Feladat

A föld hidrológiai körfolyamatában a különböző földterületek befolyásolják az időjárást és a különböző időjárások hatására a földterületek változnak. Minden földterületeknek van neve, fajtája (puszta, zöld, tavas) , tárolt vízmennyisége (km3 -ben). A földterületek feletti közös levegőnek ismerjük a páratartalmát (százalékban). Az időjárás a levegő aznapi páratartalmától függ: Ha ez meghaladja a 70%-ot, esős idő lesz, és ekkor lecsökken a páratartalom 30%-ra. 40%-os páratartalom alatt az időjárás napos lesz. 40 és 70% közötti páratartalom esetén az esős időjárásnak (páratartalom-40)*3,3 százalék az esélye, egyébként felhős időjárás lesz. (Véletlenszám generátorral állítsunk el egy számot 0 és 100 között, és ha ez kisebb, mint a (páratartalom-40)*3,3 érték, akkor esős, különben felhős időjárás legyen.)

Az egyes földterületek – a megadásuk sorrendjében – reagálnak a különböző időjárásokra: először a vízmennyiségük változik, majd befolyásolják a levegő páratartalmát. Egyetlen földterület vízmennyisége sem lehet negatív.

> Puszta: napos idő hatására a vízmennyiség 3 km3 -rel csökken, felhős idő hatására 1 km3 -rel, eső hatására 5km3 -rel nő. A levegő páratartalmát 3%-kal növeli. 15 km3 -nél több tárolt víz esetén zölddé változik.
>
> Zöld: napos idő hatására a vízmennyiség 6 km3 -rel csökken, felhős idő hatására 2 km3 -rel, eső hatására 10km3 -rel nő. A levegő páratartalmát 7%-kal növeli. 50km3 -es vízmennyiség fölött tavassá változik. 16km3 alatt pusztává változik.
>
> Tavas: napos idő hatására a vízmennyiség 10 km3 -rel csökken, felhős idő hatására 3 km3 -rel, eső hatására 15km3 -rel nő. A levegő páratartalmát 10%-kal növeli. 51km3 alatt zölddé változik.

Addig szimuláljuk a folyamatot újra és újra a földterületek megadott sorrendjében, amíg minden földterület azonos fajtájú nem lesz. Körönként mutassuk meg a földterületek összes tulajdonságát! A program egy szövegfájlból olvassa be az adatokat! Ennek első sorában a földterületek száma szerepel. A következő sorok tartalmazzák a földterületek adatait szóközökkel elválasztva: a terület tulajdonosát (szóköz nélküli sztring), fajtáját (egy karakter azonosítja: p - puszta, z - zöld, t - tavas), és a kezdeti vízmennyiségét. Az utolsó sor a földterületek feletti levegő kezdeti páratartalmát mutatja. A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát. (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

```
4
Bean t 86
Green z 26
Dean p 12
Teen z 35
```

98

Elemzés

A feladatban három különböző tájtípus van, amik különböző módon reagálnak az időjárási viszonyokra. Időjárásból szintén három féle van, napos, esős vagy felhős.
Minden tájnak van egy adott vízszintje, és ez nő vagy csökken az időjárástól függően. Ha egy táj elér egy bizonyos vízszintet, akkor megváltozik a típusa.
Ezen kívül a különböző tájak a páratartalmat is másképpen változtatják meg.

Vízszint változás	Napos idő	Esős idő	Felhős idő
Puszta	-3 km^3	+5 km^3	+1 km^3
Zöld	-6 km^3	+10 km^3	+2 km^3
Tavas	-10 km^3	+15 km^3	+3 km^3

Levegő páratartalmának változása tájanként:

Táj	**Páratartalom v.**
Puszta	+3%
Zöld	+7%
Tavas	+10%

Ezen kívül az alábbi vízmennyiségeknél változnak meg a tájak:

