Mainface.java

/\*

\* 社区主题软件（悠品）

\* version 1.0

\* 主界面

\* 2017-5-23

\* 最近修改 2017-6-23

\* 修改内容：改变界面样貌和组件

\* 源代码编码：UTF-8

\* 作者：郑思泽 zhengsz@pku.edu.cn

\*

\*/

package MainPage;

import java.awt.Color;

import java.awt.Cursor;

import java.awt.Font;

import java.awt.Image;

import java.awt.event.MouseAdapter;

import java.awt.event.MouseEvent;

import java.awt.event.WindowAdapter;

import java.awt.event.WindowEvent;

import java.awt.image.BufferedImage;

import java.io.BufferedInputStream;

import java.io.BufferedOutputStream;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.IOException;

import java.util.Random;

import javax.imageio.ImageIO;

import javax.swing.BorderFactory;

import javax.swing.ImageIcon;

import javax.swing.JButton;

import javax.swing.JFileChooser;

import javax.swing.JFrame;

import javax.swing.JLabel;

import javax.swing.JPanel;

import javax.swing.JTextField;

import Client.connection;

import Community.Community;

import Info.Usr\_Info;

/\*

\* 主界面主要类

\*/

public class MainFace extends JFrame {

private static final long serialVersionUID = 1L;

File Root, Picture, User;// 文件目录

File img;// 临时图片

int imgHeight, imgWidth;// 临时图片大小

File jf\_mem;// 存储最近一次打开目录

Usr\_Info user\_info;// 用户信息

connection con;// 与服务器连接

JFrame pthis = this;

/\*

\* 界面组件

\*/

public JPanel community, newblog, showblog;

public JLabel lbl\_LeftInfo, lbl\_MidInfo, lbl\_RightInfo;

JLabel lbl\_LuserImage, lbl\_Litems[], lbl\_Msearch, lbl\_Mitems[], lbl\_Mimages[], lbl\_Ritems[], lbl\_Rnull;

final int Litems\_num = 3, Mitems\_num = 8, Ritems\_num = 0;// 各个label所含的子label数目

JTextField txt\_MSearch;

JButton btn\_MSearch;

JLabel lbl\_warning;

String Mstring[] = { "悠品新闻", "悠品娱乐", "悠品贴吧", "悠品运动", "悠品美食", "悠品轻奢", "悠品美图", "悠品生活" };

SelfPage selfPage;

Community comPage;

Color ColorOpt[] = { Color.red, Color.GREEN, Color.pink, Color.blue, Color.CYAN, Color.darkGray, Color.green,

Color.magenta, Color.ORANGE, Color.YELLOW, Color.lightGray };

Random r = new Random();

/\*

\* 初始化函数

\*/

public void init() throws IOException {

init\_LayBoard();// 初始化底层面板

init\_SelfPage();// 初始化个人修改主页

this.setBounds(200, 150, 900, 500);

this.setLayout(null);

this.addWindowListener(new WindowAdapter() {// 关闭窗口时发送信息给服务端

public void windowClosing(WindowEvent e) {

con.close();

}

});

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

this.setVisible(true);

}

/\*

\* 底层面板初始化函数

\*/

private void init\_LayBoard() throws IOException {

/\*

\* 界面初始化部分 分为左中右三部分为三个label

\*/

lbl\_LeftInfo = new JLabel();

lbl\_LeftInfo.setBounds(0, 0, 60, 500);

lbl\_LeftInfo.setOpaque(true);

lbl\_LeftInfo.setBackground(Color.BLACK);

lbl\_LuserImage = new JLabel(); // 用户头像

lbl\_LuserImage.setBounds(5, 5, 50, 50);

img = new File(Picture.getPath() + "/user\_image0.jpg");

if (!img.exists())

img = new File(Picture.getPath() + "/default\_image.jpg");

System.out.println(Picture);

BufferedInputStream in = new BufferedInputStream(new FileInputStream(img.getPath()));

