代码清单

目录

[代码清单 1](#_Toc452293496)

[1 目录划分 3](#_Toc452293497)

[2 每个包里面所有的类以及用处 3](#_Toc452293498)

[2.1 cn.list.start 程序打开包 3](#_Toc452293499)

[2.2 cn.list.control 对于程序内部控制包 3](#_Toc452293502)

[2.3 cn.list.model 程序实体定义包，包括get与set方法 3](#_Toc452293503)

[2.4 cn.list.ui UI包 3](#_Toc452293504)

[2.5 cn.list.util 数据库连接包 3](#_Toc452293505)

[2.6 cn.list.workinterface 接口包 3](#_Toc452293506)

[3 具体代码 3](#_Toc452293507)

# 目录划分

cn.list.start 程序打开包

cn.list.control 对于程序内部控制包

cn.list.model 程序实体定义包，包括get与set方法

cn.list.ui UI包

cn.list.util 数据库连接包

cn.list.workinterface 接口包

# 每个包里面所有的类以及用处

* 1. cn.list.start 程序打开包

Start.java打开程序 9行

2. 1. cn.list.control 对于程序内部控制包

CountDown倒计时 111行

EventManage事件控制 338行

EventManageTest 事件控制的单元测试 278行

* 1. cn.list.model 程序实体定义包，包括get与set方法

Event 实体定义 101行

* 1. cn.list.ui UI包

MainUi 主界面 470行

NewEventUi 新建界面 270行

ModifyEventUi 修改界面 270行

DelEventUi 删除界面67行

ClearEventUi 清空界面63行

ErrorHint错误提示界面 38行

FinishHint 结束提示界面 105行

* 1. cn.list.util 数据库连接包

DBUtil 数据库连接21行

DbException 数据库提示包8行

BusinessException 异常包8行

BaseException异常包8行

* 1. cn.list.workinterface 接口包

IEvent 事件接口包20行

# 具体代码

cn/list/astart/start.java

package cn.list.astart;

import cn.list.ui.MainUi;

public class start {

public static void main(String[] args) {

MainUi start = new MainUi();

// start.initialize();

}

}

cn/list/control/EventManage.java

package cn.list.control;

import java.sql.Connection;

import java.util.Date;

import java.sql.SQLException;

import java.sql.Timestamp;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

import cn.list.util.\*;

import cn.list.model.Event;

import cn.list.workinterface.IEvent;

public class EventManage implements IEvent {

@Override

public void CreateEvent(String Name,Date BeginTime,Date EndTime,boolean Hint,String describe,int level) throws BusinessException, DbException {

// 创建新事件

if(Name==null){

throw new BusinessException("必须写名字");

}

if(BeginTime==null){

throw new BusinessException("必须写起始时间");

}

if(EndTime==null){

throw new BusinessException("必须写结束时间");

}

if(BeginTime.before(EndTime)==false)

{

throw new BusinessException("结束必须在起始时间之前");

}

Event event=new Event();

event.setName(Name);

event.setBeginTime(BeginTime);

event.setEndTime(EndTime);

event.setHint(Hint);

event.setComplete(false);

event.setDescribe(describe);

event.setLevel(level);

event.setDel(false);

event.setChange(false);

EventManage a=new EventManage();

a.AddEvent(event);

}

@Override

public void AddEvent(Event event) throws DbException {

// TODO Auto-generated method stub

Connection conn=null;

try {

conn=DBUtil.getConnection();

int id=0;

String sql="select count(\*) from [Event]";

java.sql.PreparedStatement pst=conn.prepareStatement(sql);

java.sql.ResultSet rs=pst.executeQuery();

while(rs.next())

{

id=rs.getInt(1)+1;

}

rs.close();

pst.close();

sql="INSERT INTO [work].[dbo].[Event] ([ID],[Name],[BeginTime],[EndTime],[Hint],[Complete],"

+ "[describe],[level],[del],[change]) VALUES(?,?,?,?,?,?,?,?,?,?)";

pst=conn.prepareStatement(sql);

pst.setInt(1, id);

pst.setString(2,event.getName());

pst.setTimestamp(3, new Timestamp(event.getBeginTime().getTime()));

pst.setTimestamp(4, new Timestamp(event.getEndTime().getTime()));

if(event.isHint()==true){

pst.setInt(5, 1);

}

else{

pst.setInt(5, 0);

}

pst.setInt(6, 0);

pst.setString(7,event.getDescribe());

pst.setInt(8,event.getLevel());

pst.setInt(9,0);

pst.setInt(10,0);

// rs.close();

pst.execute();

pst.close();

} catch (SQLException e) {

e.printStackTrace();

throw new DbException(e);

}

finally{

if(conn!=null)

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

@Override

public Event SerchEvent(int ID) throws BusinessException, DbException {

List<Event> TotalEvent=new ArrayList<Event>();

Connection conn=null;

int hint;

int complete;

int del;

try {

conn=DBUtil.getConnection();

String sql="SELECT [ID],[Name],[BeginTime],[EndTime],[Hint],[Complete],[describe],[level],[del]"

+"FROM [work].[dbo].[Event]"+

"where id=?";

java.sql.PreparedStatement pst=conn.prepareStatement(sql);

pst.setInt(1,ID);

java.sql.ResultSet rs=pst.executeQuery();

if(!rs.next())

throw new BusinessException("不存在");

Event event=new Event();

event.setID(ID);

event.setName(rs.getString(2));

Timestamp beginendtime=rs.getTimestamp(3);

java.util.Date begin=new java.util.Date(beginendtime.getTime());

event.setBeginTime(begin);

Timestamp sqlendtime=rs.getTimestamp(4);

java.util.Date end=new java.util.Date(sqlendtime.getTime());

event.setEndTime(end);

hint=rs.getInt(5);

if (hint==0){

event.setHint(false);

}

else{

event.setHint(true);

}

complete=rs.getInt(6);

if (complete==0){

event.setComplete(false);

}

else{

event.setComplete(true);

}

event.setDescribe(rs.getString(7));

event.setLevel(rs.getInt(8));

del=rs.getInt(9);

if (del==0){

event.setComplete(false);

}

else{

event.setComplete(true);

}

rs.close();

pst.execute();

pst.close();

return event;

} catch (SQLException e) {

e.printStackTrace();

throw new DbException(e);

}

finally{

if(conn!=null)

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

@Override

public void DelEvent(Event event) throws BusinessException, DbException, SQLException {

// TODO Auto-generated method stub

EventManage a=new EventManage();

if(a.SerchEvent(event.getID())!=null){

Connection conn=null;

conn=DBUtil.getConnection();

String sql="UPDATE [work].[dbo].[Event]SET [del] = 1 WHERE id=?";

java.sql.PreparedStatement pst=conn.prepareStatement(sql);

pst.setInt(1,event.getID());

pst.execute();

pst.close();

}

}

@Override

public void ModifyEvent(Event event,Event newevent)throws BusinessException, DbException, SQLException {

// TODO Auto-generated method stub

EventManage a=new EventManage();

if(a.SerchEvent(event.getID())!=null){

Connection conn=null;

conn=DBUtil.getConnection();

String sql="UPDATE [work].[dbo].[Event]"

+" SET [Name] = ? ,"

+ "[BeginTime] =? ,"

+ "[EndTime] = ?,"

+ "[Hint] = ?,"

+ "[describe] =?,"

+ "[level] = ?"

