Java Freshers

Candidate-ID	0012400053
Candidate Name	Ede Mahendra
Email	edemahendra85@gmail.com
Mobile	8688089358
Source	Naukari

Screening Test

Completed on : 30-Nov-2024 02:30 PM

#	Question	Response	Marks
1	Have you fully completed your education and prepared to work	Yes	1
2	Have you taken any long term training coure on Java Technologies	Yes	1
3	Have you recently completed your education and training, and there no is major Gap (more then one year after the education)	Yes	1
4	Are you, willing to work on-premises (there is no work from home)	Yes	1
5	Can you write simple Java Programs on your own	Yes	1
6	Are you, available immediately to join us, if we hire you	Yes	1
7	Did you complete any kind of Developer Certification in Java	Yes	1
8	Do you know how to write unit testcases using Junit	Yes	1
9	if we hire you, will you be prepared to stay and work with us at least for two years	Yes	1
10	Will you be flexible any role in Software Engineering	Yes	1
11	Explain about yourself why you are the right candiate for this Job	video response	1

Overall Score : 11/11 (Promoted)

Coding Test

Completed on : 02-Dec-2024 12:28 PM

1. Question: Implement an EMI Calculator in Java. The objective is to create a program that calculates the Equated Monthly Installment (EMI) for a given loan. The EMI is calculated based on the principal loan amount, annual rate of interest, and loan tenure. Sample Input : Principal amount: 500000 Annual rate of interest: 7.5 Loan tenure: 20 Expected Output : The calculated EMI is: 3982.71
Marks Scored : 0

```
public class EMI_Calculator {

   // Method to calculate EMI
   public double calculateEMI(double principal, double annualRateOfInterest, int loanTenure) {
      double monthlyRateOfInterest = annualRateOfInterest / (12*100);
      int loanTenureInMonths = loanTenure *12;
      double emi=(principal* monthlyRateOfInterest *Math.pow(1+ monthlyRateOfInterest, loanTenureInMonths))/
               (Math.pow(1+ monthlyRateOfInterest, loanTenureInMonths) -1);
      return emi;

   }

   // Main method to test the EMI calculation
   public static void main(String[] args) {
      // Create an instance of EMI_Calculator
      EMI_Calculator calculator = new EMI_Calculator();

      // Sample inputs
      double principal = 500000; // Example principal amount (e.g., 5,00,000)
      double annualRateOfInterest = 7.5; // Example annual interest rate (e.g., 7.5%)
      int loanTenure = 20; // Example loan tenure in years (e.g., 20 years)

      // Calculate EMI
      double emi = calculator.calculateEMI(principal, annualRateOfInterest, loanTenure);

      // Output the result
      System.out.println("The calculated EMI is: " + emi);
   }
}
```

2. Write a Java program that counts the occurrences of words in a given input string that are of a specified minimum length or longer. The program should return a map of words sorted in descending order of their frequency. If two words have the same frequency, they should be sorted alphabetically in ascending order. Sample Input : Ignore case sensitivity (e.g., "Test" and "test" should be treated as the same word). Words are sequences of alphanumeric characters. Punctuation and other symbols should not be considered part of words. Words shorter than the specified minimum length should be excluded from the count. Expected Output : Words with minimum length 8: exceptional: 2 wonderful: 1
Marks Scored : 0

```
import java.util.*;
import java.util.stream.Collectors;

public class WordCounter {

   public static Map getWordsWithMinLength(String input, int minLength) {

            // CODE HERE
      Map wordCounts = new HashMap<>();
      String[] words = input.split("\\s+");
      for (String word :words) {
         if (word.length() >= minLength){
            word = word.toLowerCase();
            wordCounts.put(word, wordCounts.getOrDefault(word,0)+1);
            }
         }
         return wordCounts;
      }
      return wordCounts;

      //return Arrays.stream(input.split("\\s+"))
        //  .filter(word -> word.length() >=minLength)
         // .map(String::toLowerCase)
         // .collect(Collectors.toMap(
          //    word -> word,
           //   word -> 1,
            //  Integer::sum
          //

   public static void main(String[] args) {
      // Example input string
      String: "This is a test string with some exceptional words like exceptional and wonderful words."

      // Set minimum length for the words
      int minLength = 8;

      // Get words with the specified minimum length and their count
      Map wordCounts = getWordsWithMinLength(input, minLength);

      // Print the result
      System.out.println("Words with minimum length " + minLength + ":");
      wordCounts.forEach((word, count) -> System.out.println(word + ": " + count));
   }
}
```

3. Question Write a program to demonstrate most simple usage, application of design pattern prototype. Expected Output: Original: Shape[type=Circle, color=Red] Cloned: Shape[type=Circle, color=Red] After modifying cloned object: Original: Shape[type=Circle, color=Red] Cloned: Shape[type=Circle, color=Blue]
Marks Scored : 6

```
// PrototypeDemo.java
class Shape implements Cloneable {
   private String type;
   private int[] dimensions;

   public Shape(String type, int[] dimensions) {
       // CODE HERE
       this.type=type;
       this.dimensions=dimensions.clone();
   }

   public String getType() {
       // CODE HERE
       return type;
   }

   public int[] getDimensions() {
       // CODE HERE
       return dimensions.clone();
   }

   @Override
   protected Shape clone() {
      try {
                  // CODE HERE
          Shape cloned=(Shape) super.clone();
          cloned.dimensions=dimensions.clone();
          return cloned;
      } catch (CloneNotSupportedException e) {
          e.printStackTrace();
          return null;
      }
   }
}

public class PrototypeDemo {
   public static void main(String[] args) {
       Shape original = new Shape("Circle", new int[]{7});
       Shape cloned = original.clone();
       cloned.getDimensions()[0] =14;
       System.out.println("Original Shape: " + original.getType() + " with dimensions " + java.util.Arrays.toString(original.getDimensions()));
       System.out.println("Cloned Shape: " + cloned.getType() + " with dimensions " + java.util.Arrays.toString(cloned.getDimensions()));
   }
}
```

Overall Score : 6/12 (Promoted)

Interview

Interviewed on : 03-Dec-2024 10:30 AM **Interviewd by : Srinivas Gadepalli**

Interview Summary	
Personal	3.33
OOPS	1.67
Java Programming	2.0
Java Technology	2.0
Average interview rating	2.25

Question wise rating:

Personal	
what is the Necessity of Job	3
Tell Me About yourself	3
Career Plan	4
Average rating	3.33

OOPS	
UML	3
Abstraction	1
Inheritance	1
Average rating	1.67

Java Programming	
OOPS Implementation	1
JUNIT?	3
Memory management	2
Average rating	2.0

Java Technology	
JVM	1
JDK/JRE	3
Average rating	2.0

Interviewer Remarks:

No Remarks

Interviewer's Feedback:

Interviewer	Decision	Notes
Srinivas Gadepalli	Not Hire	not yet hands on .. needs training good attitude