Image bi = ImageIO.read(in);

imgHeight = bi.getHeight(null);

imgWidth = bi.getWidth(null);

if (imgHeight != 50 || imgWidth != 50) {

MyImageTools.resizePicture(bi, 50, 50, img); // 将图片变为指定大小

}

in.close();

lbl\_LuserImage.setIcon(new ImageIcon(img.getPath()));

lbl\_LeftInfo.add(lbl\_LuserImage);

lbl\_Litems = new JLabel[Litems\_num];

for (int i = 0; i < Litems\_num; ++i) { // 左侧面板控制组件初始化

lbl\_Litems[i] = new JLabel();

lbl\_Litems[i].setBounds(5, 60 \* (i + 1) + 5, 50, 50);

img = new File(Picture.getPath() + "/L\_image" + i + ".jpg");

if (!img.exists())

img = new File(Picture.getPath() + "/default\_image.jpg");

in = new BufferedInputStream(new FileInputStream(img.getPath()));

bi = ImageIO.read(in);

imgHeight = bi.getHeight(null);

imgWidth = bi.getWidth(null);

if (imgHeight != 50 || imgWidth != 50) {

MyImageTools.resizePicture(bi, 50, 50, img);

}

in.close();

lbl\_Litems[i].setIcon(new ImageIcon(img.getPath()));

lbl\_Litems[i].setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));

lbl\_LeftInfo.add(lbl\_Litems[i]);

}

// 为组件添加事件响应

lbl\_Litems[0].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) { // 该按钮响应新闻界面打开事件

lbl\_Litems[0].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) { // 新闻界面初始化

lbl\_Litems[0].setBorder(null);

}

public void mouseClicked(MouseEvent e) { // 该按钮响应个人主页打开事件

lbl\_Rnull.setVisible(false);

selfPage.setVisible(false);

comPage = new Community(user\_info, con, pthis);

comPage.setBounds(0, 0, 640, 500);

comPage.setVisible(true);

lbl\_RightInfo.add(comPage);

}

});

lbl\_Litems[1].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Litems[1].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Litems[1].setBorder(null);

}

public void mouseClicked(MouseEvent e) {

lbl\_Rnull.setVisible(false);

comPage.setVisible(false);

selfPage.refresh();

selfPage.setVisible(true);

}

});

lbl\_Litems[2].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Litems[2].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Litems[2].setBorder(null);

}

public void mouseClicked(MouseEvent e) {

lbl\_Rnull.setVisible(true);

selfPage.setVisible(false);

}

});

lbl\_MidInfo = new JLabel();

lbl\_MidInfo.setBounds(60, 0, 200, 500);

lbl\_MidInfo.setOpaque(true);

lbl\_MidInfo.setBackground(Color.LIGHT\_GRAY);

lbl\_Msearch = new JLabel();

lbl\_Msearch.setBounds(0, 0, 200, 50);

txt\_MSearch = new JTextField();

txt\_MSearch.setBounds(5, 5, 140, 40);

lbl\_Msearch.add(txt\_MSearch);

btn\_MSearch = new JButton();

btn\_MSearch.setBounds(155, 5, 40, 40);

img = new File(Picture.getPath() + "/search\_image.jpg");

if (!img.exists())

img = new File(Picture.getPath() + "/default\_image.jpg");

in = new BufferedInputStream(new FileInputStream(img.getPath()));

bi = ImageIO.read(in);

imgHeight = bi.getHeight(null);

imgWidth = bi.getWidth(null);

if (imgHeight != 40 || imgWidth != 40) {

MyImageTools.resizePicture(bi, 40, 40, img);

}

in.close();

btn\_MSearch.setIcon(new ImageIcon(img.getPath()));

lbl\_Msearch.add(btn\_MSearch);

lbl\_MidInfo.add(lbl\_Msearch);

lbl\_Mitems = new JLabel[Mitems\_num];

lbl\_Mimages = new JLabel[Mitems\_num];

for (int i = 0; i < Mitems\_num; ++i) { // 中间组件初始化

lbl\_Mitems[i] = new JLabel();

lbl\_Mimages[i] = new JLabel();

lbl\_Mitems[i].setBounds(0, 50 \* (i + 1), 200, 50);

lbl\_Mimages[i].setBounds(5, 5, 40, 40);

img = new File(Picture.getPath() + "/M\_image" + i + ".jpg");

if (!img.exists())

img = new File(Picture.getPath() + "/default\_image.jpg");

in = new BufferedInputStream(new FileInputStream(img.getPath()));

bi = ImageIO.read(in);

int height = bi.getHeight(null);

int width = bi.getWidth(null);

if (height != 40 || width != 40) {

MyImageTools.resizePicture(bi, 40, 40, img);