+" WHERE id=?";

java.sql.PreparedStatement pst=conn.prepareStatement(sql);

pst.setString(1,newevent.getName());

pst.setTimestamp(2, new Timestamp(newevent.getBeginTime().getTime()));

pst.setTimestamp(3, new Timestamp(newevent.getEndTime().getTime()));

if(newevent.isHint()==true){

pst.setInt(4, 1);

}

else{

pst.setInt(4, 0);

}

pst.setString(5,newevent.getDescribe());

pst.setInt(6,newevent.getLevel());

pst.setInt(7,event.getID());

pst.execute();

pst.close();

}

}

@Override

public List<Event> LoadEvent() throws BusinessException, DbException {

List<Event> TotalEvent=new ArrayList<Event>();

Connection conn=null;

int hint;

int complete;

int del;

try {

conn=DBUtil.getConnection();

String sql="SELECT [ID],[Name],[BeginTime],[EndTime],[Hint],[Complete],[describe],[level],[del]"

+"FROM [work].[dbo].[Event]";

java.sql.PreparedStatement pst=conn.prepareStatement(sql);

java.sql.ResultSet rs=pst.executeQuery();

// if(!rs.next())

// return null;

while(rs.next()){

Event event=new Event();

event.setID(rs.getInt(1));

event.setName(rs.getString(2));

Timestamp beginendtime=rs.getTimestamp(3);

java.util.Date begin=new java.util.Date(beginendtime.getTime());

event.setBeginTime(begin);

Timestamp sqlendtime=rs.getTimestamp(4);

java.util.Date end=new java.util.Date(sqlendtime.getTime());

event.setEndTime(end);

hint=rs.getInt(5);

if (hint==0){

event.setHint(false);

}

else{

event.setHint(true);

}

complete=rs.getInt(6);

if (complete==0){

event.setComplete(false);

}

else{

event.setComplete(true);

}

event.setDescribe(rs.getString(7));

event.setLevel(rs.getInt(8));

del=rs.getInt(9);

if (del==0){

event.setComplete(false);

}

else{

event.setComplete(true);

}

TotalEvent.add(event);

}

rs.close();

pst.execute();

pst.close();

return TotalEvent;

} catch (SQLException e) {

e.printStackTrace();

throw new DbException(e);

}

finally{

if(conn!=null)

try {

conn.close();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

public void ClearEvent()throws BusinessException, DbException, SQLException {

Connection conn=null;

conn=DBUtil.getConnection();

String sql="DELETE FROM [work].[dbo].[Event] ";

java.sql.PreparedStatement pst=conn.prepareStatement(sql);

pst.execute();

pst.close();

}

}

cn/list/control/EventManageTest.java

package cn.list.control;

import static org.junit.Assert.\*;

import java.sql.SQLException;

import java.util.Date;

import org.junit.Assert;

import org.junit.Test;

import cn.list.model.Event;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

public class EventManageTest {

//正常运行

Event event=new Event();

EventManage test=new EventManage();

@Test

public void testCreateEvent() {

Date begindate=new Date();

Date enddate=new Date();

enddate.setYear(2020);

try {

test.CreateEvent("单元测试样例", begindate, enddate, true,"仅仅是测试", 1);

} catch (BusinessException e) {

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

@Test

public void testCreateEvent1() {

//缺少名字

Date begindate=new Date();

Date enddate=new Date();

enddate.setYear(2020);

try {

test.CreateEvent(null, begindate, enddate, true,"仅仅是测试", 1);

} catch (BusinessException e) {

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

fail("Not yet implemented");

}

@Test

public void testCreateEvent2() {

//缺少起始时间

Date begindate=new Date();

Date enddate=new Date();

enddate.setYear(2020);

try {

test.CreateEvent("单元测试样例", null, enddate, true,"仅仅是测试", 1);

} catch (BusinessException e) {

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

fail("Not yet implemented");

}

@Test

public void testCreateEvent3() {

//缺少结束时间

Date begindate=new Date();

Date enddate=new Date();

enddate.setYear(2020);

try {

test.CreateEvent("单元测试样例", begindate, null, true,"仅仅是测试", 1);

} catch (BusinessException e) {

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

fail("Not yet implemented");

}

@Test

public void testCreateEvent4() {

//时间之前

Date begindate=new Date();

Date enddate=new Date();

try {

test.CreateEvent("单元测试样例", begindate, enddate, true,"仅仅是测试", 1);

} catch (BusinessException e) {

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

fail("Not yet implemented");

}

@Test

public void testSerchEvent() {

//正常查询

try {

Event event =test.SerchEvent(1);

System.out.println(event.getID());

System.out.println(event.getName());

System.out.println(event.getBeginTime());

System.out.println(event.getEndTime());

System.out.println(event.getDescribe());

} catch (BusinessException e) {

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

@Test

public void testSerchEvent1() {

//找不到

try {

test.SerchEvent(0);

} catch (BusinessException e) {

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

fail("Not yet implemented");

}

@Test

public void testLoadEvent() {

try {

test.LoadEvent();

} catch (BusinessException e) {

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

@Test

public void testClearEvent() throws SQLException {

try {

test.ClearEvent();

} catch (BusinessException e) {

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

cn/list/control/Fortest.java

package cn.list.control;

import java.util.Date;

import cn.list.ui.MainUi;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

public class Fortest {

public static void main(String[] args) throws BusinessException, DbException {

Date begin=new Date();

Date end=new Date();

end.setYear(2020);

EventManage test=new EventManage();

for(int i=1;i<10;i++)

test.CreateEvent("测试样例"+i, begin, end, true, "这只是一个测试", 2);

}

public static void test() throws BusinessException, DbException {

Date begin=new Date();

Date end=new Date();

end.setYear(2020);

EventManage test=new EventManage();

for(int i=1;i<10;i++)

test.CreateEvent("测试样例"+i, begin, end, true, "这只是一个测试", 2);

}

}

cn/list/model/DataChange.java

package cn.list.model;

import java.sql.Date;

public class DataChange {

//java.util.Date curDate = new java.util.Date();//新建一个util类型的date

//java.sql.Date date = new java.sql.Date(curDate.getTime());//进行日期的转换

public java.sql.Date UtilChangSql(java.util.Date UtilData){

java.sql.Date sqldate=new java.sql.Date(UtilData.getTime());

return sqldate;

}

public java.util.Date SQlChangeUtil(java.sql.Date SqlDate){

java.util.Date Utildate=new java.sql.Date(SqlDate.getTime());

return Utildate;

}

}

cn/list/model/Event.java

package cn.list.model;

import java.util.Date;

import java.util.List;

public class Event {

private int Totalnum;//数据总数

private int ID;//事件编号

private String Name;//事件名称；

private Date BeginTime;//起始时间

private Date EndTime;//结束时间

private boolean Hint;//提示标识符

private boolean Complete;//完成标识符

private String Place;//地点描述

private String Character;//人物描述

private String describe;//事件描述

private int level;//优先级

private boolean del;//删除标识符

private boolean change;//修改标识符

public static boolean flag=true;//运行标识符

public int getTotalnum() {

return Totalnum;

