

JAVA Collections - Lambda - Streams

Self Study 6 Exercises

In this exercise, you will find 20 different exercises about Java Collection Framework and stream operations.

- 1- Visit <https://www.w3resource.com/java-exercises/collection/index.php> . You will find 126 exercises about Java Collections with solutions.
- 2- Write a program that takes an array as input and prints the frequency of elements, using HashMaps.
- 3- Given an array of integers, find if the array contains any duplicates. Your function should return true if any value appears at least twice in the array, and it should return false if every element is distinct.
- 4- Merge two sorted linked lists and return it as a new list.
- 5- Given a singly linked list, determine if it is a palindrome.
- 6- Write a recursive traversal of the project folder with stream operations.
- 7- Given a list of Strings, write a method that returns a list of all strings that start with the letter 'a' (lower case) and have exactly 3 letters. TIP: Use Java 8 Lambdas and Streams API's.
- 8- Write a method that returns a comma separated string based on a given list of integers. Each element should be preceded by the letter 'e' if the number is even, and preceded by the letter 'o' if the number is odd. For example, if the input list is (3,44), the output should be 'o3,e44'.
- 9- The WordList.java file reads a text file and makes an alphabetical list of all the words in that file. The list of words is output to another file. Improve the program so that it also keeps track of the number of times that each word occurs in the file. Write two lists to the output file. The first list contains the words in alphabetical order. The number of times that the word occurred in the file should be listed along with the word. Then write a second list to the output file in which the words are sorted according to the number of times that they occurred in the files. The word that occurred most often should be listed first.
- 10- Write a few lambda expressions and a function that returns a lambda expression as its value. Suppose that a function interface ArrayProcessor is defined as

```
public interface ArrayProcessor { double apply( double[] array ); }
```

Write a class that defines four public static final variables of type ArrayProcessor that process an array in the following ways: find the maximum value in the array, find the minimum value in an array, find the sum of the values in the array, and find the average of the values in the array. In each case, the value of the variable should be given by a lambda expression. The class should also define a function

```
public static ArrayProcessor counter( double value ) { ...
```

This function should return an ArrayProcessor that counts the number of times that value occurs in an array. The return value should be given as a lambda expression.

The class should have a `main()` routine that tests your work. The program that you write for this exercise will need access to the file ArrayProcessor.java, which defines the functional interface.

11- Solve the following question, we selected from the coursebook, using Collections:

1. Sorting Accounts : Write a class AccountComparator to implement the Comparator interface for class Account in Figure 15.9 (Textbook slides Ch15, code available ch15) based on the account's balance. Use this class in order to sort a list of accounts in descending order based on the account's balance.
2. Sorting Words with a TreeSet: Write a program that uses a String method split to tokenize a line of text input by the user and places each token in a TreeSet. Print the elements of the TreeSet. [Note: This should cause the elements to be printed in ascending sorted order.]
3. Changing a PriorityQueue's Sort Order: The output of Figure 16.14 (Textbook Slides) shows that PriorityQueue orders Double elements in ascending order. Rewrite Figure 16.14 so that it orders Double elements in descending order (i.e. 9.8 should be the highest-priority element rather than 3.2).

12- Solve the following question, we selected from the coursebook, using Streams and Lambdas:

1. Averages: Write a program to generate ten random integers between zero and 1000, then display how many of them are odd and how many are even; also display the average of: all numbers, odd numbers, and even numbers.
2. Create a Person class with instance variable name (String) and age (int), getter/setter methods, and toString method. Then write a test program that creates an array of Person which has 10 Person objects. Using streams and lambdas:
 - Create a list of Person (List) which contains all persons whose name starts with "A". Print the list.
 - Group persons by age, and print list of persons for each age.
 - Determine the average of all persons.
 - Use IntSummaryStatistics and Collectors.summarizingInt() (see java documentation) to find count, sum, min, average, and max of all persons' age.
 - Form a string that starts with "Students " continues with each persons name separated by " and ", and ending with " are in COMP132.". For example: "Students Waris and Damla are in COMP132."

Solutions:

- Solutions of 9, 10, 11 and 12 are in separate files.
- Other solutions are in the solutions.java file.

Credits:

- Exercise 9 and 10 : Hobart and William College Math Department - Java Exercises
- Exercise 7 and 8 : <https://code-exercises.com/programming/>
- Exercise 6: <http://web.mit.edu/6.031/www/fa17/classes/26-map-filter-reduce/>
- Exercise 3, 4 and 5 : <https://github.com/chrismltais/Java-Exercises/>