CS 102: Introduction to Computer Science

Project One - Strength Subsides in Strings

Student Name	Mohammad El-Abid
Student ID	8905652
Professor	Sabrina Thompson
Date	Tuesday, January 24, 2012

Table of Contents

Abstract

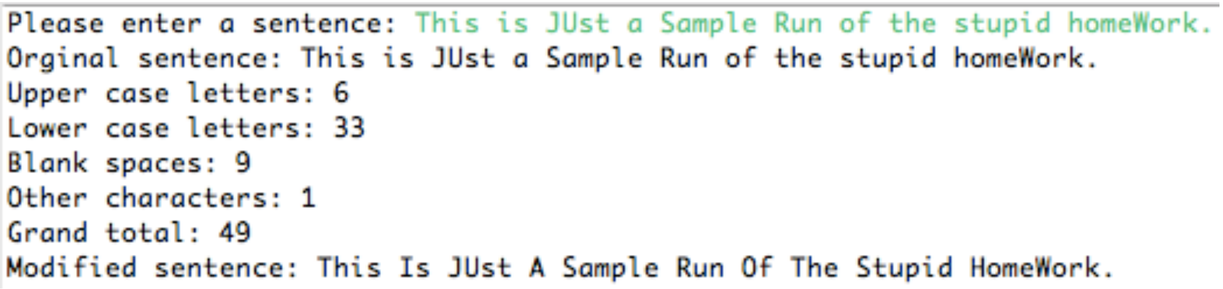
This application takes an input string from `System.in` then count the uppercase characters, lowercase characters, spaces, and "others" (or miscellaneous). The program also modifies the input string to capitalize every letter after a space.

Introduction

The application was rather simplistic requiring only the `Scanner` dependency to read from the standard input (`STDIN`). Once the string was gathered I just had to iterate over the characters in the string (using `String#charAt(int index)`) to increment the counters of the the counter variables. When iterating over a lower case letter, I check if the previous letter was a space, if it was I'll convert the character to uppercase by removing 32 from it's integral value.

Screenshots

Application output to `stdout`:



```
Please enter a sentence: This is JUst a Sample Run of the stupid homeWork.
Orginal sentence: This is JUst a Sample Run of the stupid homeWork.
Upper case letters: 6
Lower case letters: 33
Blank spaces: 9
Other characters: 1
Grand total: 49
Modified sentence: This Is JUst A Sample Run Of The Stupid HomeWork.
```

Code

File: Application.java

```
package edu.bridgeport;

import java.util.Scanner;

public class Application {

    public static void main(String[] args) {
           Scanner input = new Scanner(System.in);

           System.out.print("Please enter a sentence: ");

           String line = ""; // Stop the "may not be initialized"
error
           try {
                line = input.nextLine();
           } catch(Exception e) {
                System.out.println();
                System.out.println(e);
                System.exit(1);
           }
           StringBuilder modify = new StringBuilder(line);

           System.out.println("Original sentence: " + line);

           int uppercases, lowercases, spaces, others;
           uppercases = lowercases = spaces = others = 0;

           // Iterate through the string
           for(int i = 0; i < line.length(); i++) {
                int character = (int) line.charAt(i);

                if( character >= 65 && character <= 90) {
                      uppercases++;
                } else if( character >= 97 && character <= 122) {
                      lowercases++;
                      if(line.charAt(i-1) == ' ') modify.setCharAt(i,
(char)(character - 32));
                }
                else if( character == 32 ) {
                      spaces++;
                } else {
                      others++;
                }
            }

            // Over 9,000 reference
            if(line.toLowerCase().matches("over nine thousand") &&
line.length() <= 9000 ) {
                System.out.println("That's under nine thousand, learn
how to count.");
            } else if( line.length() > 9000 ) {
                System.out.println("IT'S OVER NINE THOUSAND
characters.");
            }

            // UPPER CASE
            System.out.println("Upper case letters: " + uppercases);
            // LOWER CASE
            System.out.println("Lower case letters: " + lowercases);
            // BLANK SPACES
            System.out.println("Blank spaces: " + spaces);
            // OTHER
            System.out.println("Other characters: " + others);
            // TOTAL
            System.out.println("Grand total: " + line.length());
            // MODIFIED
            System.out.println("Modified sentence: " +
modify.toString());
    }

}
```

Conclusion

I remembered how powerful the StringBuilder class was. But questioned why the String class has a replace method and regular expression methods, but does not have the ability to change a single character based on index.

Works Used

Java documentation of String and StringBuilder.