Java面向对象程序设计课程设计实验报告六

实验人：王维

学号：320210942931

班级：2021级计算机科学与技术二班

一：补全实验题

* 10.3.1（3）
* 源码：

|  |
| --- |
| package ExperimentSourceCode\_6**.**fill**;**  **import** java**.**util**.\*;**  class TreesetTest**{**  public static void main**(**String args**[])** **{**  TreeSet**<**Student**>** mytree**=**  **new** TreeSet**<**Student**>(new** Comparator**<**Student**>()** **{**  public int compare**(**Student a**,**Student b**)** **{**  **return** a**.**compareTo**(**b**);**  **}**  **});**  Student st1**,**st2**,**st3**,**st4**;**  st1**=new** Student**(**90**,**"zhan ying"**);**  st2**=new** Student**(**66**,**"wang heng"**);**  st3**=new** Student**(**86**,**"Liuh qing"**);**  st4**=new** Student**(**76**,**"yage ming"**);**  mytree**.**add**(**st1**);**  mytree**.**add**(**st2**);**  mytree**.**add**(**st3**);**  mytree**.**add**(**st4**);**  Iterator**<**Student**>** te**=**mytree**.**iterator**();**  **while(**te**.**hasNext**())** **{**  Student stu**=**te**.**next**();**  System**.**out**.**println**(**""**+**stu**.**name**+**" "**+**stu**.**english**);**  **}**  **}**  **}**  class Student **implements** Comparable **{**  int english**=**0**;**  String name**;**  Student**(**int e**,**String n**)** **{**  english**=**e**;**name**=**n**;**  **}**  public int compareTo**(**Object b**)** **{**  Student st**=(**Student**)**b**;**  **return** **(this.**english**-**st**.**english**);**  **}**  **}** |

