```
 import java.util.ArrayList;
import java.util.Scanner;

public class PlantCareApp {

    // Inner Plant class
    static class Plant {
        String name;
        int waterInterval;    // in days
        int fertilizeInterval; // in days
        int daysSinceWatered;
        int daysSinceFertilized;

        Plant(String name, int waterInterval, int fertilizeInterval) {
            this.name = name;
            this.waterInterval = waterInterval;
            this.fertilizeInterval = fertilizeInterval;
            this.daysSinceWatered = 0;
            this.daysSinceFertilized = 0;
        }

        void passDay() {
            daysSinceWatered++;
            daysSinceFertilized++;
        }

        void water() {
            daysSinceWatered = 0;
            System.out.println(name + " has been watered! 💧");
        }

        void fertilize() {
            daysSinceFertilized = 0;
            System.out.println(name + " has been fertilized! 🌿");
        }

        void showStatus() {
            System.out.println("\nPlant: " + name);
            System.out.println("Days since last watered: " + daysSinceWatered);
            System.out.println("Days since last fertilized: " + daysSinceFertilized);
            if (daysSinceWatered >= waterInterval) {
                System.out.println("⚠️  Reminder: Time to water " + name + "!");
            }
            if (daysSinceFertilized >= fertilizeInterval) {
```

```
            System.out.println("⚠️  Reminder: Time to fertilize " + name + "!");
        }
    }
}

// Program entry point
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    ArrayList<Plant> plants = new ArrayList<>();

    System.out.println("🌱 Welcome to the Plant Care Reminder App 🌱");
    System.out.println("Supporting SDG 15 – Life on Land");

    while (true) {
        System.out.println("\nMenu:");
        System.out.println("1. Add new plant");
        System.out.println("2. Show all plants");
        System.out.println("3. Pass one day (advance days)");
        System.out.println("4. Water a plant");
        System.out.println("5. Fertilize a plant");
        System.out.println("6. Exit");
        System.out.print("Enter your choice: ");

        // Validate integer input
        int choice;
        if (sc.hasNextInt()) {
            choice = sc.nextInt();
            sc.nextLine(); // consume newline
        } else {
            System.out.println("Please enter a number (1-6).");
            sc.nextLine(); // discard invalid token
            continue;
        }

        switch (choice) {
            case 1:
                System.out.print("Enter plant name: ");
                String name = sc.nextLine().trim();
                if (name.isEmpty()) {
                    System.out.println("Name cannot be empty.");
                    break;
                }
                System.out.print("Enter watering interval (days): ");
                int w = readPositiveInt(sc);
```

```
            System.out.print("Enter fertilizing interval (days): ");
            int f = readPositiveInt(sc);
            plants.add(new Plant(name, w, f));
            System.out.println(name + " added successfully!");
            break;

        case 2:
            if (plants.isEmpty()) {
                System.out.println("No plants added yet.");
            } else {
                for (Plant p : plants) {
                    p.showStatus();
                }
            }
            break;

        case 3:
            if (plants.isEmpty()) {
                System.out.println("No plants to advance.");
            } else {
                for (Plant p : plants) p.passDay();
                System.out.println("🌞 One day has passed for all plants.");
            }
            break;

        case 4:
            if (plants.isEmpty()) {
                System.out.println("No plants available.");
                break;
            }
            System.out.print("Enter plant name to water: ");
            String waterName = sc.nextLine().trim();
            if (!performActionOnPlant(plants, waterName, "water")) {
                System.out.println("Plant '" + waterName + "' not found.");
            }
            break;

        case 5:
            if (plants.isEmpty()) {
                System.out.println("No plants available.");
                break;
            }
            System.out.print("Enter plant name to fertilize: ");
            String fertName = sc.nextLine().trim();
```

```
                if (!performActionOnPlant(plants, fertName, "fertilize")) {
                    System.out.println("Plant '" + fertName + "' not found.");
                }
                break;

            case 6:
                System.out.println("Thank you for caring for nature! 🌳");
                sc.close();
                return;

            default:
                System.out.println("Invalid choice. Try again.");
        }
    }
}

// Helper to read a positive integer from Scanner
private static int readPositiveInt(Scanner sc) {
    while (true) {
        if (sc.hasNextInt()) {
            int v = sc.nextInt();
            sc.nextLine();
            if (v > 0) return v;
            System.out.print("Please enter a positive number: ");
        } else {
            System.out.print("Please enter a valid number: ");
            sc.nextLine();
        }
    }
}

// Helper to find plant by name (case-insensitive) and perform action
private static boolean performActionOnPlant(ArrayList<Plant> plants, String targetName,
String action) {
    for (Plant p : plants) {
        if (p.name.equalsIgnoreCase(targetName)) {
            if (action.equals("water")) p.water();
            else if (action.equals("fertilize")) p.fertilize();
            return true;
        }
    }
    return false;
}
}
```

💡 Overview

This is a console-based Java program that helps users take care of their plants — by reminding them when to water and fertilize.

It supports SDG 15 – Life on Land (protecting plants and promoting greenery).

⚙️ How It Works

1. Plant class (inner class):

Stores plant details:
name, watering interval, fertilizing interval, and days since last care.

Has methods to:

passDay() → adds one day

water() → resets watering counter

fertilize() → resets fertilizing counter

showStatus() → shows reminders

2. Main program (PlantCareApp):

Uses a menu system with choices:

1. Add new plant

2. Show all plants

3. Pass one day

4. Water a plant

5. Fertilize a plant

6. Exit

3. ArrayList<Plant> stores all added plants.

4. Helper methods:

readPositiveInt() → ensures valid number input.

performActionOnPlant() → finds plant by name to water/fertilize it.

🌿 Purpose (SDG 15 – Life on Land)

Encourages people to care for plants and the environment through consistent reminders — promoting sustainable living.