TRAN QUOC TUAN - UG2F20 MAY 2024

2ND ASSIGNMENT GROUP 10 (BORSI)

TASK:

7. Different kinds of plants live on a planet. If the nutrient of a plant runs out (its nutrient level becomes zero), the

plant wastes away. There are three kinds of radiation on the planet: alpha, delta, no radiation. The different species

of plants  react to radiation differently. The  reaction involves  a  change in the  nutrient level of the  plant and the

radiation the next day. The radiation of the next day will be alpha radiation if the sum of the demand for alpha

radiation over all plants is greater than the sum of the demand for delta radiation by at least three. If the demand

for delta radiation is greater by at least three than the demand for alpha radiation, the radiation will be delta. If the

difference is less than three, there will be no radiation. There is no radiation the first day.

Each plant has a name (string), a nutrient level (int), and a boolean that denotes whether it's alive. The plant species

are wombleroot, wittentoot and woreroot. The different plant species react to the different radiations as follows.

The level of nutrients changes first. After that, the plant can influence the radiation of the next day if it's still alive.

Wombleroot: Alpha radiation makes the nutrient level increase by 2, no radiation makes it decrease by 1, and

delta  radiation makes it decrease by 2. It demands  alpha  radiation by a  strength of 10 regardless  of the current

radiation. This plant also wastes away if its nutrient level increases above 10.

Wittentoot: Alpha radiation makes the nutrient level decrease by 3, no radiation makes it decrease by 1, delta

radiation makes it increase by 4. This plant demands delta radiation with strength 4 if its nutrient level is less than

5, with strength 1 if its nutrient level is between 5 and 10, and doesn't influence the radiation if its nutrient level is

greater than 10.

Woreroot: Its nutrient level increases by 1 if there is alpha or delta radiation, and decreases by 1 if there is no

radiation. Doesn't influence the radiation of the next day.

Simulate the ecosystem of plants and give the name of the strongest plant which is still alive after n days. Print

all the data of the plants and the level of radiation on each day.

The program should read the data of the simulation from a text file. The first line contains the number of plants.

Each of the next lines contains the data of one plant: its name, its species, and its starting nutrient level. The species

can be: wom - wombleroot, wit - wittentoot, wor - woreroot. The last line of the file contains n, the number of days

as an int. The program should ask for the filename and display the contents of the file. You can assume that the

input file is correct. A possible input file:

4

Hungry wom 7

Lanky wit 5

Big wor 4

Tall wit 3

10

Analysis:  
Plant species:

Wombleroot

* Alpha Radiation: Increases nutrient by 2.
* Delta Radiation: Decreases nutrient by 2.
* No Radiation: Decreases nutrient by 1.
* Always demands Alpha radiation with a strength of 10.

Wittentoot

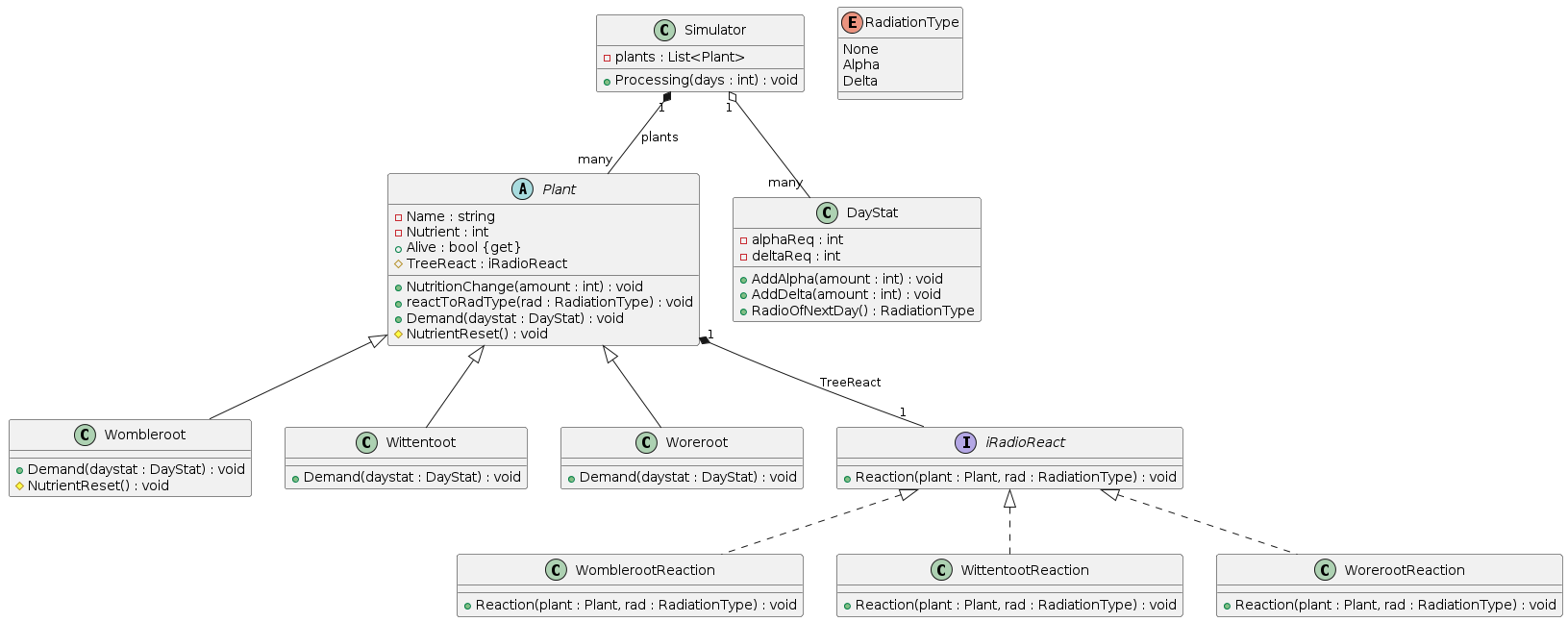
* Alpha Radiation: Decreases nutrient by 3.
* Delta Radiation: Increases nutrient by 4.
* No Radiation: Decreases nutrient by 1.
* Demands Delta radiation based on its nutrient level.

