CSSSKL 142 - Lab7
Strings
University of Washington, Bothell

Summary

Today we will write some programs involving String objects. Strings are widely used not only in Java, but in many other programming languages, as well. You can think of a string as a sequence of characters; strings and are treated as objects in Java and Java platform provides the String class to create and manipulate strings.

Exercise 1 - Capitalizing First Letters of Input Text

1.1 Write a method that capitalizes the first letter of each word in a sentence, named `capitalizeFirstLetters`. This method prints a capitalized version of a String to the screen - that is, it changes the first letter of each word in the string to upper case (if it is not already upper case). So "I really enjoy attending lab!" becomes "I Really Enjoy Attending Lab!". The string to be printed should be a parameter to the method, and the method should have a void return. You should test your method in main by getting a line of input from the user and applying the method to it.

Public static void capitalizeFirstLetters(String input) {

        //TODO: parse input for words and capitalize each one.
}

Hint: You may want to browse the Java String docs for useful methods on strings, such as split().
Hint: You can use a new Scanner to get next() words out of a String.
Hint: StringTokenizer is a class that can do the same thing.

1.2 Write a method called nameInitials that takes one String as argument, pertaining to somebody's full name and returns a String of the name's initials. Usage example,

```
String initials = nameInitials("Bertrand Arthur William Russell");
System.out.println(initials);          //should print B.A.W.R.
```

1.3 Write a method called letterCount that takes two String arguments, one containing some text and the other containing a single letter. This method should then compute and return the count of all of the occurrences of that letter in the given text. Usage example,

```
int countMs;
countMs = letterCount("I am the very model of a modern major general", "m");
System.out.println(countMs);        //should print 4
```

2. Finding the Lexicographic or by Length Largest String

2.1 Write a method called `lexLargest` that takes a string object argument containing some text. Your method should return the lexicographical largest word in the String object (that is, the one that would appear latest in the dictionary).

```
String large = lexLargest ("I am the very model of a modern major general");
System.out.println(large);                    // should print : very
```

2.2 Write a method called `largestBy` that takes a string object argument containing some text. Your method should return the largest-by-length word in the String object. In case of ties, return the first occurrence of the largest word. Note the return type is String. Example,

```
String large ;
large = largestBy("I am the very model of a modern major general");
System.out.println(large);                    // should print general
```

2.3 (Optional) Now, using above two methods, see if you can combine them in one single method and still produce the same outputs?
NOTE: use an integer to determine which algorithm your code will execute, e.g.,
`String largestByAction(String inputText, int action)`
If integer action is 1, then run first algorithm and if integer action is 2, then run the 2nd algorithm. Also, if the integer action is anything other than 1 or 2, return the empty string "".

3. Finding the position of characters in a String

3.1 Write a Java program named `CharsIndex` to get the index of all the characters of the alphabet. Ask user to input a string. The sample output was produced by this string: `"The quick brown fox jumps over the lazy dog."`

Sample Output:

```
  a   b  c   d e   f   g h i  j
==========================
36 10 7 40 2 16 42 1 6 20

k   l   m   n   o   p q   r   s   t
==========================
8 35 22 14 12 23 4 11 24 31

u   v   w   x   y   z
================
5 27 13 18 38 37
```

3.2 Try the following string as input to your code. Do you see anything unusual in your output? If yes, explain what causes that and document your observation as comments in your code.

`"I love programming languages, especially Java.`