```
package a2;

import java.util.Scanner;

/**
 *  @author Connor Cousineau
 */

public class MethodCollection
{


      public static void main(String[] args)
      {
      System.out.println("There are " + countTokens("random words go here
and then some") + " tokens.");
      System.out.println("There are " + countTokens("there are four
words") + " tokens.");
      System.out.println("There are " + countTokens("there are five words
here") + " tokens.");
      System.out.println("There are " + countTokensThatAreNotInt("my name
is bob 14") + " tokens that are not Ints.");
      System.out.println("There are " + countTokensThatAreNotInt("14 15 my
name is David") + " tokens that are not Ints.");
      System.out.println("There are " + countTokensThatAreNotInt("12 24 12
12 42") + " tokens that are not Ints.");
      System.out.println("The number is: " + describeSign(5));
      System.out.println("The number is: " + describeSign(-10));
      System.out.println("The number is: " + describeSign(0));
      System.out.println("Is divisable by 7: " +
isEvenlyDivisibleBySeven(7));
      System.out.println("Is divisable by 7: "
+isEvenlyDivisibleBySeven(20));
      System.out.println(makeSquare(2));
      System.out.println(makeSquare(10));
      System.out.println(capitalizeLastCharacter("word"));
      System.out.println(capitalizeLastCharacter("sadness"));
      System.out.println(capitalizeLastCharactersInSentence("This Is An
Article"));
      System.out.println(capitalizeLastCharactersInSentence("Pointlessly
long sentence for the purpose of proving it does it's job"));

      }



      /**
       * @param sentence Brings in a sentence.
       *
       * @return Returns the number of tokens.
       */
      public static int countTokens(String sentence)
      {
                  int count = 0;
                  Scanner scanner = new Scanner(sentence);
```

```
                while (scanner.hasNext())
                {
                            scanner.next();
                            count ++;
                }

                return count;
        }


        /**
         * @param sentence Brings in a sentence.
         *
         * @return Returns the number of non-numbers.
         */
        public static int countTokensThatAreNotInt(String sentence)
        {

                int numberOfNotInts = 0;

                Scanner scanner = new Scanner(sentence);

                while (scanner.hasNext())
                {
                            if (scanner.hasNextInt())
                            {
                                  scanner.hasNext();
                            }
                            else
                            {
                                  numberOfNotInts ++;
                            }

                            scanner.next();

                }

                return numberOfNotInts;
        }


        /**
         * @param value Brings in a value.
         *
         * @return returns a "negative" or "non-negative" based on the sign
of value.
         */
        public static String describeSign(int value)
        {

                if( value >= 0)
                {
                        return "non-negative";
                }
                else
                {

                        return "negative";
                }

        }


        /**
         * @param value Brings in a value.
         *
         * @return returns a true or false statement.
         */
        public static boolean isEvenlyDivisibleBySeven(int value)
        {

                if (value % 7 <= 0)
                {
                        return true;
                }

                else
                {
                        return false;
                }
        }


        /**
         * @param edge Defines the character that represents the character
edge.
         *
         * @param inner Defines the character that represents the character
inner.
         *
         * @param width Defines the variable width.
         *
         * @return Returns a line representation based on the information
provided.
         */
        public static String makeLine(char edge, char inner, int width)
        {
                String line = "";
                int i = 0;
                while (i < width - 2)
                {
                        line = line + inner;
                        i = i + 1;
                }
                return edge + line + edge;
        }
```

```
      /**
       * @param width Defines the value that represents the width of the
square.
       *
       * @return Returns the full square.
       */
      public static String makeSquare(int width)
      {
                  int count = 0;
                  String body = "";

                  if(width == 2)
                  {
                              body = body  + makeLine('+','-', width) +
"\n";
                              body = body + makeLine('+','-', width);
                              return body;
                  }
                  else

                  {
                  while (count < width - 2)
                  {
                              body = body + makeLine('+','-', width);
                              while (count<width-2)
                              {
                              count++;
                              body = body + "\n" + makeLine('|', ' ',
width);
                              }
                              body = body + "\n" + makeLine('+','-',
width);
                  }
                  return body;
                  }

      }


      /**
       * @param word Brings in a word.
       *
       * @return Returns an altered word.
       */
      public static String capitalizeLastCharacter(String word)
      {
                  String strangeWord = "";
                  strangeWord = word.substring(0, word.length()-1);
                  char bigLetter = word.charAt(word.length()-1);
                  bigLetter = Character.toUpperCase(bigLetter);
                  return strangeWord + bigLetter;
      }
```

```
	/**
	 * @param sentence Brings in the sentence to be converted.
	 *
	 * @return Gives back the altered sentence.
	 */
	public static String capitalizeLastCharactersInSentence(String 
sentence)
	{
			String strangeSentence = "";
			String strangeWord = "";
			String word;
			Scanner scanner = new Scanner(sentence);

	while (scanner.hasNext())
	{
			if (scanner.hasNext())
			{
			word = scanner.next();
			char bigLetter = word.charAt(word.length()-1);
			word = word.substring(0, word.length()-1);
			bigLetter = Character.toUpperCase(bigLetter);
			strangeWord = word + bigLetter + " ";
			}
			strangeSentence = strangeSentence + strangeWord;
	}
			// Change or remove this statement as needed
			return strangeSentence;
	}

}
```