Homework Turnin

Email:	rgalanos@fcps.edu
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	01-05
Receipt ID:	a8878a0149af0407cb3e43a615c77611

Replacing prior submission from Wed 2016/09/21 11:15am.

Turnin Successful!

The following file(s) were received:

Sentence_Driver.java (3913 bytes)

```
//Name:          date:
import java.util.*;
public class Sentence_Driver
{
   public static void main(String[] args)
   {
      System.out.println("PALINDROME TESTER");
      Sentence s = new Sentence( "\"Hello there!\" she said." );
      System.out.println( s.getSentence() );
      System.out.println( s.getNumWords() );
      System.out.println( s.isPalindrome() );
      System.out.println();

      s = new Sentence( "A Santa lived as a devil at NASA." );
      System.out.println( s.getSentence() );
      System.out.println( s.getNumWords() );
      System.out.println( s.isPalindrome() );
      System.out.println();


      s = new Sentence( "Flo, gin is a sin! I golf." );
      System.out.println( s.getSentence() );
      System.out.println( s.getNumWords() );
      System.out.println( s.isPalindrome() );
      System.out.println();


      s = new Sentence( "Eva, can I stab bats in a cave?" );
      System.out.println( s.getSentence() );
      System.out.println( s.getNumWords() );
      System.out.println( s.isPalindrome() );
      System.out.println();

      s = new Sentence( "Madam, I'm Adam." );
      System.out.println( s.getSentence() );
      System.out.println( s.getNumWords() );
      System.out.println( s.isPalindrome() );
      System.out.println();

   // Lots more test cases.  Test every line of code.  Test
   // the extremes, test the boundaries.  How many test cases do you need?

   }
}
class Sentence
{
   private String mySentence;
   private int myNumWords;
```

```
   //Constructor.  Creates sentence from String str.
   //               Finds the number of words in sentence.
   //Precondition:  Words in str separated by exactly one blank.
   public Sentence( String str )
   {
      mySentence = str;
      myNumWords = 0;
      StringTokenizer  st = new StringTokenizer(str);
      while(st.hasMoreTokens())
      {
         st.nextToken();
         myNumWords++;
      }

   }

   public int getNumWords()
   {
      return myNumWords;
   }

   public String getSentence()
   {
      return mySentence;
   }

   //Returns true if mySentence is a palindrome, false otherwise.
   public boolean isPalindrome()
   {
      String sentence = mySentence;
      sentence = removeBlanks(sentence);
      sentence = lowerCase(sentence);
      sentence = removePunctuation(sentence);
      return isPalindrome(sentence, 0, sentence.length()-1);
   }
   //Precondition: s has no blanks, no punctuation, and is in lower case.
   //Returns true if s is a palindrome, false otherwise.
   private static boolean isPalindrome( String s, int start, int end )
   {
      if((start == end || start == end-1)&&s.charAt(start)==s.charAt(end))
      {
         return true;
      }
      else if(s.charAt(start)!=s.charAt(end))
      {
         return false;
      }
      else
      {
         return isPalindrome(s, start+1, end-1);
      }
   }
   //Returns copy of String s with all blanks removed.
   //Postcondition:  Returned string contains just one word.
   private static String removeBlanks( String s )
   {
      return s.replaceAll("\\s","");
   }

   //Returns copy of String s with all letters in lowercase.
   //Postcondition:  Number of words in returned string equals
   //                number of words in s.
   private static String lowerCase( String s )
   {
      return s.toLowerCase();
   }

   //Returns copy of String s with all punctuation removed.
   //Postcondition:  Number of words in returned string equals
   //                number of words in s.
   private static String removePunctuation( String s )
   {
      final String punct = ".,?!:;\"(){}[]<>";
      int i = 0;
      while(i<s.length())
      {
         if(punct.indexOf(s.charAt(i))!=-1)
            s = s.substring(0,i)+s.substring(i+1);
         else
            i++;
```

```
        }
        return s;
    }
}
```