CSE 222/505 Homework 4 - Deep Space Planetary System Analysis

Class Structure and Field

Node star

- **The root of the system tree.**
- **Represents the central star, to which all planets are attached.**
- **Every celestial body is represented using a Node object containing name, type, and sensor data.**

createPlanetSystem(String name, double temperature, double pressure, double humidity, double radiation)

- **Initializes the planetary system by creating a star node.**
- **Humidity must be exactly 0.0 (as stars don't have humidity).**
- **All parameters are validated using checkRange()**

```java
private boolean checkRange(double humidity, double temperature, double pressure, double radiation){
    if(humidity<0.0 || humidity >100.0 || temperature<0.0 || pressure<0.0 || radiation<0 ){
        return false;
    }
    return true;
}

/**
 * Creates the main star (root) of the planetary system.
 *
 * @param name        name of the star
 * @param temperature temperature in Kelvin
 * @param pressure    pressure in Pascals
 * @param humidity    humidity (must be 0)
 * @param radiation   radiation level in Sieverts
 * @return true if creation was successful, false otherwise
 */
public boolean createPlanetSystem(String name, double temperature,double pressure, double humidity, double radiation){
    if (humidity != 0) {
        System.err.println("Humidity must be 0!");
        return false ;
    }
    else if(!checkRange(humidity,temperature,pressure,radiation)){
        System.err.println("Data of Planet System out of range.Can not created.");
        return false;
    }
    else{
        star = new Node(name,type:"Star",temperature,pressure,humidity,radiation);
    }
    return true;
}
```

addPlanet(String planetName, String parentName, double temperature, double pressure, double humidity, double radiation)

- **Adds a planet under the given parent node**
- **Only one planet can be directly added under a parent node (equidistance check).**
- **If a planet with the same name exists, or the parent is a moon, the operation is denied.**
- **Validates all sensor values.**

```
public void addPlanet(String planetName , String parentName, double temperature,double pressure, double humidity, double radiation ){
    Node parent = findNode(star, parentName);
    if(parent == null){
        System.err.println("Parent node not found");
    }
    else if(!checkRange(humidity,temperature,pressure,radiation)){
        System.err.println("Data out of range.");
    }
    else if(parent.getType().equals("Moon")){
        System.err.println( "This is a satellite. A planet cannot be added to this satellite.");
    }
    else if(findNode(star, planetName)!=null && findNode(star, planetName).getType().equals("Planet")){
        System.err.println("This planet already exists.");
    }
    else if(checkPlanet(parent)){
        System.err.println("The planets cannot be equidistant from the star.");
    }
    else{
        System.out.println("Planet added.");
        parent.addChild(new Node(planetName,type:"Planet",temperature,pressure,humidity,radiation));
    }
}
```

addSatellite(String satelliteName, String parentName, double temperature, double pressure, double humidity, double radiation)

- **Adds a moon/satellite to a specified planet.**
- **Fails if the parent is not a planet or the satellite already exists.**
- **Sensor data is validated before insertion.**

```
public void addSatellite(String satelliteName , String parentName, double temperature,double pressure, double humidity, double radiation ){
    Node parent = findNode(star, parentName);
    if(parent == null){
        System.err.println("Parent node not found");
    }
    else if(!checkRange(humidity,temperature,pressure,radiation)){
        System.err.println("Data out of range.");
    }
    else if(!parent.getType().equals("Planet")){
        System.err.println( "Parent type must be planet.");
    }
    else if(findNode(star,satelliteName)!=null && findNode(star,satelliteName).getType().equals("Moon")){
        System.err.println("This satellite already exists.");
    }
    else{
        System.out.println("Satellite added.");

        parent.addChild(new Node(satelliteName,type:"Moon",temperature,pressure,humidity,radiation));
    }
```

findRadiationAnomalies(double threshold)

- **Searches the entire system recursively for nodes with radiation levels exceeding the given threshold.**
- **Returns a List<Node> of matching anomalies.**
- **Delegates to findRadiationAnomaliesR() for recursive traversal.**

```java
public List<Node> findRadiationAnomalies(double threshold){
    List<Node> anomalies = new ArrayList<>();
    findRadiationAnomaliesR(star, threshold , anomalies);
    return anomalies;
}

/**
 * Recursive helper to find radiation anomalies.
 *
 * @param current   the current node
 * @param threshold radiation threshold
 * @param anomalies list to store found anomalies
 */
private void findRadiationAnomaliesR(Node current, double threshold, List<Node> anamolies){
    if(current.getSensorData().getRadiation()>threshold){
        anamolies.add(current);
    }
    for(Node temp : current.getChildren()){
        findRadiationAnomaliesR(temp, threshold, anamolies);
    }
}
```

getPathTo(String name)

