Saipidin kyzy Upol **2nd assignment/8. Task** 22 May 2022
NEPTUNCODE
ihlg15@inf.elte.hu
Group 7

Task
8. 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 until there is no radiation on two consecutive 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 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:

Analysis[1]
Independent objects in the task are the plants. They can be divided into 3 different groups:Wombleroot, Wittentoot and Woreroot.
All of them have a name and a nutrient level that can be got.

Reaction to radiation:

Womberoot:

Radiation	Nutrient change
Alpha	+2
Delta	-2
No radiation	-1

Case: It demands alpha radiation of strength 10 without any conditions.
Dies also: If nutrient > 10

Wittentoot:

Radiation	Nutrient change
Alpha	-3
Delta	+4
No radiation	-1

Case 1: Demands delta radiation of strength 4 if nutrient < 5
Case 2: Demands delta radiation of strength 1 if 5<= nutrient < 10
Case 3: If nutrient > 10 does not influence radiation

Woreroot:

Radiation	Nutrient change
Alpha	+1
Delta	+1
No radiation	-1

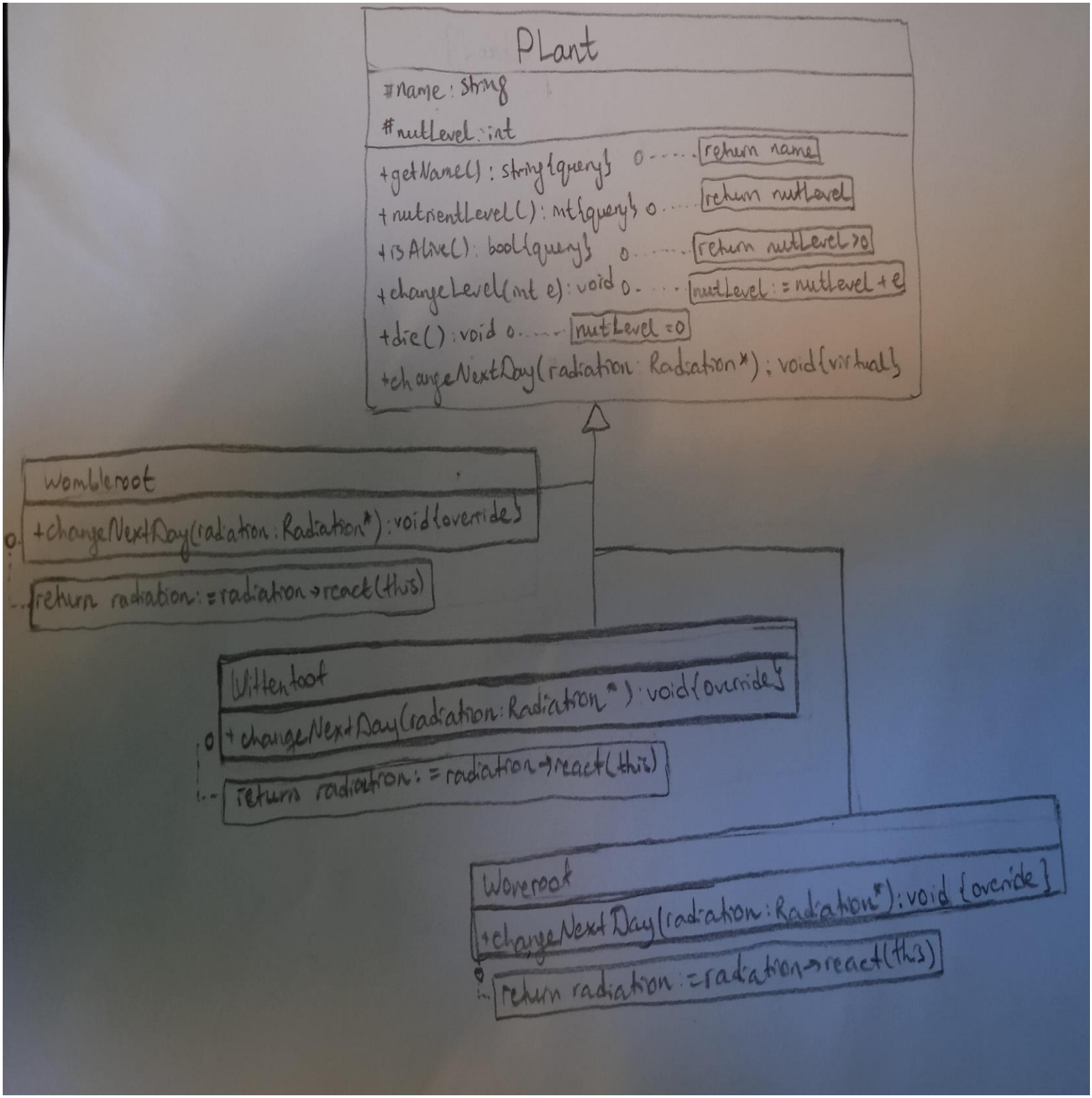
Does not influence the radiation of the next day at all.

