ЛАБОРАТОРНАЯ РАБОТА **№1**

**Работа с Web-ресурсами**

**Цель работы.** Познакомиться с возможностями поиска информации в удаленных URL-ресурсах.

**Краткое теоретическое содержание**. В теоретическом материале представлены сведения по работе с URL−ресурсами. Нас интересует поиск различных информационных документов по набору ключевых слов. В настоящей работе следует создать небольшое приложение, которое позволяет ввести ключевые слова и найти любой (один) html-документ соответствующий (или, как говорят, релевантный) введенному набору ключевых слов. Исходную форму для поисковика можно реализовать таким образом

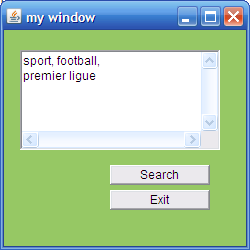


Рисунок 1. Окно приложения

В текстовой области вводим ключевые слова. Нажимаем кнопку Search. Выполняется поиск по документам, записанным в пакете source\_html (рисунок 2).

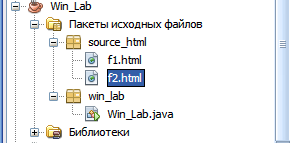


Рисунок 2. Определение области поиска

Найденный документ следует открыть в броузере (например, Opera) , как показано на рисунке 3.

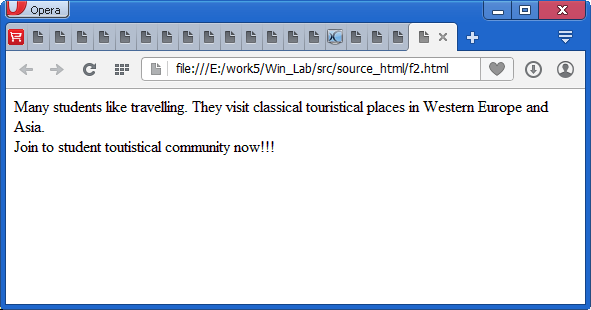


Рисунок 3. Открытие документа в броузере

Необходимо создать простое приложение Java.

Помещаем «заготовку» приложения, которую следует доработать.

package win\_lab;

import java.awt.\*;

import java.awt.event.\*;

import java.io.\*;

import java.util.\*;

import java.io.BufferedInputStream;

import java.io.File;

import java.io.FileOutputStream;

import java.io.IOException;

import java.net.URL;

import java.net.URLConnection;

import java.nio.charset.MalformedInputException;

public class Win\_Lab extends Frame implements ActionListener{

Button bex=new Button("Exit");

Button sea=new Button("Search");

TextArea txa = new TextArea();

public Win\_Lab()

{

super("my window");

setLayout(null);

setBackground(new Color(150,200,100));

setSize(450,250);

add(bex);

add(sea);

add(txa);

bex.setBounds(110,190,100,20);

bex.addActionListener(this);

sea.setBounds(110,165,100,20);

sea.addActionListener(this);

txa.setBounds(20,50,300,100);

this.show();

this.setLocationRelativeTo(null);

}

public void actionPerformed(ActionEvent ae)

{

if(ae.getSource()==bex)

System.exit(0);

else

if (ae.getSource()==sea)

{

String [] keywords=txa.getText().split(",");

for (int j=0;j<keywords.length;j++)

{

System.out.println(keywords[j]);

}

File f = new File("e:/work5/Win\_lab/src/source\_html");

ArrayList<File> files =

new ArrayList<File>(Arrays.asList(f.listFiles()));

txa.setText("");

for (File elem : files)

{

int zcoincidence = test\_url(elem,keywords);

txa.append("\n"+elem+" :"+zcoincidence);

}}}

public static int test\_url(File elem, String [] keywords)

{

int res=0;

URL url = null;

URLConnection con = null;

int i;

try

{

String ffele=""+elem;

url = new URL("file:/"+ffele.trim());

con = url.openConnection();

File file = new File("e:/work5/rezult.html");

BufferedInputStream bis = new BufferedInputStream(

con.getInputStream());

BufferedOutputStream bos = new BufferedOutputStream(

new FileOutputStream(file));

