Master the ArrayList!

Objectives

Use of Java Strings and ArrayList. For this assignment, you are allowed to use ArrayList only.

Details

Inside an ArrayListMethods class, create methods for each of the following with the appropriate inputs and outputs.

[1]. Uniqueness

Write a method unique() which returns true if all the items in the supplied List are unique. All the items are unique if none of them are the same. Return false otherwise.

[2]. All Multiples

Write a method allMultiples() which returns a new List of integers that contains all the integers in the List argument that are multiples of the given int. For example, if the List is [1; 25; 2; 5; 30; 19; 57; 2; 25] and 5 was provided, the new list should contain [25; 5; 30; 25].

[3]. All Strings of Size

Write a method named allStringsOfSize() which returns a new List<String> which contains all the Strings from the List argument that are length characters long. For example, if the inputs are ["I", "like", "to", "eat", "eat", "eat", "apples", "and", "bananas"] and 3, the new list is ["eat", "eat", "eat", "and"].

[4]. Permutations

Write a method isPermutation() which return true if the two List arguments are permutations of each other, otherwise it returns

false. Two lists are permutations if they contain the same elements, including the same number of duplicates, but they don't have to contain the elements in the same order. For example, [1,2,4] and [2,1,4] are permutations of each other.

[5]. String Tokenization[1]

Write a method tokenize() which returns a List of words from the input string. We assume that each word in the input string is separated by whitespace.[2] If our input String is "Hello, world!", then the output should be ["Hello,", "world!"].

[6]. Remove All

Write a method removeAll() which removes a particular item from the given Listand returns the new List. For example, if the method is passed the List<Integer> [1, 4, 5, 6, 5, 5, 2] and an Integer 5, the method removes all 5's from the List. The List then becomes [1, 4, 6, 2].

---

[1] *Tokenization* is the process of demarcating and possibly classifying sections of a string of input characters. The resulting tokens are then passed on to some other form of processing. Tokenization is often used to parse inputs in lexical processing for natural and computer programming processing. Real-world lexical processing has much more complex rules than our simple method.

[2] Check the String class Javadoc to find a method that might help.