Name: Josaiah Murfeal Dkhar(2447125)

# **LAB EXERCISE 7**

**Question**:.

Create a JAVA program for a Smart Waste Disposal System which includes functionalities such as adding waste processing method for **Organic Waste** that calculates compost weight. The compost weight is 80% of the original organic waste weight. Add a method that lists the waste items that are eligible for composting and their compost weight.

**Case Study**: Smart Waste Disposal System

A city implements a **Smart Waste Disposal System** to manage waste disposal efficiently. The system categorizes waste into different types (e.g., Organic, Plastic, Metal, etc.) and calculates the disposal cost and recyclable waste weight. During a pilot run, the city realized:

* 60% of the waste was recyclable, with plastics contributing significantly.
* Organic waste required composting rather than typical disposal methods.
* Citizens suggested improving system functionality by displaying compost-eligible items and enabling real-time cost calculations.

Analyze the system's efficiency and suggest how functional programming in Java enhances the modularity and scalability of the application.

**Summary:**

# The Smart Waste Disposal System demonstrates the efficient management of waste using Java's functional programming constructs like Predicate, Function, Consumer, and Supplier. It simplifies processing waste by categorizing and calculating total weights, costs, and recyclable contributions. The system ensures scalability for additional waste processing needs, such as composting organic waste.Key components:

# **WasteType Enum:** Classifies waste into ORGANIC, PLASTIC, METAL, PAPER, and GLASS.

# **WasteItem Class:** Stores a waste item’s description and category.

# **WasteDisposalSystem Class:** Maintains a list of waste items and provides methods to add and display them.

# **Main Class:** Handles user input and integrates with the waste disposal system.

# **Diagram:**

# **A screenshot of a computer Description automatically generated**

# **Code:**

**WasteType.java**

public enum WasteType {

    ORGANIC,

    PLASTIC,

    METAL,

    GLASS,

    PAPER,

    OTHER

}

**Waste.java**

public class Waste {

    private WasteType type;

    private double weight;

    public Waste(WasteType type, double weight) {

        this.type = type;

        this.weight = weight;

    }

    public WasteType getType() {

        return type;

    }

    public double getWeight() {

        return weight;

    }

    @Override

    public String toString() {

        return "Waste{" +

                "type=" + type +

                ", weight=" + weight +

                '}';

    }

}

**WasteDisposalSystem.java**

import java.util.ArrayList;

import java.util.List;

import java.util.function.Consumer;

import java.util.function.Function;

import java.util.function.Predicate;

import java.util.function.Supplier;

import java.util.stream.Collectors;

public class WasteDisposalSystem {

    private List<Waste> wasteList = new ArrayList<>();

    public void addWaste(Waste waste) {

        wasteList.add(waste);

    }

    public void processWaste() {

        // Add a new waste item using Supplier

        addNewWasteItem(() -> new Waste(WasteType.PLASTIC, 2.5));

        // Define predicates, functions, and consumers

        Predicate<Waste> isRecyclable = waste -> waste.getType() == WasteType.PLASTIC;

        Predicate<Waste> isHeavy = waste -> waste.getWeight() > 2.0;

        Function<Waste, Double> disposalCost = waste -> waste.getWeight() \* 1.5;

        Function<Waste, String> categorizeWaste = waste -> isRecyclable.test(waste) ? "Recyclable" : "Non-Recyclable";

        Consumer<Waste> printWaste = waste -> System.out.println("Processing waste: " + waste);

        Consumer<Waste> printCategory = waste -> System.out.println("Category: " + categorizeWaste.apply(waste));

        // Calculate and print total weight of recyclable waste

        double totalRecyclableWeight = calculateTotalWeight(isRecyclable);

        System.out.println("Total weight of recyclable waste: " + totalRecyclableWeight + " kg");

        // Process each waste item

        wasteList.stream()

                .filter(isRecyclable.and(isHeavy))

                .forEach(waste -> {

                    printWaste.accept(waste);

                    printCategory.accept(waste);

                    double cost = disposalCost.apply(waste);

                    System.out.println("Disposal cost: $" + cost);

                });

    }

    private void addNewWasteItem(Supplier<Waste> wasteSupplier) {

        Waste newWaste = wasteSupplier.get();

        addWaste(newWaste);

    }

    private double calculateTotalWeight(Predicate<Waste> filter) {

        return wasteList.stream()

                .filter(filter)

                .mapToDouble(Waste::getWeight)

                .sum();

    }

}

**Main.java**

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        WasteDisposalSystem system = new WasteDisposalSystem();

        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the number of waste items:");

        int numberOfItems = scanner.nextInt();

        scanner.nextLine(); // Consume newline

        for (int i = 0; i < numberOfItems; i++) {

            System.out.println("Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):");

            String typeInput = scanner.nextLine();

            WasteType type = WasteType.valueOf(typeInput.toUpperCase());

            System.out.println("Enter waste weight:");

            double weight = scanner.nextDouble();

            scanner.nextLine(); // Consume newline

            system.addWaste(new Waste(type, weight));

        }

        system.processWaste();

        scanner.close();

    }

}

[Scroll down for Output]

OUTPUT:

PS D:\2MCA\JAVA\JAVA\Lab-Exercise-07> javac Main.java

PS D:\2MCA\JAVA\JAVA\Lab-Exercise-07> java Main

Enter the number of waste items:

3

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

plastic

Enter waste weight:

2.5

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

metal

Enter waste weight:

3

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

GLASS

Enter waste weight:

5.9

Total weight of recyclable waste: 5.0 kg

Processing waste: Waste{type=PLASTIC, weight=2.5}

Category: Recyclable

Disposal cost: $3.75

Processing waste: Waste{type=PLASTIC, weight=2.5}

Category: Recyclable

Disposal cost: $3.75

PS D:\2MCA\JAVA\JAVA\Lab-Exercise-07> java Main

PS D:\2MCA\JAVA\JAVA\Lab-Exercise-07> java Main

Enter the number of waste items:

6

6

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

ORGANIC

Enter waste weight:

2.6

2.6

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

PLASTIC

PLASTIC

Enter waste weight:

8.6

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

METAL

Enter waste weight:

7.2

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

paper

Enter waste weight:

88.2

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

Other

Enter waste weight:

77.2

Enter waste type (ORGANIC, PLASTIC, METAL, GLASS, PAPER, OTHER):

Metal

Enter waste weight:

88.8

Total weight of recyclable waste: 11.1 kg

Processing waste: Waste{type=PLASTIC, weight=8.6}

Category: Recyclable

Disposal cost: $12.899999999999999

Processing waste: Waste{type=PLASTIC, weight=2.5}

Category: Recyclable

Disposal cost: $3.75

PS D:\2MCA\JAVA\JAVA\Lab-Exercise-07>

**Inference**

From the implementation and case study, it is inferred that functional programming and stream-based processing significantly improve system modularity and efficiency. Integrating predictive analytics (e.g., compost weight prediction) can enhance the system's decision-making capabilities for waste management.