Saifeddin Alkurdi
o9ffdc
2nd OOP assignment

ASSIGNMENT 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.

How does the program generally work

1. The program initiates by reading input data from a file, which includes the number of plants along with their respective properties such as name, type, and nutrient level. It performs validation on the input data to ascertain its correctness.
2. For each plant, the program creates objects based on its type (Wombleroot, Wittentoot, or Woreroot) using the provided data.
3. The program simulates daily radiation over a span of 10 days, with each day featuring a distinct radiation level as specified in the input file. On each day, the program iterates over all the plants, applying the radiation simulation based on their type using the designated radiation form.
4. The Radiation Form calculates the day's radiation by totaling the radiation demand for each type of plant. It then determines the radiation type for the following day by evaluating whether the difference between alpha and delta radiation, or vice versa, exceeds three. A plant is included in this calculation if it is either a Wombleroot or Wittentoot and is still alive.
5. Each day, the Radiation Form updates the plants' nutrient levels according to the type of radiation applied. If a plant's nutrient level reaches zero or, in some cases with Wombleroot, exceeds ten, the plant is deemed to have died.
6. After processing each day, the program displays updated information for each plant, including its type and nutrient level.
7. Following 10 rounds of simulation, the program identifies the plant with the maximum nutrient level among those still alive and outputs its name.
8. The program then compiles a simulation summary, detailing the type of radiation for each day and the corresponding nutrient levels of the plants and most importantly the winner and survivor.

Overall, the program uses the factory method pattern to initiate objects based on the plant type , updates their nutrient level accordingly, and provides a summary of the simulation results.

Now in the following tables it will show how different radiation types affect each plant differently

FOR Wombleroot:

Radiation	Nutrient level effect
Alpha	+2
Delta	-2
No Radiation	-1

FOR Wittletoot:

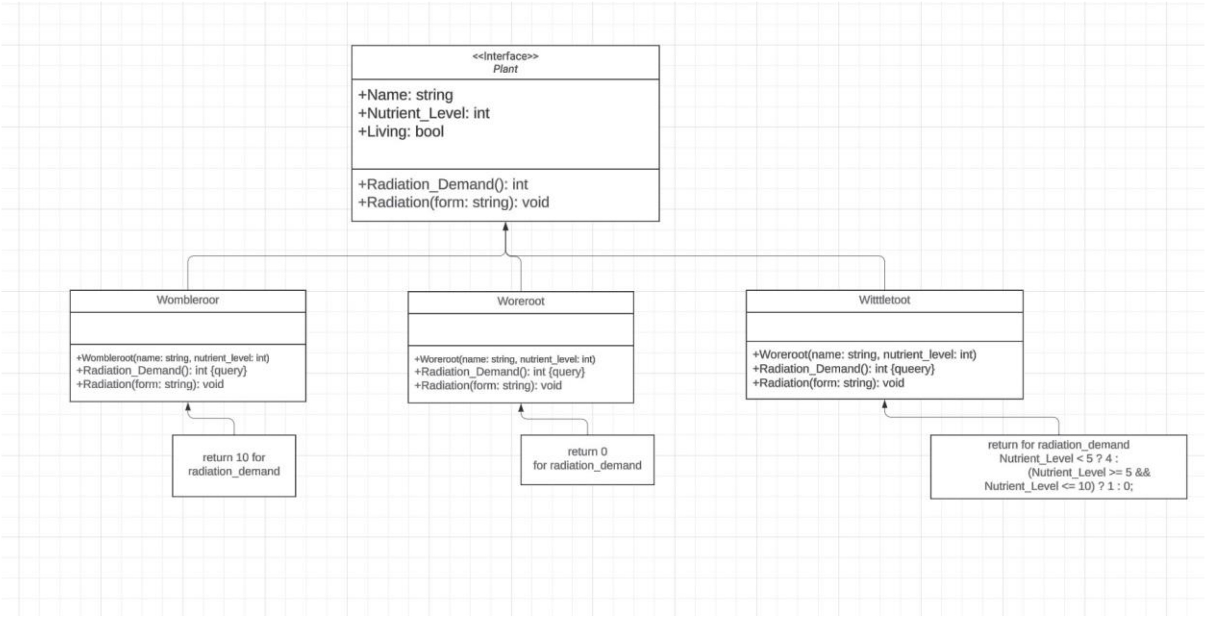
Radiation	Nutrient Level effect
Alpha	-3
Delta	+4
No Radiation	-1