}

public void setTotalnum(int totalnum) {

Totalnum = totalnum;

}

public int getID() {

return ID;

}

public void setID(int iD) {

ID = iD;

}

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public Date getBeginTime() {

return BeginTime;

}

public void setBeginTime(Date beginTime) {

BeginTime = beginTime;

}

public Date getEndTime() {

return EndTime;

}

public void setEndTime(Date endTime) {

EndTime = endTime;

}

public boolean isHint() {

return Hint;

}

public void setHint(boolean hint) {

Hint = hint;

}

public boolean isComplete() {

return Complete;

}

public void setComplete(boolean complete) {

Complete = complete;

}

public String getPlace() {

return Place;

}

public void setPlace(String place) {

Place = place;

}

public String getCharacter() {

return Character;

}

public void setCharacter(String character) {

Character = character;

}

public String getDescribe() {

return describe;

}

public void setDescribe(String describe) {

this.describe = describe;

}

public int getLevel() {

return level;

}

public void setLevel(int level) {

this.level = level;

}

public boolean isDel() {

return del;

}

public void setDel(boolean del) {

this.del = del;

}

public boolean isChange() {

return change;

}

public void setChange(boolean change) {

this.change = change;

}

}

cn/list/model/Tip.java

package cn.list.model;

import java.sql.Date;

public class Tip {

private int Totalnum;//数据总数

private int ID;//事件编号

private String Name;//事件名称；

private Date BeginTime;//起始时间

private Date EndTime;//结束时间

private boolean Hint;//提示标识符

private boolean Complete;//完成标识符

private String describe;//事件描述

private boolean del;//删除标识符

private boolean change;//修改标识符

public int getTotalnum() {

return Totalnum;

}

public void setTotalnum(int totalnum) {

Totalnum = totalnum;

}

public int getID() {

return ID;

}

public void setID(int iD) {

ID = iD;

}

public String getName() {

return Name;

}

public void setName(String name) {

Name = name;

}

public Date getBeginTime() {

return BeginTime;

}

public void setBeginTime(Date beginTime) {

BeginTime = beginTime;

}

public Date getEndTime() {

return EndTime;

}

public void setEndTime(Date endTime) {

EndTime = endTime;

}

public boolean isHint() {

return Hint;

}

public void setHint(boolean hint) {

Hint = hint;

}

public boolean isComplete() {

return Complete;

}

public void setComplete(boolean complete) {

Complete = complete;

}

public String getDescribe() {

return describe;

}

public void setDescribe(String describe) {

this.describe = describe;

}

public boolean isDel() {

return del;

}

public void setDel(boolean del) {

this.del = del;

}

public boolean isChange() {

return change;

}

public void setChange(boolean change) {

this.change = change;

}

}

cn/list/ui/ClearEventUi.java

package cn.list.ui;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.SQLException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.SwingConstants;

import cn.list.control.EventManage;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

public class ClearEventUi {

private JFrame frame;

public void ClearEventUi() {

frame = new JFrame();

frame.setBounds(100, 100, 300, 200);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("ÊÇ·ñÇå¿Õ£¿");

lblNewLabel.setFont(new Font("SimSun", Font.BOLD, 30));

lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel.setBounds(70, 23, 161, 37);

frame.getContentPane().add(lblNewLabel);

JButton btnNewButton = new JButton("È·¶¨");

btnNewButton.setBounds(35, 120, 93, 23);

frame.getContentPane().add(btnNewButton);

btnNewButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

EventManage eventManage=new EventManage();

try {

eventManage.ClearEvent();

} catch (BusinessException | DbException | SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

frame.setVisible(false);

}

});

JButton btnNewButton\_1 = new JButton("\u53D6\u6D88");

btnNewButton\_1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

frame.setVisible(false);

}

});

btnNewButton\_1.setBounds(171, 120, 93, 23);

frame.getContentPane().add(btnNewButton\_1);

frame.setVisible(true);

}}

cn/list/ui/DelEventUi.java

package cn.list.ui;

import java.awt.EventQueue;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.SQLException;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.SwingConstants;

import cn.list.control.EventManage;

import cn.list.model.Event;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

public class DelEventUi {

private JFrame frame;

public void DelEventUi(final Event event) {

frame = new JFrame();

frame.setBounds(100, 100, 300, 200);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("\u662F\u5426\u5220\u9664");

lblNewLabel.setFont(new Font("SimSun", Font.BOLD, 30));

lblNewLabel.setHorizontalAlignment(SwingConstants.CENTER);

lblNewLabel.setBounds(70, 23, 161, 37);

frame.getContentPane().add(lblNewLabel);

JButton btnNewButton = new JButton("È·¶¨");

btnNewButton.setBounds(35, 120, 93, 23);

frame.getContentPane().add(btnNewButton);

JButton btnNewButton\_1 = new JButton("\u53D6\u6D88");

btnNewButton\_1.setBounds(171, 120, 93, 23);

frame.getContentPane().add(btnNewButton\_1);

frame.setVisible(true);

btnNewButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

EventManage a=new EventManage();

try {

a.DelEvent(event);

} catch (BusinessException | DbException | SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

frame.setVisible(false);

}

});

btnNewButton\_1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

frame.setVisible(false);

}

});

}

}

cn/list/ui/MainUi.java

package cn.list.ui;

import java.awt.EventQueue;

import javax.swing.JFrame;

import javax.swing.JMenu;

import javax.swing.JMenuBar;

import javax.swing.JMenuItem;

import javax.swing.JOptionPane;

import javax.swing.JPanel;

import java.awt.BorderLayout;

import java.awt.GridBagLayout;

import javax.swing.JButton;

import java.awt.GridBagConstraints;

import java.awt.Insets;

import java.awt.event.ActionListener;

import java.util.List;

import java.util.Vector;

import java.awt.event.ActionEvent;

import javax.swing.JTextField;

import java.awt.FlowLayout;

import javax.swing.JSplitPane;

import javax.swing.JTabbedPane;

import javax.swing.JScrollPane;

import javax.swing.JLayeredPane;

import java.awt.Color;

import java.awt.GridLayout;

import javax.swing.JLabel;

import javax.swing.JToolBar;

import javax.swing.RootPaneContainer;

import javax.swing.table.DefaultTableModel;

import javax.swing.table.TableModel;

import cn.list.control.EventManage;

import cn.list.control.Fortest;

import cn.list.model.Event;

import cn.list.ui.\*;

import cn.list.util.BaseException;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

import cn.list.waste.ToDolistamain;

import javax.swing.JTextPane;

import javax.swing.JTable;

public class MainUi {

private JFrame frame;

public JTable table\_1;

//

Object tblEventData[][];

List<Event> allEvent=null;

DefaultTableModel tabStepModel=new DefaultTableModel();

Vector<Vector<Object>> eventData = new Vector<Vector<Object>>();

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

DefaultTableModel tablmod=new DefaultTableModel();

Object tblData[][];

Object tblTitle[];

JButton button\_1;

public void reload() throws BaseException{

EventManage load =new EventManage();

try {

allEvent=load.LoadEvent();

} catch (BusinessException | DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

