Intro to Java

7 - Strings / Solutions: Strings

Solutions: Strings

Counting characters

```
class CharCount {

    public static void main(String[] args) {

        String s = "Als Gregor Samsa eines Morgens aus unruhigen Träumen erwachte " +
                "fand er sich in seinem Bett zu einem ungeheueren Ungeziefer verwandelt.";

        int numberOfE = count(s, 'e');
        int numberOfU = count(s, 'u');

        System.out.println("There are " + numberOfE + " 'e's and " + numberOfU + " 'u's.");
    }

    /* Count how many times a character is present in a string */
    public static int count(String text, char c) {

        int charCount = 0;
        int n = text.length();

        for(int i = 0; i < n; i++) {

            char t = text.charAt(i);
            if (t == c) {
                charCount++;
            }

            // Also possible:
            // if (text.charAt(i) == c) ...
        }

        return charCount;
    }
}
```

Bonus: counting all letters

```
class Main {

    public static void main(String[] args) {
        String s = "mississippi";

        for (int i = 0; i < s.length(); i++) {
            char letter = s.charAt(i);
            int count = count(s, letter);
            int firstOccurance = s.indexOf(letter);
            if (firstOccurance == i) {
                System.out.println(letter + " is occuring " + count + " times");
            }
        }
    }

    /* Count how many times a character is present in a string */
    public static int count(String text, char c) {

        int charCount = 0;
        int n = text.length();

        for(int i = 0; i < n; i++) {

            char t = text.charAt(i);
            if (t == c) {
                charCount++;
            }

            // Also possible:
```

```
            // if (text.charAt(i) == c) ...
        }

        return charCount;
    }
}
```

Reverse

Two possibilities:

```
public static String reverse(String s) {
    String r = "";
    for (int i = 0; i < s.length(); i++) {
        r = s.charAt(i) + r;
    }
    return r;
}
```

Or:

```
public static String reverse(String s) {
    String r = "";
    for (int i = s.length() - 1; i >= 0; i--) {
        r += s.charAt(i);
    }
    return r;
}
```

Palindrome

It’s very simple if we can reuse `reverse`:

```
public static boolean isPalindrome(String word) {
    return word.equalsIgnoreCase(reverse(word));
}
```

Secret messages

```
import java.util.Scanner;


class Main {
  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    String secret = scanner.next(); // here we read the message from the user, remember to define the Sca


    String message = ""; // this is where we'll concatenate latters as we decode them
    // below we need a loop over all the characters in the `secret`, lookup String method for length
    // loop will run from 0 to the length - 1 of the given `secret`
    for (int i = 0; i < secret.length(); i++) {
      char currentLetter = secret.charAt(i); // find a method on String to give us character at i


      // since we're given the secret text to get the original message for each letter we need to
      // fetch the previous letter, but there is one exception
      char previousLetter;
      if (currentLetter == 'A') { // change the ?, for one letter we can't look at the previous letter
        previousLetter = 'Z'; // change the ?, we need to substitute that letter with another one
      } else {
        previousLetter = (char) (currentLetter - 1); // this is how we lookup the previous letter
      }


      // we need to put the previousLetter to the end of message form the message
      message = message + previousLetter; // change this
    }


    System.out.println(message);
  }
}
```

Made with ❤ by teachers at ReDI School.