La Salle College
Advanced Object Oriented Programming – 420PA3AS

Assignment 1 – Java Strings
Michelle M. Khalifé

Create a switch menu that runs the chosen exercise. You can go by numbers or strings for the case labels.
Nest your switch in a do-while to allow the user multiple runs. Make sure s/he has an option to quit.

#1 Palindrome

a. Write a method `reverseString()` that takes a string argument and returns the string in reverse order.
b. A palindrome is a string that can be read the same way forward and backward. Write a method `isPalindrome()` that accepts a string and returns `true` if the string is a palindrome, `false` otherwise. Version1 uses the `reverseString()` method. Version2 should do without generating a new string.
c. In the palindrome switch case, prompt the user for a number *n* followed by that many strings and prints out each string and the boolean value indicating whether the string is a palindrome or not.
d. How can you ensure that the number matches the number of strings and vice versa?

```
Input                                   Output
5 madam racer toot java programming     madam, palindrome: true
                                        racer, palindrome: true
                                        toot, palindrome: true
                                        java, palindrome: false
                                        programming, palindrome: false
```

#2 Random String

a. Write a method `generateRandomString()` that accepts a number *n* and a string *s* and returns a new random string. The method picks *n* random characters from the original string and uses them to construct and return this new random string. A character can only be selected once.
b. In the switch case, prompt the user for a number and a string. Make sure the number is less than the length of the string.

```
Input                                   Output
3 I love java                           jlo
5 superbowl                             psrlu
```

#3 Substrings

a. Write a method `howMany()` that accepts two strings `s1` and `s2` and returns the number of times string s2 appears in s1.
b. Write a method `longestSubstring()` that accepts a string and returns the longest substring in which characters appear in alphabetical order, as well as its length,

```
Input                                   Output
booboo, boo                             booboo, boo: 2
abcdxyzklmnop                           abcdxyzklmnop: abcdxyz, 6
```

#4 Digits Range

a. Write a method `getDigitsRange()` that accepts an integer, finds the largest and smallest digits within it, and returns their range. The latter is determined by adding 1 to the difference of the highest and lowest digits.
b. In the switch case, prompt the user for an integer that has at least 2 digits and no leading 0s. Use a do-while to validate the input before calling `getRangeOfDigits()`

```
Input                                     Output
57391                                     57391, range: 9
3648                                      3648, range: 6
90                                        90, range: 10
```

#5 Split String (Toptal interview question)

a. Write a method `splitString()` that accepts a string *s* consisting of the letters *a* and *b*, and returns the number of ways *s* can be split into 3 parts of different or equal length, such that each part contains the same number of *a's*:

```
Input                                     Output
babaa                                     ba|ba|a, bab|a|a; total 2
ababa                                     a|ba|ba, ab|ab|a, a|bab|a; total 3
aba                                       n/a; total: 0
bbbbb                                     b|b|bbb, b|bb|bb, b|bbb|b, bb|b|bb,
                                          bb|bb|b, bbb|b|b; total 6
```