Object tblTitle[]={"ID","姓名","起始时间","结束时间","描述","是否提示","优先级"};

if(allEvent.size()<=15)

tblData =new Object[16][7];

else

tblData =new Object[allEvent.size()][7];

int i=0;

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

if(allEvent.get(j).isDel()==false&&allEvent.get(j).isComplete()==false){

tblData[i][0]=allEvent.get(j).getID();

tblData[i][1]=allEvent.get(j).getName();

tblData[i][2]=allEvent.get(j).getBeginTime();

tblData[i][3]=allEvent.get(j).getEndTime();

tblData[i][4]=allEvent.get(j).getDescribe();

tblData[i][5]=allEvent.get(j).isHint();

tblData[i][6]=allEvent.get(j).getLevel();

i++;

}

}

tablmod.setDataVector(tblData,tblTitle);

table\_1.validate();

table\_1.repaint();

}

public static void main(String[] args) {

EventQueue.invokeLater(new Runnable() {

public void run() {

try {

MainUi window = new MainUi();

window.frame.setVisible(true);

} catch (Exception e) {

e.printStackTrace();

}

}

});

}

/\*\*

\* Create the application.

\* @wbp.parser.entryPoint

\*/

public MainUi() {

initialize();

}

public void initialize() {

frame = new JFrame("timetable");

frame.setBounds(200, 200, 450, 450);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.getContentPane().setLayout(new BorderLayout(0, 0));

JPanel panel = new JPanel();

frame.getContentPane().add(panel, BorderLayout.EAST);

GridBagLayout gbl\_panel = new GridBagLayout();

gbl\_panel.columnWidths = new int[]{0, 0, 0, 0, 0, 10, 0};

gbl\_panel.rowHeights = new int[]{10, 0, 0, 0, 0, 0, 0, 0, 0, 0};

gbl\_panel.columnWeights = new double[]{1.0, 0.0, 1.0, 1.0, 0.0, 0.0, Double.MIN\_VALUE};

gbl\_panel.rowWeights = new double[]{0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, Double.MIN\_VALUE};

panel.setLayout(gbl\_panel);

JButton b1 = new JButton("\u65B0\u5EFA");

GridBagConstraints gbc\_b1 = new GridBagConstraints();

gbc\_b1.insets = new Insets(0, 0, 5, 5);

gbc\_b1.gridx = 4;

gbc\_b1.gridy = 2;

panel.add(b1, gbc\_b1);

JButton btnNewButton\_1 = new JButton("修改");

btnNewButton\_1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

}

});

GridBagConstraints gbc\_btnNewButton\_1 = new GridBagConstraints();

gbc\_btnNewButton\_1.insets = new Insets(0, 0, 5, 5);

gbc\_btnNewButton\_1.gridx = 4;

gbc\_btnNewButton\_1.gridy = 3;

panel.add(btnNewButton\_1, gbc\_btnNewButton\_1);

JButton btnNewButton\_2 = new JButton("删除");

GridBagConstraints gbc\_btnNewButton\_2 = new GridBagConstraints();

gbc\_btnNewButton\_2.insets = new Insets(0, 0, 5, 5);

gbc\_btnNewButton\_2.gridx = 4;

gbc\_btnNewButton\_2.gridy = 4;

panel.add(btnNewButton\_2, gbc\_btnNewButton\_2);

JButton btnNewButton\_3 = new JButton("清空");

GridBagConstraints gbc\_btnNewButton\_3 = new GridBagConstraints();

gbc\_btnNewButton\_3.insets = new Insets(0, 0, 5, 5);

gbc\_btnNewButton\_3.gridx = 4;

gbc\_btnNewButton\_3.gridy = 5;

panel.add(btnNewButton\_3, gbc\_btnNewButton\_3);

button\_1 = new JButton("\u5237\u65B0\u5B9E\u9A8C");

GridBagConstraints gbc\_button\_1 = new GridBagConstraints();

gbc\_button\_1.insets = new Insets(0, 0, 5, 5);

gbc\_button\_1.gridx = 4;

gbc\_button\_1.gridy = 6;

panel.add(button\_1, gbc\_button\_1);

JPanel panel\_4 = new JPanel();

GridBagConstraints gbc\_panel\_4 = new GridBagConstraints();

gbc\_panel\_4.gridwidth = 3;

gbc\_panel\_4.insets = new Insets(0, 0, 5, 5);

gbc\_panel\_4.fill = GridBagConstraints.BOTH;

gbc\_panel\_4.gridx = 0;

gbc\_panel\_4.gridy = 7;

panel.add(panel\_4, gbc\_panel\_4);

JButton button = new JButton("\u6DFB\u52A0\u6D4B\u8BD5\u6570\u636E");

GridBagConstraints gbc\_button = new GridBagConstraints();

gbc\_button.insets = new Insets(0, 0, 5, 5);

gbc\_button.gridx = 4;

gbc\_button.gridy = 7;

panel.add(button, gbc\_button);

JPanel panel\_1 = new JPanel();

frame.getContentPane().add(panel\_1, BorderLayout.SOUTH);

GridBagLayout gbl\_panel\_1 = new GridBagLayout();

gbl\_panel\_1.columnWidths = new int[]{47, 0, 0, 0, 0, 10, 0, 0, 0};

gbl\_panel\_1.rowHeights = new int[]{0, 0, 0, 20, 0, 0, 36, 0, 0, 10, 0};

gbl\_panel\_1.columnWeights = new double[]{1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 1.0, Double.MIN\_VALUE};

gbl\_panel\_1.rowWeights = new double[]{1.0, 1.0, 1.0, 0.0, 0.0, 1.0, 1.0, 0.0, 0.0, 1.0, Double.MIN\_VALUE};

panel\_1.setLayout(gbl\_panel\_1);

JButton btnNewButton\_4 = new JButton("\u751F\u6210\u4FBF\u7B7E");

btnNewButton\_4.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

}

});

GridBagConstraints gbc\_btnNewButton\_4 = new GridBagConstraints();

gbc\_btnNewButton\_4.insets = new Insets(0, 0, 5, 5);

gbc\_btnNewButton\_4.gridx = 1;

gbc\_btnNewButton\_4.gridy = 3;

panel\_1.add(btnNewButton\_4, gbc\_btnNewButton\_4);

JButton btnNewButton\_5 = new JButton("\u4FBF\u7B7E\u7BA1\u7406");

btnNewButton\_5.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

}

});

GridBagConstraints gbc\_btnNewButton\_5 = new GridBagConstraints();

gbc\_btnNewButton\_5.insets = new Insets(0, 0, 5, 5);

gbc\_btnNewButton\_5.gridx = 2;

gbc\_btnNewButton\_5.gridy = 3;

panel\_1.add(btnNewButton\_5, gbc\_btnNewButton\_5);

JButton btnNewButton\_6 = new JButton("\u8BBE\u7F6E");

GridBagConstraints gbc\_btnNewButton\_6 = new GridBagConstraints();

gbc\_btnNewButton\_6.insets = new Insets(0, 0, 5, 5);

gbc\_btnNewButton\_6.gridx = 3;

gbc\_btnNewButton\_6.gridy = 3;

panel\_1.add(btnNewButton\_6, gbc\_btnNewButton\_6);