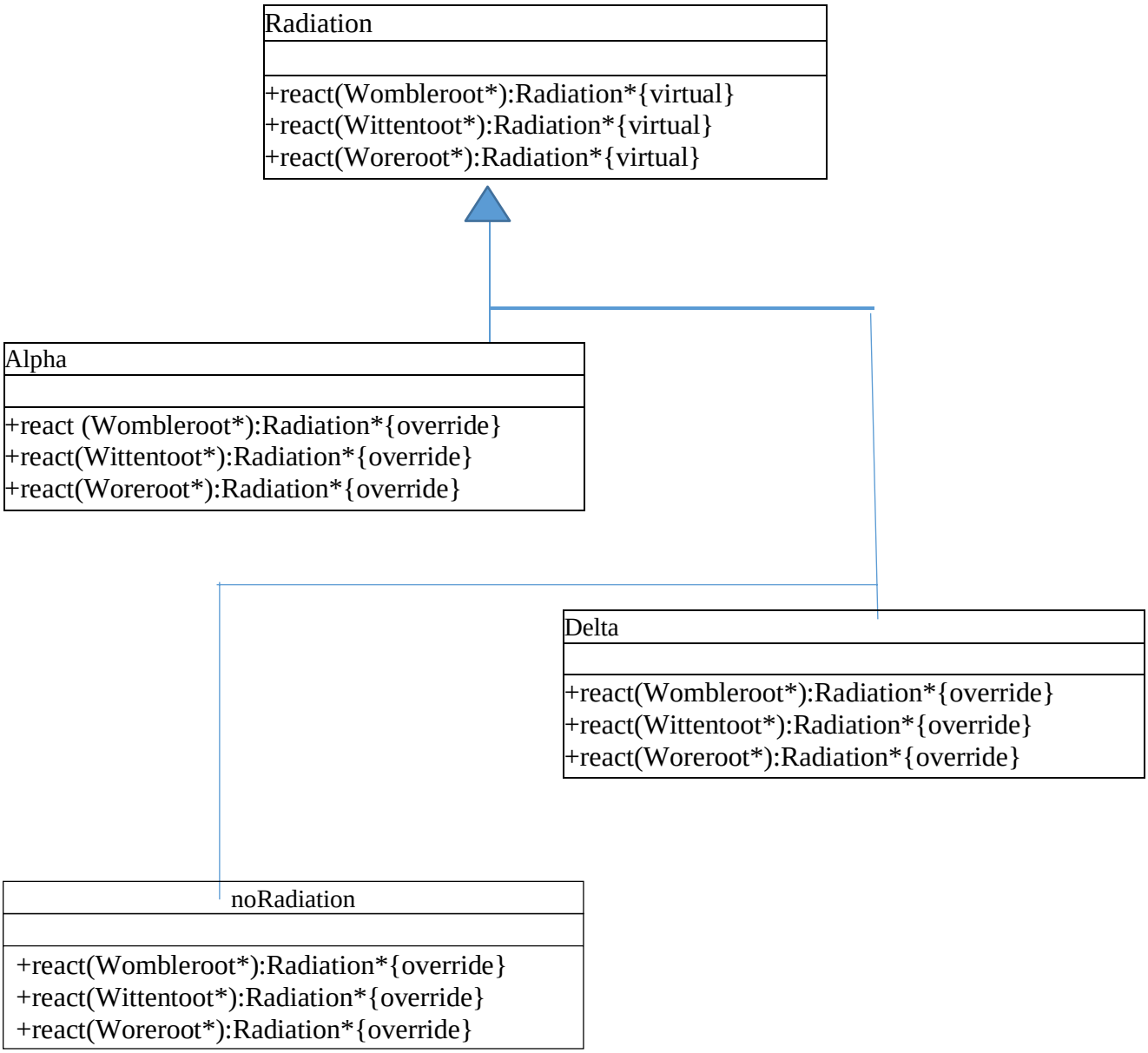
Next Day Radiation:

