THIRD ASSIGNMENT DOCUMENTATION

Bruce Shiundu

EPX2YD

Task

A program showing the transformation of atmospheric gases due to simulation from atmospheric variables.

The layers of gases are of three types (ozone, oxygen, carbon dioxide). Every layer has a name denotation (Z – Ozone, X – Oxygen, C - Carbon dioxide) and its thickness (a natural number). The thickness is at least 0.5 km. Atmospheric variables that cause transformation of gases are categorised as either thunderstorm, sunshine or other effects. When a variable passes through a gas layer, it may transform a given percentage of it forming a new layer type while reducing its thickness.

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.

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.

Analysis[1]

Independent objects in the task are the atmospheric variables.

They can be divided into 3 different groups: sunshine, thunderstorm and other effects. It can be examined what happens when they cross a given atmospheric gas layer.

The effects are as follows:

Thunderstorm:

Atmospheric gas	*Percentage change*	*Gas change*
Ozone	-	-
Oxygen	50	Ozone
Carbon dioxide	-	-

Sunshine

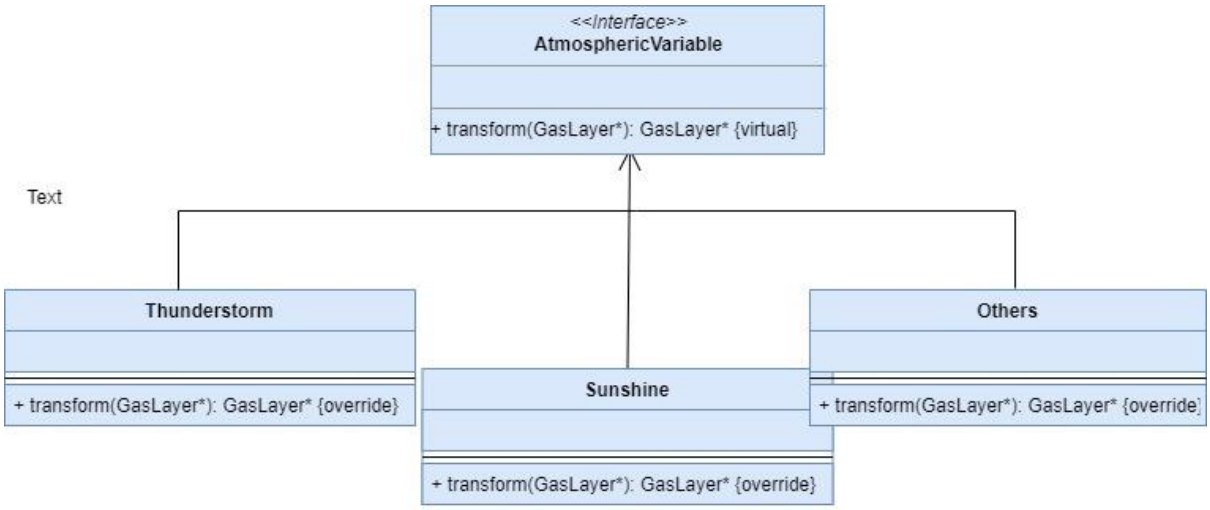
Atmospheric gas	*Percentage change*	*Gas change*
Ozone	-	-
Oxygen	5	Ozone
Carbon dioxide	5	Oxygen

Others

Atmospheric gas	*Percentage change*	*Gas change*
Ozone	5	Oxygen
Oxygen	10	Carbon dioxide
Carbon dioxide	-	-

Plan

AtmosphericVariable is an in interface with the function ***transform ()***, which shows the effect of the atmospheric variable on a given atmospheric gas layer. The function is implemented in the child classes ***Thunderstorm, Sunshine*** and ***Othesr*** making it a pure abstract class.