JButton btnNewButton\_7 = new JButton(" \u5173\u4E8E ");

GridBagConstraints gbc\_btnNewButton\_7 = new GridBagConstraints();

gbc\_btnNewButton\_7.insets = new Insets(0, 0, 5, 5);

gbc\_btnNewButton\_7.gridx = 4;

gbc\_btnNewButton\_7.gridy = 3;

panel\_1.add(btnNewButton\_7, gbc\_btnNewButton\_7);

JPanel panel\_3 = new JPanel();

GridBagConstraints gbc\_panel\_3 = new GridBagConstraints();

gbc\_panel\_3.gridwidth = 8;

gbc\_panel\_3.gridheight = 5;

gbc\_panel\_3.insets = new Insets(0, 0, 5, 0);

gbc\_panel\_3.fill = GridBagConstraints.BOTH;

gbc\_panel\_3.gridx = 0;

gbc\_panel\_3.gridy = 4;

panel\_1.add(panel\_3, gbc\_panel\_3);

JPanel panel\_2 = new JPanel();

GridBagConstraints gbc\_panel\_2 = new GridBagConstraints();

gbc\_panel\_2.gridwidth = 8;

gbc\_panel\_2.fill = GridBagConstraints.BOTH;

gbc\_panel\_2.gridx = 0;

gbc\_panel\_2.gridy = 9;

panel\_1.add(panel\_2, gbc\_panel\_2);

JTabbedPane tp = new JTabbedPane(JTabbedPane.TOP);

frame.getContentPane().add(tp, BorderLayout.CENTER);

JPanel contentPene1 =new JPanel(); //文字中层ok

contentPene1.setBackground(Color.WHITE);

JPanel contentPene4 =new JPanel(); //按钮中层ok

//中层over

//表格

table\_1=new JTable(tablmod);

try {

this.reload();

} catch (BaseException e3) {

e3.printStackTrace();

}

JScrollPane scrollPane = new JScrollPane(table\_1);

scrollPane.setBackground(Color.white);

tp.addTab("按优先级显示", null, scrollPane, null);

tp.addTab("p1", contentPene1);

contentPene1.setLayout(new BorderLayout(0, 0));

JTextPane txtpnAsdfasfd = new JTextPane();

txtpnAsdfasfd.setText("asdfasfd");

contentPene1.add(txtpnAsdfasfd, BorderLayout.NORTH);

tp.setEnabledAt(0,true);

tp.setTitleAt(1," 按日显示 ");

// tp.setPreferredSize(new Dimension(100,200));

tp.setTabPlacement(JTabbedPane.TOP);

tp.setTabLayoutPolicy(JTabbedPane.WRAP\_TAB\_LAYOUT);

JScrollPane scrollPane\_1 = new JScrollPane();

tp.addTab("按星期显示", null, scrollPane\_1, null);

JMenuBar menu=new JMenuBar();

frame.setJMenuBar(menu);

JMenu m1=new JMenu("file");

JMenu m2=new JMenu("edit");

JMenu m3=new JMenu("help");

menu.add(m1);

menu.add(m2);

menu.add(m3);

JMenuItem mi1=new JMenuItem("new");

JMenuItem mi2=new JMenuItem("open");

JMenuItem mi3=new JMenuItem("close");

JMenuItem mi4=new JMenuItem("save");

JMenuItem mi5=new JMenuItem("save as");

JMenuItem mi6=new JMenuItem("quit");

m1.add(mi1);

m1.add(mi2);

m1.add(mi3);

m1.addSeparator();

m1.add(mi4);

m1.add(mi5);

m1.addSeparator();

m1.add(mi6);

mi1.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e) {

ToDolistamain b=new ToDolistamain();

b.todilist();

}

});

frame.setVisible(true);

//以下是按键触发事件

b1.addActionListener(new ActionListener(){

public void actionPerformed(ActionEvent e) {

NewEventUi a=new NewEventUi();

a.newEventui();

try {

reload();

} catch (BaseException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

}

});

btnNewButton\_1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

ModifyEventUi a=new ModifyEventUi();

EventManage load =new EventManage();

int selectRows=table\_1.getSelectedRow();

if(selectRows==-1)

try {

throw new BusinessException("必须选中一个事件");

} catch (BusinessException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

int eventid=(int) table\_1.getValueAt(selectRows, 0) ;

Event event = null;

try {

event = load.SerchEvent(eventid);

} catch (BusinessException | DbException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

a.ModifyEventUi(event);

try {

reload();

} catch (BaseException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

}

});

btnNewButton\_2.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

System.out.println(1);

DelEventUi a=new DelEventUi();

EventManage load =new EventManage();

int selectRows=table\_1.getSelectedRow();

if(selectRows==-1)

try {

throw new BusinessException("必须选中一个事件");

} catch (BusinessException e2) {

e2.printStackTrace();

}

int eventid=(int) table\_1.getValueAt(selectRows, 0) ;

System.out.println(2);

Event event = null;

try {

event = load.SerchEvent(eventid);

System.out.println(3);

} catch (BusinessException | DbException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

a.DelEventUi(event);

System.out.println(4);

try {

reload();

} catch (BaseException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

}

});

btnNewButton\_3.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

ClearEventUi a=new ClearEventUi();

a.ClearEventUi();

try {

reload();

} catch (BaseException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

}

});

button.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

Fortest a=new Fortest();

try {

a.test();

try {

reload();

} catch (BaseException e2) {

// TODO Auto-generated catch block

e2.printStackTrace();

}

} catch (BusinessException | DbException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

button\_1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

try {

reload();

} catch (BaseException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

}

}

cn/list/ui/ModifyEventUi.java

package cn.list.ui;

import java.awt.BorderLayout;

import java.awt.Dimension;

import java.awt.FlowLayout;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.SQLException;

import java.util.Date;

import java.util.Locale;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import javax.swing.JComboBox;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JRadioButton;

import javax.swing.JScrollPane;

import javax.swing.JTextArea;

import javax.swing.JTextField;

import com.eltima.components.ui.DatePicker;

import cn.list.control.EventManage;

import cn.list.model.Event;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

public class ModifyEventUi {

private JFrame frame;

private JPanel north;

private JLabel label;

private JPanel south;

private JLabel lblNewLabel;

private JButton btnNewButton;

private JLabel lblNewLabel\_1;

private JButton btnNewButton\_1;

private JTextField nametext;

private JTextField begintimetext;

private JTextField endtimetext;

private JLabel level;

private DatePicker datepickbegintime;

private DatePicker datepickendtime;

private static final String DefaultFormat = "yyyy-MM-dd HH:mm:ss";

private Font font=new Font("Times New Roman", Font.BOLD, 14);

private Dimension dimension=new Dimension(177,24);

public boolean iscreat=false;

