Name: Tuana Melisa Aksoi

NO: 230104004903

Data 222 | Hw4 Report

Project Structure

The system was developed using the following classes:

- PlanetSystem: Main controller that processes commands and manages the tree structure.
- Node: Represents each celestial body (star, planet, moon) and stores its name, type, sensor data, and children.
- SensorData: Stores environmental data like temperature, pressure, humidity, and radiation.
- Main: Provides a terminal interface to enter commands and control the planetary system.

Design Decisions

- A **tree** structure was used, where each node could have multiple children (non-binary).
- **Recursive traversal** was used to search the tree and find radiation anomalies.
- A **stack** was used to trace the path from the root to a target node.
- **Error handling** was done using System.err.println() to show warnings without stopping the program.

Node.java

➤ **What It Does:**

Represents a **celestial object** in the system — like a **star**, **planet**, or **moon**.

➤ **Main Properties:**

- name: name of the object (e.g., "Earth")
- type: "Star", "Planet", or "Moon"
- sensorData: holds data like temperature, pressure, etc.
- children: list of other Node objects orbiting this node

➤ **Main Methods:**

- addChild(Node child): adds a child node (planet or moon)
- getChildren(): returns list of children
- Simple getters/setters

SensorData.java

Stores environmental measurements for celestial bodies.

Structure:

- A simple data container class that manages four key environmental metrics
- Used by Node objects to store measurement data

Key Components:

- Fields:
    - temperature: Measured in Kelvin
    - pressure: Measured in Pascals
    - humidity: Percentage from 0-100
    - radiation: Measured in Sieverts
- Methods:
    - Constructor that initializes all measurements
    - Getters and setters for all properties

Implementation Details:

- Uses double precision floating-point values for accuracy
- Follows Java bean pattern with private fields and accessor methods

PlanetSystem.java

- createPlanetSystem

This method creates the foundation of our planetary system with a star as the root.

Validate that humidity is 0 (stars must have 0% humidity)

Create a new SensorData object with provided measurements

Create a new Node with name, type "Star", and the SensorData

Set this node as the root of the planetary system

- findParent

If current node is null, return null (base case for recursion)

If current node's name matches target name, return current node

For each child of the current node: a. Recursively call findParent with the child node b. If result is not null, return the result

If no match found in any branch, return null

- addPlanet

Check if root exists; if not, print error and return

Validate humidity is between 0-100; if not, print error and return

Find parent node by name using findParent method

If parent not found, print error and return

If parent is a moon, print error (cannot add planet to a moon) and return

Check for duplicate name among parent's children a. If duplicate found, print error and return

Create new SensorData object with provided measurements

Create new Node with name, type "Planet", and SensorData

Add new planet node to parent's children list

Print confirmation message

- addSatellite

Check if root exists; if not, print error and return

Validate humidity is between 0-100; if not, print error and return

Find parent node by name using findParent method

If parent not found, print error and return

If parent is a moon, print error (cannot add satellite to another satellite) and return

Check for duplicate name among parent's children a. If duplicate found, print error and return

Create new SensorData object with provided measurements

Create new Node with name, type "Moon", and SensorData

Add new satellite node to parent's children list

- findRadiationAnomalies

Check if root exists; if not, print error and return

Validate threshold is non-negative; if negative, print error and return

Create empty ArrayList to store results

Call helper method findAnomalies with root, threshold, and result list

For each node in result list: a. Print node name and radiation level

- findAnomalies (helper method)

If current node is null, return (base case for recursion)

If current node's radiation level exceeds threshold: a. Add current node to result list

For each child of current node: a. Recursively call findAnomalies with child, threshold, and result list

- getPathTo

Check if root exists; if not, print error and return

Create empty Stack to store path

Call helper method findPath with root, target name, and path stack

If findPath returns false, print error (name not found) and return

Convert stack to array for easier printing

For each node name in path array: a. Print indentation based on depth in path b. Print node name with tree branch symbol

- findPath (helper method)

Push current node's name onto path stack

If current node's name matches target name: a. Return true (path found)

For each child of current node: a. Recursively call findPath with child, target name, and path stack b. If result is true, return true (propagate success up call stack)

