Java mini project – waste Management System

-SANTHOSH.S

(231401093)

CSBS-B, 2ND YEAR

**AIM:**

To develop a **Waste Management System** to effectively store, track, and manage waste records for a given location.

**ALGORITHM:**

1. **Define the Purpose**:  
   The system must enable users to record waste types, quantities, and locations systematically.
2. **Problem Identification**:
   * Existing manual waste management methods are inefficient.
   * Lack of proper tracking leads to mismanagement.
3. **Input Requirements**:
   * Accept waste type as a string.
   * Accept quantity in kilograms as a number.
   * Accept location as a string.
4. **Storage Mechanism**:
   * Use an internal data structure (e.g., ArrayList) to store waste records, each with the fields: type, quantity, and location.
5. **User Interaction**:
   * Implement a menu-driven interface for:
     + Adding new waste records.
     + Viewing all stored records.
     + Deleting a specific record by index.
     + Exiting the system.
6. **Processing Logic**:
   * Match user actions with appropriate functionalities:
     + **Add**: Store the input data in the ArrayList.
     + **View**: Retrieve and display all records.
     + **Delete**: Remove a record from the list based on user input.
7. **Validation**:
   * Ensure inputs are valid (e.g., non-negative quantities, non-empty waste types).
   * Handle incorrect menu options or record indices gracefully.
8. **Output Results**:
   * Display success or error messages for operations.
   * Provide a list of records in a clear, user-friendly format.
9. **Repeat Until Exit**:
   * Loop the program until the user chooses the "Exit" option.

**PROGRAM:**

**import java.util.ArrayList;**

**import java.util.Scanner;**

**// Class to represent a waste record**

**class WasteRecord {**

**String wasteType;**

**double quantity; // in kilograms**

**String location;**

**WasteRecord(String wasteType, double quantity, String location) {**

**this.wasteType = wasteType;**

**this.quantity = quantity;**

**this.location = location;**

**}**

**@Override**

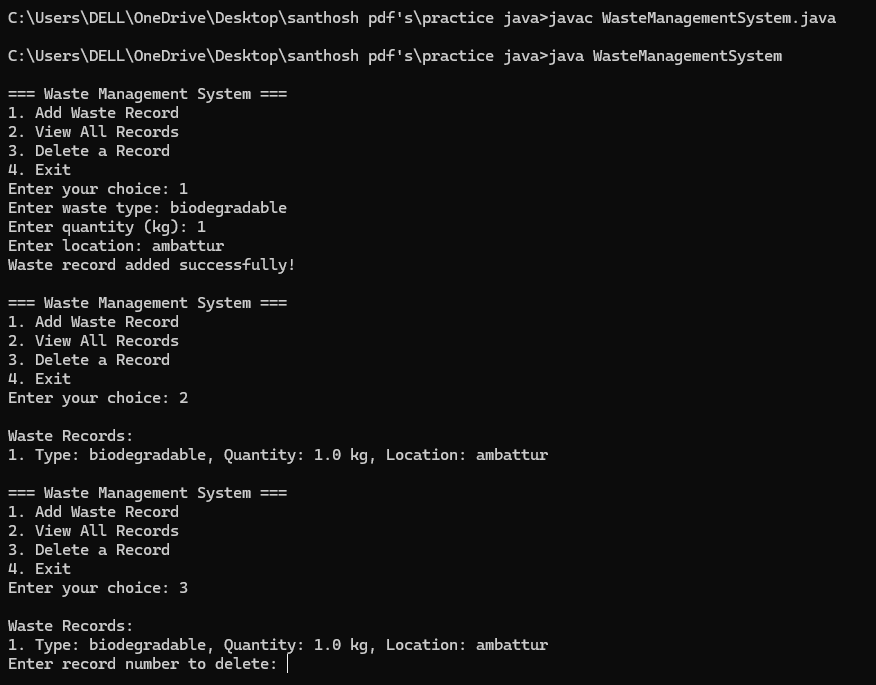
**public String toString() {**

**return "Type: " + wasteType + ", Quantity: " + quantity + " kg, Location: " + location;**

**}**

**}**

**OUTPUT:**

****

**// Main class with the main method**

**public class WasteManagementSystem {**

**private static ArrayList<WasteRecord> wasteRecords = new ArrayList<>();**

**private static Scanner scanner = new Scanner(System.in);**

**public static void main(String[] args) {**

**while (true) {**

**System.out.println("\n=== Waste Management System ===");**

**System.out.println("1. Add Waste Record");**

**System.out.println("2. View All Records");**

**System.out.println("3. Delete a Record");**

**System.out.println("4. Exit");**

**System.out.print("Enter your choice: ");**

**int choice = scanner.nextInt();**

**scanner.nextLine(); // Consume newline**

**switch (choice) {**

**case 1:**

**addRecord();**

**break;**

**case 2:**

**viewRecords();**

**break;**

**case 3:**

**deleteRecord();**

**break;**

**case 4:**

**System.out.println("Exiting the system. Goodbye!");**

**return;**

**default:**

**System.out.println("Invalid choice! Please try again.");**

**}**

**}**

**}**

**// Method to add a waste record**

**private static void addRecord() {**

**System.out.print("Enter waste type: ");**

**String wasteType = scanner.nextLine();**

**System.out.print("Enter quantity (kg): ");**

**double quantity = scanner.nextDouble();**

**scanner.nextLine(); // Consume newline**

**System.out.print("Enter location: ");**

**String location = scanner.nextLine();**

**wasteRecords.add(new WasteRecord(wasteType, quantity, location));**

**System.out.println("Waste record added successfully!");**

**}**

**// Method to view all waste records**

**private static void viewRecords() {**

**if (wasteRecords.isEmpty()) {**

**System.out.println("No records found.");**

**} else {**

**System.out.println("\nWaste Records:");**

**for (int i = 0; i < wasteRecords.size(); i++) {**

**System.out.println((i + 1) + ". " + wasteRecords.get(i));**

**}**

**}**

**}**

**// Method to delete a specific waste record**

**private static void deleteRecord() {**

**viewRecords();**

**if (!wasteRecords.isEmpty()) {**

**System.out.print("Enter record number to delete: ");**

**int recordNumber = scanner.nextInt();**

**scanner.nextLine(); // Consume newline**

**if (recordNumber > 0 && recordNumber <= wasteRecords.size()) {**

**wasteRecords.remove(recordNumber - 1);**

**System.out.println("Record deleted successfully!");**

**} else {**

**System.out.println("Invalid record number!");**

**}**

**}**

**}**

**}**

**RESULT:**

The java program for waste management system executed successfully