}

in.close();

lbl\_Mimages[i].setIcon(new ImageIcon(img.getPath()));

lbl\_Mitems[i].add(lbl\_Mimages[i]);

lbl\_Mitems[i].setFont(new Font("宋体", Font.BOLD, 18));

lbl\_Mitems[i].setText(" " + Mstring[i]);

lbl\_Mitems[i].setBorder(BorderFactory.createEtchedBorder());

lbl\_Mitems[i].setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));

lbl\_MidInfo.add(lbl\_Mitems[i]);

}

// 为中间组件添加事件响应

lbl\_Mitems[0].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Mitems[0].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Mitems[0].setBorder(BorderFactory.createEtchedBorder());

}

public void mouseClicked(MouseEvent e) {

// 功能定义，具体尚未实现

}

});

lbl\_Mitems[1].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Mitems[1].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Mitems[1].setBorder(BorderFactory.createEtchedBorder());

}

public void mouseClicked(MouseEvent e) {

// 功能定义，具体尚未实现

}

});

lbl\_Mitems[2].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Mitems[2].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Mitems[2].setBorder(BorderFactory.createEtchedBorder());

}

public void mouseClicked(MouseEvent e) {

// 功能定义，具体尚未实现

}

});

lbl\_Mitems[3].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Mitems[3].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Mitems[3].setBorder(BorderFactory.createEtchedBorder());

}

public void mouseClicked(MouseEvent e) {

// 功能定义，具体尚未实现

}

});

lbl\_Mitems[4].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Mitems[4].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Mitems[4].setBorder(BorderFactory.createEtchedBorder());

}

public void mouseClicked(MouseEvent e) {

// 功能定义，具体尚未实现

}

});

lbl\_Mitems[5].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Mitems[5].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Mitems[5].setBorder(BorderFactory.createEtchedBorder());

}

public void mouseClicked(MouseEvent e) {

// 功能定义，具体尚未实现

}

});

lbl\_Mitems[6].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Mitems[6].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Mitems[6].setBorder(BorderFactory.createEtchedBorder());

}

public void mouseClicked(MouseEvent e) {

// 功能定义，具体尚未实现

}

});

lbl\_Mitems[7].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_Mitems[7].setBorder(BorderFactory.createLineBorder(ColorOpt[r.nextInt(ColorOpt.length)]));

}

public void mouseExited(MouseEvent e) {

lbl\_Mitems[7].setBorder(BorderFactory.createEtchedBorder());

}

public void mouseClicked(MouseEvent e) {

// 功能定义，具体尚未实现

}

});

// 右侧部分初始化

lbl\_RightInfo = new JLabel();

lbl\_RightInfo.setBounds(260, 0, 640, 500);

lbl\_RightInfo.setOpaque(true);

// 初始化设置界面

lbl\_Rnull = new JLabel();

lbl\_Rnull.setBounds(0, 0, 640, 500);

img = new File(Picture.getPath() + "/logo\_image.jpeg");

if (!img.exists())

img = new File(Picture.getPath() + "/default\_image.jpg");

in = new BufferedInputStream(new FileInputStream(img.getPath()));

bi = ImageIO.read(in);

int height = bi.getHeight(null);

int width = bi.getWidth(null);

if (height != 500 || width != 640) {

MyImageTools.resizePicture(bi, 640, 500, img);

}

in.close();

lbl\_Rnull.setIcon(new ImageIcon(img.getPath()));

lbl\_RightInfo.add(lbl\_Rnull);

// 用于提醒重要信息的窗口

lbl\_warning = new JLabel();

lbl\_warning.setBounds(200, 0, 200, 20);

lbl\_warning.setOpaque(true);

lbl\_warning.setBackground(Color.red);

lbl\_warning.setForeground(Color.black);

lbl\_warning.setFont(new Font("华文新魏", Font.PLAIN, 18));

lbl\_warning.setVisible(false);

// 将三部分界面添加到窗口容器中

lbl\_RightInfo.add(lbl\_warning);

this.add(lbl\_LeftInfo);

this.add(lbl\_MidInfo);

this.add(lbl\_RightInfo);

}

/\*

\* 初始化个人信息主页

\*/

private void init\_SelfPage() throws IOException {

selfPage = new SelfPage();

selfPage.setBounds(0, -5, 640, 500);

selfPage.setVisible(false);

lbl\_RightInfo.add(selfPage);

}

/\*

\* 构造函数 参数1：界面访问信息所用根目录 参数2：界面访问 图片所用目录 参数3：界面访问用户资料所用目录 参数4：用户信息 参数5：与服务器链接

\*

\* 抛出异常：IOException

\*/

public MainFace(File Root\_, File Picture\_, File User\_, Usr\_Info user\_info\_, connection con\_) throws IOException {

super("悠品");