- **Finds and returns the path from the star to a target node using a Stack<String>.**
- **Represents the path from root to the desired node.**
- **Uses the recursive findPath() method.**

```java
public Stack<String> getPathTo(String name){
    Stack<String> path = new Stack<>();
    if(findPath(star, name, path)){
        return path;
    }
    return null;
}

/**
 * Recursive helper to find the path to a specific node.
 *
 * @param current current node
 * @param name    target node name
 * @param path    stack used to build the path
 * @return true if the node was found, false otherwise
 */
private boolean findPath(Node current, String name, Stack<String> path){
    if(current.getName().equals(name)){
        path.push(current.getName());
        return true;
    }
    for(Node child : current.getChildren()){
        if(findPath(child, name, path)){
            path.push(current.getName());
            return true;
        }
    }
    return false;
}
```

printMissionReport()

- **Prints a report of all nodes in the system in a recursive, tree-like fashion.**
- **Outputs include each node's name, type, and sensor data (e.g., "300 Kelvin, 101 Pascals").**

```java
public void printMissionReport(){
    printMissionReportR(star);
}

/**
 * Prints the mission report for a specific node by name.
 *
 * @param name the name of the node to report
 */
public void printMissionReport(String name){
    Node result = findNode(star, name);
    if(result!=null){
        System.out.println(result);
    }
    else{
        System.err.println("Node not found.");
    }
}
```

printMissionReport(String name)

- **Prints information only for a specific node.**
- **If the node does not exist, an error message is printed**

```java
/**
 * Prints the mission report for a specific node by name.
 *
 * @param name the name of the node to report
 */
public void printMissionReport(String name){
    Node result = findNode(star, name);
    if(result!=null){
        System.out.println(result);
    }
    else{
        System.err.println("Node not found.");
    }
}
```

Helper functions :

findNode(Node current, String name)

- **Recursively traverses the system tree to find a node by its name.**
- **Implements depth-first search (DFS).**

```
private Node findNode (Node current ,String name){
    if(current.getName().equals(name)){
        return current;
    }
    for(Node child : current.getChildren()){
       Node result = findNode(child, name);
       if(result!=null){
        return result;
       }
    }
    return null;
}

/**
```

checkPlanet(Node node)

- **Checks whether a given node already has a planet as a direct child.**
- **Enforces the rule that planets must not be equidistant under the same parent.**

```
private boolean checkPlanet(Node node){
    boolean flag = false;
    for(Node child : node.getChildren()){
        if(child.getType().equals("Planet")){
            flag = true;
        }
    }
    return flag;
}
```

INPUT1 Example :

input.txt

```
create planetSystem Sun 20 10 21 3
create planetSystem Sun 20 10 0 3
addSatellite das Sun 21 3 4 32
addPlanet Mercury  Sun 21 3 0 5
addPlanet Venus Mercury 21 3 4 9
addPlanet Earth Venus 21 3 4 2
addPlanet Mars Earth 21 3 4 231
addSatellite Moon Earth 21 3 4 32
addSatellite Phobos Mars 21 3 4 5
addSatellite Deimos Mars 22.7 3 4 7
addSatellite Deimos Mars 21 3 4 32
addSatellite  yusuf Deimos 21 3 4 32
addPlanet Jupiter  Mars 21 3 4 11
addPlanet yusuf Mercury 21 3 4 22
getPathTo Mars
getPathTo Deimos
printMissionReport
printMissionReport Deimos
exit

```

OUTPUT1 Example:

```
Humidity must be 0!
------------------------------------------------
Planet system created.
------------------------------------------------
Parent type must be planet.
------------------------------------------------
Planet added.
------------------------------------------------
Planet added.
------------------------------------------------
Planet added.
------------------------------------------------
Planet added.
------------------------------------------------
Satellite added.
------------------------------------------------
Satellite added.
------------------------------------------------
Satellite added.
------------------------------------------------
This satellite already exists.
------------------------------------------------
Parent type must be planet.
------------------------------------------------
Planet added.
------------------------------------------------
The planets cannot be equidistant from the star.
------------------------------------------------
Sun -> Mercury -> Venus -> Earth -> Mars
------------------------------------------------
Sun -> Mercury -> Venus -> Earth -> Mars -> Deimos
------------------------------------------------
Sun (Star):  20.00 Kelvin,  10.00 Pascals,  %0.00,  3.00 Sieverts.
Mercury (Planet):  21.00 Kelvin,  3.00 Pascals,  %0.00,  5.00 Sieverts.
Venus (Planet):  21.00 Kelvin,  3.00 Pascals,  %4.00,  9.00 Sieverts.
Earth (Planet):  21.00 Kelvin,  3.00 Pascals,  %4.00,  2.00 Sieverts.
Mars (Planet):  21.00 Kelvin,  3.00 Pascals,  %4.00,  231.00 Sieverts.
Phobos (Moon):  21.00 Kelvin,  3.00 Pascals,  %4.00,  5.00 Sieverts.
Deimos (Moon):  22.70 Kelvin,  3.00 Pascals,  %4.00,  7.00 Sieverts.
Jupiter (Planet):  21.00 Kelvin,  3.00 Pascals,  %4.00,  11.00 Sieverts.
Moon (Moon):  21.00 Kelvin,  3.00 Pascals,  %4.00,  32.00 Sieverts.
------------------------------------------------
Deimos (Moon):  22.70 Kelvin,  3.00 Pascals,  %4.00,  7.00 Sieverts.
------------------------------------------------
Exiting...
```

