ConceptsOfCS

Mastering Java Streams API: Top Interview Programming Questions & Answers

1. ***Given a list of integers, find out all the numbers starting with 1 using Stream functions.***

```
public static List<Integer> getNumbersStartsWithOne(java.util.List<java.lang.Integer>
numbers){
   return  numbers.stream().filter(x->x.toString().startsWith("1"))

.collect(Collectors.toList());
}
```

2. find duplicate elements in the list.

```
public static List<Integer> findDuplicateElements(List<Integer> list){
    Set<Integer> tempSet = new HashSet<>();
   return list.stream().filter(x -> !tempSet.add(x)).collect(Collectors.toList());
}
```

3. find largest element of the list using streams.

```
public static Integer findLargestElementInTheList(List<Integer> list){
   Integer maxValue = list.stream().max((x,y) -> x.compareTo(y)).get();
   return maxValue;
}
```

4. In a Given String, find the first non-repeated character using streams API.

```
public static char findFirstNonRepeatativeCharacter(String s1){
  return s1.chars().mapToObj(x -> (char)x).filter( x-> s1.indexOf(x) ==
s1.lastIndexOf(x)).findFirst().orElse('0');
}
```

5. In a Given String, find the first repeated character using streams API.

```
public static char findFirstNonRepeatativeCharacter(String s1){
  return s1.chars().mapToObj(x -> (char)x).filter( x-> s1.indexOf(x) !=
s1.lastIndexOf(x)).findFirst().orElse('0');
}
```

6. sort the elements of a list in descending order using Streams API.

```
public static void  sortTheListInReverseOrder(List<Integer> list){
   list.stream()
         .sorted(Collections.reverseOrder())
         .forEach(System.out::println);;
}
```

7. check if a list contains duplicate elements or if any number appears appears twice in a list.

```
public static boolean  checkIfElementExistTwice(List<Integer> list){
   return !(list.stream().distinct().collect(Collectors.toList()).size() == 
list.size());
}
```

8. Calculate the cube of each element present in the list and filter the elements whose value is greater than 50 using streams API.

```
public static void 
calculateCubeAndFilterNumberGreaterThan50(List<Integer> list){
   list.stream().map(x -> x*x*x).filter(x -> x > 50 ).forEach(System.out::println);
}
```

9. Program to convert list to Map Using Stream API .

```
public static void convertListOfObjectsIntoMap(){
   List<User> users = new ArrayList<>();
   users.add(new User(1, "Ashutosh"));
   users.add(new User(2, "Mohit"));
   Map<Integer, String> usersMap = users.stream().collect(Collectors.toMap(x -> 
x.age(), x -> x.name()));
   System.out.println(usersMap);
}
```

10. Calculate the length of each word present in the list .

```
public static void calLengthOfEachWordPresentInTheList(){
   List<String> names = Arrays.asList("AA", "BB", "AA", "CC");
   Map<String, Integer> countMap = 
names.stream().distinct().collect(Collectors.toMap(Function.identity(), 
String::length));
   System.out.println(countMap);
}
```

11. Find the nth largest number in the list.

```
public static Integer findNthLargestElementInTheList(Integer n){
    List<Integer> list3 = Arrays.asList(10,15);
    if(list3.size() >= n && n > 0){
        return list3.stream()
                .distinct()
                .sorted(Comparator.reverseOrder())
                .skip(n-1)
                .findFirst().get();
    }
    else {
        return -1;
    }
}
```

12. Count occurrences of each character in a string.

```
public Map<Character, Long> getOccurrencesOfEachCharacterInString(String s1){
    Map<Character, Long> frequencyMap = s1.chars()
            .mapToObj(c -> (char) c)
            .collect(Collectors.groupingBy(Function.identity(), Collectors.counting()));
    return  frequencyMap;
```

13. Find the longest word in the sentence.

```
public String getLongestWordFromSentence(String sentence) {
    return Arrays.stream(sentence.split(" "))
            .max(Comparator.comparingInt(String::length)).get();
}
```

14. Find the sum of all even numbers in the list.

```java
public Integer findTheSumOfAllEvenNumbers(List<Integer> numbers) {
   return numbers.stream()
         .filter(n -> n % 2 == 0)
         .mapToInt(Integer::intValue)
         .sum();
}
```

15. Sort a list of employees by salary.

```java
public Integer sortEmployeesBySalary(List<Employee> employees) {
   return employees.stream().sorted(Comparator.comparingInt(e -> e.salary))
.forEach(System.out::println);
}
```

16. How to flatten a list of list?

```java
public List<String> flatenTheListOfList(List<List<String>> lists){
   return lists.stream().flatMap(List::stream).collect(Collectors.toList());
}
```

17. Find common elements in both the list.

```java
public List<String> findCommonElementsInBothTheList(List<String> list1,
List<String> list2){
   return list1.stream()
         .filter(list2::contains)
         .collect(Collectors.toList());
}
```

18 Partition a list into even and odd number

```java
public Map<Boolean, List<Integer>>  partitionAListIntoEvenOdd(List<Integer>
numbers){
   return numbers.stream() .collect(Collectors.partitioningBy(n -> n % 2 == 0));
}
```

19. Convert a list of numbers to a comma-separated string

```java
public String  getCommaSeperatedStringFromAList(List<Integer> numbers)
{
   return numbers.stream() .map(String::valueOf)
.collect(Collectors.joining(", "));;
}
```

20. Find total Number of characters in a list of String

```java
public Integer  getTotalNumberOfCharInAListOfString(List<String> list){
   return list.stream()
          .mapToInt(String::length)
          .sum();
}
```

21. Find the most repeated character of a string

```java
public Character findMostRepetativeCharacter(String str){
   Optional<Character> mostRepeatedChar = str.chars()
          .mapToObj(c -> (char) c)
          .collect(Collectors.groupingBy(Function.identity(), Collectors.counting()))
          .entrySet().stream()
          .max(Map.Entry.comparingByValue())
          .map(Map.Entry::getKey);
   return mostRepeatedChar.get();
}
```