Root = Root\_;

Picture = Picture\_;

User = User\_;

user\_info = user\_info\_;

con = con\_;

init();

System.out.println("client" + user\_info.getName());

}

/\*

\* 内部类，个人主页 JPanel子类

\*/

class SelfPage extends JPanel {

private static final long serialVersionUID = 1L;

JLabel lbl\_change\_back; // 用于充当背景，存放其他组件

JTextField txt\_change[]; // 信息内容

JLabel lbl\_image[]; // 属性图像与用户头像

JLabel lbl\_image\_ins; // 提示信息

JLabel lbl\_id, lbl\_level;

JButton btn\_save; // 保存按钮

/\*

\* 初始化函数 初始化界面组件

\*/

void init() throws IOException {

lbl\_change\_back = new JLabel();

// 设置背景图片

img = new File(Picture.getPath() + "/change\_back\_image.jpg");

if (!img.exists())

img = new File(Picture.getPath() + "/default\_image.jpeg");

BufferedInputStream in = new BufferedInputStream(new FileInputStream(img.getPath()));

Image bi = ImageIO.read(in);

imgHeight = bi.getHeight(null);

imgWidth = bi.getWidth(null);

if (imgHeight != 500 || imgWidth != 640) {

MyImageTools.resizePicture(bi, 640, 500, img);// 修改成指定大小

}

in.close();

lbl\_change\_back.setIcon(new ImageIcon(img.getPath()));

lbl\_image = new JLabel[8];

for (int i = 0; i < 7; ++i) { // 个人信息属性组件

lbl\_image[i] = new JLabel();

lbl\_image[i].setBounds(70, 80 + i \* 50, 100, 40);

img = new File(Picture.getPath() + "/change\_image" + i + ".jpg");

if (!img.exists())

img.createNewFile();

in = new BufferedInputStream(new FileInputStream(img.getPath()));

bi = ImageIO.read(in);

imgHeight = bi.getHeight(null);

imgWidth = bi.getWidth(null);

if (imgHeight != 40 || imgWidth != 100) {

MyImageTools.resizePicture(bi, 100, 40, img);

}

in.close();

lbl\_image[i].setIcon(new ImageIcon(img.getPath()));

JLabel tmp = lbl\_image[i];

lbl\_image[i].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

tmp.setBorder(BorderFactory.createRaisedBevelBorder());

}

public void mouseExited(MouseEvent e) {

tmp.setBorder(null);

}

});

lbl\_change\_back.add(lbl\_image[i]);

}

lbl\_image[7] = new JLabel(); // 用户头像加载

lbl\_image[7].setBounds(400, 80, 90, 90);

lbl\_image[7].setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));

img = new File(Picture.getPath() + "/user\_image1.jpg");

if (!img.exists())

img.createNewFile();

in = new BufferedInputStream(new FileInputStream(img.getPath()));

bi = ImageIO.read(in);

imgHeight = bi.getHeight(null);

imgWidth = bi.getWidth(null);

if (imgHeight != 90 || imgWidth != 90) {

MyImageTools.resizePicture(bi, 90, 90, img);

}

in.close();

lbl\_image[7].setIcon(new ImageIcon(img.getPath()));

lbl\_image[7].addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

lbl\_image\_ins.setVisible(true);

lbl\_image[7].setBorder(BorderFactory.createLineBorder(Color.blue));

}

public void mouseExited(MouseEvent e) {

lbl\_image\_ins.setVisible(false);

lbl\_image[7].setBorder(null);

}

public void mouseClicked(MouseEvent e) { // 响应头像更改事件

JFileChooser jf = new JFileChooser();

jf.setDialogTitle("选取头像");

jf.setAcceptAllFileFilterUsed(false);

if (jf\_mem != null)

jf.setCurrentDirectory(jf\_mem);

// 设置文件筛选

jf.addChoosableFileFilter(new javax.swing.filechooser.FileFilter() {

@Override

public boolean accept(File f) {

String s = f.getName().toLowerCase();

if (s.endsWith(".jpg") || f.isDirectory())

return true;

return false;

}

// 描述内容为\*.JPG

@Override

public String getDescription() {

// TODO Auto-generated method stub

return new String("\*.JPG");

}

});

