Szobonya Péter Csaba - YKMM49

3. Beadandó

**Feladat**

Ismerjük a légkör egymás felett elhelyezkedő ózon, oxigén, és széndioxid anyagú légrétegeit, amelyek vastagsága a légköri viszonyoktól (zivataros, napos, egyéb) függően változik. Amikor egy légköri réteg anyagának egy része átalakul, akkor ez az anyagmennyiség fölszáll, és vastagítja a felette lévő első ugyanolyan anyagú réteget. Ha nincs fölötte ilyen réteg, akkor a légkör legtetején új réteget képez. Egy rétegnek sem csökkenhet a vastagsága fél kilométer alá. Ha ez mégis megtörténne, akkor ez a réteg is felszáll, és egyesül a fölötte lévő első ugyanilyen anyagú réteggel. Ha azonban nincs ilyen, akkor megszűnik.

A folyamat során először egymástól függetlenül reagálnak az egyes légrétegek az aktuális időjárási viszonyra, utána rétegenként alulról felfelé haladva felszállnak az újonnan keletkeztek anyagmennyiségek, illetve a túl vékony rétegek.

A következőkben megadjuk, hogy az egyes anyagok miként reagálnak a különböző időjárási viszonyokra.

| anyag | zivatar | napos | más |
| --- | --- | --- | --- |
| ózon | - | - | 5% oxigénné |
| oxigén | 50% ózonná | 5% ózonná | 10% széndioxiddá |
| széndioxid | - | 5% oxigénné | - |

Szimuláljuk a folyamatot, amíg a légköri rétegek száma a bemeneti rétegszám háromszorosára nem nő, vagy három alá csökken. Körönként mutassuk meg a légrétegek összes tulajdonságát!A program egy szövegfájlból olvassa be a légkör adatait! Az első sorban a légrétegek száma szerepel. A következő sorok tartalmazzák alulról felfelé haladva a légrétegek adatait szóközökkel elválasztva: anyaga (ezt egy karakter azonosítja: z -ózon, x -oxigén, s -széndioxid), és vastagsága. A rétegeketleíró részt követő sorban a változó légköri viszonyok találhatók egy karaktersorozatban (z -zivatar, n -napos, m –más). Ha a szimuláció a karaktersorozat végére ér, az elejéről folytatja. A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát. (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

| 4  z 5  x 0.8  s 3  x 4  mmmmnnznnmm |
| --- |

**Elemzés**

A feladat önálló objektumai a különféle elemek, amelyek három elemre bonthatók: ózon, oxigén és széndioxid.

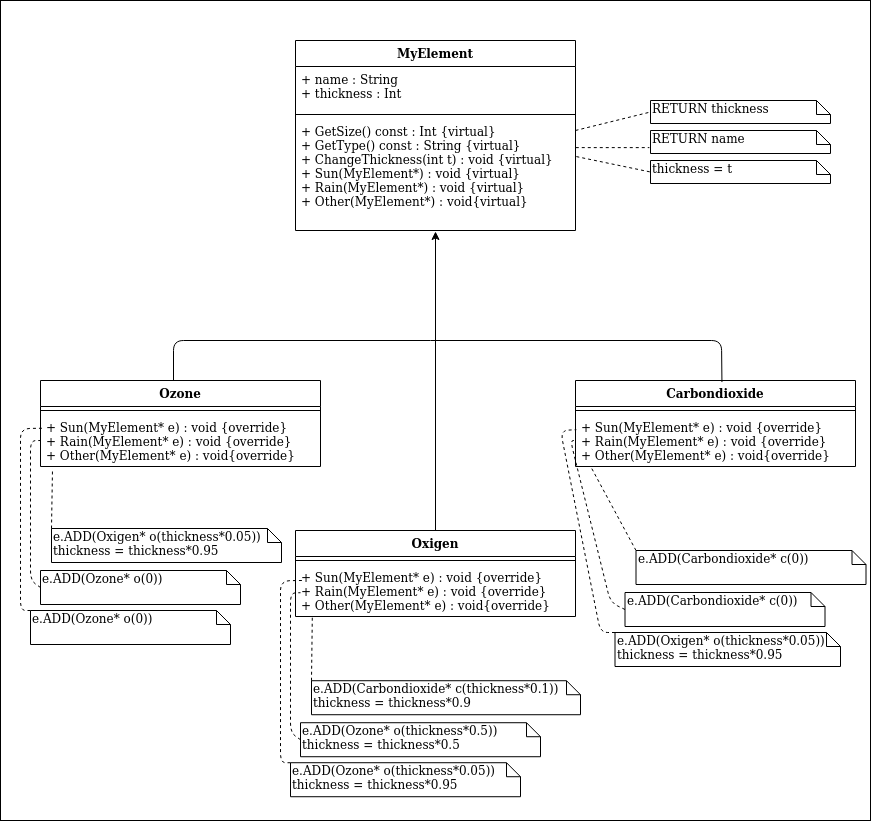
Mindegyiknek van neve, vastagsága és meg lehet kérdezni ezeket róla. Továbbá meg lehet vizsgálni, hogy mi történik vele különböző időjárásokban. Az időjárások felbontják a rétegeket és az új rétegeket felszállítják, illetve a túl vékony rétegeket eltávolítják.

