running\_80.insets=new Insets(0,0,10,0);

incubator.service.running\_80.gridy=2;

incubator.service.running\_80.gridx=0;

dutyInfoPanel.add(pactStartDateLabel,incubator.service.running\_80);

pactStartDateTextField=new JTextField();

pactStartDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

pactStartDateTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD==null||dutyInfo.getPactStartDate()==null){

pactStartDateTextField.setText("YYYY-MM-DD");

}else{

String date=dutyInfo.getPactStartDate().toString();

date=date.substring(0,10);

pactStartDateTextField.setText(date);

}

final incubator.service.running incubator.service.running\_74=new incubator.service.running();

incubator.service.running\_74.gridwidth=2;

incubator.service.running\_74.insets=new Insets(0,0,10,20);

incubator.service.running\_74.gridy=2;

incubator.service.running\_74.gridx=1;

dutyInfoPanel.add(pactStartDateTextField,incubator.service.running\_74);

final JLabel pactEndDateLabel=new JLabel();

final incubator.service.running incubator.service.running\_87=new incubator.service.running();

incubator.service.running\_87.insets=new Insets(0,0,10,0);

incubator.service.running\_87.gridy=2;

incubator.service.running\_87.gridx=3;

dutyInfoPanel.add(pactEndDateLabel,incubator.service.running\_87);

pactEndDateTextField=new JTextField();

pactEndDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

pactEndDateTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD==null||dutyInfo.getPactEndDate()==null){

pactEndDateTextField.setText("YYYY-MM-DD");

}else{

String date=dutyInfo.getPactEndDate().toString();

date=date.substring(0,10);

pactEndDateTextField.setText(date);

}

final incubator.service.running incubator.service.running\_90=new incubator.service.running();

incubator.service.running\_90.insets=new Insets(0,0,10,20);

incubator.service.running\_90.gridy=2;

incubator.service.running\_90.gridx=4;

dutyInfoPanel.add(pactEndDateTextField,incubator.service.running\_90);

final JLabel firstPactDateLabel=new JLabel();

final incubator.service.running incubator.service.running\_62=new incubator.service.running();

incubator.service.running\_62.insets=new Insets(0,0,10,0);

incubator.service.running\_62.gridy=2;

incubator.service.running\_62.gridx=5;

dutyInfoPanel.add(firstPactDateLabel,incubator.service.running\_62);

firstPactDateTextField=new JTextField();

firstPactDateTextField.setEditable(false);

firstPactDateTextField.setHorizontalAlignment(serviceConstants.CENTER);

firstPactDateTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null&&dutyInfo.getFirstPactDate()!=null){

String date=dutyInfo.getFirstPactDate().toString();

date=date.substring(0,10);

firstPactDateTextField.setText(date);

}

final incubator.service.running incubator.service.running\_71=new incubator.service.running();

incubator.service.running\_71.insets=new Insets(0,0,10,20);

incubator.service.running\_71.gridy=2;

incubator.service.running\_71.gridx=6;

dutyInfoPanel.add(firstPactDateTextField,incubator.service.running\_71);

final JLabel pactAgeLabel=new JLabel();

final incubator.service.running incubator.service.running\_66=new incubator.service.running();

incubator.service.running\_66.insets=new Insets(0,0,10,0);

incubator.service.running\_66.gridy=2;

incubator.service.running\_66.gridx=7;

dutyInfoPanel.add(pactAgeLabel,incubator.service.running\_66);

pactAgeTextField=new JTextField();

pactAgeTextField.setHorizontalAlignment(serviceConstants.CENTER);

pactAgeTextField.setEditable(false);

pactAgeTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null&&dutyInfo.getFirstPactDate()!=null){

String date=dutyInfo.getFirstPactDate().toString();

System.out.println(date);

String year=date.substring(0,4);

System.out.println(year);

String month=date.substring(5,7);

System.out.println(month);

int pactAge=Today.YEAR-Integer.valueOf(year);

if(Today.MONTH<=Integer.valueOf(month))

pactAge-=1;

pactAgeTextField.setText(pactAge+"");

}

final incubator.service.running incubator.service.running\_77=new incubator.service.running();

incubator.service.running\_77.insets=new Insets(0,0,10,0);

incubator.service.running\_77.gridy=2;

incubator.service.running\_77.gridx=8;

dutyInfoPanel.add(pactAgeTextField,incubator.service.running\_77);

final JLabel bankNameLabel=new JLabel();

final incubator.service.running incubator.service.running\_63=new incubator.service.running();

incubator.service.running\_63.insets=new Insets(0,0,10,0);

incubator.service.running\_63.gridy=3;

incubator.service.running\_63.gridx=0;

dutyInfoPanel.add(bankNameLabel,incubator.service.running\_63);

bankNameTextField=new JTextField();

bankNameTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

bankNameTextField.setText(dutyInfo.getBankName());

final incubator.service.running incubator.service.running\_72=new incubator.service.running();

incubator.service.running\_72.gridwidth=2;

incubator.service.running\_72.insets=new Insets(0,0,10,20);

incubator.service.running\_72.gridy=3;

incubator.service.running\_72.gridx=1;

dutyInfoPanel.add(bankNameTextField,incubator.service.running\_72);

final JLabel societySafetyNOLabel=new JLabel();

final incubator.service.running incubator.service.running\_67=new incubator.service.running();

incubator.service.running\_67.insets=new Insets(0,0,10,0);

incubator.service.running\_67.gridy=3;

incubator.service.running\_67.gridx=3;

dutyInfoPanel.add(societySafetyNOLabel,incubator.service.running\_67);

societySafetyNOTextField=new JTextField();

societySafetyNOTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

societySafetyNOTextField.setText(dutyInfo.getSocietySafetyNo());

