< ==========================By Shwetha===Java8====================🡺

How do you find the average length of strings containing 'e' in an array using streams?

OptionalDouble averageLength = Arrays.stream(strings)

                                   .filter(s -> s.contains("e"))

                                  .mapToInt(String::length)

                                   .average();

How do you find the longest common prefix among all strings in an array using streams?

String longestCommonPrefix = Arrays.stream(strings)

                                  .reduce((s1, s2) -> {

                                      int minLength = Math.min(s1.length(), s2.length());

                                      int i = 0;

                                      while (i < minLength && s1.charAt(i) == s2.charAt(i)) {

                                          i++;

                                      }

                                      return s1.substring(0, i);

                                  })

                                 .orElse("");

How do you find the number of strings with unique characters in an array using streams?

long uniqueCharStringsCount = Arrays.stream(strings)

                                   .filter(s -> s.chars().distinct().count() == s.length())

                                   .count();

How do you find the number of strings containing at least one digit in an array using streams?

long digitContainingStringsCount = Arrays.stream(strings)

                                      .filter(s -> s.matches(".\*\\d.\*"))

                                      .count();

How do you find the strings containing all vowels in an array using streams?

List<String> allVowelsStrings = Arrays.stream(strings)

                                    .filter(s -> s.replaceAll("[aeiouAEIOU]", "").isEmpty())

                                   .collect(Collectors.toList());

How do you find the strings with characters in alphabetical order in an array using streams?

List<String> alphabeticalOrderStrings = Arrays.stream(strings)

                                            .filter(s -> s.chars().sorted().mapToObj(c -> (char) c).collect(Collectors.toList()).equals(s.chars().mapToObj(c -> (char) c).collect(Collectors.toList())))

                                            .collect(Collectors.toList());

How do you find the strings with all distinct characters in an array using streams?

java

Copy code

List<String> distinctCharStrings = Arrays.stream(strings)

                                       .filter(s -> s.chars().distinct().count() == s.length())

                                       .collect(Collectors.toList());

How do you find the strings with characters in reverse alphabetical order in an array using streams?

List<String> reverseAlphabeticalOrderStrings = Arrays.stream(strings)

                                                  .filter(s -> s.chars().sorted().mapToObj(c -> (char) c).collect(Collectors.toList()).equals(s.chars().sorted().mapToObj(c -> (char) c).collect(Collectors.toList())))

                                                  .collect(Collectors.toList());

How do you find the strings containing all uppercase letters in an array using streams?

List<String> uppercaseStrings = Arrays.stream(strings)

                                    .filter(s -> s.chars().allMatch(Character::isUpperCase))

                                   .collect(Collectors.toList());

How do you find the strings with characters in ascending ASCII order in an array using streams?

List<String> ascendingASCIIStrings = Arrays.stream(strings)

                                         .filter(s -> s.chars().sorted().mapToObj(c -> (char) c).collect(Collectors.toList()).equals(s.chars().mapToObj(c -> (char) c).collect(Collectors.toList())))

                                         .collect(Collectors.toList());

How do you find the strings containing exactly two vowels in an array using streams?

List<String> twoVowelsStrings = Arrays.stream(strings)

                                    .filter(s -> s.chars().filter(c -> "aeiouAEIOU".indexOf(c) != -1).count() == 2)

                                   .collect(Collectors.toList());

==============================================================🡺

// 1. List Of Candidates

long count = candidates.stream().count();  
System.out.println("1. List Of Candidates: "+count);

// 2. List of Candidate for City

Map<String, List<Candidate>> cityMap = candidates.stream().collect(Collectors.groupingBy(Candidate::getCity));  
System.out.println("2. List of Candidates for City: \n"+cityMap);

// 3. Highest Experienced Candidate

Candidate candidate = candidates.stream().sorted((c1, c2) -> c2.getYearOfExperience() - c1.getYearOfExperience()).findFirst().get();  
System.out.println("3. Highest Experienced Candidate: "+candidate);  
// 4. Candidate count per technology"  
Map<String, Long> canCountPerTech = candidates.stream().collect(Collectors.groupingBy(Candidate::getTechnicalExpertise, Collectors.counting()));  
System.out.println("4. Candidate count per technology: \n"+canCountPerTech);

// 5. Candidate City and skills

Map<String, Set<String>> cityAndSkills = candidates.stream()  
.collect(Collectors.groupingBy(Candidate::getCity, Collectors.mapping(Candidate::getTechnicalExpertise, Collectors.toSet())));  
System.out.println("5. Candidate City and skills:\n"+cityAndSkills);

// 6. Sort Candidates based on years

List<Candidate> years = candidates.stream().sorted((c1, c2) -> c1.getYearOfExperience() - c2.getYearOfExperience()).collect(Collectors.toList());  
System.out.println("6. Sort Candidates based on years:\n"+years);