Assignment 3

Name : Rizwan Hussain **Neptun:** ULLBQG

Task

In the hydrological cycle of the Earth, various areas affect the weather as well as areas are also affected by various weathers. Areas involved in the simulation: plain, grassland, lakes region. Each area has a name, and the amount of water stored in the certain area is also given in km3. The humidity of the air over the areas is also given in percentage.
The possible types of weather are the following**: sunny**, **cloudy, rainy**, depending on the humidity of the air. In case the humidity exceeds 70%, the weather gets rainy and the humidity decreases to 30%.
In case the humidity is between 40-70%, the calculation of the chance of rainy weather is: (humidity-30)*3,3%, otherwise the weather is cloudy. Humidity below 40% leads to sunny weather.
In the following, we declare how the certain areas respond to the different type of weathers. First the amount of water stored by the area varies then the weather will be affected. There is no type of areas with negative amount of water stored.
In case the type is *plain*, if the weather is sunny, the amount of water will be decreased by 3 km3; if cloudy, it will be decreased by 1 km3; for rainy weather it will be increased by 20 km3. The humidity of the air is increased by 5%. If the amount of the stored water is greater than 15 km3, the plain area changes into grassland.
In case of type *grassland*: in sunny weather, the amount of water is decreased by 6 km3, for cloudy it will be decreased by 2 km3, but and for rainy, it will be increased by 15 km3. The humidity of the air is increased by 10%. The area becomes lakes region obtaining amount of water over 50 km3, whereas in case the amount of stored water goes below 16 km3, the area changes to plain.
In case of type *lakes region*: in sunny weather, the amount of water is decreased by 10 km3, for cloudy it will be decreased by 3 km3, for rainy it will be increased by 20 km3. The humidity will be increased by 15%. Beyond an amount of water of 51 km3the area changes into grassland.
The program reads data from a text file. The first line of the file contains a single integer N indicating the number of areas. Each of the following N lines contains the attributes of an area separated by spaces: the owner of the area, the type of the area, and the amount of water stored by the area. In the last line, the humidity of the air is given in percentage. The type is identified by a character: P –plain, G –grassland, L –lakes region.
After 10 simulation rounds, determine the owner of the area which is storing the greatest amount of water. The amount of water is also required to be determined. The program should print all attributes of the certain areas by simulation rounds!
The program should ask for a filename, then print the content of the input file. You can assume that the input file is correct. Sample input:

```
4
Mr Bean L 86
Mr Green G 26
Mr Dean P 12
Mr Teen G 3598
```

Plan

There are two main abstract classes Area and Weather. Three child classes have been introduced for each of them. Area has two attributes name and amount of water for which it has getters and setters. Area class has some virtual functions like transform and transmute which can be overridden in its child classes. The transmute function in all the child classes takes a Weather and changes its own water level on the basis of that weather and then changes the humidity of that weather as well. The transform function yields a new Area on the basis of the current water level. The constructor of Area is protected which can only be accessed in its child classes. Area has an enumeration namely Exceptions which makes sure that initially the water level is positive.

Weather class has type and humidity as its attributes and it has getters and setters for both of them. Weather has a transform function which transforms one weather into other, depending upon the amount of humidity it has. It is same as the constructor of the Weather class. All the child classes of Weather have a static instance of their own type which is initialized as null but can be changed by another static method. They have a method to destroy this instance as well. They also have a public constructor as well. Here is how different types of areas react to the different types of Weather.

Plain:

Plain area reacts to different weathers as follows:

Weather Type	**Water Change**	**Humidity Change**
Sunny	-3	-
Cloudy	-1	-
Rainy	20	5

If the amount of water exceeds 15 km^3, the area changes to Grassland.

Grassland:

Grassland behaves as follows:

Weather Type	**Water Change**	**Humidity Change**
Sunny	-6	**-**
Cloudy	-2	**-**
Rainy	15	10

Grassland converts into Lake if the water amount goes beyond 50 km^3. And if the water level falls below 16 km^3, Grassland changes to Plain.

