Thet Naing Soe 2nd assignment/0. Task 19th May 2023

NEPTUNCODE : x0flxy
x0flxy@inf.elte.hu
Group 1
Task (No4)

Layers of gases are given, with certain type (ozone, oxygen, carbon dioxide) and thickness, affected by atmospheric variables (thunderstorm, sunshine, other effects). When a part of one layer changes into another layer due to an atmospheric variable, the newly transformed layer ascends and engrosses the first identical type of layer of gases over it. In case there is no identical layer above, it creates a new layer on the top of the atmosphere. In the following we declare, how the different types of layers react to the different variables by changing their type and thickness. No layer can have a thickness less than 0.5 km, unless it ascends to the identical-type upper layer. In case there is no identical one, the layer perishes.

	thunderstorm	sunshine	other
ozone	-	-	5% turns to oxygen
oxygen	50% turns to ozone	5% turns to ozone	10% turns to carbon di-oxide
carbon dioxide	-	5% turns to oxygen	-

The program reads data from a text file. The first line of the file contains a single integer N indicating the number of layers. Each of the following N lines contains the attributes of a layer separated by spaces: type and thickness. The type is identified by a character: Z – ozone, X – oxygen, C – carbon dioxide. The last line of the file represents the atmospheric variables in the form of a sequence of characters: T – thunderstorm, S – sunshine, O – others. In case the simulation is over, it continues from the beginning.

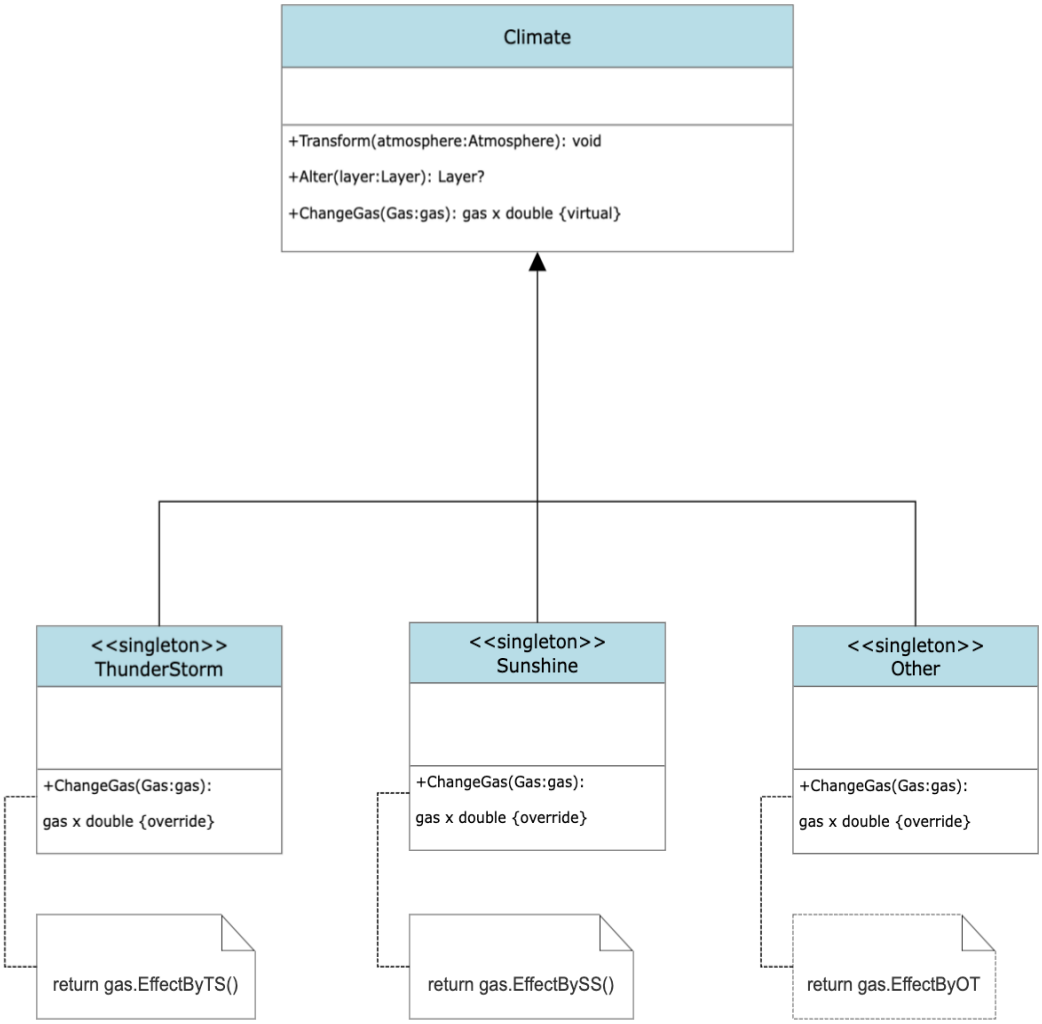
The program should continue the simulation until the number of layers is the triple of the initial number of layers or is less than three. The program should print all attributes of the layers by simulation rounds!