If no path found in children, pop current node from stack (backtrack)

Return false (no path found in this branch)

- printMissionReport

  Check if root exists; if not, print error and return

  Call helper method printRecursive with root and depth 0

- printRecursive (helper method)

  If current node is null, return (base case for recursion)

  Print indentation based on depth parameter

  Print node information: name, type, and all sensor data (formatted)

  For each child of current node: a. Recursively call printRecursive with child and depth+1

- printMissionReport (with nodeName parameter)

  Check if root exists; if not, print error and return

  Find target node using findNodeDFS method

  If target node not found, print error and return

  Print report header with node name

  Call helper method printNodeData with target node

- printNodeData (helper method)

  Get SensorData from node parameter

  Print node name and type

  Print all sensor measurements formatted with units

- findNodeDFS

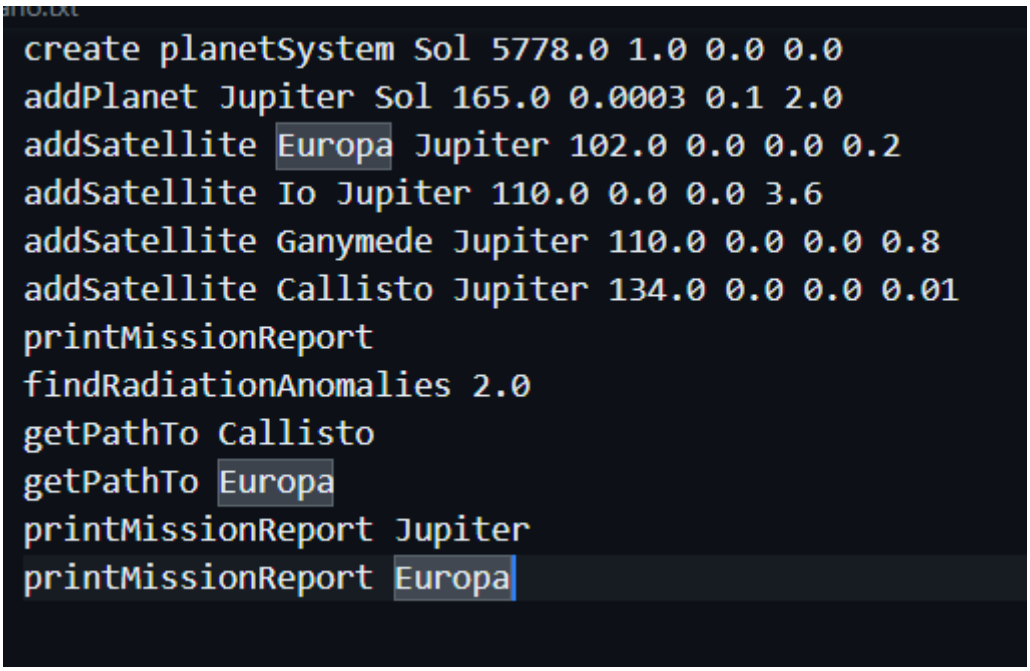
  If current node's name matches target name: a. Return current node (found)

  For each child of current node: a. Recursively call findNodeDFS with child and target name b. If result is not null, return result (propagate found node up call stack)

  If no match found in any branch, return null

OUTPUTS

Input no:1


create planetSystem Sol 5778.0 1.0 0.0 0.0
addPlanet Jupiter Sol 165.0 0.0003 0.1 2.0
addSatellite Europa Jupiter 102.0 0.0 0.0 0.2
addSatellite Io Jupiter 110.0 0.0 0.0 3.6
addSatellite Ganymede Jupiter 110.0 0.0 0.0 0.8
addSatellite Callisto Jupiter 134.0 0.0 0.0 0.01
printMissionReport
findRadiationAnomalies 2.0
getPathTo Callisto
getPathTo Europa
printMissionReport Jupiter
printMissionReport Europa

