Name: Josaiah Murfeal Dkhar(2447125)

**LAB EXERCISE 3**

**Question:** Create a console application for your domain using Class, Inheritance, Abstract Classes, Final and Static Keywords, Constructors with Constructor Chaining, Method Chaining, and StringBuilder. The application should allow user input to define different waste disposal types, such as recyclable and hazardous items, and generate a report for each type using efficient string manipulation with StringBuilder. Create a case study for your domain and implement the above mentioned concepts.

**Case Study**: Smart Waste Disposal System

The **Smart Disposal Management System** is a Java-based console application designed to streamline and categorize waste management processes. This application allows waste management operators to input, categorize, and manage waste disposal details for recyclable and hazardous waste types. The system emphasizes object-oriented programming principles, enabling efficient handling of user-defined waste details while maintaining clear structure and organization in code.Summary:

# **Summary:**

The Smart Disposal Management System is a console-based application that enables effective waste categorization and disposal management through object-oriented principles. The application allows users to define and categorize waste items interactively, with a focus on recyclable and hazardous types, each requiring distinct handling methods. Leveraging Java features like inheritance, abstract classes, final and static keywords, constructor and method chaining, and efficient string manipulation with StringBuilder, the system efficiently captures and processes user-defined data.

Prompting users to input details for each waste item, such as type, weight, and disposal method, the application dynamically generates reports to aid in decision-making for proper waste handling. This case study demonstrates a scalable, user-friendly solution that enhances waste management processes, offering an adaptable framework suitable for various waste categories while maintaining secure and efficient data management.

# **Diagram**

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

# **Code:**

DisposalItem.java

public abstract class DisposalItem {

protected final String itemId;

protected final String itemType;

protected final int weight;

public DisposalItem(String itemId, String itemType, int weight) {

this.itemId = itemId;

this.itemType = itemType;

this.weight = weight;

}

public abstract void process();

public abstract String generateReport(); // Abstract method for generating a report

public final String getItemInfo() {

return "ID: " + itemId + ", Type: " + itemType + ", Weight: " + weight + "kg";

}

}

**HazardousItem.java**

public class HazardousItem extends DisposalItem {

private String disposalMethod;

public HazardousItem(String itemId, String itemType, int weight, String disposalMethod) {

super(itemId, itemType, weight); // Constructor chaining with super

this.disposalMethod = disposalMethod;

}

@Override

public void process() {

System.out.println("Processing hazardous item: " + itemType + " with ID: " + itemId);

System.out.println("Using disposal method: " + disposalMethod);

}

public HazardousItem setDisposalMethod(String disposalMethod) { // Method chaining example

this.disposalMethod = disposalMethod;

return this;

}

public final String generateReport() { // Using final to prevent overriding

StringBuilder report = new StringBuilder();

report.append("Hazardous Item Report\n")

.append(getItemInfo()).append("\n")

.append("Disposal Method: ").append(disposalMethod);

return report.toString();

}

}

**RecyclableItem.java**

public class RecyclableItem extends DisposalItem {

private String recyclingMethod;

public RecyclableItem(String itemId, String itemType, int weight, String recyclingMethod) {

super(itemId, itemType, weight);

this.recyclingMethod = recyclingMethod;

}

@Override

public void process() {

System.out.println("Processing recyclable item: " + itemType + " with ID: " + itemId);

System.out.println("Using recycling method: " + recyclingMethod);

}

public RecyclableItem setRecyclingMethod(String recyclingMethod) { // Method chaining example

this.recyclingMethod = recyclingMethod;

return this;

}

public String generateReport() {

StringBuilder report = new StringBuilder();

report.append("Recyclable Item Report\n")

.append(getItemInfo()).append("\n")

.append("Recycling Method: ").append(recyclingMethod);

return report.toString();

}

}

**DisposalTest.java**

import java.util.Scanner;

public class DisposalTest {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Welcome to the Smart Disposal Management System");

System.out.println("Choose the type of disposal:");

System.out.println("1 - Recyclable (e.g., Plastic, Metal, Glass)");

System.out.println("2 - Hazardous (e.g., Battery, Chemicals, Medical Waste)");

System.out.print("Enter disposal type (1 or 2): ");

int disposalType = scanner.nextInt();

scanner.nextLine(); // Consume newline

DisposalItem item = null;

if (disposalType == 1) {

System.out.print("Enter Recyclable Item ID (e.g., R001): ");

String itemId = scanner.nextLine();

System.out.print("Enter recyclable material type (e.g., Plastic, Metal, Glass): ");

String itemType = scanner.nextLine();

System.out.print("Enter weight in kilograms (e.g., 5): ");

int weight = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter recycling method (e.g., Thermal Recycling, Chemical Recycling): ");

String recyclingMethod = scanner.nextLine();

item = new RecyclableItem(itemId, itemType, weight, recyclingMethod);

} else if (disposalType == 2) {

System.out.print("Enter Hazardous Item ID (e.g., H001): ");

String itemId = scanner.nextLine();

System.out.print("Enter hazardous material type (e.g., Battery, Chemicals, Medical Waste): ");

String itemType = scanner.nextLine();

System.out.print("Enter weight in kilograms (e.g., 3): ");

int weight = scanner.nextInt();

scanner.nextLine(); // Consume newline

System.out.print("Enter disposal method (e.g., Incineration, Secure Landfill): ");

String disposalMethod = scanner.nextLine();

item = new HazardousItem(itemId, itemType, weight, disposalMethod);

} else {

System.out.println("Invalid choice. Please restart the program and select a valid option.");

scanner.close();

return;

}

// Processing the item and generating a report

System.out.println("\nProcessing item...");

item.process();

System.out.println("\nGenerated Report:");

System.out.println(item.generateReport());

scanner.close();

}

}

[Scroll down for Output]

OUTPUT:

A screenshot of a computer program

Description automatically generated