-Puszta: vízmennyiség > 15 → Zöld

-Zöld: vízmennyiség > 50 → Tavas, vízmennyiség < 16 → Puszta

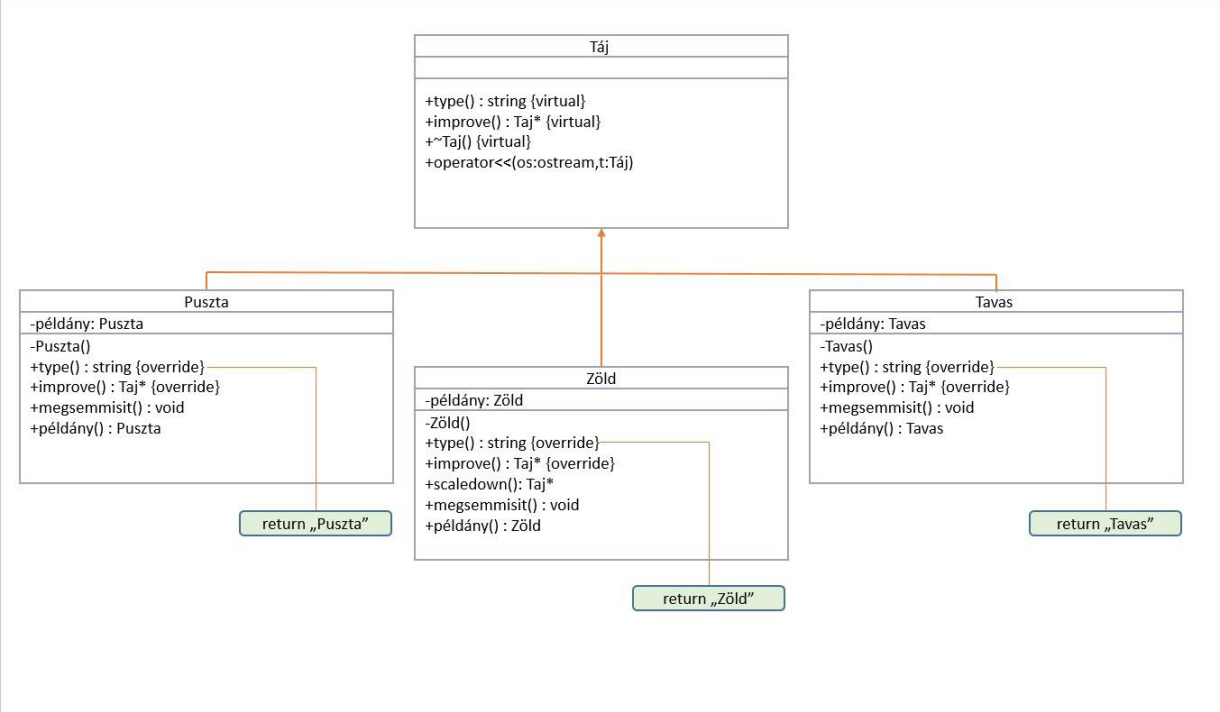
-Tavas: vízmennyiség < 51 → Zöld

Terv

Létrehozunk egy Taj nevű ősosztályt, ami tartalmazza az összes virtuális függvényt, mint pl a type() és az improve().

Ebből az ősosztályból leszármaztatunk 3 külön ösztályt, Puszta, Zold és Tavas néven.