Output no:1

```
find src -name "*.java" > sources.txt
javac -d build @sources.txt
=== Deep Space Planetary System ===
Available commands:
 -create planetSystem <name> <temp> <pressure> <humidity> <radiation>
 -addPlanet <name> <parent> <temp> <pressure> <humidity> <radiation>
 -addSatellite <name> <parent> <temp> <pressure> <humidity> <radiation>
 -findRadiationAnomalies <threshold>
 -getPathTo <name>
 -printMissionReport
 -printMissionReport <name>
 -exit

> Created planetary system with star: 'Sol'.

> Added planet 'Jupiter' under 'Sol'.

> Added satellite 'Europa' under 'Jupiter'.

> Added satellite 'Io' under 'Jupiter'.

> Added satellite 'Ganymede' under 'Jupiter'.

> Added satellite 'Callisto' under 'Jupiter'.

> Sol (Star): 5778.00 Kelvin, 1.0000 Pascal, 0.00% humidity, 0.00 Sievert
  Jupiter (Planet): 165.00 Kelvin, 0.0003 Pascal, 0.10% humidity, 2.00 Sievert
    Europa (Moon): 102.00 Kelvin, 0.0000 Pascal, 0.00% humidity, 0.20 Sievert
    Io (Moon): 110.00 Kelvin, 0.0000 Pascal, 0.00% humidity, 3.60 Sievert
    Ganymede (Moon): 110.00 Kelvin, 0.0000 Pascal, 0.00% humidity, 0.80 Sievert
    Callisto (Moon): 134.00 Kelvin, 0.0000 Pascal, 0.00% humidity, 0.01 Sievert

> Anomalies:
Io: 3.60 Sieverts

> Path to Callisto:
└── Sol
    └── Jupiter
        └── Callisto

> Path to Europa:
└── Sol
    └── Jupiter
        └── Europa

>
===== MISSION REPORT FOR JUPITER =====
Jupiter (Planet)
Temperature: 165.0 Kelvin
Pressure: 3.0E-4 Pascals
Humidity: 0.1%
Radiation: 2.0 Sieverts
======================================

>
===== MISSION REPORT FOR EUROPA =====
Europa (Moon)
Temperature: 102.0 Kelvin
Pressure: 0.0 Pascals
Humidity: 0.0%
Radiation: 0.2 Sieverts
======================================
Mission control terminated.
```

Input no:2

```
create planetSystem Kerbol 5843.42 16402.1 0 63354
addPlanet Moho Kerbol 572.3 95.45 34.3 0.023
addPlanet Eve Moho 423.32 155.32 0.0001 0.4
addSatellite Gilly Eve 123.2 32.65 0.0 0.0
addPlanet Kerbin Eve 300.22 101.346 64.4 0.006
addSatellite Mun Kerbin 113.2 32.65 0.0 3.3
addSatellite Minmus Kerbin 123.2 32.65 2.0 12.0
addPlanet Duna Kerbin 300.22 101.346 64.4 0.006
addSatellite Ike Duna 123.2 32.65 2.0 0.0
addPlanet Dres Duna 300.22 101.346 64.4 44.006
addPlanet Jool Dres 300.22 101.346 64.4 212.006
addSatellite Laythe Jool 113.2 32.65 0.0 0.3
addSatellite Vall Jool 113.2 32.65 0.0 5.3
addSatellite Tylo Jool 113.2 32.65 0.0 0.3
addSatellite Bop Jool 113.2 32.65 0.0 22.3
addSatellite Pol Jool 113.2 32.65 0.0 112.3
addPlanet Eeloo Jool 572.3 95.45 34.3 0.023
getPathTo Bop
getPathTo Mun
findRadiationAnomalies 0
findRadiationAnomalies 120
printMissionReport Tylo
printMissionReport Eve
printMissionReport
exit
```