JTextArea describetextArea ;

public void ModifyEventUi(final Event event) {

frame = new JFrame();

frame.setBounds(100, 100, 400, 400);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.getContentPane().setLayout(new BorderLayout(0, 0));

north = new JPanel();

frame.getContentPane().add(north, BorderLayout.NORTH);

label = new JLabel("ÐÞ¸Ä");

label.setFont(new Font("SimSun", Font.BOLD, 30));

north.add(label);

south = new JPanel();

frame.getContentPane().add(south, BorderLayout.SOUTH);

south.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));

btnNewButton = new JButton("ÐÞ¸Ä");

south.add(btnNewButton);

lblNewLabel\_1 = new JLabel(" ");

south.add(lblNewLabel\_1);

btnNewButton\_1 = new JButton("È¡Ïû");

south.add(btnNewButton\_1);

btnNewButton\_1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

frame.setVisible(false);

}

});

JPanel Center = new JPanel();

frame.getContentPane().add(Center, BorderLayout.CENTER);

Center.setLayout(null);

JLabel name = new JLabel("\u540D\u79F0");

name.setBounds(10, 8, 54, 15);

Center.add(name);

nametext = new JTextField(event.getName());

nametext.setBounds(74, 8, 284, 21);

Center.add(nametext);

nametext.setColumns(10);

JLabel begintime = new JLabel("\u8D77\u59CB\u65F6\u95F4");

begintime.setBounds(10, 38, 54, 15);

Center.add(begintime);

JLabel endtime = new JLabel("\u7ED3\u675F\u65F6\u95F4");

endtime.setBounds(10, 68, 54, 15);

Center.add(endtime);

Date date=new Date();

datepickbegintime = new DatePicker(event.getBeginTime(),DefaultFormat,font,dimension);

datepickbegintime.setBounds(74, 38, 284, 21);

datepickbegintime.setLocale(Locale.CHINA);

datepickbegintime.setTimePanleVisible(true);

Center.add(datepickbegintime);

// endtimetext = new JTextField();

// endtimetext.setBounds(74, 68, 284, 21);

// Center.add(endtimetext);

// endtimetext.setColumns(10);

datepickendtime = new DatePicker(event.getEndTime(),DefaultFormat,font,dimension);

datepickendtime.setBounds(74, 68, 284, 21);

datepickendtime.setLocale(Locale.CHINA);

datepickendtime.setTimePanleVisible(true);

Center.add(datepickendtime);

level = new JLabel("\u4F18\u5148\u7EA7");

level.setBounds(10, 98, 54, 15);

Center.add(level);

JPanel leveljpane = new JPanel();

leveljpane.setBounds(74, 90, 150, 24);

Center.add(leveljpane);

leveljpane.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));

final JRadioButton one = new JRadioButton("1");

leveljpane.add(one);

final JRadioButton two = new JRadioButton("2");

leveljpane.add(two);

final JRadioButton three = new JRadioButton("3");

leveljpane.add(three);

final JRadioButton four = new JRadioButton("4");

leveljpane.add(four);

ButtonGroup levelBG=new ButtonGroup();

levelBG.add(one);

levelBG.add(two);

levelBG.add(three);

levelBG.add(four);

if(event.getLevel()==1)

{

one.setSelected(true);

}

if(event.getLevel()==2)

{

two.setSelected(true);

}

if(event.getLevel()==3)

{

three.setSelected(true);

}

if(event.getLevel()==4)

{

four.setSelected(true);

}

JLabel label\_1 = new JLabel("\u662F\u5426\u63D0\u793A");

label\_1.setBounds(10, 128, 54, 15);

Center.add(label\_1);

JPanel noticepanel = new JPanel();

noticepanel.setBounds(73, 118, 89, 24);

Center.add(noticepanel);

noticepanel.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));

final JRadioButton noticeyes = new JRadioButton("\u662F");

noticepanel.add(noticeyes);

JRadioButton noticeno = new JRadioButton("\u5426");

noticepanel.add(noticeno);

ButtonGroup repeat=new ButtonGroup();

repeat.add(noticeyes);

repeat.add(noticeno);

if(event.isHint()==true){

noticeyes.setSelected(true);

}

else{

noticeno.setSelected(true);

}

JLabel notice = new JLabel("\u63D0\u793A\u65B9\u5F0F");

notice.setBounds(172, 128, 54, 15);

Center.add(notice);

JComboBox noticecomboBox = new JComboBox();

noticecomboBox.setBounds(236, 125, 120, 21);

Center.add(noticecomboBox);

JLabel describe = new JLabel("\u63CF\u8FF0");

describe.setBounds(10, 159, 54, 15);

Center.add(describe);

JScrollPane scrollPane = new JScrollPane();

scrollPane.setBounds(20, 184, 338, 90);

Center.add(scrollPane);

describetextArea = new JTextArea(event.getDescribe());

describetextArea.setLineWrap(true);

describetextArea.setWrapStyleWord(true);

scrollPane.setViewportView(describetextArea);

frame.setVisible(true);

btnNewButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

Event newevent =new Event();

String DefaultFormat = "yyyy-MM-dd HH:mm:ss";

newevent.setName(nametext.getText());

newevent.setBeginTime((Date)datepickbegintime.getValue());

newevent.setEndTime((Date)datepickendtime.getValue());

boolean hint;

if(noticeyes.isSelected()==true)

newevent.setHint(true);

else

newevent.setHint(false);

int level = 0;

if(one.isSelected()==true)

newevent.setLevel(1);

if(two.isSelected()==true)

newevent.setLevel(2);

if(three.isSelected()==true)

newevent.setLevel(3);

if(four.isSelected()==true)

newevent.setLevel(4);

newevent.setDescribe(describetextArea.getText());

EventManage a=new EventManage();

try {

a.ModifyEvent(event, newevent);

frame.setVisible(false);

} catch (BusinessException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (DbException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (SQLException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

}

}

cn/list/ui/NewEventUi.java

package cn.list.ui;

import java.awt.EventQueue;

import javax.swing.JFrame;

import java.awt.FlowLayout;

import javax.swing.JLabel;

import java.awt.GridBagLayout;

import java.awt.GridBagConstraints;

import java.awt.Insets;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.util.Date;

import java.util.Locale;

import javax.swing.JTextField;

import java.awt.BorderLayout;

import java.awt.Dimension;

import javax.swing.JPanel;

import javax.swing.ButtonGroup;

import javax.swing.JButton;

import javax.swing.JComboBox;

import java.awt.GridLayout;

import com.jgoodies.forms.layout.FormLayout;

import com.eltima.components.ui.DatePicker;

import com.jgoodies.forms.layout.ColumnSpec;

import com.jgoodies.forms.layout.FormSpecs;

import com.jgoodies.forms.layout.RowSpec;

import cn.list.control.EventManage;

import cn.list.util.BaseException;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

import javax.swing.JRadioButton;

import java.awt.Font;

import javax.swing.JTextPane;

import javax.swing.JList;

import javax.swing.JTextArea;

import javax.swing.JScrollPane;

public class NewEventUi {

private JFrame frame;

private JPanel north;

private JLabel label;

private JPanel south;

private JLabel lblNewLabel;

private JButton btnNewButton;

private JLabel lblNewLabel\_1;

private JButton btnNewButton\_1;

private JTextField nametext;

private JTextField begintimetext;

private JTextField endtimetext;

private JLabel level;

private DatePicker datepickbegintime;

private DatePicker datepickendtime;

private static final String DefaultFormat = "yyyy-MM-dd HH:mm:ss";

private Font font=new Font("Times New Roman", Font.BOLD, 14);

private Dimension dimension=new Dimension(177,24);

public boolean iscreat=false;

public void newEventui() {

frame = new JFrame();

frame.setBounds(100, 100, 400, 400);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.getContentPane().setLayout(new BorderLayout(0, 0));

north = new JPanel();

frame.getContentPane().add(north, BorderLayout.NORTH);

label = new JLabel("\u65B0\u5EFA");

label.setFont(new Font("SimSun", Font.BOLD, 30));

north.add(label);

south = new JPanel();

frame.getContentPane().add(south, BorderLayout.SOUTH);

south.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));

btnNewButton = new JButton("ÐÂ½¨");

south.add(btnNewButton);

lblNewLabel\_1 = new JLabel(" ");

south.add(lblNewLabel\_1);

btnNewButton\_1 = new JButton("È¡Ïû");

south.add(btnNewButton\_1);