Woreroot

* Alpha/Delta Radiation: Increases nutrient by 1.
* No Radiation: Decreases nutrient by 1.
* Does not influence the radiation of the next day.

Radiation Types

* Alpha Radiation: Selected if the sum of the demand for Alpha radiation over all plants exceeds the sum of the demand for Delta radiation by at least three.
* Delta Radiation: Selected if the sum of the demand for Delta radiation exceeds the sum of the demand for Alpha radiation by at least three.
* No Radiation: Selected if the difference in demands between Alpha and Delta radiation is less than three.

****

In the plant and radiation code, the abstract **Plant** class encapsulates essential attributes like name and nutrient level, and uses the **iRadioReact** interface to delegate radiation response strategies to different plant types—Wombleroot, Wittentoot, and Woreroot. This implementation uses the design pattern allowing each plant species to adapt dynamically to environmental changes without modifying the base class:

* Wombleroot: consistently demands Alpha radiation, influencing daily radiation outcomes significantly.
* Wittentoot: varies its demand for Delta radiation based on its nutrient level, adapting its strategy to optimize survival.
* Woreroot: remains neutral, providing a stable response in varying radiation conditions.

The Simulator class manages the ecosystem's daily operations, using the DayStat class to calculate the dominant radiation type based on the plants' cumulative demands. This design adheres to the open/closed principle, facilitating maintainability and easy integration of new plant types or behaviors.

Reaction:

* Plant x RadiationType→ Plant x RadiationType: Give the change of plant nutrition based on raditation type.
* A = (RadiationType: Radiation, plants: plant)
* Pre = Radiation = RadiationType ∧ plants = Plant
* Post = (Radiation=RadiationType ∧
* ∀i∈[1..n]:plants[i],Radiation[i]= reactToRadType(plants0[i],RadiationType[i−1])

alive=⨁𝑖=1..𝑛 < plants[𝑖]. 𝑛𝑎𝑚𝑒 >

{plants[i].Alive()}

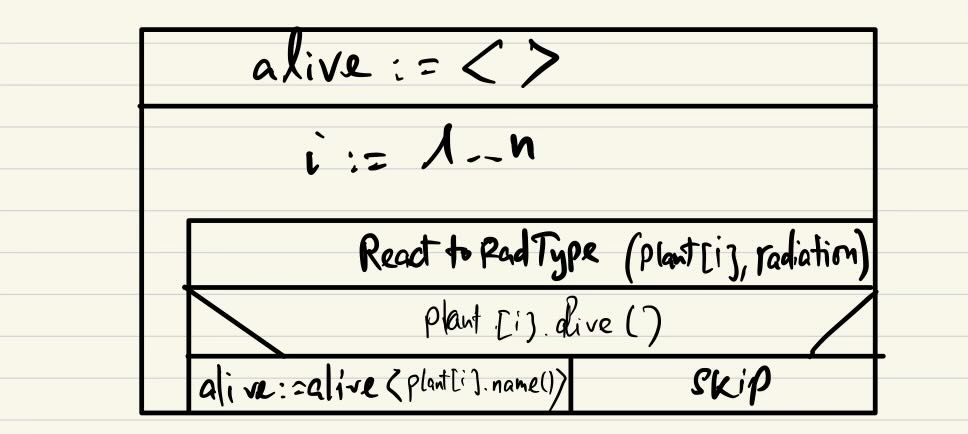
Analogy:

