

Szemesi Gábor	ql10n6	OEP komplex beadandó	2022.05.
---------------	--------	----------------------	----------

## 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 felszáll, 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	-

Addig szimuláljuk a folyamatot, amíg el nem fogy valamelyik anyag teljesen a légkörből. Körönként mutassuk meg a légrétegek összes tulajdonságát!

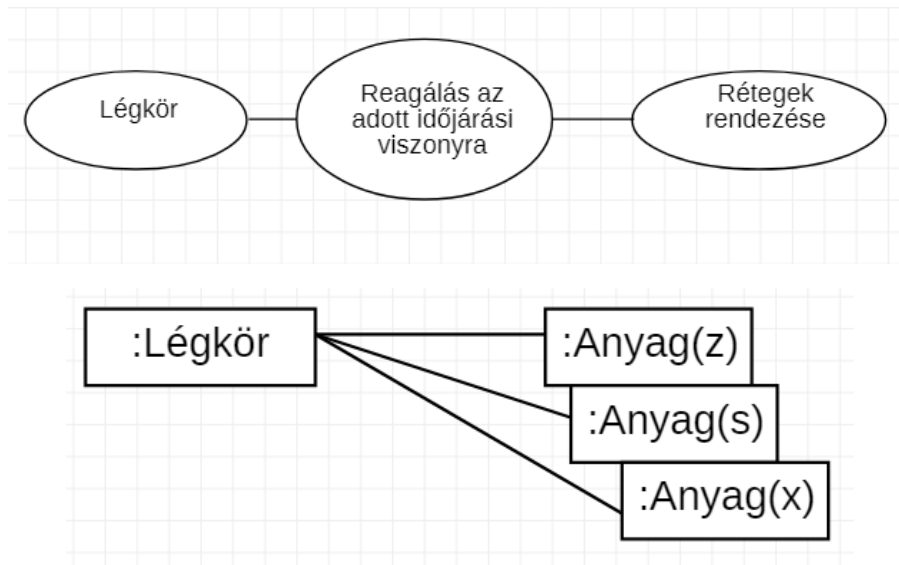
A program egy szövegfájlból olvassa be a légkör adatait! Az első sorban a légrétegek száma szerepel. A következő sorok tartalmazzák alulról felfelé haladva a légrétegek adatait szóközzel elválasztva: anyaga (ezt egy karakter azonosítja: z - ózon, x - oxigén, s - széndioxid), és vastagsága. A rétegeket leíró 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:

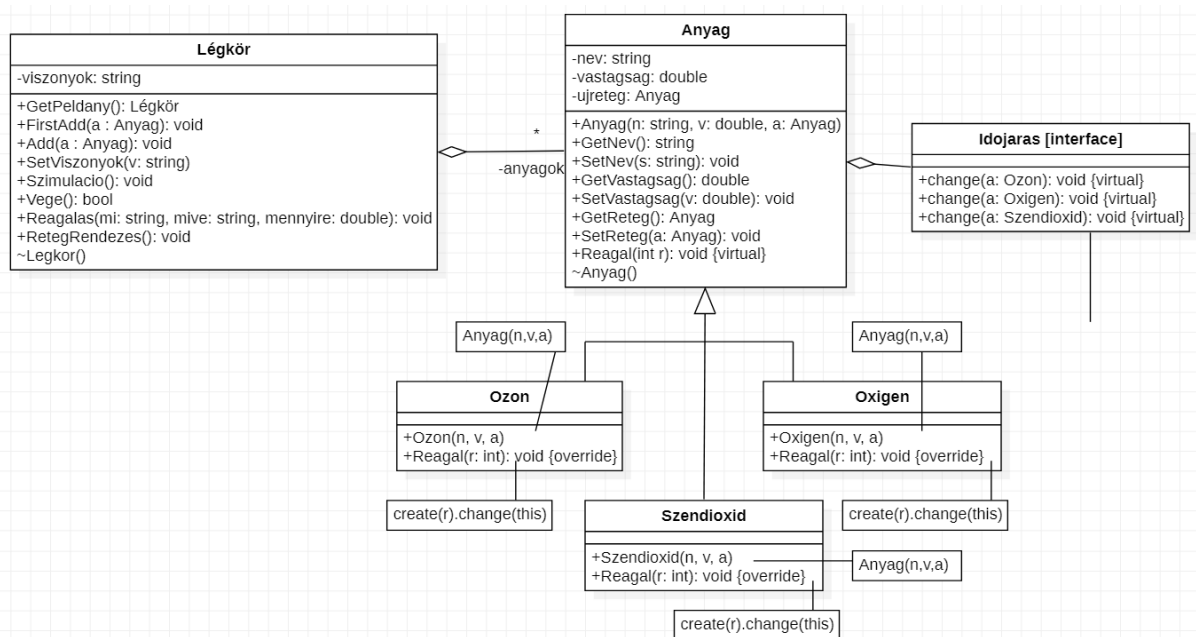
```
4
z 5
x 0.8
s 3
x 4
mmmmnnznnmm
```

## Terv

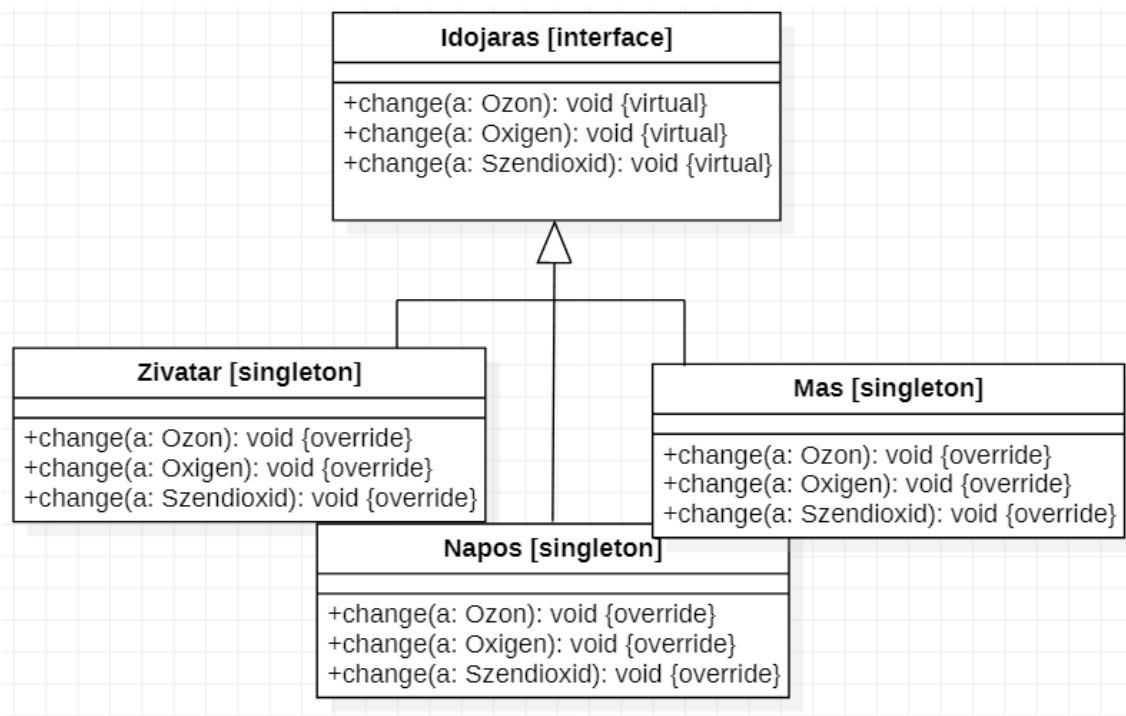
A feladatban a megadott időjárési viszonyokra reagál a légkör. A szimulációt az Szimulacio() metódus végzi, amíg mind három anyag megtalálható a légkörben (Ózon,Szendioxid,Oxigén).



A anyagok között megkülönböztetünk Ózont, Oxigént, és Széndioxidot. Az OpenClose elv betartása céljából ezeket származtatással definiáljuk



Mindegyiknek van egy Reagal() metódusa, amely az anyagnak a paraméterként megkapott viszonyt írja le úgy, hogy meghívja az időjárásnak a réteg típusához illeszkedő change() metódusát. Itt a „látogató” tervmintát alkalmazzuk.



A konkrét időjárési viszonyok osztályait is származtatással definiáljuk. Ezeket az osztályokat az „egyke” tervmintának megfelelően valósítjuk meg. A change() metódus az alábbi (a feladat szövegéből kinyerhető) táblázatnak megfelelően reagálnak a rétegek az időjárési viszonyokra.

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

### Léggör:

```

GetPeldany(): void
-----
return new Legkor();
  
```

Léggör példányosítása.

```

FirstAdd(a :Anyag): void
-----
anyagok.AddFront(a);
  
```

A légréteget a struktúránk elejére szúrja be.

```

Add(a :Anyag): void
-----
anyagok.AddBack(a);
  
```

A légréteget a struktúránk végére szúrja be.