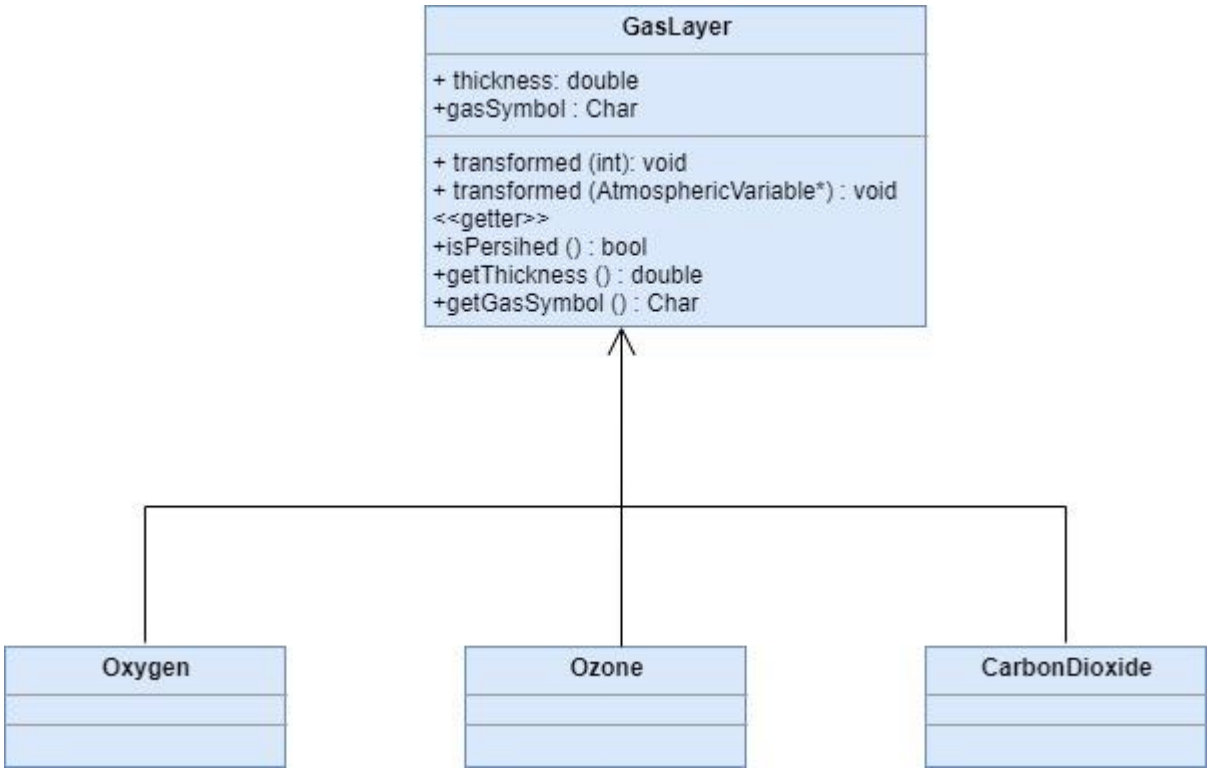


The base class ***GasLayer*** provides a general description of how gas layers are transformed through an overloaded method ***transformed().***

The method either takes a percentage _change(Integer) which shows by what percentage the gas layer (its thickness) changes when simulated by a given atmospheric variable as a parameter or a pointer to a **newly_ transformed** gas layer which indicates that it **merges** with the given new layer which ascends from the bottom thus increasing in thickness.

Getter functions ***isPerished(), getThickness() and getGasSymbol ()*** are provided described within the class. A given gas perishes when its thickness is less than 0.5km.

Concrete gas layers: ***Ozone***, ***Oxygen***, and ***CarbonDioxide***. Are inherited from the base class ***GasLayer***



Plan of the main program

An enumerator enumerates a vector of gas layers after every round of simulation which is then displayed on the screen.

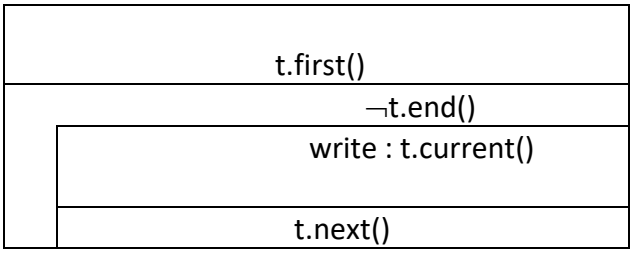
A round of simulation is provided through file input which basically is a sequence of atmospheric variable denotations.

The state of the atmospheric gas layers is displayed after every simulation round.