Case 1: if sum of the demand for alpha radiation is greater by at least 3 than the sum of the demand for delta then the next day is going to have alpha radiation.

Case 2: if sum of the demand for delta radiation is greater by at least 3 than the sum of the demand foralpha then the next day is going to have delta radiation.

Case 3: if sum of the demand for alpha radiation – sum of the demand for delta radiation < 3 then the next day is going to have no radiation.


Plant
#name : string
#nutLevel : int
+getName() : string{query} o---- return name
+nutrientLevel() : nt{query} o---- return nutLevel
+isAlive() : bool{query} o---- return nutLevel>0
+changeLevel(int e) : void o---- nutLevel := nutLevel + e
+die() : void o---- nutLevel = 0
+changeNextDay(radiation: Radiation*) : void{virtual}
Womberoot
+changeNextDay(radiation: Radiation*) : void{override}
return radiation := radiation->react(this)
Vittentoot
+changeNextDay(radiation: Radiation*) : void{override}
return radiation := radiation->react(this)
Woveroot
+changeNextDay(radiation: Radiation*) : void {override}
return radiation := radiation->react(this)



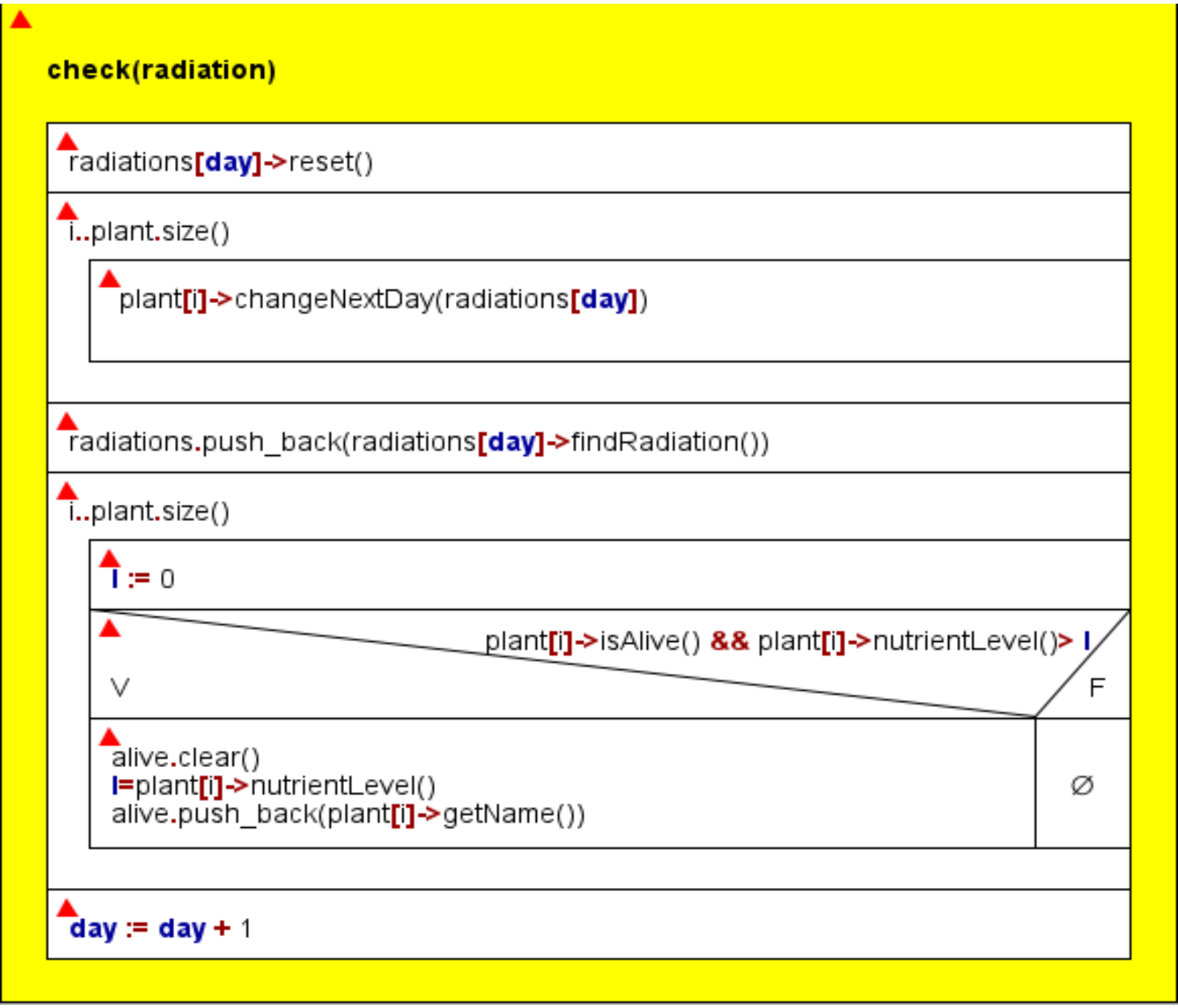
All the classes of the Radiation are realized based on the Singleton design pattern, as it is enough to create one object for each class

