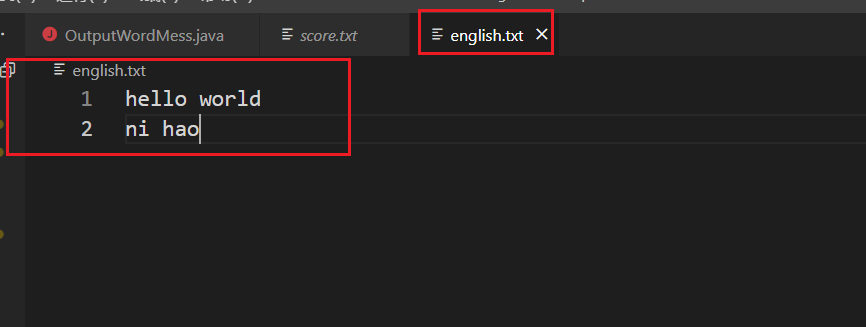
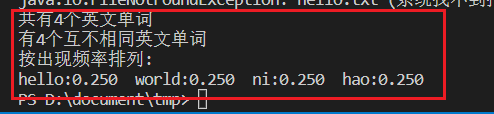
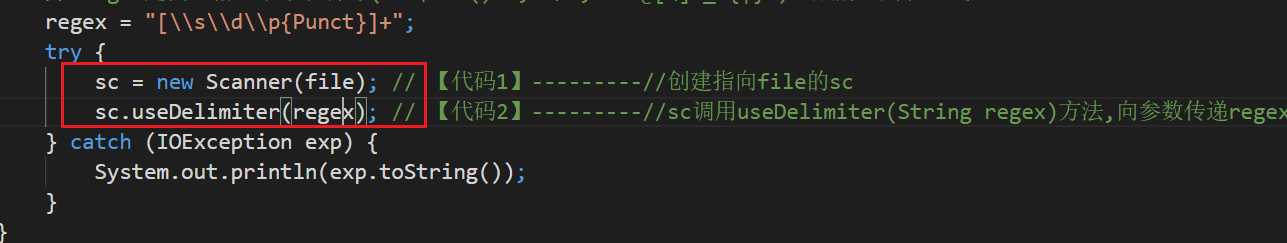
## 结果截图

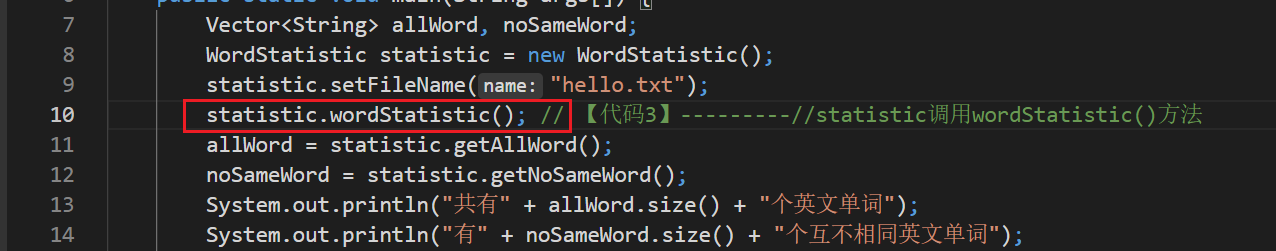
压缩包里没有给english.txt文件，自己建了一个。





## 代码截图





## 源码

package chapter10\_2;

import java.util.\*;

public class OutputWordMess {

public static void main(String args[]) {

Vector<String> allWord, noSameWord;

WordStatistic statistic = new WordStatistic();

statistic.setFileName("hello.txt");

statistic.wordStatistic(); // 【代码3】---------//statistic调用wordStatistic()方法

allWord = statistic.getAllWord();

noSameWord = statistic.getNoSameWord();

System.out.println("共有" + allWord.size() + "个英文单词");

System.out.println("有" + noSameWord.size() + "个互不相同英文单词");

System.out.println("按出现频率排列:");

int count[] = new int[noSameWord.size()];

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

String s1 = noSameWord.elementAt(i);

for (int j = 0; j < allWord.size(); j++) {

String s2 = allWord.elementAt(j);

if (s1.equals(s2))

count[i]++;

}

}

for (int m = 0; m < noSameWord.size(); m++) {

for (int n = m + 1; n < noSameWord.size(); n++) {

if (count[n] > count[m]) {

String temp = noSameWord.elementAt(m);

noSameWord.setElementAt(noSameWord.elementAt(n), m);

noSameWord.setElementAt(temp, n);

int t = count[m];

count[m] = count[n];

count[n] = t;

}

}

}

for (int m = 0; m < noSameWord.size(); m++) {

double frequency = (1.0 \* count[m]) / allWord.size();

System.out.printf("%s:%-7.3f", noSameWord.elementAt(m), frequency);

}

}

}

//

package chapter10\_2;

import java.io.\*;

import java.util.\*;

public class WordStatistic {

Vector<String> allWord, noSameWord;

File file = new File("english.txt");

Scanner sc = null;

String regex;

WordStatistic() {

allWord = new Vector<String>();

noSameWord = new Vector<String>();

// regex是由空格、数字和符号(!"#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~)组成的正则表达式

regex = "[\\s\\d\\p{Punct}]+";

try {

sc = new Scanner(file); // 【代码1】---------//创建指向file的sc

sc.useDelimiter(regex); // 【代码2】---------//sc调用useDelimiter(String regex)方法,向参数传递regex

} catch (IOException exp) {

System.out.println(exp.toString());

}

}

void setFileName(String name) {

file = new File(name);

try {

sc = new Scanner(file);

sc.useDelimiter(regex);

} catch (IOException exp) {

System.out.println(exp.toString());

}

}

public void wordStatistic() {

try {

while (sc.hasNext()) {

String word = sc.next();

allWord.add(word);

if (!noSameWord.contains(word))

noSameWord.add(word);

}

} catch (Exception e) {

}

}

public Vector<String> getAllWord() {

return allWord;

}

public Vector<String> getNoSameWord() {

return noSameWord;

}

}