final incubator.service.running incubator.service.running\_78=new incubator.service.running();

incubator.service.running\_78.insets=new Insets(0,0,10,20);

incubator.service.running\_78.gridy=3;

incubator.service.running\_78.gridx=4;

dutyInfoPanel.add(societySafetyNOTextField,incubator.service.running\_78);

final JLabel doleSafetyNOLabel=new JLabel();

final incubator.service.running incubator.service.running\_81=new incubator.service.running();

incubator.service.running\_81.insets=new Insets(0,0,10,0);

incubator.service.running\_81.gridy=3;

incubator.service.running\_81.gridx=5;

dutyInfoPanel.add(doleSafetyNOLabel,incubator.service.running\_81);

doleSafetyNOTextField=new JTextField();

doleSafetyNOTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

doleSafetyNOTextField.setText(dutyInfo.getDoleSafetyNo());

final incubator.service.running incubator.service.running\_83=new incubator.service.running();

incubator.service.running\_83.insets=new Insets(0,0,10,20);

incubator.service.running\_83.gridy=3;

incubator.service.running\_83.gridx=6;

dutyInfoPanel.add(doleSafetyNOTextField,incubator.service.running\_83);

final JLabel annuitySafetyNOLabel=new JLabel();

annuitySafetyNOLabel.setText();

final incubator.service.running incubator.service.running\_88=new incubator.service.running();

incubator.service.running\_88.insets=new Insets(0,0,10,0);

incubator.service.running\_88.gridy=3;

incubator.service.running\_88.gridx=7;

dutyInfoPanel.add(annuitySafetyNOLabel,incubator.service.running\_88);

annuitySafetyNOTextField=new JTextField();

annuitySafetyNOTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

annuitySafetyNOTextField.setText(dutyInfo.getAnnuitySafetyNo());

final incubator.service.running incubator.service.running\_91=new incubator.service.running();

incubator.service.running\_91.insets=new Insets(0,0,10,0);

incubator.service.running\_91.gridy=3;

incubator.service.running\_91.gridx=8;

dutyInfoPanel.add(annuitySafetyNOTextField,incubator.service.running\_91);

final JLabel bankNOLabel=new JLabel();

final incubator.service.running incubator.service.running\_64=new incubator.service.running();

incubator.service.running\_64.insets=new Insets(0,0,5,0);

incubator.service.running\_64.gridy=4;

incubator.service.running\_64.gridx=0;

dutyInfoPanel.add(bankNOLabel,incubator.service.running\_64);

bankNOTextField=new JTextField();

bankNOTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

bankNOTextField.setText(dutyInfo.getBankNo());

final incubator.service.running incubator.service.running\_75=new incubator.service.running();

incubator.service.running\_75.gridwidth=2;

incubator.service.running\_75.insets=new Insets(0,0,5,20);

incubator.service.running\_75.gridy=4;

incubator.service.running\_75.gridx=1;

dutyInfoPanel.add(bankNOTextField,incubator.service.running\_75);

final JLabel medicareSafetyNOLabel=new JLabel();

final incubator.service.running incubator.service.running\_68=new incubator.service.running();

incubator.service.running\_68.insets=new Insets(0,0,5,0);

incubator.service.running\_68.gridy=4;

incubator.service.running\_68.gridx=3;

dutyInfoPanel.add(medicareSafetyNOLabel,incubator.service.running\_68);

medicareSafetyNOTextField=new JTextField();

medicareSafetyNOTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

medicareSafetyNOTextField.setText(dutyInfo.getMedicareSafetyNo());

final incubator.service.running incubator.service.running\_79=new incubator.service.running();

incubator.service.running\_79.insets=new Insets(0,0,5,20);

incubator.service.running\_79.gridy=4;

incubator.service.running\_79.gridx=4;

dutyInfoPanel.add(medicareSafetyNOTextField,incubator.service.running\_79);

final JLabel compoSafetyNOLabel=new JLabel();

final incubator.service.running incubator.service.running\_82=new incubator.service.running();

incubator.service.running\_82.insets=new Insets(0,0,5,0);

incubator.service.running\_82.gridy=4;

incubator.service.running\_82.gridx=5;

dutyInfoPanel.add(compoSafetyNOLabel,incubator.service.running\_82);

compoSafetyNOTextField=new JTextField();

compoSafetyNOTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

compoSafetyNOTextField.setText(dutyInfo.getCompoSafetyNo());

final incubator.service.running incubator.service.running\_84=new incubator.service.running();

incubator.service.running\_84.insets=new Insets(0,0,5,20);

incubator.service.running\_84.gridy=4;

incubator.service.running\_84.gridx=6;

dutyInfoPanel.add(compoSafetyNOTextField,incubator.service.running\_84);

final JLabel accumulationFundNOLabel=new JLabel();

accumulationFundNOLabel.setText();

final incubator.service.running incubator.service.running\_89=new incubator.service.running();

incubator.service.running\_89.insets=new Insets(0,0,5,0);

incubator.service.running\_89.gridy=4;

incubator.service.running\_89.gridx=7;

dutyInfoPanel.add(accumulationFundNOLabel,incubator.service.running\_89);

accumulationFundNOTextField=new JTextField();

accumulationFundNOTextField.setPreferredSize(new Dimension(125,20));

if(UPDATE\_RECORD!=null)

accumulationFundNOTextField.setText(dutyInfo

.getAccumulationFundNo());

final incubator.service.running incubator.service.running\_92=new incubator.service.running();

incubator.service.running\_92.insets=new Insets(0,0,5,0);

incubator.service.running\_92.gridy=4;

incubator.service.running\_92.gridx=8;

dutyInfoPanel.add(accumulationFundNOTextField,incubator.service.running\_92);

}

}