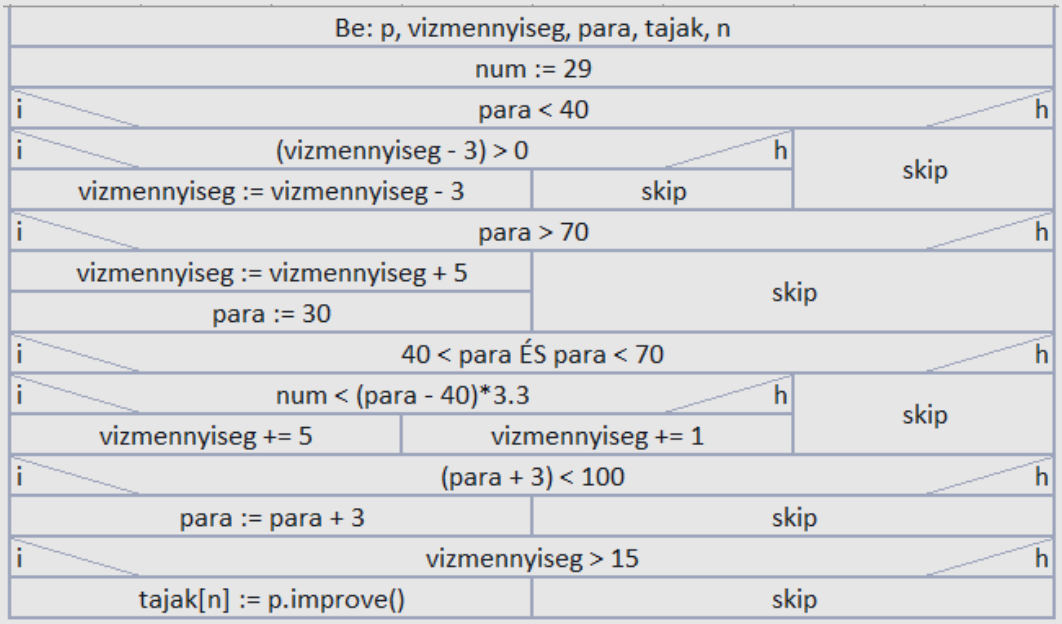
Ezek az osztályok mint az egyke tervezési mintára épülnek, hogy könnyebben lehessen a memóriát felszabadítani. A Zold osztály még rendelkezik az alap függvényeken kívül egy scaledown() függvénnyel is, mivel ez a táj két féle képpen tud megváltozni.



Az UML diagramon szereplő függvényyeken kívül még létre lett hozva egy, a szimuláció függvényeit tartalmazó file. Ez a függvény a látogatók tervmintát valósítja meg, a három különböző talajtípusra máshogy működik:

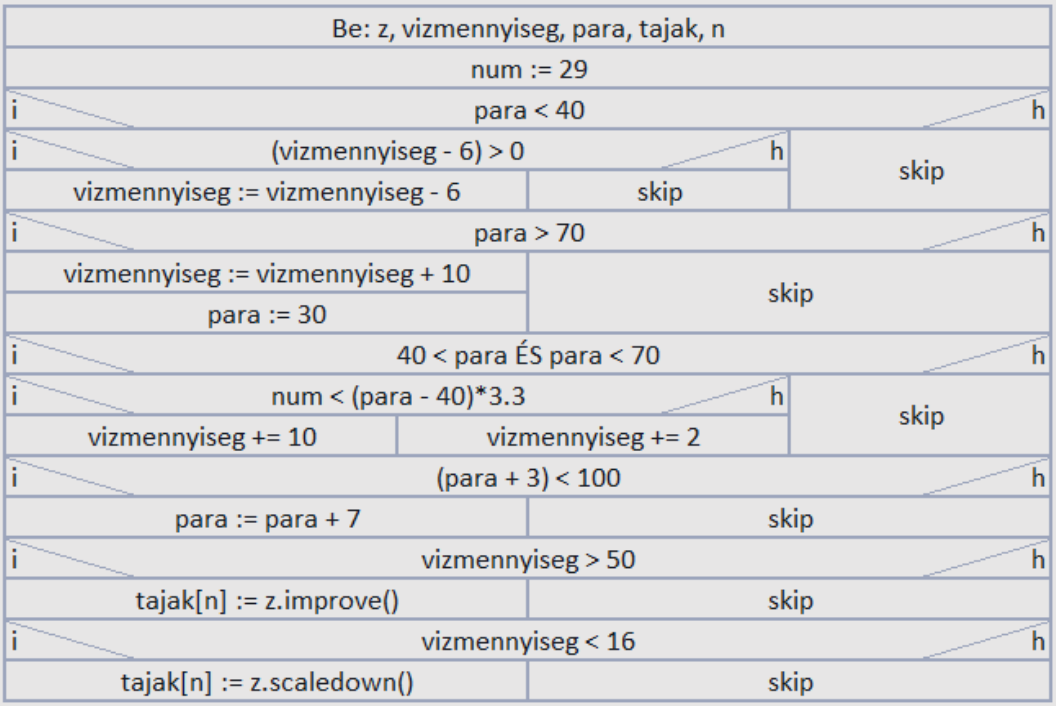
Puszta

A= (p: Puszta*, vizmennyiseg: Egész, para: Egész, tajak: Taj* vector, n: Egész)