**Terv**

Az elemek leírásához bevezetünk négy osztályt: az elemek általános tulajdonságait megadó MyElement osztályt és ebből származtatjuk az Ozone, Oxigen és Carbondioxide osztályokat. A származtatott osztályok mindegyikétől meg lehet kérdezni, hogy mi történik vele a különböző időjárásokban.

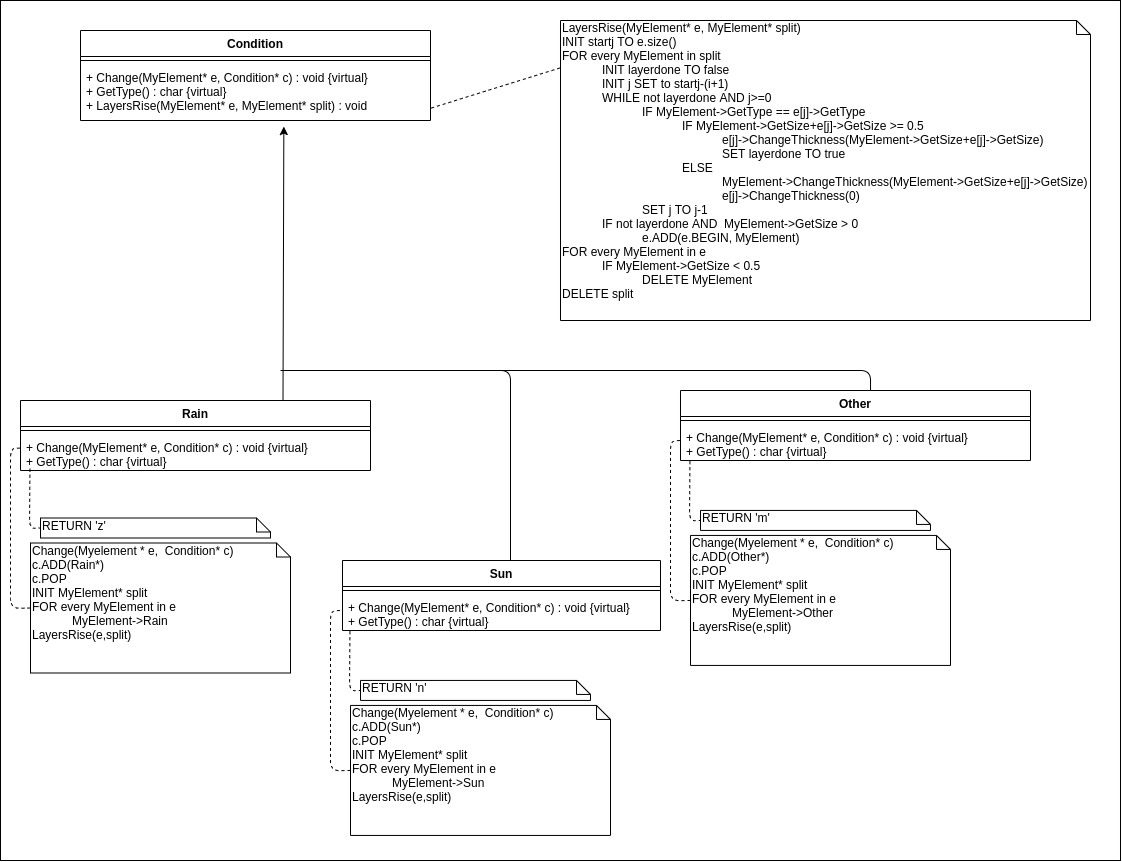
Az időjárások leírásához szintén négy osztályt vezetünk be: egy ősosztályt Condition néven és ennek három származtatott osztályát Rain, Sun, Other néven. Ezek az osztályok felelősek azért, hogy a rétegek helyesen legyenek felbontva és elrendezve.