|  |  |
| --- | --- |
| enor(E) | i = 1 .. n |
| f(e) | reactToRadType(Plants[i], radiation) |
| s | plants |
| H, +, 0 | Plant\* , ⊜, plants[i] |

|  |  |
| --- | --- |
| enor(E) | i = 1 .. n |
| f(e) | Demand(plants[i], now) |
| s | daystat |
| H, +, 0 | RadiationType\* , ⊜, daystat |

|  |  |
| --- | --- |
| enor(E) | i = 1 .. n |
| f(e) | if plants[i].Alive() |
| s | Alive |
| H, +, 0 | Plant\*, ⊕, <> |

Merging into a loop:



Each plant in the program will go through n+1 process in n days of exposure to types of radiation, which changing within the simulation day after day. The 0th state of plants[i] is the plant's initial state (plants[i=0]), and the nth state is the plant after interacting with radiation across n days (plants[i=n]).

The state of the ith plant before reacting to the jth day's radiation is represented by plants(𝑗−1)[𝑖], from which the jth state is shaped by the method Reaction()(plants𝑗[𝑖]), alongside the environmental conditions of the day, Daystat𝑗.

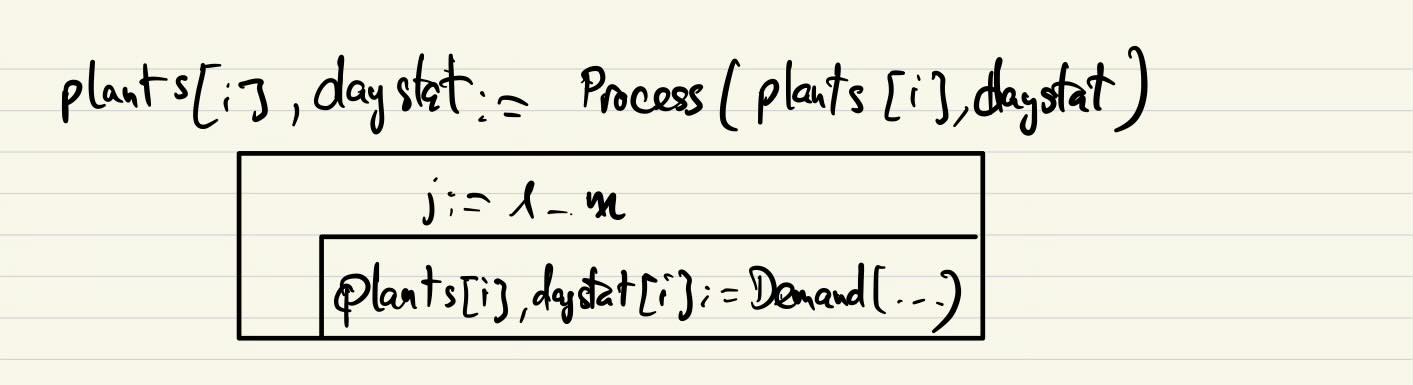
The task after to solvie will be:

∀j∈[1..m]: plantsj[i], daystat[j] = Demand(plantsj-1[i], daystati-1[j]) ∧ plants[i] = plantsm[i])