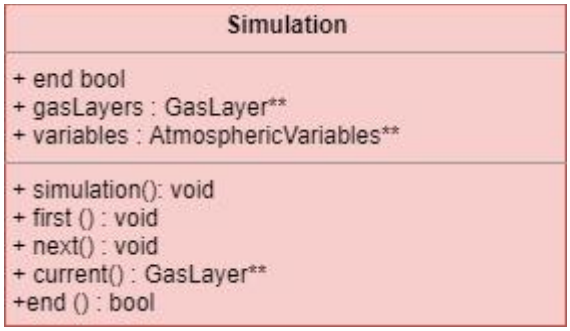
Enumeration is done by the class **Simulation**

A = (t:enor(Simulation))

Pre = (t=t')



The Simulation



The simulation receives a file input during construction.

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 which are used to instantiate gasLayer (A vector of gas layers).

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 which are used to instantiate the variables (a vector of atmospheric variables)

Simulation	first()	next()	current()	end()	simulate()	searchUpperGasLayers ()
gasLayers : GasLayers** variables :AtmospericVariables end : $\mathbb{L}$	simulate()	simulate()	return gasLayers	return end()	See below	See below

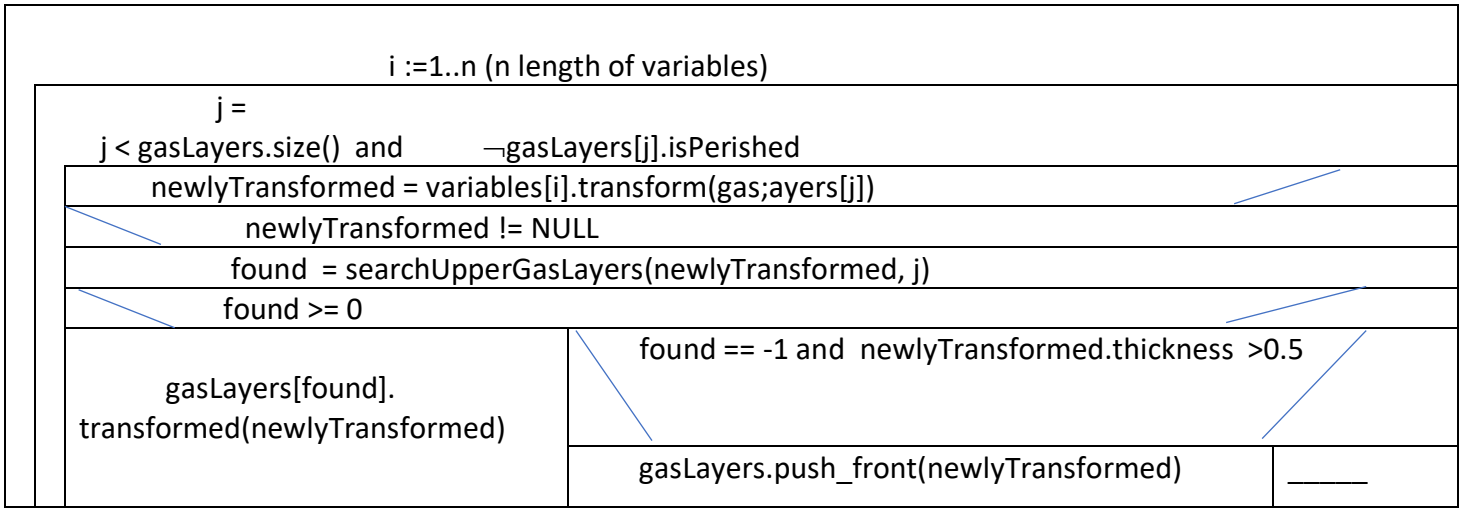
searchUpperGasLayers ()

Takes a newly transformed gas layer (and the starting altitude) and searches in the upper layers if there is a gas layer of the same type. Returns the position of the matching gas layer if found else returns -1 if not found.

simulate ()

The simulate function transforms every gas layer by the atmospheric variable. If a new gas layer is formed, it is added to a matching upper layer of similar type if exists else it forms a new layer on top if it has a thickness of at least 0.5 km.

It uses the function **searchUpperGasLayers ()** to find the existing of a matching upper gas layer.



The simulation stops when any gas layer is perished from the atmosphere.

Testing

Testing is based on unit testing. Individual components have been validated.

Individual units have also been combined and tested.

1. The effect of each atmospheric variable on each atmospheric gas layer has been tested.

2. The simulation has also been tested to come to a completion only when there is a single gas layer perished from the atmosphere.