Ezekre az osztályokra épül rá a Simulate osztály ami kezeli a lépéseit a szimulációnak. Itt lehetőség van arra, hogy lépésenként haladjunk a szimulációval illetve,hogy kiírjük az éppen jelenlegi állapotot.

*A MyElement és származtatottjai*

A Sun, Rain, Other metódusok egy MyElement listát várnak amibe bele tudják írni a szétbontott elemeket.

*A Condition osztály és származtatottjai*



***A Simulate Class***

Ez az a class ami irányítja az idójárási eseteket és az elemek változását. Ennek az osztálynak a konstruktorában beolvassuk a szimulálni készült adatokat, amikkel majd dolgozni fog a program.

A classnak továbbá van még egy Next(), egy End() és egy PrintCurrent() függvénye. Amik egy MyElement típusú vektorra és egy Condition típusú sorra támaszkodnak. Ezek tárolják a légrétegeket és a sorban következő időjárásokat.

*A simulate class konstruktora*

Itt olvassuk be a szimuláció adatait. A feladat leírás alapján feltehetjük, hogy a bemenet mindig helyes.

| read(f,startLayers) | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| layers = MyElement\*, conditions = Condition\* | | | | | | |
|  | i=1..startLayers | | | | | |
|  | read(f,tempchar,tempfloat) | | | | | |
|  | tempchar == ‘z’ | | | tempchar == ‘x’ | tempchar == ‘s’ | |
|  | layers.push\_back(Ozone\* (tempfloat)) | | | layers.push\_back(Oxigen\* (tempfloat)) | layers.push\_back(Carbondioxide\* (tempfloat)) | |
| read(f,ss) | | | | | | |
|  | | i=1..ss.size() | | | | |
|  | | tempc=ss.at(i) | | | | |
|  | | tempc == ‘m’ | tempc == ‘n’ | | | tempc ==’z’ |
|  | | conditions.push(Other\*) | conditions.push(Sun\*) | | | conditions.push(Rain\*) |

*A simulate class Next függvénye*

Itt dolgozunk fel egy időjárási lehetőséget az összes rétegre.

A=(layers:MyElement\*, conditions:Conditions\*, end:Boolean)

Ef: -

Uf: A jelenlegi rétegek felbontása és helyes sorbarendezése a feladat alapján

| end = layers.size > startLayers\*3 or layer.size < 3 |
| --- |
| condition.front -> Change(layers,conditions) |

*A simulate class End függvénye*

RETURN end

*A simulate class PrintCurrent függvénye*

A függvény feladata, hogy kiírja az éppen jelenlegi állását a szimulációnak.

**Tesztelési terv**

1. Tesztelés túl kevés adattal.

2. Egy Bemenet végigszámolása és az eredmények egyeztetése.