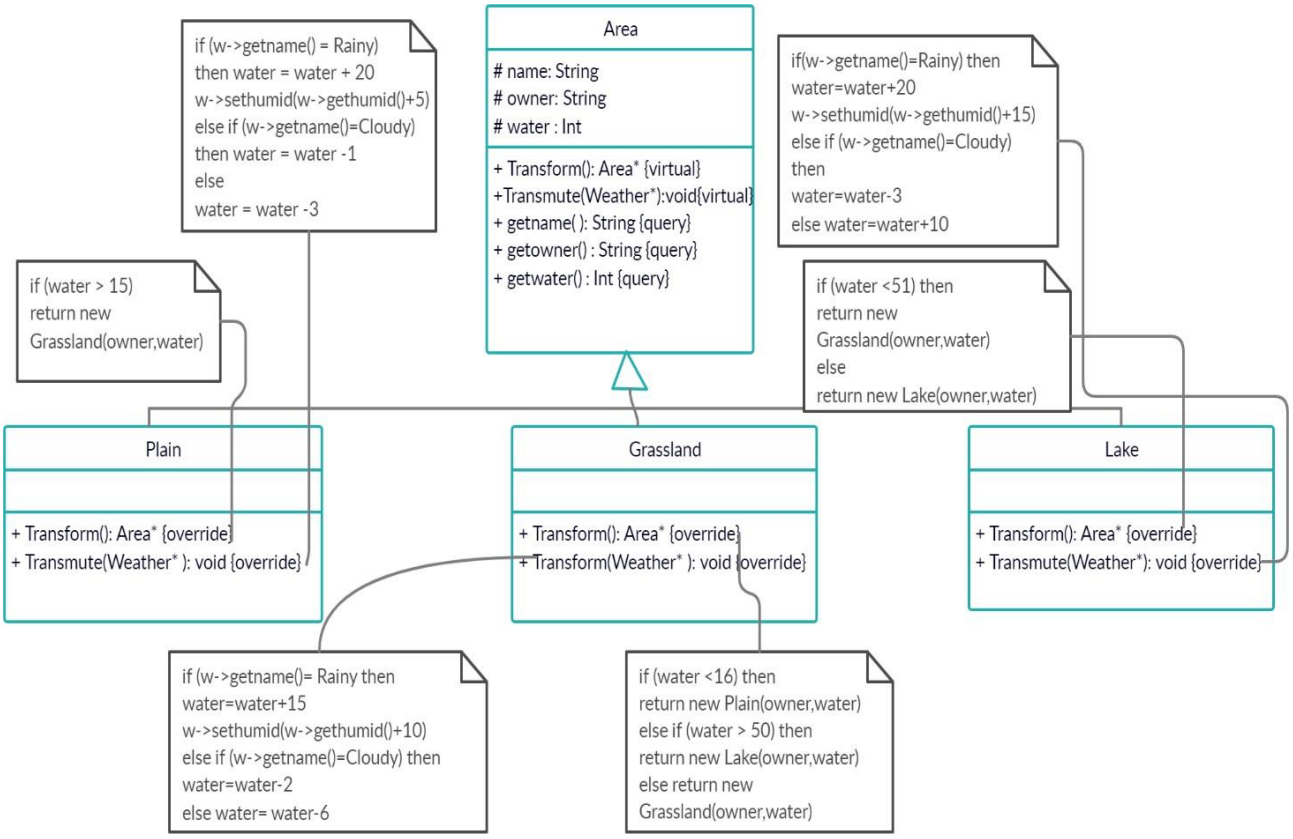
Lake:

Lake is affected by different weathers as follows:

