|  |
| --- |
| package soft; |
|  |  |
|  | import java.io.BufferedReader; |
|  | import java.io.File; |
|  | import java.io.FileNotFoundException; |
|  | import java.io.FileReader; |
|  | import java.io.FileWriter; |
|  | import java.io.IOException; |
|  | import java.util.ArrayList; |
|  | import java.util.Collections; |
|  | import java.util.Comparator; |
|  | import java.util.HashMap; |
|  | import java.util.List; |
|  | import java.util.Map; |
|  | import java.util.Map.Entry; |
|  | import java.util.Scanner; |
|  | import java.util.StringTokenizer; |
|  | import java.util.TreeMap; |
|  | import java.util.regex.Matcher; |
|  | import java.util.regex.Pattern; |
|  |  |
|  |  |
|  | public class zuoye4 |
|  | { |
|  | //函数说明：正则表达式法判断字符是否为数字 |
|  | public static boolean isNumeric(CharSequence charSequence) |
|  | { |
|  | Pattern pattern = Pattern.compile("[0-9]\*"); |
|  | Matcher isNum = pattern.matcher(charSequence); |
|  | if( !isNum.matches() ) |
|  | { |
|  | return false; |
|  | } |
|  | return true; |
|  | } |
|  | // TODO Auto-generated method stub |
|  | public static void main(String[] args) throws Exception |
|  | { |
|  |  |
|  |  |
|  | Map<String,Integer> map=new HashMap<String,Integer>(); |
|  | //读取文档 |
|  | File file = new File("e:\\文档\\Gone\_with\_the\_wind.txt"); |
|  | FileReader reader = new FileReader(file); |
|  | int fileLen = (int)file.length(); |
|  | char[] chars = new char[fileLen]; |
|  | reader.read(chars); |
|  | String es= String.valueOf(chars); |
|  |  |
|  | //为了不区分大小写，把所有字母转换成小写 |
|  | es= es.toLowerCase(); |
|  | //这个类会将字符串分解 |
|  | StringTokenizer token=new StringTokenizer(es); |
|  | //遍历 |
|  | while(token.hasMoreTokens()) |
|  | { |
|  | //按照,[空格] ? . !，双引号，单引号，换行符去分割 |
|  | String word=token.nextToken(", ?.!:\"\"\''\n"); |
|  | //取单词的第一个字符，调用函数isNumeric判断word的首字符是否为数字,并且判断单词长度是否大于3 |
|  | if( !isNumeric(word.subSequence(0,1))&&word.length()>=4) |
|  | { |
|  | //统计单词的个数 |
|  | if(map.containsKey(word)) |
|  | { |
|  | int count=map.get(word); |
|  | //如果HashMap已有这个单词，则使它的数量加1 |
|  | map.put(word, count+1); |
|  | } |
|  | //如果没有第二个相同单词，数量为1 |
|  | else |
|  | map.put(word, 1); |
|  | } |
|  |  |
|  | } |
|  |  |
|  |  |
|  | List<Map.Entry<String, Integer>> infoids = new ArrayList<Map.Entry<String, Integer>>(map.entrySet()); |
|  |  |
|  | //排序 |
|  | Collections.sort(infoids, new Comparator<Map.Entry<String, Integer>>() |
|  | { |
|  | //如果单词频率不同，降序排序，频率相同，按词典顺序排序 |
|  | public int compare(Map.Entry<String, Integer> o1, Map.Entry<String, Integer> o2) |
|  | { |
|  | if(o1.getValue()<o2.getValue()) |
|  |  |
|  | return 1; |
|  | else if(o1.getValue()>o2.getValue()) |
|  | return -1; |
|  | else |
|  | return(o1.getKey().compareTo(o2.getKey())); |
|  | } |
|  | }); |
|  |  |
|  |  |
|  |  |
|  | //文件写入 |
|  | FileWriter writer = new FileWriter("E:\\Result.txt.txt"); |
|  | for (int i = 0; i <infoids.size(); i++) |
|  | { |
|  | //一次写入一个单词以及对应的频率 |
|  | Entry<String, Integer> id =infoids.get(i); |
|  | writer.write(id.getKey()+' '+id.getValue()+"\r\n"); |
|  |  |
|  | } |
|  | //全部写入 |
|  | writer.flush(); |
|  | writer.close(); |
|  |  |
|  |  |
|  | } |
|  |  |
|  | } |
|  |  |
|  |  |
|  |  |