String bhtml=""; //file content in byte array

while ((i = bis.read()) != -1) {

bos.write(i);

bhtml+=(char)i;

}

bos.flush();

bis.close();

String htmlcontent=

(new String(bhtml)).toLowerCase(); //file content in string

System.out.println("New url content is: "+htmlcontent);

for (int j=0;j<keywords.length;j++)

{

if(htmlcontent.indexOf(keywords[j].trim().toLowerCase())>=0)

res++;

}}

catch (MalformedInputException malformedInputException)

{

System.out.println("error "+malformedInputException.getMessage());

return -1;

}

catch (IOException ioException)

{

System.out.println("error "+ioException.getMessage());

return -1;

}

catch(Exception e)

{

System.out.println("error "+e.getMessage());

return -1;

}

return res;

}

public static void main(String[] args)

{

new Win\_Lab();

}

}

В конструкторе создается сама форма и ее визуальные компоненты (кнопки и текстовая область). Нас в первую очередь интересует кнопка поиска (sea). Ее обработчик такой

if (ae.getSource()==sea)

{

String [] keywords=txa.getText().split(",");

for (int j=0;j<keywords.length;j++)

{

System.out.println(keywords[j]);

}

File f = new File("e:/work5/Win\_lab/src/source\_html");

ArrayList<File> files =

new ArrayList<File>(Arrays.asList(f.listFiles()));

txa.setText("");

for (File elem : files)

{

int zcoincidence = test\_url(elem,keywords);

txa.append("\n"+elem+" :"+zcoincidence);

}}

Сначала формируем массив ключевых слов (которые должны разделяться запятыми) :

String [] keywords=txa.getText().split(",");

Затем формируем список файлов в указываемой директории:

File f = new File("e:/work5/Win\_lab/src/source\_html");

ArrayList<File> files =

new ArrayList<File>(Arrays.asList(f.listFiles()));

После этого для каждого файла в цикле получаем с помощью метода test\_url число совпадений. Если представленную выше «заготовку» программы выполнить с ключевыми словами students, like, travelling, то получим следующий результат в окне программы (рисунок 4).

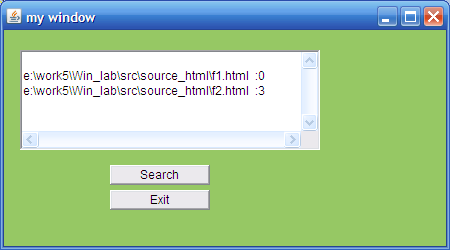


Рисунок 4. Результат работы программы

Напротив имени каждого файла указано число совпадений с ключевыми словами.

**Задание**.

1. Подготовить собственный набор html-документов (согласовать с преподавателем).
2. Завершить заготовку программы так, чтобы файл с наибольшим числом совпадений открывался броузером.

***Порядок выполнения работы***

1. Изучить теоретическую часть.
2. Получить индивидуальное задание у преподавателя из прилагаемого списка.
3. Создать и отладить приложение.
4. Написать отчет.
5. Защитить работу.

***Индивидуальные задания***

**Вариант 1.**

Выполнить поиск цены запрашиваемого товара (товар вводится по названию). Список товаров представлен в файле .html

**Вариант 2.**

Выполнить поиск названия лекарства (вводится набор из 1-3 симптомов). Список лекарств и симптомов представлен в текстовом файле на сервере

**Вариант 3.**

Выполнить поиск имени автора (вводится набор из 1-3 слов из названия книги). Список книг и авторов представлен в текстовом файле на сервере

**Вариант 4.**

Выполнить поиск столицы государства (вводится название государства, возможно с ошибкой). Список государств и столиц представлен в текстовом файле на сервере

**Вариант 5.**

Выполнить поиск породы собаки (вводится набор из 1-3 слов из качеств собаки). Список собак и их характеристик представлен в текстовом файле на сервере

**Контрольные вопросы**.

1. Что такое URL?
2. Как программно получить список всех файлов в папке?
3. Как программно запустить броузер и открыть html-файл?
4. Объяснить использование потоковых классов в вашем приложении.