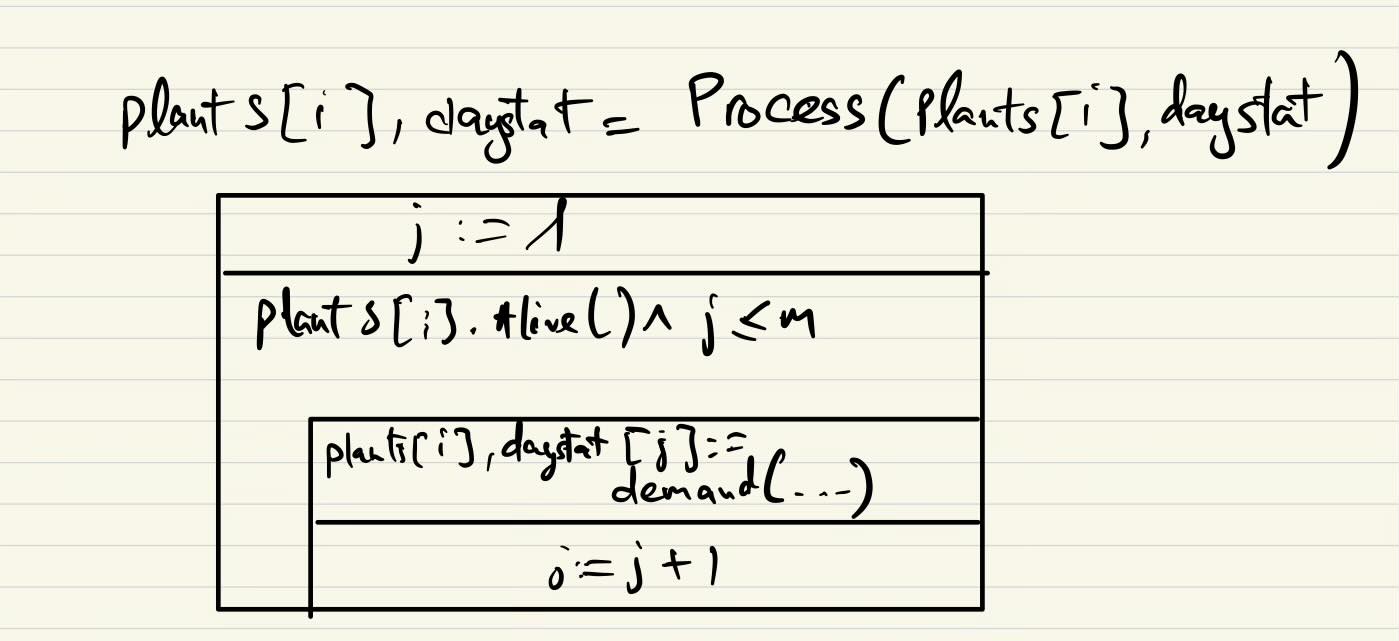
The interaction of a plant with radiation represents a continuous change in the plant's condition in the simulation, influencing both its internal nutrient value and its external influence on the radiation value.

|  |  |
| --- | --- |
| enor(E) | j = 1 .. n |
| f(e) | Demand(plants[i], daystat[j])1 |
| s | plants [i] |
| H, +, 0 | Plant\*, ⊜, plants[i] |

|  |  |
| --- | --- |
| enor(E) | j = 1 .. m |
| f(e) | Demand (plants [i], daystat [j])2 |
| s | daystat [j] |
| H, +, 0 | RadiationType\*, ⊕, <> |

Merging to a loop:  


If it is determined that there are no changes to either the plant's condition or the radiation environment following a plant's death, the effectiveness of the previous algorithm could be improved.



**Testing:**

Grey box test cases:

**Outer Loop (Simulation Over Days)**

Length-based:

* Zero Plants: Tests the simulator with no plants to ensure no processing errors or exceptions occur, and no plants are mistakenly added or processed.
* Single Plant: Checks if a single plant can survive through multiple days of simulation, verifying plant death mechanics when nutrient levels potentially fall below zero due to continuous radiation exposure.

First and Last:

* Initial Plant State: Verifies that the nutrient level of a plant remains unchanged when no days of simulation are processed.
* Plant After One Day: Ensures that nutrient levels change after a day of simulation, reflecting the impact of radiation.

**Inner Loop (Reaction to Radiation Types)**

Length-based:

* Zero Day Simulation: Confirms that no changes happen to a plant's nutrient level when the simulation spans zero days, ensuring that initial conditions are preserved correctly.
* One Day Simulation: Tests the change in nutrient levels after one day, indicating whether radiation effects are correctly applied.

First and Last:

* Initial Response to Radiation: Evaluates how a plant reacts immediately to each type of radiation (Alpha, Delta, None), ensuring that the reaction logic for each radiation type is correctly implemented.
* Last Known Condition: Tests how plants behave under extreme conditions (e.g., starting with the maximum integer value for nutrients) to check for any overflow errors or incorrect nutrient calculations.

Examination of Specific Functions and Conditions

* Test Plant Responses to Radiation Types: Iteratively tests a single plant's response to each type of radiation, checking the immediate effects of each radiation type on the plant's nutrient levels.
* Test Extreme Conditions: Assesses how the simulator handles a plant starting with the maximum possible nutrient value, verifying that the system properly manages large numbers and does not cause integer overflow or other errors.