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
Z 5
X 0.8
C 3
X 4
OOOOSSTSSOO
```

Plan

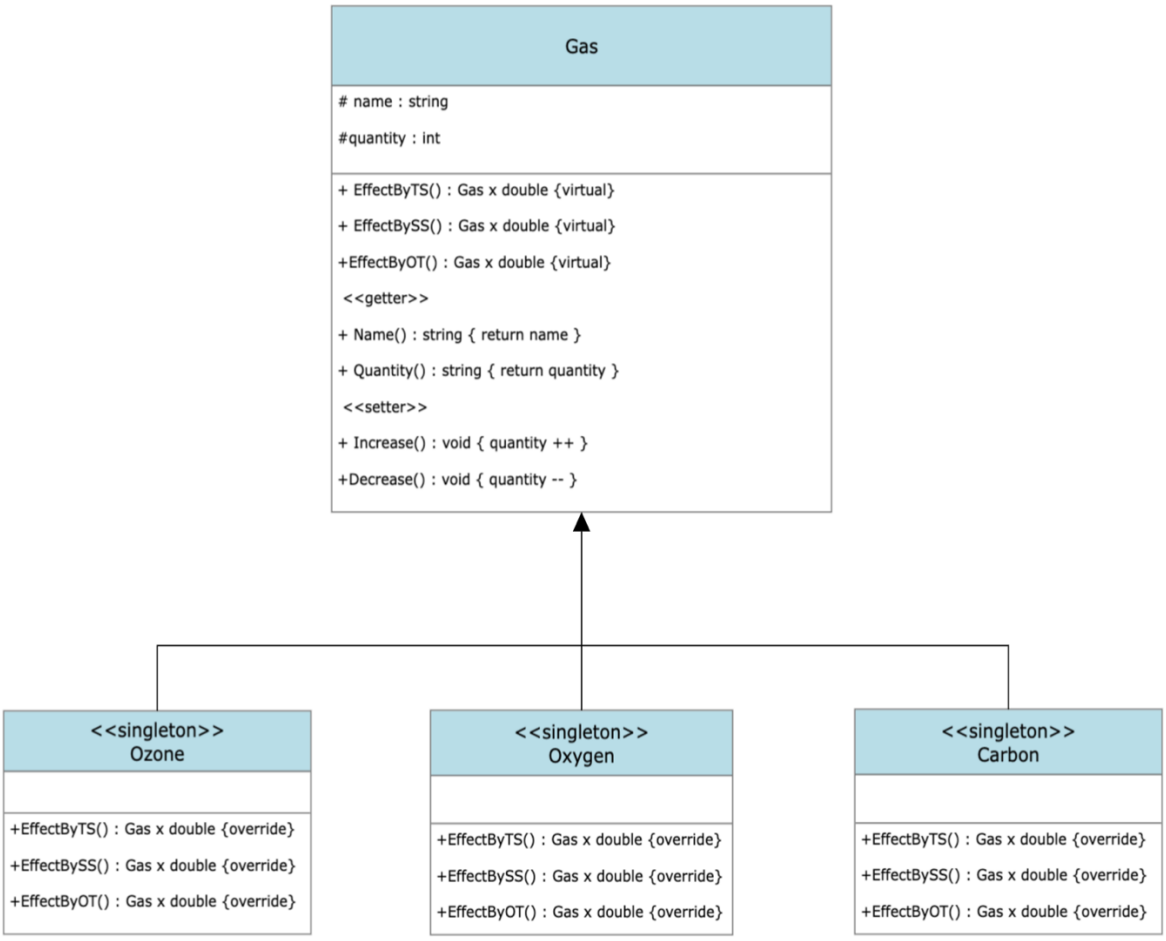
The Class Diagram of Climate is as follows.



Method ChangeGas() of the concrete Climates expect a Gas object as an input parameter as a visitor and call the methods which correspond to the species of the Climate.

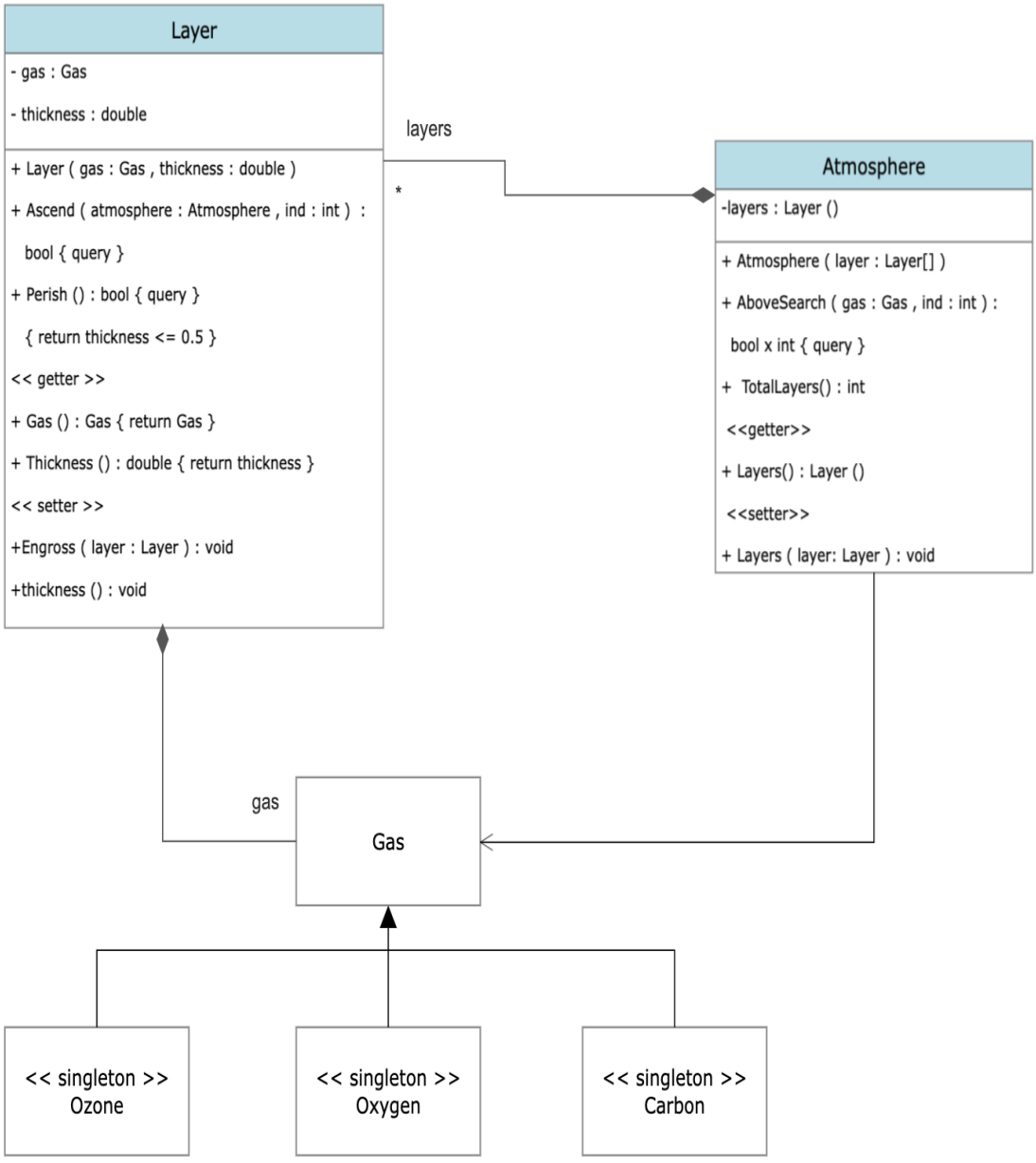
All the classes of the Climate are created based on the Singleton design pattern, as it is enough to create one object for each class.

The Class diagram of Gas is the following.



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

The class diagram of Atmosphere and Layer are the following.



In the specification, it is necessary to calculate with the n+1 versions of the atmosphere as every climate changes it. The 0th version is the initial atmosphere. The transformation of one climate is denoted by function Alter: Layer × Climate^n → Layer × Layer? which gives the changed layer. ith version of the atmosphere is denoted by atmosi, which the program is not going to show, it is going to be just a temporal value of variable atmosphere.

And Concatenation of the layers (after transformation by climate) is a Summations just as the assortment of the layers. As all of them are based on the same enumerator, they can be merged into the same loop (i=1 .. n).

The original number of layers in atmosphere is retrived from a text file by ReadInt method and assigned to the variable “orglayers”, representing the initial state of the atmospheric composition.

