Rajalakshmi Engineering College

Name: Nishant Gurung
Email: 241501131@rajalakshmi.edu.in
Roll no: 241501131
Phone: 8670069246
Branch: REC
Department: AI & ML - Section 3
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q1

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : COD

1. Problem Statement

A city traffic management system needs to track vehicles entering a toll booth. Each vehicle is uniquely identified by its registration number. The system should allow adding vehicles to a record, ensuring that no duplicate registration numbers exist. The vehicles should be stored in a HashSet, which does not guarantee any specific order.

Your task is to implement a program using a HashSet that allows adding vehicle details and displaying the records.

Input Format

The first line of input contains an integer N - the number of vehicles.

The next N lines contain details of each vehicle in the format: "RegNumber

OwnerName VehicleType"

1. RegNumber (String) - A unique registration number (Alphanumeric).
2. OwnerName (String) - The name of the vehicle owner.
3. VehicleType (String, Car, Bike, or Truck) - The type of vehicle.

If a vehicle with the same registration number is already present, ignore the duplicate entry.

Output Format

The output prints the unique vehicle records in any order (since HashSet does not maintain order).

Output format: "RegNumber OwnerName VehicleType"

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 5
KA01AB1234 John Car
MH02CD5678 Alice Bike
DL03EF9012 Bob Truck
TN04GH3456 Mike Car
KA01AB1234 John Car
Output: TN04GH3456 Mike Car
KA01AB1234 John Car
MH02CD5678 Alice Bike
DL03EF9012 Bob Truck
```

Answer

```
import java.util.*;

class Vehicle {
   String regNumber, ownerName, vehicleType;

   public Vehicle(String regNumber, String ownerName, String vehicleType) {
      this.regNumber = regNumber;
      this.ownerName = ownerName;
      this.vehicleType = vehicleType;
```

```
    }

    @Override
    public boolean equals(Object o) {
        if (this == o) return true;
        if (!(o instanceof Vehicle)) return false;
        Vehicle v = (Vehicle) o;
        return regNumber.equals(v.regNumber);
    }

    @Override
    public int hashCode() {
        return regNumber.hashCode();
    }

    @Override
    public String toString() {
        return regNumber + " " + ownerName + " " + vehicleType;
    }
}

class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int N = Integer.parseInt(sc.nextLine());
        HashSet<Vehicle> vehicles = new HashSet<>();
        for (int i = 0; i < N; i++) {
            String[] parts = sc.nextLine().split(" ");
            vehicles.add(new Vehicle(parts[0], parts[1], parts[2]));
        }
        for (Vehicle v : vehicles) {
            System.out.println(v);
        }
    }
}
```

Status : Correct **_Marks : 10/10_**

Rajalakshmi Engineering College

Name: Nishant Gurung
Email: 241501131@rajalakshmi.edu.in
Roll no: 241501131
Phone: 8670069246
Branch: REC
Department: AI & ML - Section 3
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q2

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : COD

1. Problem Statement

John is organizing a fruit festival, and the quantities of various fruits are stored in a HashMap where fruit names are keys and quantities are values.

Help him develop a program to find the total quantity of fruits for the festival by summing up the values in the HashMap.

Input Format

The input consists of fruit quantities in the format 'fruitName:quantity', where fruitName is the name of the fruit(a string), and quantity is a double value representing the quantity.

The input is terminated by entering "done".

Output Format

The output prints a double value, representing the sum of values in the HashMap, rounded off to two decimal places.

If the value is not numeric, print "Invalid input".

If any special characters other than ':' are entered, print "Invalid format".

Refer to the sample output for formatting specifications.

Sample Test Case

Input: Banana:15.2
Orange:56.3
Mango:47.3
done

Output: 118.80

Answer

```
import java.util.*;

class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in); // Create a Scanner to read user input

      HashMap<String, Double> fruits = new HashMap<>();

      // Infinite loop to continuously read input until user types "done"
      while (true) {
         String input = sc.nextLine(); // Read one line of input from the user

