package reptile;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.\*;

import java.net.\*;

import java.util.\*;

import javax.swing.\*;

import javax.swing.text.BadLocationException;

import javax.swing.text.DefaultHighlighter.DefaultHighlightPainter;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class rep extends JFrame implements ActionListener{

private JPanel jPanel=new JPanel();

private JPanel jpl1=new JPanel();

private JPanel jpl2=new JPanel();

private JPanel jpl3=new JPanel();

private JPanel jpl4=new JPanel();

private JPanel jpl5=new JPanel();

private JPanel jpl6=new JPanel();

private JPanel jpl7=new JPanel();

private JPanel jpl8=new JPanel();

private JPanel jpl9=new JPanel();

private JPanel jpl10=new JPanel();

private JLabel siteWarn=new JLabel("输入网址:");

private JTextField siteField=new JTextField(25);

private JScrollPane siteSPane=new JScrollPane(siteField);

private JButton goSpider=new JButton("开始爬取");

private JTextArea htmlArea=new JTextArea(15, 25);

private JScrollPane htmlSPane=new JScrollPane(htmlArea);

private JTextArea textArea=new JTextArea(15,25);

private JScrollPane textSPane=new JScrollPane(textArea);

private JTextArea senArea=new JTextArea(15, 25);

private JScrollPane senSPane=new JScrollPane(senArea);

private JTabbedPane tabPane=new JTabbedPane();

private JTextArea sensWord=new JTextArea(8,25);

private JScrollPane wordPane=new JScrollPane(sensWord);

private JButton openLib=new JButton(" 导入敏感词库");

private JButton match=new JButton("匹配");

private JButton siteLib=new JButton("导入网址库");

private JButton senin=new JButton("输入");

private JButton seout=new JButton("导出");

private JComboBox<String> charset=new JComboBox<String>();

private String textType="UTF-8";

private ArrayList<String> wordList=new ArrayList<String>(); //保存敏感词

private ArrayList<Integer> wordNum=new ArrayList<Integer>();

private String regExHtml="<[^>]+>"; //匹配标签

private String regExScript = "<script[^>]\*?>[\\s\\S]\*?<\\/script>"; //匹配script标签

private String regExStyle = "<style[^>]\*?>[\\s\\S]\*?<\\/style>"; //匹配style标签

private String regExSpace="[\\s]{2,}"; //匹配连续空格或回车等

private String regExImg="&[\\S]\*?;+"; //匹配网页上图案的乱码 //定义正则表达式

private Pattern pattern3=Pattern.compile(regExHtml, Pattern.CASE\_INSENSITIVE);

private Pattern pattern1=Pattern.compile(regExScript,Pattern.CASE\_INSENSITIVE);

private Pattern pattern2=Pattern.compile(regExStyle,Pattern.CASE\_INSENSITIVE);

private Pattern pattern4=Pattern.compile(regExSpace, Pattern.CASE\_INSENSITIVE);

private Pattern pattern5=Pattern.compile(regExImg,Pattern.CASE\_INSENSITIVE);

public rep() throws IOException { //设置界面风格

try { UIManager.setLookAndFeel(UIManager.getSystemLookAndFeelClassName()); }

catch (ClassNotFoundException e1) { // TODO Auto-generated catch block

e1.printStackTrace(); }

catch (InstantiationException e1) { // TODO Auto-generated catch block

e1.printStackTrace();

} catch (IllegalAccessException e1) { // TODO Auto-generated catch block

e1.printStackTrace(); }

catch (UnsupportedLookAndFeelException e1) { // TODO Auto-generated catch block

e1.printStackTrace(); }

this.setTitle("Reptile");

this.setLocation(400, 200);

this.setSize(600, 500);

this.setDefaultCloseOperation(EXIT\_ON\_CLOSE);

jPanel.setLayout(new BorderLayout());

charset.addItem("UTF-8");

charset.addItem("GBK");

charset.setEditable(false); //设置为不可编辑 //处理其事件,更新编码方式

charset.addActionListener(new ActionListener() { //获取选择的编码方式,默认情况下为UTF-8

@Override

public void actionPerformed(ActionEvent e) {

textType=(String)charset.getSelectedItem(); } });