A = atmosphere: Atmosphere , climate: Climate ;

Pre = atmosphere = $atmosphere_0 \wedge$ climate = $climate_{0;}$

Post = $\forall i \in [1..n]$: $atmosphere_n$. TotalLayers() < 3 ||
$atmosphere_n$. TotalLayers() = $atmosphere_0$ * 3;
$Atmosphere_i$ = Transform(climate[(i-1)%n + 1] , $atmosphere_{n-1}$)

Analogy : Summation with as long as condition

enor(E) : i = 1 .. as long as !atmosphere.TotalLayers() < 3 || atmosphere.TotalLayers() = orglayers * 3.

f(e) : Transform(climate[(n – 1) % n + 1], atmosphere)

s : atmosphere.layers

H, + , 0 : Atmosphere, ⊖ , atmosphere

i := 1	
!(atmosphere.TotalLayers() < 3 || atmosphere.TotalLayers == 12)	
	Atmosphere = Transform (climate[(i – 1) % n+1] , atmosphere
	i := i + 1

The ith layer is going to have m+1 states as transformation by the climate
which is, in essence, rebuilt. 0th state of atmosphere.Layers[i] is the given layer (atmos.Layers0[i]), while the nth state is the layer after deforming the whole atmosphere (atmos.Layersm[i] =atmosphere.Layers[i]). The i_{th} layer before transformed by the jth climate is denoted by atmosphere.Layersj-1[i] from which the i_{th} state is created by method Alter() (atmos.Layersj[i]).

Testing

Grey box test cases
TestClimate (Summation)
1. length-based:

- one each layer on 3 different climates
- 3 layers

TestSimulation(Counting)
1. length-based:

- total number of layers in atmosphere