         // If user types "done", exit the loop
         if (input.equals("done")) break;

         // Validate input: must contain exactly one ":" separator
         // If it doesn't contain ":" OR has more than one ":", it's invalid
         if (!input.contains(":") || input.indexOf(":") != input.lastIndexOf(":")) {
            System.out.println("Invalid format"); // Print error message
            return; // Stop program execution
         }
```

```
            // Split the input into two parts (fruit name and quantity)
            String[] parts = input.split(":");

            // If splitting doesn't give exactly two parts, it's invalid
            if (parts.length != 2) {
                System.out.println("Invalid format");
                return; // Stop program execution
            }

            try {
                // Convert the second part (quantity) to a double
                double qty = Double.parseDouble(parts[1]);

                fruits.put(parts[0], qty);
            }
            catch (NumberFormatException e) {
                // If conversion fails (non-numeric quantity), show error and exit
                System.out.println("Invalid input");
                return;
            }
        }

        // Calculate the total quantity of all fruits
        double sum = 0;
        for (double val : fruits.values()) {
            sum += val; // Add each fruit’s quantity to the sum
        }

        // Print the total sum rounded to two decimal places
        System.out.printf("%.2f\n", sum);
    }
}
```

Status : Correct ***Marks : 10/10***

Rajalakshmi Engineering College

Name: Nishant Gurung
Email: 241501131@rajalakshmi.edu.in
Roll no: 241501131
Phone: 8670069246
Branch: REC
Department: AI & ML - Section 3
Batch: 2028
Degree: B.E - AI & ML

Scan to verify results

2024_28_III_OOPS Using Java Lab

2028_REC_OOPS using Java_Week 10_Q3

Attempt : 1
Total Mark : 10
Marks Obtained : 10

Section 1 : COD

1. Problem Statement

Priya is analyzing encrypted messages in a research project. She wants to analyze the frequency of each character in a given paragraph. The characters should be stored in a TreeMap so that the output is sorted in ascending order of characters automatically.

You are required to build a Java program that:

Uses a TreeMap<Character, Integer> to count how many times each character appears in the message.Ignores spaces and considers only alphabets (case-sensitive).Outputs the frequencies of characters in sorted order.

You must use a TreeMap in the class named MessageAnalyzer.

Input Format

The first line of input contains an integer n, the number of lines in the message.

The next n lines each contain a string (the encrypted message line).

Output Format

The first line of output prints: "Character Frequency:"

Then print each character and its frequency in the format: "<character>: <count>"

Refer to the sample output for formatting specifications.

Sample Test Case

```
Input: 2
Hello World
Java
Output: Character Frequency:
H: 1
J: 1
W: 1
a: 2
d: 1
e: 1
l: 3
o: 2
r: 1
v: 1
```

Answer

```
// You are using Java
import java.util.*;

class MessageAnalyzer {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in); // Create a Scanner object to read
input

        // Read the number of lines (messages) to analyze
        int n = Integer.parseInt(sc.nextLine());
        TreeMap<Character, Integer> map = new TreeMap<>();

        // Loop to read 'n' lines from input
        for (int i = 0; i < n; i++) {
            String line = sc.nextLine(); // Read a full line of text

            // Convert the line into an array of characters and loop through each one
            for (char ch : line.toCharArray()) {
                // Check if the character is a letter (ignores digits, spaces, punctuation,
etc.)
                if (Character.isLetter(ch)) {
                    // Update the frequency count in the map
                    // If 'ch' already exists, increment its count by 1
                    // If not, set its count to 1 (default value 0 + 1)
                    map.put(ch, map.getOrDefault(ch, 0) + 1);
                }
            }
        }

        // Display the frequency of each character
        System.out.println("Character Frequency:");

        for (Map.Entry<Character, Integer> entry : map.entrySet()) {
            System.out.println(entry.getKey() + ": " + entry.getValue());
        }
    }
}
```

Status : Correct ***Marks : 10/10***