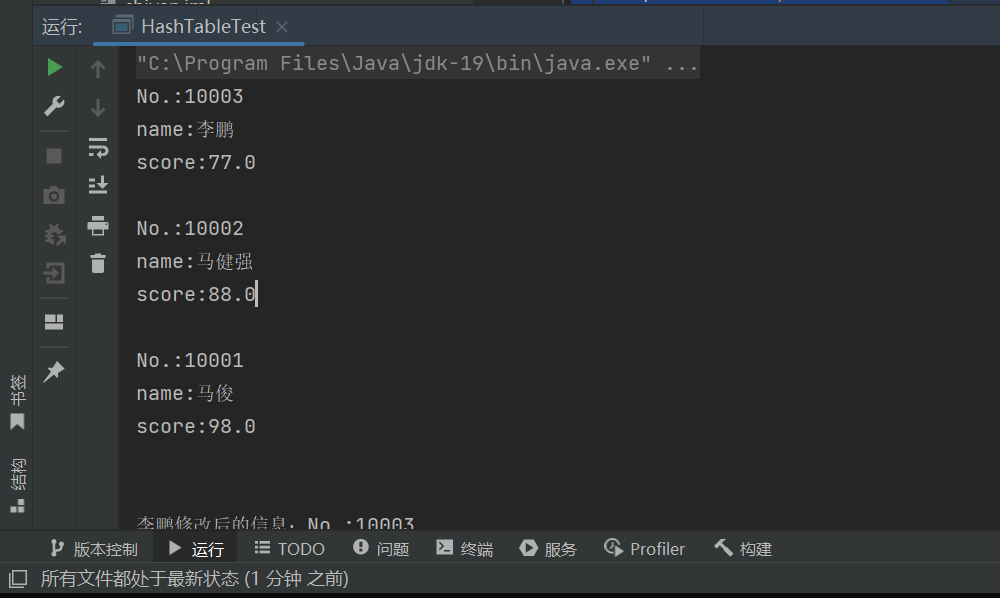
* 运行结果：



* 10.3.1（4）
* 源码：

|  |
| --- |
| package ExperimentSourceCode\_6**;**  **import** java**.**util**.\*;**  class Student **{**  String name**;**  int number**;**  float score**;**  Student**(**String name**,**int number**,**float score**){**  **this.**name**=**name**;**  **this.**number**=**number**;**  **this.**score**=**score**;**  **}**  public String toString**(){**  **return** "No.:"**+**number**+**"\nname:"**+**name**+**"\nscore:"**+**score**+**"\n"**;**  **}**  **}**  class HashTableTest **{**  public static void main**(**String args**[])** **{**  Hashtable h **=** **new** Hashtable**();**  Enumeration e**;**  Student stu**;**  String str**;**  h**.**put**(**"10001"**,** **new** Student**(**"马俊"**,**10001**,**98**));**  h**.**put**(**"10002"**,** **new** Student**(**"马健强"**,**10002**,**88**));**  h**.**put**(**"10003"**,** **new** Student**(**"李鹏"**,**10003**,**77**));**  e **=** h**.**keys**();**  **while** **(**e**.**hasMoreElements**())** **{**  str **=** **(**String**)**e**.**nextElement**();**  System**.**out**.**println**(**""**+(**Student**)**h**.**get**(**str**));**  **}**  System**.**out**.**println**();**  float score**=** **((**Student**)** h**.**get**(**"10003"**)).**score**;**  h**.**put**(**"10003"**,** **new** Student**(**"李鹏"**,**10003**,**score**+**15**));**  System**.**out**.**println**(**"李鹏修改后的信息：" **+** **(**Student**)**h**.**get**(**"10003"**));**  **}**  **}** |

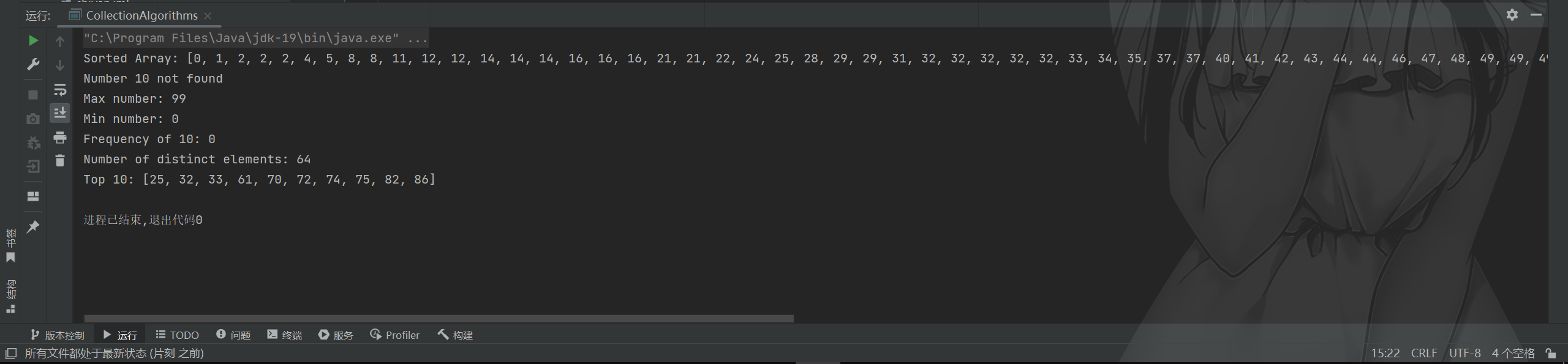
* 运行结果：



* 10.3.2（3）
* 源码：

|  |
| --- |
| package ExperimentSourceCode\_6**.**fill**;**  **import** java**.**util**.\*;**  public class CollectionAlgorithms **{**  public static void main**(**String**[]** args**){**  List**<**Integer**>** list**=new** ArrayList**<>();**  **for(**int i**=**0**;**i**<**100**;**i**++){**  list**.**add**((**int**)(**Math**.**random**()\***100**));**//添加元素  **}**  Collections**.**sort**(**list**);** //对集合进行排序  System**.**out**.**println**(**"Sorted Array: "**+**list**);**  int testNumber**=**10**;**  int index**=**Collections**.**binarySearch**(**list**,**testNumber**);**  **if(**index**>=**0**){**  System**.**out**.**println**(**"Number "**+**testNumber**+**" found at index: "**+**index**);**  **}else{**  System**.**out**.**println**(**"Number "**+**testNumber**+**" not found"**);**  **}**  System**.**out**.**println**(**"Max number: "**+**Collections**.**max**(**list**));**  System**.**out**.**println**(**"Min number: "**+**Collections**.**min**(**list**));**  System**.**out**.**println**(**"Frequency of "**+**testNumber**+**": "**+**Collections**.**frequency**(**list**,**testNumber**));**//testNumber出现的频率  Set**<**Integer**>** sortedList**=new** HashSet**<>();**  sortedList**.**addAll**(**list**);**  System**.**out**.**println**(**"Number of distinct elements: "**+**sortedList**.**size**());**  list**.**clear**();**  list**.**addAll**(**sortedList**);**  Collections**.**shuffle**(**list**);** //搅乱集合元素顺序  List**<**Integer**>** topTenList**=**list**.**subList**(**0**,**10**);**  Collections**.**sort**(**topTenList**);**  System**.**out**.**println**(**"Top 10: "**+**topTenList**);**  **}**  **}** |