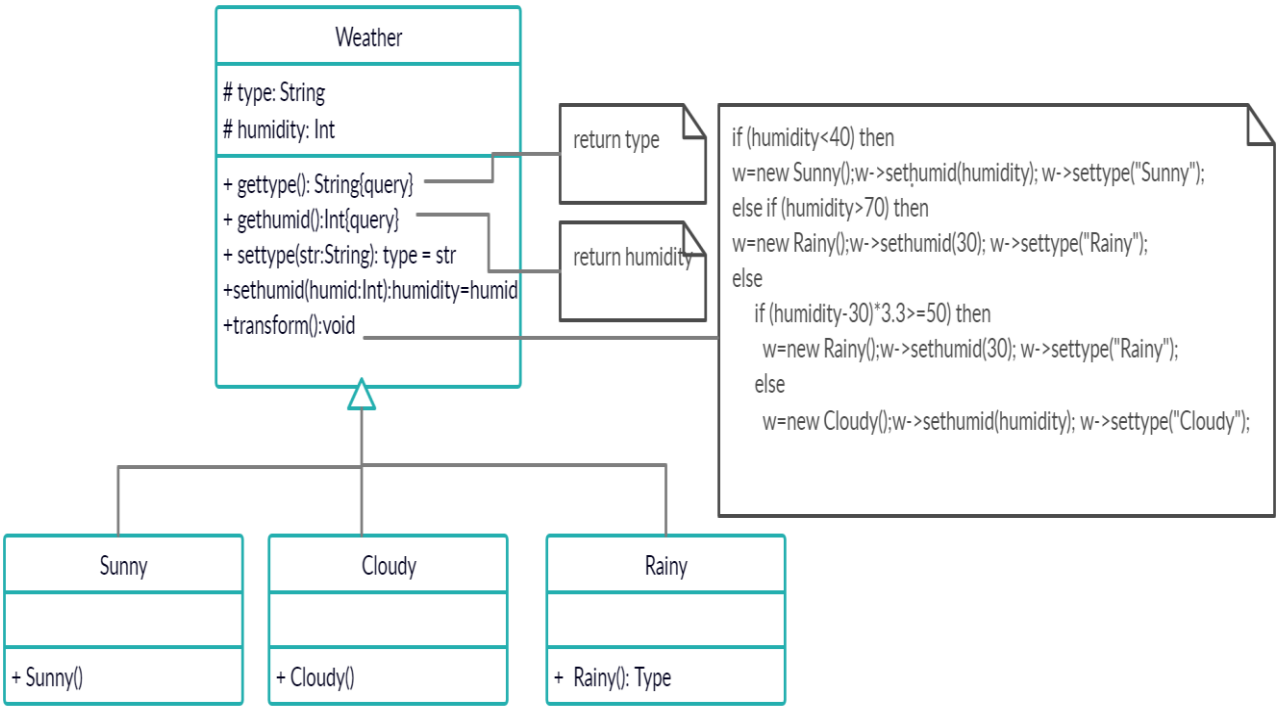
Weather Type	**Water Change**	**Humidity Change**
Sunny	-10	**-**
Cloudy	-3	**-**
Rainy	20	15

Below the amount of 51 km^3 of water, Lake is converted to a Grassland.

Here is the UML class diagram of the Class Area. Transform function of Area returns a new Area on the basis water level. Transmute function expects a Weather as a parameter and it changes its own water level and humidity of that Weather.



UML diagram of Weather class is as follows. Transform function Weather changes the Weather w attribute of Weather in accordance with the humidity level.



In the specification, it is necessary to calculate with the n+1 versions of the owners as every area transmutes the given weather. In every simulation the *transmute* function of area takes the given weather and on the basis of type of this weather, it changes its water level accordingly. Then if the water level falls below zero, it shows an exception. Otherwise the transform function changes the area according to its water level.
Then it transforms the weather as well at the end of every simulation.
Then comes the maxLand procedure, which checks the landlord who has the most water at the end of 10 simulations.

Specification:

A = *owners*: *Owner**, *LandLord*: *String, maxWater:Integer*
Owner = rec[name: String , area* : Area]