jpl1.setLayout(new BorderLayout());

siteWarn.setPreferredSize(new Dimension(70,30));

siteSPane.setPreferredSize(new Dimension(300, 30));

goSpider.setPreferredSize(new Dimension(90, 30));

jpl5.setLayout(new GridLayout(1, 2, 10,10));

jpl5.add(goSpider);

jpl5.add(charset);

jpl1.add(siteWarn,BorderLayout.WEST);

jpl1.add(siteSPane,BorderLayout.CENTER);

jpl1.add(jpl5,BorderLayout.EAST); //源代码文本,以及处理后的文本框设置

htmlArea.setEditable(false);

htmlArea.setLineWrap(true);

htmlArea.setFont(new Font("宋体", Font.PLAIN, 14));

jpl2.setLayout(new BorderLayout());

jpl2.add(htmlSPane,BorderLayout.CENTER); //设置布局

jpl10.setLayout(new BorderLayout());

jpl10.add(senSPane,BorderLayout.CENTER);

jpl10.add(seout,BorderLayout.SOUTH);

jpl8.setLayout(new GridLayout(2, 1, 10,5));

jpl8.add(siteLib);

jpl8.add(openLib);

jpl9.setLayout(new GridLayout(2, 1, 10,5));

jpl9.add(match);

jpl9.add(senin);

jpl3.setLayout(new BorderLayout());

sensWord.setLineWrap(true);

sensWord.setEditable(false);

wordPane.setPreferredSize(new Dimension(6, 400));

jpl3.add(jpl8,BorderLayout.NORTH);

jpl3.add(wordPane,BorderLayout.CENTER);

jpl3.add(jpl9,BorderLayout.SOUTH);

textArea.setFont(new Font("宋体", Font.PLAIN, 14));

textArea.setEditable(false);

textArea.setLineWrap(true);

jpl4.setLayout(new BorderLayout());

jpl4.add(textSPane,BorderLayout.CENTER);

tabPane.add("html源代码", jpl2);

tabPane.add("网页文本",jpl4);

tabPane.add("敏感词统计",jpl10);

jpl7.setLayout(new BorderLayout());

jpl7.add(tabPane,BorderLayout.CENTER);

jpl6.setLayout(new BorderLayout());

jpl6.add(jpl7,BorderLayout.CENTER);

jpl6.add(jpl3,BorderLayout.EAST);

jPanel.add(jpl1,BorderLayout.NORTH);

jPanel.add(jpl6,BorderLayout.CENTER);

this.add(jPanel);

this.setVisible(true); //事件处理

goSpider.addActionListener(this);

siteLib.addActionListener(this);

openLib.addActionListener(this);

match.addActionListener(this);

senin.addActionListener(this);

seout.addActionListener(this);

}

public String getHtml(String website)

{

String str=null;

String text=""; //保存网页的内容

try

{

URL url=new URL(website); //建立对应的URL对象

URLConnection urlConne=url.openConnection(); //连接

urlConne.connect(); //获取输入流

BufferedReader br=new BufferedReader(new InputStreamReader(urlConne.getInputStream(),textType));

System.out.println("开始爬取");

while(true) { //爬取到结束

str=br.readLine();

if(str==null) break;

text+=(str+"\n");

}

br.close(); //关闭输入流

}

catch (Exception e)

{ // TODO: handle exception

JOptionPane.showMessageDialog(null, website+"爬取源代码失败");

System.out.println("爬取结束");

}

return text;

//返回html代码文本

}

public String getText(String str)

{

Matcher matcher=pattern1.matcher(str);

str=matcher.replaceAll(""); //匹配普通标签

matcher=pattern2.matcher(str);

str=matcher.replaceAll(""); //匹配script标签

matcher=pattern3.matcher(str);

str=matcher.replaceAll(""); //匹配style标签

matcher=pattern4.matcher(str);

str=matcher.replaceAll("\n"); //匹配连续回车或空格

matcher=pattern5.matcher(str);

str=matcher.replaceAll(""); //匹配网页图案出现的乱码

return str; //返回文本

}

public void getLib()