Output no: 2

```
> Created planetary system with star: 'Kerbol'.

> Added planet 'Moho' under 'Kerbol'.

> Added planet 'Eve' under 'Moho'.

> Added satellite 'Gilly' under 'Eve'.

> Added planet 'Kerbin' under 'Eve'.

> Added satellite 'Mun' under 'Kerbin'.

> Added satellite 'Minmus' under 'Kerbin'.

> Added planet 'Duna' under 'Kerbin'.

> Added satellite 'Ike' under 'Duna'.

> Added planet 'Dres' under 'Duna'.

> Added planet 'Jool' under 'Dres'.

> Added satellite 'Laythe' under 'Jool'.

> Added satellite 'Vall' under 'Jool'.

> Added satellite 'Tylo' under 'Jool'.

> Added satellite 'Bop' under 'Jool'.

> Added satellite 'Pol' under 'Jool'.

> Added planet 'Eeloo' under 'Jool'.

> Path to Bop:
└── Kerbol
    └── Moho
        └── Eve
            └── Kerbin
                └── Duna
                    └── Dres
                        └── Jool
                            └── Bop

> Path to Mun:
└── Kerbol
    └── Moho
        └── Eve
            └── Kerbin
                └── Mun
```

```
> Anomalies:
Kerbol: 63354.00 Sieverts
Moho: 0.02 Sieverts
Eve: 0.40 Sieverts
Kerbin: 0.01 Sieverts
Mun: 3.30 Sieverts
Minmus: 12.00 Sieverts
Duna: 0.01 Sieverts
Dres: 44.01 Sieverts
Jool: 212.01 Sieverts
Laythe: 0.30 Sieverts
Vall: 5.30 Sieverts
Tylo: 0.30 Sieverts
Bop: 22.30 Sieverts
Pol: 112.30 Sieverts
Eeloo: 0.02 Sieverts

> Anomalies:
Kerbol: 63354.00 Sieverts
Jool: 212.01 Sieverts

>
===== MISSION REPORT FOR TYLO =====
Tylo (Moon)
Temperature: 113.2 Kelvin
Pressure: 32.65 Pascals
Humidity: 0.0%
Radiation: 0.3 Sieverts
=======================================

>
===== MISSION REPORT FOR EVE =====
Eve (Planet)
Temperature: 423.32 Kelvin
Pressure: 155.32 Pascals
Humidity: 1.0E-4%
Radiation: 0.4 Sieverts
=======================================

> Kerbol (Star): 5843.42 Kelvin, 16402.1000 Pascal, 0.00% humidity, 63354.00 Sievert
  Moho (Planet): 572.30 Kelvin, 95.4500 Pascal, 34.30% humidity, 0.02 Sievert
    Eve (Planet): 423.32 Kelvin, 155.3200 Pascal, 0.00% humidity, 0.40 Sievert
      Gilly (Moon): 123.20 Kelvin, 32.6500 Pascal, 0.00% humidity, 0.00 Sievert
      Kerbin (Planet): 300.22 Kelvin, 101.3460 Pascal, 64.40% humidity, 0.01 Sievert
        Mun (Moon): 113.20 Kelvin, 32.6500 Pascal, 0.00% humidity, 3.30 Sievert
        Minmus (Moon): 123.20 Kelvin, 32.6500 Pascal, 2.00% humidity, 12.00 Sievert
        Duna (Planet): 300.22 Kelvin, 101.3460 Pascal, 64.40% humidity, 0.01 Sievert
          Ike (Moon): 123.20 Kelvin, 32.6500 Pascal, 2.00% humidity, 0.00 Sievert
          Dres (Planet): 300.22 Kelvin, 101.3460 Pascal, 64.40% humidity, 44.01 Sievert
            Jool (Planet): 300.22 Kelvin, 101.3460 Pascal, 64.40% humidity, 212.01 Sievert
              Laythe (Moon): 113.20 Kelvin, 32.6500 Pascal, 0.00% humidity, 0.30 Sievert
              Vall (Moon): 113.20 Kelvin, 32.6500 Pascal, 0.00% humidity, 5.30 Sievert
              Tylo (Moon): 113.20 Kelvin, 32.6500 Pascal, 0.00% humidity, 0.30 Sievert
              Bop (Moon): 113.20 Kelvin, 32.6500 Pascal, 0.00% humidity, 22.30 Sievert
              Pol (Moon): 113.20 Kelvin, 32.6500 Pascal, 0.00% humidity, 112.30 Sievert
              Eeloo (Planet): 572.30 Kelvin, 95.4500 Pascal, 34.30% humidity, 0.02 Sievert

> Mission control terminated.
```