Pre = $owners = owners_0$

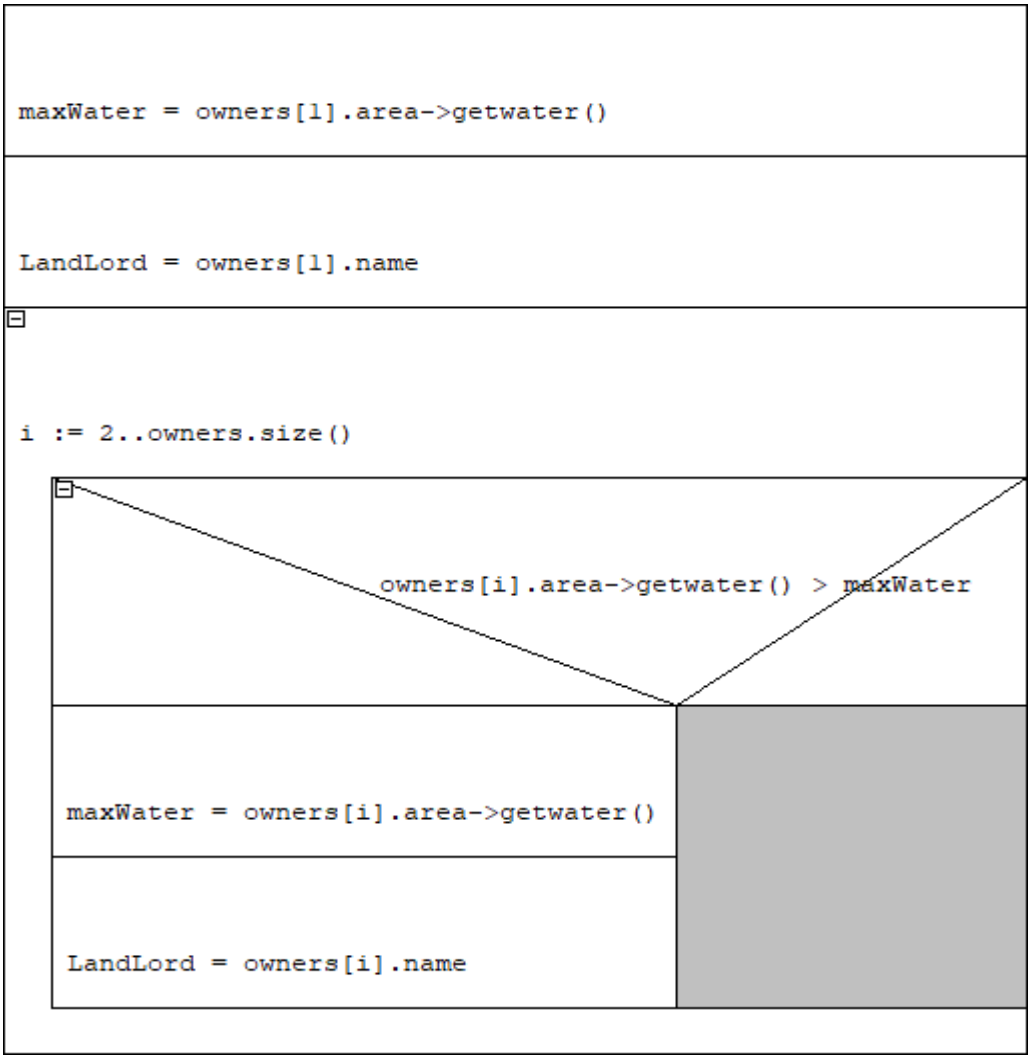
Post = $(LandLord,maxWater) = MAX_{i=1..owners.size()}(owners[i].area->getwater())$

Finding the landlord with most water after 10 simulations is the maximum search.

Analogy:

enor(E)	$i = 1 .. owners.size()$
$f(e)$	$owners[i].area->getwater()$
$H,<$	$Z,<$

Algorithm:


maxWater = owners[1].area->getwater()
LandLord = owners[1].name
i := 2..owners.size()
owners[i].area->getwater() > maxWater
maxWater = owners[i].area->getwater()
LandLord = owners[i].name

Testing

Grey box test cases:

Max Selection:

-first Landlord has the most water
-last Landlord has the most water
-There is only 1 landlord
-Landlord with most water is in middle

Simulate Function on file:

-When water level remains above zero during simulation process
-Water level falls below 0 during the simulation.

Exceptions:

-When there is at least one input area with negative water level

Transform function of Weather:

-Testing transform function of Weather using a Plain
-Testing transform function of Weather using a Grassland
-Testing transform function of Weather using a Lake

Transform and Transmute Functions of Area and Weather:

- Testing transmute and transform functions of Area and transform function of Weather using a Plain
- Testing transmute and transform functions of Area and transform function of Weather using a Grassland
- Testing transmute and transform functions of Area and transform function of Weather using a Lake

Simulate Function using Vector of Owners:

- Testing Simulate function on a single Plain Area
- Testing Simulate function on a single Grassland Area
- Testing Simulate function on a single Grassland Area when its water level falls below 0
- Testing Simulate function on a single Lake Area
- Testing Simulate function on all types of Area