In the Specification we show how the nutrient level depends on the radiation and how the radiation of the next day depends on the nutrient level of the current day.

Specification for the stimulation of the program

A = (radiations: Radiation^m , plants: Plant^n , alive: String*)
Pre = (plants = plants^0 ∧ radiations = radiations$^{0)}$
Post = (radiations = radiations^0 ∧ (plants, radiations, alive) = SEARCH(∀i∈[1..n]: plants[i], radiations$_i$ = react(plants^0[i], radiations[i]) ∧ radiations[i] == noRadiation::instance() ∧ radiations[i+1] == n oRadiation::instance() ^ radiations[i]==radiations[i+1]) ∧ alive = $\oplus i$=1..n < plants[i]. $name$)

Algorithm for the Linear Search function:



Testing

Grey box test cases:

1) Test cases for empty file
   No element;

2) Test cases for inp.txt (more elements);
   a. First element
   b. Middle element
   c. Last element

3) Test cases for inp1.txt (more elements);
   a. First element
   b. Middle element
   c. Last element

4) Test cases for methods of plants;
   a. isAlive()
   b. getName()

5) Test cases for attributes of radiation class;
   a. getAlphaDemand()
   b. getDeltaDemand()

6) Test cases for instances/types of radiation class;
   a. Alpha::instance()
   b. Delta::instance()
   c. noRadiation::instance()

7) Examination of function react()
   Different cases depending on the plant and the radiation