# 实验报告三

编写一个英文词频统计的程序。统计时不区分大小写，忽略所有的数字和标点符号。要求能处理普通的txt文本和html文本。在处理html文本时，要求过滤掉所有的html标签。 （10分）

1. 题目分析
2. 代码实现

package test03;

import java.io.BufferedReader;//缓冲区读取内容，避免中文乱码

import java.io.FileNotFoundException;//拒绝访问解决方法

import java.io.FileReader;//读取文件

import java.io.IOException;//处理读取操作失误

import java.util.HashMap;//哈希表

import java.util.Iterator;//迭代器，对文件内容进行遍历

import java.util.StringTokenizer;//字符串分解器，根据分隔符把字符串分割成标记（Token），然后按照请求返回各个标记

public class frequency {

public static void main(String[] args) {

//用HashMap存放<单词:词频>这样一个映射关系

HashMap<String, Integer> hashMap=new HashMap<String, Integer>();

//用正则表达式来过滤字符串中的所有标点符号

String regex = "[【】、.。,\"!--;:?\'\\]]";

try {

//读取要处理的文件

BufferedReader br=new BufferedReader(new FileReader("D:\\java\_project\\English composition.txt"));

String value;

while((value=br.readLine())!=null){

value=value.replaceAll(regex, " ");

//使用StringTokenizer来分词(StringTokenizer详见JDK文档)

StringTokenizer tokenizer = new StringTokenizer(value);

while(tokenizer.hasMoreTokens()){

String word=tokenizer.nextToken();

if(!hashMap.containsKey(word)){

hashMap.put(word, new Integer(1));

}else{

int k=hashMap.get(word).intValue()+1;

hashMap.put(word, new Integer(k));

}

}

}

//遍历HashMap,输出结果

Iterator iterator=hashMap.keySet().iterator();

while(iterator.hasNext()){

String word=(String) iterator.next();

System.out.println(word+":\t"+hashMap.get(word));

}

} catch (FileNotFoundException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

}

}

1. 运行结果