Szimulacio(): void

```
-----  
iterate=0  
WHILE Vege() LOOP  
  Reagalas(viszonyok[iterate])  
  RetegRendezes()  
  IF iterate < |viszonyok| THEN iterate=iterate+1;  
  ELSE iterate = 0  
  ENDIF  
ENDLOOP
```

A szimuláció. Amíg mind három anyag megtalálható a légkörben elvégzi az adott időjárásnak megfelelő átalakulásokat. Ha elfogyna az időjárásai viszony akkor ellőről kezd.

Vege(): bool

```
-----  
z=SEARCH (e∈anyagok) e.GetNev()="z"  
n=SEARCH (e∈anyagok) e.GetNev()="n"  
m=SEARCH (e∈anyagok) e.GetNev()="m"  
return (z ∧ n ∧ m)
```

A Vege() metódus ellenőrzi, hogy mind három anyag megtalálható a légkörben. Ha igen true-val, ha nem false-al tér vissza.

Reagalas(ido: string): void

```
-----  
int n;  
IF ido = 'z'  
  n = 0  
ELSE IF ido = 'n'  
  n = 1  
ELSE IF ido = 'm'  
  n = 2  
ENDIF  
FORALL a IN anyagok LOOP  
  anyagok[j].Reagal(n)  
ENDLOOP
```

Az anyagok reagálnak.

RetegRendezes(): void

```
-----  
FOR (i = 1..|anyagok|) LOOP  
  IF anyagok[i].GetReteg() NOT NULL THEN  
    l = false;  
    FOR (j = i..|anyagok|) LOOP  
      IF anyagok[j].GetNev() = anyagok[i].GetReteg().GetNev() ∧ l=false) THEN  
        anyagok[j].SetVastagsag(anyagok[i].GetReteg().GetVastagsag())  
        l = true  
      ENDIF  
    ENDLOOP  
    IF l=false THEN  
      Add(Anyag(anyagok[i].GetReteg().GetNev(), anyagok[i].GetReteg().GetVastagsag(), NULL));  
    ENDIF  
    anyagok[i].SetReteg(nullptr);  
  ENDIF  
  IF (anyagok[i].GetVastagsag() < 0.5)  
    l = false;  
    FOR (j = i..|anyagok|) LOOP  
      IF (anyagok[j].GetNev() == anyagok[i].GetNev() ∧ l=false)  
        anyagok[j].SetVastagsag(anyagok[i].GetVastagsag())  
        l = true  
      ENDIF  
    ENDFOR  
    anyagok.delete(i);  
    i=i-1  
  ENDIF  
ENDFOR
```

A RetegRendezes() metódus kezeli az új és/vagy túl vékony (<0.5) rétegeket.

Az új rétegnél lineáris keresését végzünk. Először új réteget keresünk. Ha találtunk a réteg felett keresünk hozzá hasonlót. Ha találtunk akkor hozzá kapcsoljuk az új réteget. Ha nem találtunk a légkör legtetején létre hozunk egyet. Ez lehet kisebb, mint 0.5, de ezt majd lekezeljük a vékony rétegeknél (a feladat megengedi, hogy megmaradjon, mert lehet tud egyesülni másik felszálló réteggel).

A vékony rétegnél lineáris keresését végzünk, hogy megtaláljuk a 0.5 km-nél kisebb vastagságú rétegeket. Ha találunk ilyen, akkor felszáll és egyesül a felette lévő első anyaggal. Ha nem akkor megsemmisül a réteg.

```
~Legkor()
-----
FORALL a IN anyagok LOOP
  delete a
ENDLOOP
```

Destruktor.

### **Anyag:**

```
Anyag(n:string, v:double, a:Anyag)
-----
nev, vastagsag, ujreteg=n, v, a
```

Konstruktor.

<<getter,setter>>

```
GetNev(): string
-----
return nev
```

```
SetNev(s: string): void
-----
nev=s
```

```
GetVastagsag(): double
-----
return vastagsag
```

```
SetVastagsag(in v:double): void
-----
vastagsag=v
```

```
GetReteg(): Anyag
-----
return ujreteg
```

```
SetReteg(a: Anyag): void
-----
ujreteg=a
```

```
~Anyag()
-----
delete ujreteg
```

Destruktor.

## Tesztelési terv

Meg kell vizsgálni minden időjárási viszonyban jól reagálnak-e a rétegek. Továbbá új anyag létrejött-e és helyesen anyagú. Tesztet kell még írni a vékony réteg megszűnésére, felszállására.