Homework Turnin

Email: rgalanos@fcps.edu
Section: 6G
Course: TJHSST APCS 2016-17
Assignment: 05-05

Receipt ID: 10ef7cf24777d3d577a4dfe16dd8b72d

Turnin Successful!

The following file(s) were received:

IndexMaker.java (4011 bytes)

```java
//  Name:      date:
//  This program takes a text file, creates an index (by line numbers)
//  for all the words in the file and writes the index
//  into the output file.  The program prompts the user for the file names.

import java.util.*;
import java.io.*;

public class IndexMaker
{
   public static void main(String[] args) throws IOException
   {
      Scanner keyboard = new Scanner(System.in);
      System.out.print("\nEnter input file name: ");
      String inFileName = keyboard.nextLine().trim();
      Scanner inputFile = new Scanner(new File(inFileName));
      String outFileName = "fishIndex.txt";
      PrintWriter outputFile = new PrintWriter(new FileWriter(outFileName));
      indexDocument(inputFile, outputFile);
      inputFile.close();
      outputFile.close();
      System.out.println("Done.");
   }
   public static void indexDocument(Scanner inputFile, PrintWriter outputFile)
   {
      DocumentIndex index = new DocumentIndex();
      String line = null;
      int lineNum = 0;
      while(inputFile.hasNextLine())
      {
         lineNum++;
         index.addAllWords(inputFile.nextLine(), lineNum);
      }
      for(IndexEntry entry : index)
         outputFile.println(entry);
   }
}
class DocumentIndex extends ArrayList<IndexEntry>
{

      //constructors
   public DocumentIndex()
   {
      super();
   }

   public DocumentIndex(int x)
   {
      super(x);
   }
   /** calls foundOrInserted, which returns an index.  At that position,
   updates that IndexEntry's list of line numbers with num. */
   public void addWord(String word, int num)
   {
      this.get(foundOrInserted(word)).add(num);
   }

    /** extracts all the words from str, skipping punctuation and whitespace
    and for each word calls addWord(word, num).  A good way to skip punctuation
    and whitespace is to use String's split method, e.g., split("[., \"!?]") */
   public void addAllWords(String str, int num)
   {
      String[] words = str.split("[., \"!?]");
      for(String x: words)
         if(!x.equals(""))
            addWord(x, num);
   }

    /** traverses this DocumentIndex and compares word to the words in the
    IndexEntry objects in this list, looking for the correct position of
    word. If an IndexEntry with word is not already in that position, the
    method creates and inserts a new IndexEntry at that position. The
    method returns the position of either the found or the inserted
    IndexEntry.*/
   private int foundOrInserted(String word)
   {
      for(int index = 0; index<this.size();index++)
      {

         if(this.get(index).getWord().equalsIgnoreCase(word))
            return index;
         else if(this.get(index).getWord().compareTo(word.toUpperCase())>0)
         {
            this.add(index, new IndexEntry(word));
            return index;
         }
      }
      this.add(this.size(), new IndexEntry(word));
      return this.size()-1;


   }
}

class IndexEntry implements Comparable<IndexEntry>
{
     //fields
   private ArrayList<Integer> numList;
   private String word;

     //constructors
   public IndexEntry(String w)
   {
      word = w.toUpperCase();
      numList = new ArrayList<Integer>();
   }
     /**  appends num to numsList, but only if it is not already in that list.
          */
   public void add(int num)
   {
      if(numList.size()<1)
         numList.add(num);
      else if(numList.get(numList.size()-1)!=num)
         numList.add(num);
   }

      /** this is a standard accessor method  */
   public String getWord()
   {
      return word;
   }

     /**  returns a string representation of this Index Entry.  */
   public String toString()
   {
      String str = word + " ";
      for(int i=0;i<numList.size()-1;i++)
         str += numList.get(i) + ", ";
      str += numList.get(numList.size()-1) + "";
      return str;
   }
```

```java

    public int compareTo(IndexEntry e)
    {
       return word.compareTo(e.getWord());
    }
}


```