JPanel Center = new JPanel();

frame.getContentPane().add(Center, BorderLayout.CENTER);

Center.setLayout(null);

JLabel name = new JLabel("\u540D\u79F0");

name.setBounds(10, 8, 54, 15);

Center.add(name);

nametext = new JTextField();

nametext.setBounds(74, 8, 284, 21);

Center.add(nametext);

nametext.setColumns(10);

JLabel begintime = new JLabel("\u8D77\u59CB\u65F6\u95F4");

begintime.setBounds(10, 38, 54, 15);

Center.add(begintime);

JLabel endtime = new JLabel("\u7ED3\u675F\u65F6\u95F4");

endtime.setBounds(10, 68, 54, 15);

Center.add(endtime);

// begintimetext = new JTextField();

// begintimetext.setBounds(74, 38, 284, 21);

// Center.add(begintimetext);

// begintimetext.setColumns(10);

//

Date date=new Date();

datepickbegintime = new DatePicker(date,DefaultFormat,font,dimension);

datepickbegintime.setBounds(74, 38, 284, 21);

datepickbegintime.setLocale(Locale.CHINA);

datepickbegintime.setTimePanleVisible(true);

Center.add(datepickbegintime);

// endtimetext = new JTextField();

// endtimetext.setBounds(74, 68, 284, 21);

// Center.add(endtimetext);

// endtimetext.setColumns(10);

datepickendtime = new DatePicker(date,DefaultFormat,font,dimension);

datepickendtime.setBounds(74, 68, 284, 21);

datepickendtime.setLocale(Locale.CHINA);

datepickendtime.setTimePanleVisible(true);

Center.add(datepickendtime);

level = new JLabel("\u4F18\u5148\u7EA7");

level.setBounds(10, 98, 54, 15);

Center.add(level);

JPanel leveljpane = new JPanel();

leveljpane.setBounds(74, 90, 150, 24);

Center.add(leveljpane);

leveljpane.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));

final JRadioButton one = new JRadioButton("1");

one.setSelected(true);

leveljpane.add(one);

final JRadioButton two = new JRadioButton("2");

leveljpane.add(two);

final JRadioButton three = new JRadioButton("3");

leveljpane.add(three);

final JRadioButton four = new JRadioButton("4");

leveljpane.add(four);

ButtonGroup levelBG=new ButtonGroup();

levelBG.add(one);

levelBG.add(two);

levelBG.add(three);

levelBG.add(four);

JLabel label\_1 = new JLabel("\u662F\u5426\u63D0\u793A");

label\_1.setBounds(10, 128, 54, 15);

Center.add(label\_1);

JPanel noticepanel = new JPanel();

noticepanel.setBounds(73, 118, 89, 24);

Center.add(noticepanel);

noticepanel.setLayout(new FlowLayout(FlowLayout.CENTER, 5, 5));

final JRadioButton noticeyes = new JRadioButton("\u662F");

noticeyes.setSelected(true);

noticepanel.add(noticeyes);

JRadioButton noticeno = new JRadioButton("\u5426");

noticepanel.add(noticeno);

ButtonGroup repeat=new ButtonGroup();

repeat.add(noticeyes);

repeat.add(noticeno);

JLabel notice = new JLabel("\u63D0\u793A\u65B9\u5F0F");

notice.setBounds(172, 128, 54, 15);

Center.add(notice);

JComboBox noticecomboBox = new JComboBox();

noticecomboBox.setBounds(236, 125, 120, 21);

Center.add(noticecomboBox);

JLabel describe = new JLabel("\u63CF\u8FF0");

describe.setBounds(10, 159, 54, 15);

Center.add(describe);

JScrollPane scrollPane = new JScrollPane();

scrollPane.setBounds(20, 184, 338, 90);

Center.add(scrollPane);

final JTextArea describetextArea = new JTextArea();

describetextArea.setLineWrap(true);

describetextArea.setWrapStyleWord(true);

scrollPane.setViewportView(describetextArea);

frame.setVisible(true);

//ÐÂ½¨µÄÊÂ¼þ

btnNewButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

String DefaultFormat = "yyyy-MM-dd HH:mm:ss";

String name =nametext.getText();

Date begin=(Date)datepickbegintime.getValue();

Date end=(Date)datepickendtime.getValue();

if(name!=null&&begin!=null&&end!=null){

iscreat=true;

}

boolean hint;

if(noticeyes.isSelected()==true)

hint=true;

else

hint=false;

int level = 0;

if(one.isSelected()==true)

level=1;

if(two.isSelected()==true)

level=2;

if(three.isSelected()==true)

level=3;

if(four.isSelected()==true)

level=4;

String describe=describetextArea.getText();

EventManage a=new EventManage();

try {

a.CreateEvent(name, begin, end, hint, describe, level);

iscreat=true;

System.out.println(iscreat);

frame.setVisible(false);

} catch (BusinessException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

} catch (DbException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

}

});

btnNewButton\_1.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

frame.setVisible(false);

}

});

}

}

cn/list/util/BaseException.java

package cn.list.util;

public class BaseException extends Exception {

public BaseException(String msg){

super(msg);

}

}

cn/list/util/BusinessException.java

package cn.list.util;

public class BusinessException extends BaseException {

public BusinessException(String msg){

super(msg);

}

}

cn/list/util/DBUtil.java

package cn.list.util;

import java.sql.Connection;

public class DBUtil {

private static final String jdbcUrl="jdbc:jtds:sqlserver://127.0.0.1:1433/work";

private static final String dbUser="root";

private static final String dbPwd="zucc";

static{

try {

Class.forName("net.sourceforge.jtds.jdbc.Driver");

} catch (ClassNotFoundException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

public static Connection getConnection() throws java.sql.SQLException{

return java.sql.DriverManager.getConnection(jdbcUrl, dbUser, dbPwd);

}

}

cn/list/util/DbException.java

package cn.list.util;

public class DbException extends BaseException {

public DbException(java.lang.Throwable ex){

super("Êý¾Ý¿â²Ù×÷´íÎó£º"+ex.getMessage());