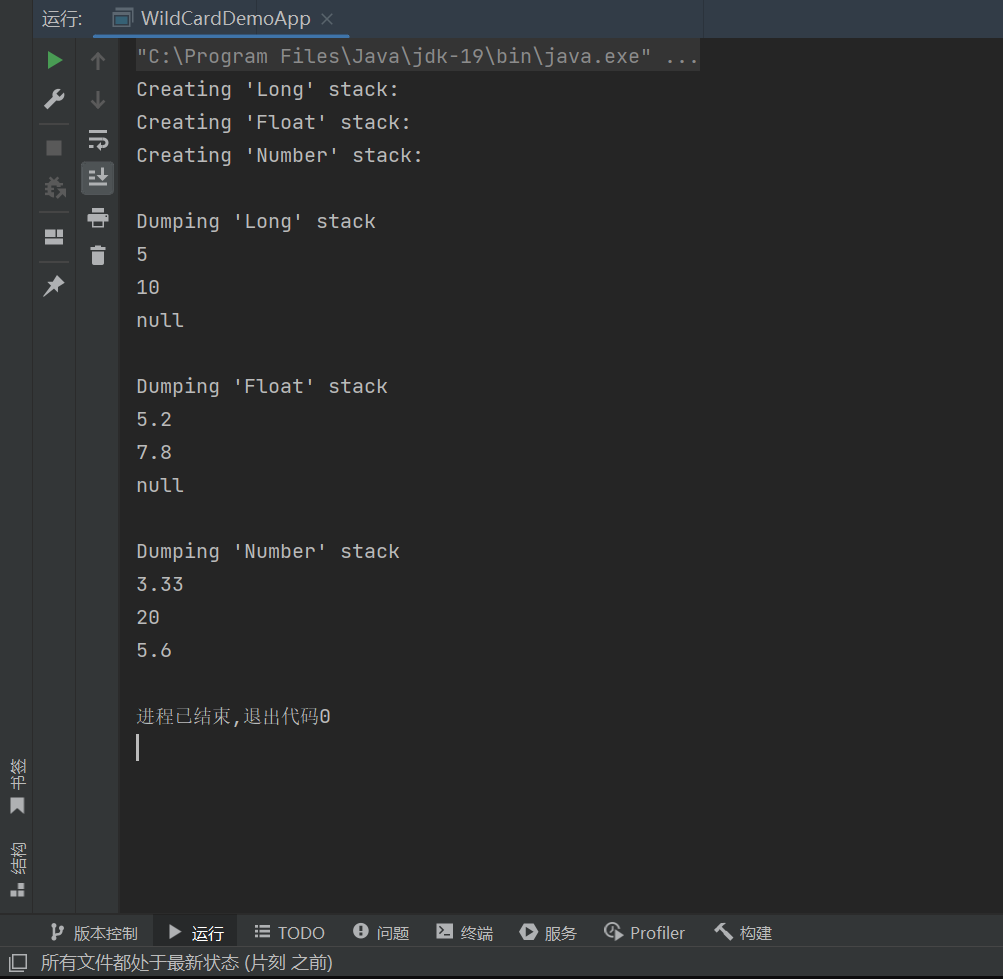
* 运行结果：



* 10.3.2（4）
* 源码：

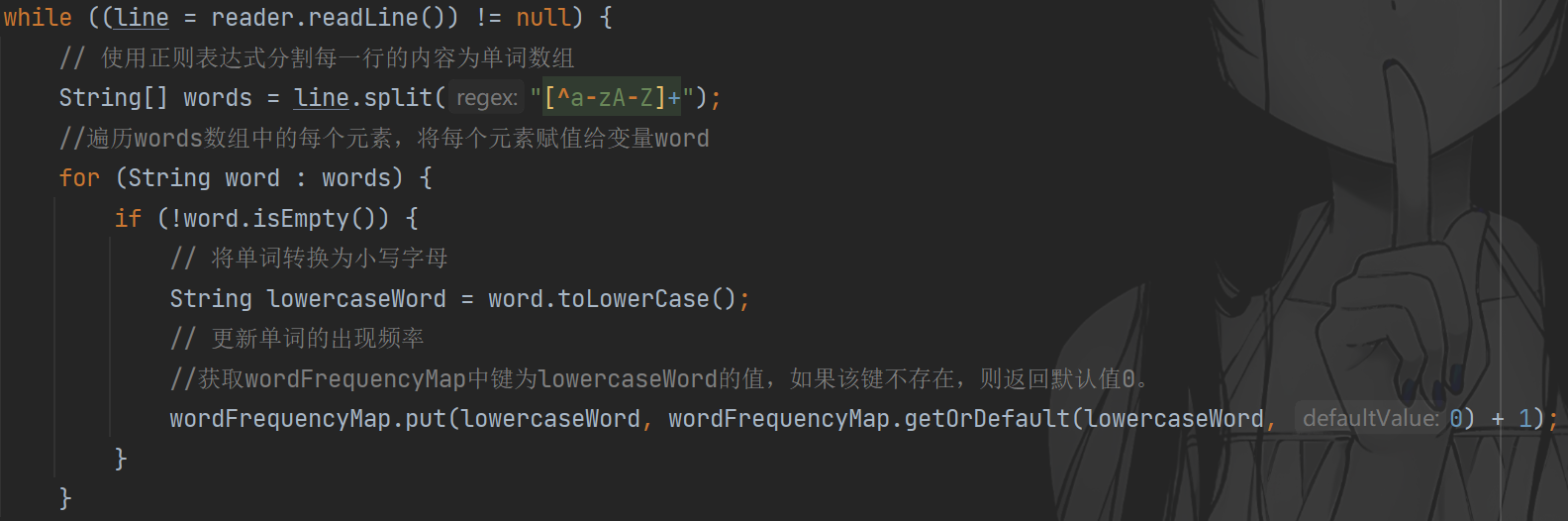
|  |
| --- |
| package ExperimentSourceCode\_6**.**fill**;**  public class WildCardDemoApp **{**  public static void main**(**String args**[]){**  System**.**out**.**println**(**"Creating 'Long' stack:"**);**  NumberStack**<**Long**>** longStack**=new** NumberStack**<**Long**>();**  longStack**.**push**(**5L**);**  longStack**.**push**(**10L**);**  System**.**out**.**println**(**"Creating 'Float' stack:"**);**  NumberStack**<**Float**>** floatStack**=new** NumberStack**<**Float**>();**  floatStack**.**push**(**5.2f**);**  floatStack**.**push**(**7.8f**);**  System**.**out**.**println**(**"Creating 'Number' stack:"**);**  NumberStack**<**Number**>** numberStack**=new** NumberStack**<**Number**>();**  numberStack**.**push**(**3.33**);**  numberStack**.**push**(**20L**);**  numberStack**.**push**(**5.6f**);**  System**.**out**.**println**(**"\nDumping 'Long' stack"**);**  dumpStack**(**longStack**);**  System**.**out**.**println**(**"\nDumping 'Float' stack"**);**  dumpStack**(**floatStack**);**  System**.**out**.**println**(**"\nDumping 'Number' stack"**);**  dumpStack**(**numberStack**);**  **}**  static void dumpStack**(**NumberStack**<?>** stack**){**  **for(**Number n**:**stack**.**getStack**()){**  System**.**out**.**println**(**n**);**  **}**  **}**  **}**  class NumberStack**<**T **extends** Number**>** **{**  private Number stack**[]=new** Number**[**3**];**  private int ptr**=-**1**;**  public Number**[]** getStack**(){**  **return** stack**;**  **}**  void push**(**T data**){**  ptr**++;**  stack**[**ptr**]=**data**;**  **}**  T pop**(){**  **return** **(**T**)** stack**[**ptr**--];**  **}**  **}** |