INPUT2 Example:

input.txt

```
create planetSystem Solaris 5800 100000 12 5
create planetSystem Solaris 5800 100000 0 5
addPlanet Terra Solaris 288 101325 30 3
addPlanet Vulcan Terra 320 95000 20 7
addPlanet Gaia Vulcan 275 100000 25 9
addPlanet Erebos Gaia 260 97000 28 4
addSatellite Luna Terra 250 80000 15 3
addSatellite Io Vulcan 240 90000 19 8
addSatellite Titan Gaia 230 85000 22 12
addSatellite Echo Erebos 210 70000 10 2
addSatellite Luna Terra 250 80000 15 3
addSatellite Echo Erebos 210 70000 10 2
addPlanet Erebos Gaia 260 97000 28 4
addSatellite Sol Solaris 200 60000 5 1
addPlanet Kronos Echo 300 100000 50 5
addSatellite Mist Kronos 220 80000 10 3
addPlanet Mercury Solaris -100 100000 10 5
addSatellite Ghost Vulcan 240 -1 19 8
addSatellite Specter Vulcan 240 80000 105 8
addSatellite Null Vulcan 240 80000 19 -2
findRadiationAnomalies -10
findRadiationAnomalies 7
getPathTo Titan
getPathTo Mist
getPathTo Atlantis
printMissionReport
printMissionReport Io
printMissionReport Pluto
exit

```

OUTPUT2 Example:

```
Humidity must be 0!
-------------------------------------------------
Planet system created.
-------------------------------------------------
Planet added.
-------------------------------------------------
Planet added.
-------------------------------------------------
Planet added.
-------------------------------------------------
Planet added.
-------------------------------------------------
Satellite added.
-------------------------------------------------
Satellite added.
-------------------------------------------------
Satellite added.
-------------------------------------------------
Satellite added.
-------------------------------------------------
This satellite already exists.
-------------------------------------------------
This satellite already exists.
-------------------------------------------------
This planet already exists.
-------------------------------------------------
Parent type must be planet.
-------------------------------------------------
This is a satellite. A planet cannot be added to this satellite.
-------------------------------------------------
Parent node not found
-------------------------------------------------
Data out of range.
-------------------------------------------------
Data out of range.
-------------------------------------------------
Data out of range.
-------------------------------------------------
Data out of range.
-------------------------------------------------
threshold must be greater than 0.
-------------------------------------------------
Gaia (Planet):  275.00 Kelvin,  100000.00 Pascals,  25.00%,  9.00 Sieverts.
Titan (Moon):  230.00 Kelvin,  85000.00 Pascals,  22.00%,  12.00 Sieverts.
Io (Moon):  240.00 Kelvin,  90000.00 Pascals,  19.00%,  8.00 Sieverts.
-------------------------------------------------
Solaris -> Terra -> Vulcan -> Gaia -> Titan
-------------------------------------------------
Mist is not found.
-------------------------------------------------
Atlantis is not found.
-------------------------------------------------
Solaris (Star):  5800.00 Kelvin,  100000.00 Pascals,  0.00%,  5.00 Sieverts.
Terra (Planet):  288.00 Kelvin,  101325.00 Pascals,  30.00%,  3.00 Sieverts.
Vulcan (Planet):  320.00 Kelvin,  95000.00 Pascals,  20.00%,  7.00 Sieverts.
Gaia (Planet):  275.00 Kelvin,  100000.00 Pascals,  25.00%,  9.00 Sieverts.
Erebos (Planet):  260.00 Kelvin,  97000.00 Pascals,  28.00%,  4.00 Sieverts.
Echo (Moon):  210.00 Kelvin,  70000.00 Pascals,  10.00%,  2.00 Sieverts.
Titan (Moon):  230.00 Kelvin,  85000.00 Pascals,  22.00%,  12.00 Sieverts.
Io (Moon):  240.00 Kelvin,  90000.00 Pascals,  19.00%,  8.00 Sieverts.
Luna (Moon):  250.00 Kelvin,  80000.00 Pascals,  15.00%,  3.00 Sieverts.
-------------------------------------------------
Io (Moon):  240.00 Kelvin,  90000.00 Pascals,  19.00%,  8.00 Sieverts.
-------------------------------------------------
Node not found.
-------------------------------------------------
Exiting...
```