{

JFileChooser fChooser=new JFileChooser(); //文件选择框

int ok=fChooser.showOpenDialog(this);

if(ok!=JFileChooser.APPROVE\_OPTION)

return; //判断是否正常选择

wordList.clear(); //清空之前的记录

sensWord.setText("");

File choosenLib=fChooser.

getSelectedFile(); //获取选择的文件

BufferedReader br=null;

try { //读取选中文件中的记录

InputStreamReader read = new InputStreamReader (new FileInputStream(choosenLib),"UTF-8");

br=new BufferedReader(read);

while(true) {

String str=br.readLine();

if(str==null) break;

wordList.add(str); //添加到记录中

wordNum.add(0); //设置对应的初始值

sensWord.append(str+"\n"); //添加到界面中

}

br.close(); //关闭文件流

}

catch (FileNotFoundException e1) { // TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, "文件不存在");

e1.printStackTrace(); }

catch (IOException e1)

{ // TODO Auto-generated catch block

JOptionPane.showMessageDialog(null, "文件读取失败");

e1.printStackTrace(); }

}

public void showSensword()

{

javax.swing.text.Highlighter hg=textArea.getHighlighter();

//设置文本框的高亮显示

hg.removeAllHighlights(); //清除之前的高亮显示记录

String text=textArea.getText(); //得到文本框的文本

DefaultHighlightPainter painter=new DefaultHighlightPainter(Color.RED);

int size=wordList.size();

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

String str=wordList.get(i) ;

int index=0,temp=0;

while((index=text.indexOf(str,index))>=0) {

try {

wordNum.set(i,++temp);

hg.addHighlight(index, index+str.length(), painter); //高亮显示匹配到的词语

index+=str.length(); //更新匹配条件继续匹配

} catch (BadLocationException e) { // TODO Auto-generated catch block

e.printStackTrace(); } } }

}

public static void main(String[] args) throws IOException

{

new rep();

}

public void actionPerformed(ActionEvent e)

{ // TODO Auto-generated method stub

JButton j=(JButton)e.getSource(); //判断操作来源

if(j==goSpider) { //爬取单个网址

String website=siteField.getText();

new SpiderOne(this,website).start(); }

else if (j==openLib) { //打开敏感词库

getLib(); }

else if (j==match){ //匹配单个网址的敏感词高亮显示

showSensword();

int size=wordList.size();

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

senArea.append(wordList.get(i)+"出现次数为:"+wordNum.get(i)+"\n");

}

}

else if(j==siteLib) { //爬取网址库中的全部网址

spiderAll();

}

else if(j==senin) {

//spiderAll();

String str = JOptionPane.showInputDialog(

this,

"输入关键字:（如果有多个以/号分开）"

);

System.out.println(str);

String[] s =str.split("/");

for(String a:s) {

wordList.add(a); //添加到记录中

wordNum.add(0); //设置对应的初始值

sensWord.append(a+"\n"); }//添加到界面中

}

else if(j==seout) { //爬取网址库中的全部网址

JFileChooser fChooser=new JFileChooser(); //文件选择框

int ok=fChooser.showOpenDialog(this);

if(ok!=JFileChooser.APPROVE\_OPTION)

return; //判断是否正常选择

File choosenLib=fChooser.

getSelectedFile(); //获取选择的文件

try {

String s=senArea.getText();

PrintStream ps=new PrintStream(choosenLib);

ps.println(s);

ps.close(); }

catch (Exception e1) { // TODO: handle exception

JOptionPane.showMessageDialog(null, "写入文件失败"); }

}

}

public void spiderAll() {

if(wordNum.size()<=0) { //判断是否选择了敏感词库

JOptionPane.showMessageDialog(null, "请先选择敏感词库");

return; }

JFileChooser fChooser=new JFileChooser(); //选择网库文件

int ok=fChooser.showOpenDialog(this);

if(ok!=JFileChooser.APPROVE\_OPTION) return;

File file=fChooser.getSelectedFile();

new SpiderAll(this, file).start(); //开启线程爬取 }

}

class SpiderOne extends Thread

{

private String website=null; //网页链接

private MyProgressBar mpb=null; //进度条 //构造函数初始化

public SpiderOne(JFrame fa,String s) {

website=s;

mpb=new MyProgressBar(fa, "Spiding"); }

public void run() {

if(website.length()<=0)

