Rajalakshmi Engineering College

Name: Darshan S
Email: 241801040@rajalakshmi.edu.in
Roll no: 241801040
Phone: 7305911089
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

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;
   String ownerName;
   String vehicleType;

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

```
        this.ownerName = ownerName;
        this.vehicleType = vehicleType;
    }

    // Two vehicles are the same if they have the same registration number
    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Vehicle other = (Vehicle) obj;
        return regNumber.equals(other.regNumber);
    }

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

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

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int N = sc.nextInt();
        HashSet<Vehicle> vehicles = new HashSet<>();

        for (int i = 0; i < N; i++) {
            String regNumber = sc.next();
            String ownerName = sc.next();
            String vehicleType = sc.next();

            Vehicle v = new Vehicle(regNumber, ownerName, vehicleType);
            vehicles.add(v);  // Duplicate regNumber ignored automatically
        }

        // Print all unique vehicle records line by line
        for (Vehicle v : vehicles) {
```

```
            System.out.println(v);
        }

        sc.close();
    }
}
```

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

Rajalakshmi Engineering College

Name: Darshan S
Email: 241801040@rajalakshmi.edu.in
Roll no: 241801040
Phone: 7305911089
Branch: REC
Department: AI & DS - Section 4
Batch: 2028
Degree: B.E - AI & DS

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

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

public class Main {
   public static void main(String[] args) {
      Scanner sc = new Scanner(System.in);
      HashMap<String, Double> fruits = new HashMap<>(); //    Whitelist
requirement

      double total = 0.0;
      boolean invalidInput = false;
      boolean invalidFormat = false;

      while (true) {
         String input = sc.next();

         if (input.equalsIgnoreCase("done")) {
            break;
         }

         // Check for invalid characters or missing colon
```

```
            if (!input.contains(":") || input.matches(".*[^a-zA-Z0-9:.].*")) {
                invalidFormat = true;
                break;
            }

            String[] parts = input.split(":");
            if (parts.length != 2) {
                invalidFormat = true;
                break;
            }

            String fruit = parts[0];
            String qtyStr = parts[1];

            try {
                double quantity = Double.parseDouble(qtyStr);
                fruits.put(fruit, quantity);
            } catch (NumberFormatException e) {
                invalidInput = true;
                break;
            }
        }

        sc.close();

        if (invalidFormat) {
            System.out.println("Invalid format");
        } else if (invalidInput) {
            System.out.println("Invalid input");
        } else {
            for (double value : fruits.values()) {
                total += value;
            }
            DecimalFormat df = new DecimalFormat("0.00");
            System.out.println(df.format(total));
        }
    }
}
```

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