FOR Woreroot

Radiation	Nutrient Level effect
Alpha	+1
Delta	+1
No Radiation	-1

Note if that nutrient level reaches 0 for all and above 10 for Wobbleroot its dead and they can not determine next day's radiation based on their demand.

UML diagram of the two classes that i have made



<<Interface>>
Plant
+Name: string
+Nutrient_Level: int
+Living: bool
+Radiation_Demand(): int
+Radiation(form: string): void
Wombleroor
+Wombleroot(name: string, nutrient_level: int)
+Radiation_Demand(): int {query}
+Radiation(form: string): void
return 10 for radiation_demand
Woreroot
+Woreroot(name: string, nutrient_level: int)
+Radiation_Demand(): int {query}
+Radiation(form: string): void
return 0 for radiation_demand
Witttletoot
+Woreroot(name: string, nutrient_level: int)
+Radiation_Demand(): int {queery}
+Radiation(form: string): void
return for radiation_demand Nutrient_Level < 5 ? 4 : (Nutrient_Level >= 5 && Nutrient_Level <= 10) ? 1 : 0;

Pseudocode :

```
Abstract Class Plant
  Properties:
    Name: String
    Nutrient_Level: Integer
    Living: Boolean

  Methods:
    Radiation_Demand(): Integer
    Radiation(form: String): Void
--------------------------------------------
Class Wombleroot Inherits Plant
  Constructor(name: String, nutrient_level: Integer)
    Set Name to name
    Set Nutrient_Level to nutrient_level
    Set Living to true
  Radiation_Demand()
    Return 10

  Radiation(form: String)
    If form is "alpha"
      Increase Nutrient_Level by 2
    Else If form is "no_radiation"
      Decrease Nutrient_Level by 1
    Else If form is "delta"
      Decrease Nutrient_Level by 2

    If Nutrient_Level > 10 or Nutrient_Level <= 0
      Set Living to false
      Set Nutrient_Level to 0
----------------------------------------
Class Woreroot Inherits Plant
  Constructor(name: String, nutrient_level: Integer)
    Set Name to name
    Set Nutrient_Level to nutrient_level
    Set Living to true

  Radiation_Demand()
    Return 0
  Radiation(form: String)
    If form is "alpha" or form is "delta"
      Increase Nutrient_Level by 1
    Else If form is "no_radiation"
      Decrease Nutrient_Level by 1

    If Nutrient_Level <= 0
      Set Living to false
      Set Nutrient_Level to 0
---------------------------------------------
Class Wittentoot Inherits Plant
  Constructor(name: String, nutrient_level: Integer)
    Set Name to name
    Set Nutrient_Level to nutrient_level
    Set Living to true

  Radiation_Demand()
    If Nutrient_Level < 5
      Return 4
    Else If Nutrient_Level >= 5 and Nutrient_Level <= 10
      Return 1
    Else
      Return 0

  Radiation(form: String)
```

```
    If form is "alpha"
        Decrease Nutrient_Level by 3
    Else If form is "no_radiation"
        Decrease Nutrient_Level by 1
    Else If form is "delta"
        Increase Nutrient_Level by 4

    If Nutrient_Level <= 0
        Set Living to false
        Set Nutrient_Level to 0
```

Testcases :

White cases:

Type of plant must be from the three types mentioned

Black cases:

If the input file is empty