}

}

cn/list/workinterface/IEvent.java

package cn.list.workinterface;

import java.util.Date;

import java.util.List;

import java.sql.SQLException;

import cn.list.model.Event;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

public interface IEvent {

public void CreateEvent(String Name,Date BeginTime,Date EndTime,boolean Hint,String describe,int level)throws BusinessException, DbException;

public void AddEvent(Event event)throws DbException;

public void DelEvent(Event event)throws BusinessException,DbException, SQLException;

public void ModifyEvent(Event event,Event newevent)throws BusinessException, DbException, SQLException ;

public Event SerchEvent(int ID)throws BusinessException, DbException;

public List<Event> LoadEvent() throws BusinessException, DbException;

public void ClearEvent()throws BusinessException, DbException, SQLException;

}

cn/list/workinterface/ITip.java

package cn.list.workinterface;

import cn.list.model.Tip;

public interface ITip {

public void CreateTip();

public void BuildTip(int ID);

public void DelTip(int ID);

public void ModifyTip(int ID);

public Tip SearchTip(int ID);

}

package cn.list.ui;

import java.awt.Font;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.SQLException;

import java.text.SimpleDateFormat;

import java.util.Date;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import cn.list.control.EventManage;

import cn.list.model.Event;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

public class FinishHint {

private JFrame frame;

public void FinishHintUi(Event event) {

frame = new JFrame("完成");

frame.setBounds(100, 100, 310, 195);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("\u4EFB\u52A1\u5B8C\u6210\uFF01");

lblNewLabel.setFont(new Font("宋体", Font.BOLD, 30));

lblNewLabel.setBounds(70, 10, 160, 49);

frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("\u63A5\u4E0B\u6765\u7684\u4EFB\u52A1\u662F ");

lblNewLabel\_1.setFont(new Font("宋体", Font.BOLD, 18));

lblNewLabel\_1.setBounds(10, 61, 148, 31);

frame.getContentPane().add(lblNewLabel\_1);

JLabel eventname = new JLabel(event.getName());

eventname.setFont(new Font("宋体", Font.BOLD, 18));

eventname.setBounds(168, 61, 127, 31);

frame.getContentPane().add(eventname);

JButton btnNewButton = new JButton("\u786E\u5B9A");

btnNewButton.setBounds(95, 123, 93, 23);

frame.getContentPane().add(btnNewButton);

JLabel label = new JLabel("\u5F00\u59CB\u65F6\u95F4\u662F ");

label.setFont(new Font("宋体", Font.BOLD, 18));

label.setBounds(10, 88, 148, 31);

frame.getContentPane().add(label);

SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

Date date=new Date();

String sdate=sdf.format(event.getBeginTime());

JLabel Begintime = new JLabel(sdate);

Begintime.setFont(new Font("宋体", Font.BOLD, 14));

Begintime.setBounds(122, 90, 162, 31);

frame.getContentPane().add(Begintime);

frame.setVisible(true);

btnNewButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

frame.setVisible(false);

}

});

}

public void FinishHintUiNoNext() {

frame = new JFrame("完成!!!");

frame.setBounds(100, 100, 310, 195);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JLabel lblNewLabel = new JLabel("\u4EFB\u52A1\u5B8C\u6210\uFF01");

lblNewLabel.setFont(new Font("宋体", Font.BOLD, 30));

lblNewLabel.setBounds(70, 10, 160, 49);

frame.getContentPane().add(lblNewLabel);

JLabel lblNewLabel\_1 = new JLabel("接下来没有任务");

lblNewLabel\_1.setFont(new Font("宋体", Font.BOLD, 18));

lblNewLabel\_1.setBounds(10, 61, 148, 31);

frame.getContentPane().add(lblNewLabel\_1);

//

// JLabel eventname = new JLabel(event.getName());

// eventname.setFont(new Font("宋体", Font.BOLD, 18));

// eventname.setBounds(168, 61, 127, 31);

// frame.getContentPane().add(eventname);

//

JButton btnNewButton = new JButton("\u786E\u5B9A");

btnNewButton.setBounds(95, 123, 93, 23);

frame.getContentPane().add(btnNewButton);

btnNewButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

frame.setVisible(false);

}

});

//

// JLabel label = new JLabel("\u5F00\u59CB\u65F6\u95F4\u662F ");

// label.setFont(new Font("宋体", Font.BOLD, 18));

// label.setBounds(10, 88, 148, 31);

// frame.getContentPane().add(label);

//

// SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");

// Date date=new Date();

// String sdate=sdf.format(event.getBeginTime());

//

// JLabel Begintime = new JLabel(sdate);

// Begintime.setFont(new Font("宋体", Font.BOLD, 14));

// Begintime.setBounds(122, 90, 162, 31);

// frame.getContentPane().add(Begintime);

frame.setVisible(true);

}

}

package cn.list.ui;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import javax.swing.JButton;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.SwingConstants;

import cn.list.model.Event;

public class ErrorHint {

private JFrame frame;

public void ErrorHintUI(String str) {

frame = new JFrame("错误！！！");

frame.setBounds(100, 100, 210, 140);

frame.setDefaultCloseOperation(JFrame.DISPOSE\_ON\_CLOSE);

frame.getContentPane().setLayout(null);

JLabel hint = new JLabel(str);

hint.setVerticalAlignment(SwingConstants.CENTER);

hint.setBounds(10, 10, 174, 58);

frame.getContentPane().add(hint);

JButton btnNewButton = new JButton("\u786E\u5B9A");

btnNewButton.setBounds(47, 68, 93, 23);

frame.getContentPane().add(btnNewButton);

btnNewButton.addActionListener(new ActionListener()

{

public void actionPerformed(ActionEvent e)

{

frame.setVisible(false);

}

});

frame.setVisible(true);

}

}

package cn.list.control;

import java.sql.SQLException;

import java.util.Date;

import java.util.List;

import cn.list.model.Event;

import cn.list.ui.FinishHint;

import cn.list.util.BusinessException;

import cn.list.util.DbException;

public class CountDown implements Runnable {

public long time=1;

Event Nextevent;

Event HintEvent;

EventManage eventmanage =new EventManage();

public CountDown(){

}

@Override

public void run() {

int i=0;

int j=0;

long end;

while(true){

HintEvent=null;

try {

HintEvent=eventmanage.SerchHintEvent();

} catch (BusinessException | DbException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

while(HintEvent==null){

//锁线程暂停1秒

try {

Thread.sleep(1000);

System.out.println("暂停");

try {

HintEvent=eventmanage.SerchHintEvent();

} catch (BusinessException | DbException e1) {

// TODO Auto-generated catch block

e1.printStackTrace();

}

} catch (InterruptedException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

end=HintEvent.getEndTime().getTime();

Date date=new Date();

long now=date.getTime();

time=(end-now)/1000;

while (time>=0)

{

//倒计时

if(time<=0){

try {

eventmanage.CompleteEvent(HintEvent);

} catch (BusinessException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (SQLException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

//缺少个调用结束的界面

try {

Nextevent=eventmanage.SerchHintEvent();

} catch (BusinessException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (DbException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

FinishHint finishhint =new FinishHint();

if(Nextevent==null){

finishhint.FinishHintUiNoNext();

break;

}

System.out.println(Nextevent.getID());

finishhint.FinishHintUi(Nextevent);

break;

}

System.out.println(HintEvent.getName()+"还剩： " + time + " 秒");

time--;

try

{

Thread.sleep(1000);

}

catch (InterruptedException e)

{

e.printStackTrace();

}

}

}

}

}