{ //判断网址是否正常

JOptionPane.showMessageDialog(null, "网址不能为空");

return; }

htmlArea.setText(""); //清除文本

textArea.setText("");

mpb.setText("爬取"+website+"中..."); //设置进度条界面标题

mpb.setVisible(true); //显示进度条

String html=getHtml(website); //开始爬取

mpb.dispose(); //关闭进度条

if(html.length()>0) { //若爬取正常

JOptionPane.showMessageDialog(null, "爬取完毕"); //提示完成

htmlArea.append(html); //显示html源代码

String text=getText(html); //匹配网页文本

textArea.append(text); //显示网页文本 //爬取网址库

}

}

}

class SpiderAll extends Thread

{

private File file=null; //网址库文本文件

private MyProgressBar mpb=null; //进度条 //构造函数初始化

public SpiderAll(JFrame fa,File f) {

file=f;

mpb=new MyProgressBar(fa, "Spiding"); }

public void run() {

try { //读取网址库中的网址

InputStreamReader read = new InputStreamReader (new FileInputStream(file),"UTF-8");

BufferedReader brr=new BufferedReader(read); //将匹配数据写入文本中

PrintStream ps=new PrintStream(new File("E:\\java\\大二下课设\\d2.txt"));

ps.println("敏感词记录如下:");

int size=wordList.size();

mpb.setVisible(true); //显示进度条

while(true) {

String website=brr.readLine();

System.out.println(website);

if(website==null) break;

mpb.setText("爬取"+website+"中..."); //设置进度条界面标题

ps.println(website+"数据如下: ");

String html=getHtml(website); //获取html代码

String text=getText(html); //匹配网页文本

for(int i=0;i<size;i++) { //在网页文本中进行匹配

String word=wordList.get(i);

int index=0,account=0,len=word.length();

while((index=text.indexOf(word,index))>=0) {

account++;

int temp=wordNum.get(i); //更新数据

wordNum.set(i,++temp);

index+=len; //更新匹配条件

}

ps.println(word+" 出现 "+account+"次"); //写入当前数据

}

ps.println(); }

brr.close(); //关闭文件流

System.out.println("爬取完毕");

ps.println("总数据如下: "); //写入总数据

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

ps.println(wordList.get(i)+" 出现 "+wordNum.get(i)+"次"); }

ps.close(); //关闭文件流

JOptionPane.showMessageDialog(null, "爬取完毕！请打开文件查看!"); }

catch (Exception e) { // TODO: handle exception

JOptionPane.showMessageDialog(null, "爬取失败"); }

finally {

mpb.dispose(); //关闭进度条

} } }

class MyProgressBar extends JDialog {

private static final long serialVersionUID = 1L;

private JPanel jPanel=new JPanel();

private JProgressBar jpb=new JProgressBar(); //进度条

private JLabel curSpiding=new JLabel(); //显示当前网址 //构造函数初始化,设置父窗口以及标题

public MyProgressBar(JFrame f,String title) {

super(f,title);

this.setLocation(f.getWidth()/2+(int)f.getLocation().getX()/2-80, f.getHeight()/2+(int)f.getLocation().getY()/2-30);

this.setSize(400,100);

this.setDefaultCloseOperation(DISPOSE\_ON\_CLOSE);

jpb.setString("玩命加载中...");

jpb.setIndeterminate(true); //设置进度条为不确定模式

jpb.setStringPainted(true);

jpb.setBorderPainted(false);

jpb.setForeground(Color.RED); //设置进度条颜色

jpb.setBackground(Color.WHITE); //设置背景

curSpiding.setPreferredSize(new Dimension(400, 30)); //界面布局

jPanel.setLayout(new BorderLayout());

jPanel.add(curSpiding,BorderLayout.NORTH);

jPanel.add(jpb,BorderLayout.CENTER);

this.add(jPanel); } //设置当前爬取网址接口

public void setText(String text) {curSpiding.setText(text);}

}

}