;

jf.showOpenDialog(new JLabel());

img = jf.getSelectedFile();

if (img != null && img.exists()) {

// 头像更改涉及两部分图片改变，fp是个人界面头像更改，fpp是主界面头像更改

File fp = new File(Picture.getPath() + "/user\_image1.jpg");

File fpp = new File(Picture.getPath() + "/user\_image0.jpg");

try {

BufferedInputStream in = new BufferedInputStream(new FileInputStream(img.getPath()));

Image bi = ImageIO.read(in);

MyImageTools.resizePicture(bi, 90, 90, fp);

MyImageTools.resizePicture(bi, 40, 40, fpp);

String s = img.getParent();

jf\_mem = new File(s);

// 采用随机数作为文件名，避免因路径的一致导致更新头像失败

File tmp = new File(s + "/" + r.nextInt() + "YoPin\_tmp.jpg");

if (!tmp.exists())

tmp.createNewFile();

MyImageTools.resizePicture(bi, 90, 90, tmp);

lbl\_image[7].setIcon(new ImageIcon(tmp.getAbsolutePath()));

in = new BufferedInputStream(new FileInputStream(img.getPath()));

bi = ImageIO.read(in);

MyImageTools.resizePicture(bi, 90, 90, fp);

MyImageTools.resizePicture(bi, 40, 40, fpp);

s = img.getParent();

jf\_mem = new File(s);

// 采用随机数作为文件名，避免因路径的一致导致更新头像失败

tmp = new File(s + "/" + r.nextInt() + "YoPin\_tmp.jpg");

if (!tmp.exists())

tmp.createNewFile();

MyImageTools.resizePicture(bi, 40, 40, tmp);

lbl\_LuserImage.setIcon(new ImageIcon(tmp.getAbsolutePath()));

tmp.delete();

in.close();

} catch (IOException ex) {

ex.printStackTrace();

}

}

}

});

lbl\_change\_back.add(lbl\_image[7]);

// 初始化提醒信息窗口

lbl\_image\_ins = new JLabel();

lbl\_image\_ins.setBounds(500, 30, 90, 20);

lbl\_image\_ins.setText("点击修改头像");

lbl\_image\_ins.setVisible(false);

lbl\_image\_ins.setOpaque(true);

lbl\_image\_ins.setBackground(Color.white);

lbl\_image\_ins.setForeground(Color.BLACK);

lbl\_image\_ins.setFont(new Font("华文新魏", Font.PLAIN, 15));

lbl\_change\_back.add(lbl\_image\_ins);

btn\_save = new JButton();

btn\_save.setOpaque(true);

btn\_save.setBackground(Color.WHITE);

btn\_save.setBounds(500, 400, 100, 25);

btn\_save.setText("保存");

btn\_save.addMouseListener(new MouseAdapter() {

public void mouseEntered(MouseEvent e) {

// 进入事件

}

public void mouseExited(MouseEvent e) {

// 离开事件

}

public void mouseClicked(MouseEvent e) {// 响应点击事件

boolean ok = true;

int a = txt\_change[4].getText().indexOf("@");

int len = txt\_change[4].getText().length();

if (a <= 0 || a >= len - 1 || len < 3) {

ok = false;

lbl\_warning.setSize(200, 20);

lbl\_warning.setText("邮箱地址格式不正确");

lbl\_warning.setVisible(true);

}

if (txt\_change[0].getText().length() == 0) {

ok = false;

lbl\_warning.setSize(130, 20);

lbl\_warning.setText("不能有空昵称");

lbl\_warning.setVisible(true);

}

if (ok) {// 合理信息，进行保存

lbl\_warning.setVisible(false);

user\_info.setName(txt\_change[0].getText());

user\_info.setHobby(txt\_change[2].getText());

user\_info.setAge(txt\_change[3].getText());

user\_info.setMail(txt\_change[4].getText());

user\_info.setStar(txt\_change[5].getText());

user\_info.setJob(txt\_change[6].getText());

// 发送给服务器

con.usr\_info\_change(user\_info);

}

}

});

// 信息填写框初始化

txt\_change = new JTextField[7];

for (int i = 0; i < 7; ++i) {

txt\_change[i] = new JTextField();

txt\_change[i].setBounds(180, 80 + 50 \* i, 200, 40);

txt\_change[i].setOpaque(true);

txt\_change[i].setBackground(Color.white);

txt\_change[i].setForeground(Color.BLACK);

txt\_change[i].setFont(new Font("黑体", Font.PLAIN, 15));

lbl\_change\_back.add(txt\_change[i]);