* 运行结果：

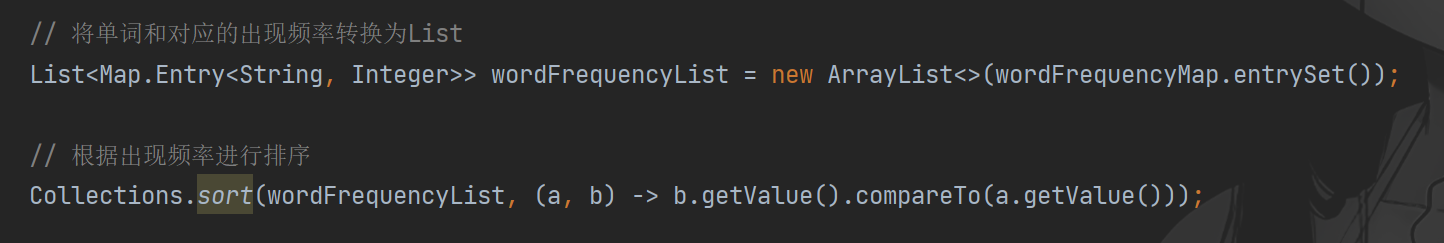


二．设计实验：请将一本英文小说或一篇英文文章，整理成文本文件，然后试编写一个Java程序，用来统计单词的出现频率，建议使用特定的数据集合，并说明算法原理。

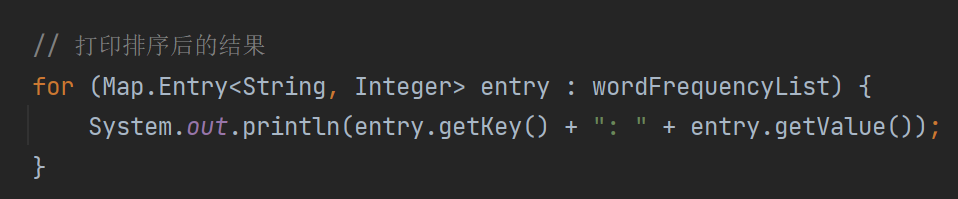
1. **整体思路：**由题，将一个文本文件内的所有英文单词读取，并统计每个单词的出现频率，很容易想到一个英文单词对应一个频率，即HashMap中的value – key 用来整理数据。
2. **具体步骤：**
3. 读取文件：异常处理。
4. 遍历文件：读取每一行的数据并由正则表达式分割出其中的英文字符串，存入HashMap中，并将key+1。
5. 关闭文件：判断异常。
6. **具体实现细节：**
7. 遍历文件：



1. 转换为List后进行排序：使用了Lambda表达式，按字典序对ab的值进行比较，在sort方法中默认会依次将表中元素依次赋值给ab



1. 遍历并打印结果：



4．**源代码：**

|  |
| --- |
| package ExperimentSourceCode\_6**.**NovelStatistics**;**  **import** java**.**io**.**BufferedReader**;**  **import** java**.**io**.**FileReader**;**  **import** java**.**util**.\*;**  public class WordFrequencyCounter **{**  public static void main**(**String**[]** args**)** **{**  String filePath **=** "src/ExperimentSourceCode\_6/data/text.txt"**;**  // 创建一个HashMap来存储单词和对应的出现频率  Map**<**String**,** Integer**>** wordFrequencyMap **=** **new** HashMap**<>();**  **try** **{**  // 读取文件  BufferedReader reader **=** **new** BufferedReader**(new** FileReader**(**filePath**));**  String line**;**  **while** **((**line **=** reader**.**readLine**())** **!=** **null)** **{**  // 使用正则表达式分割每一行的内容为单词数组  String**[]** words **=** line**.**split**(**"[^a-zA-Z]+"**);**  //遍历words数组中的每个元素，将每个元素赋值给变量word  **for** **(**String word **:** words**)** **{**  **if** **(!**word**.**isEmpty**())** **{**  // 将单词转换为小写字母  String lowercaseWord **=** word**.**toLowerCase**();**  // 更新单词的出现频率  //获取wordFrequencyMap中键为lowercaseWord的值，如果该键不存在，则返回默认值0。  wordFrequencyMap**.**put**(**lowercaseWord**,** wordFrequencyMap**.**getOrDefault**(**lowercaseWord**,** 0**)** **+** 1**);**  **}**  **}**  **}**  reader**.**close**();**  **}** **catch** **(**Exception e**)** **{**  e**.**printStackTrace**();**  **}**  // 将单词和对应的出现频率转换为List  List**<**Map**.**Entry**<**String**,** Integer**>>** wordFrequencyList **=** **new** ArrayList**<>(**wordFrequencyMap**.**entrySet**());**  // 根据出现频率进行排序  Collections**.**sort**(**wordFrequencyList**,** **(**a**,** b**)** **->** b**.**getValue**().**compareTo**(**a**.**getValue**()));**  // 打印排序后的结果  **for** **(**Map**.**Entry**<**String**,** Integer**>** entry **:** wordFrequencyList**)** **{**  System**.**out**.**println**(**entry**.**getKey**()** **+** ": " **+** entry**.**getValue**());**  **}**  **}**  **}** |

1. **运行结果：**