}

txt\_change[1].setVisible(false);

lbl\_id = new JLabel();

lbl\_id.setBounds(180, 130, 200, 40);

lbl\_id.setOpaque(true);

lbl\_id.setForeground(Color.black);

lbl\_id.setBackground(Color.WHITE);

lbl\_id.setFont(new Font("黑体", Font.PLAIN, 15));

lbl\_change\_back.add(lbl\_id);

lbl\_change\_back.add(btn\_save);

this.add(lbl\_change\_back);

this.setBounds(0, 0, 640, 500);

}

/\*

\* 用于刷新用户信息

\*/

public void refresh() {

txt\_change[0].setText(user\_info.getName());

lbl\_id.setText(user\_info.getID());

txt\_change[2].setText(user\_info.getHobby());

txt\_change[3].setText(user\_info.getAge());

txt\_change[4].setText(user\_info.getMail());

txt\_change[5].setText(user\_info.getStar());

txt\_change[6].setText(user\_info.getJob());

}

SelfPage() throws IOException {

init();

}

}

/\*

\* 内部类，设置界面 JPanel子类

\*/

class SetPage extends JPanel {

JLabel lbl\_back, lbl\_return, lbl\_return\_ins, lbl\_msg;

public void init() throws IOException {

lbl\_back = new JLabel();

lbl\_back.setBounds(0, 0, 700, 500);

img = new File(Picture.getPath() + "/set\_image.jpg");

if (!img.exists())

img = new File(Picture.getPath() + "/default\_image.jpg");

BufferedInputStream in = new BufferedInputStream(new FileInputStream(img.getPath()));

Image bi = ImageIO.read(in);

int height = bi.getHeight(null);

int width = bi.getWidth(null);

if (height != 500 || width != 700) {

MyImageTools.resizePicture(bi, 700, 500, img);

}

in.close();

lbl\_back.setIcon(new ImageIcon(img.getPath()));

lbl\_return = new JLabel();

lbl\_return.setBounds(670, 0, 30, 500);

lbl\_return.setCursor(Cursor.getPredefinedCursor(Cursor.HAND\_CURSOR));

lbl\_return.addMouseListener(new MouseAdapter() {

public void mouseClicked(MouseEvent e) {

lbl\_return\_ins.setVisible(false);

}

public void mouseEntered(MouseEvent e) {

lbl\_return\_ins.setVisible(true);

}

public void mouseExited(MouseEvent e) {

lbl\_return\_ins.setVisible(false);

}

});

lbl\_back.add(lbl\_return);

lbl\_return\_ins = new JLabel();

lbl\_return\_ins.setBounds(700, 0, 100, 20);

lbl\_return\_ins.setText("点击返回");

lbl\_return\_ins.setVisible(false);

lbl\_return\_ins.setOpaque(true);

lbl\_return\_ins.setForeground(Color.black);

lbl\_return\_ins.setBackground(Color.white);

lbl\_return\_ins.setFont(new Font("华文新魏", Font.PLAIN, 18));

// lbl\_board.add(lbl\_return\_ins);

lbl\_msg = new JLabel();

lbl\_msg.setBounds(50, 50, 260, 20);

lbl\_msg.setText("软件仍在建设中，感谢支持！");

lbl\_msg.setFont(new Font("华文新魏", Font.PLAIN, 20));

lbl\_msg.setOpaque(true);

lbl\_msg.setBackground(Color.white);

lbl\_msg.setForeground(Color.black);

lbl\_back.add(lbl\_msg);

this.add(lbl\_back);

}

/\*

\* 构造函数

\*/

SetPage() throws IOException {

init();

}

}

}

/\*

\* 图像调整辅助类

\*/

class MyImageTools {

/\*

\* 用于改变图像大小并存储 参数1：原图像 参数2：目标图像宽 参数3：目标图像高 参数4：目标图像 抛出异常：IOException

\*/

public static void resizePicture(Image in, int width, int height, File outfile) throws IOException {

// 创建指定大小缓存

BufferedImage tag = new BufferedImage(width, height, BufferedImage.TYPE\_INT\_RGB);

// 存入图片

tag.getGraphics().drawImage(in, 0, 0, width, height, null);

BufferedOutputStream out = new BufferedOutputStream(new FileOutputStream(outfile.getPath()));

// 输出图像

ImageIO.write(tag, "JPG", out);

out.close();

}

}