Zöld

A= (z: Zold*, vizmennyiseg: Egész, para: Egész, tajak: Taj* vector, n: Egész)



Tavas

A= (t: Tavas*, vizmennyiseg: Egész, para: Egész, tajak: Taj* vector, n: Egész)

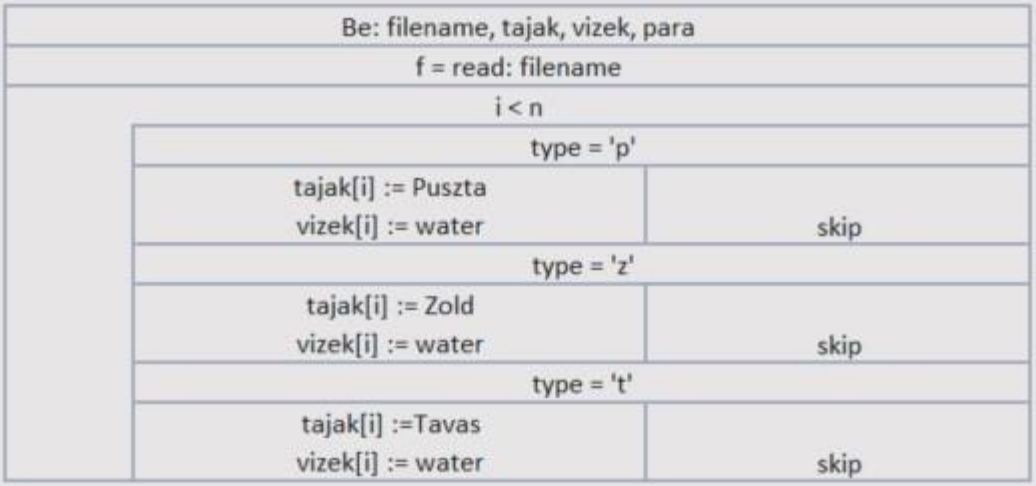
A main.cpp fileon belül találhatóak meg a create, simulate_whole, allequal, test_run, isempty, deleteVector és deleteTajak függvények.

Create függvény

A create függvény beolvassa a paraméterül kapott filet, és ebből a fileból olvas be sorokat, ez alapján tölti fel a vektorokat.

Először beolvas egy számot, n-t. Ez a szám azt jelzi, hány sor van a fileban. Ezután minden sorban megnézi, milyen betű van és szám van. A betű alapján a vektor i-dik eleme lehet Puszta, Zöld vagy Tavas. A számok pedig a vizek vektor i-ik elemei lesznek, ezek az adott tájak vízszintjét jelzik.

A= (filename: File, tajak: Taj* vektor, vizek: Egész vektor, para: Egész)



$$\forall i \in [1..n] : type = 'p' \rightarrow tajak[i] := Puszta$$
$$type = 'z' \rightarrow tajak[i] := Zold$$
$$type = 't' \rightarrow tajak[i] := Tavas$$
$$vizek[i] := water$$

Simulate_whole függvény

A simulate_whole függvény végigmegy a tajak Taj* tartalmú vektoron, és minden elemre meghívja a már fentebb vázolt simulate függvények egyikét. A= (tajak: Taj* vektor, vizek: Egész vektor, para: Egész)

$$\forall i \in [1..size(tajak)] : tajak[i] \rightarrow simulates(tajak,vizek,para)$$

Allequal függvény

Az allequal függvény azt nézi, hogy a tajak vektor minden eleme egyenlő típusú-e, azaz a szimuláció végetért-e.

Végigmegy a vektoron, és ha már mind ugyan olyan, true-val tér vissza.

A= (tajak: Taj* vektor, igaze: Logikai)

∀i ∈ [1..size(tajak)] : igaze ≔ tajak[i].type = tajak[0].type

Test_run és isempty függvények

Ez a két függvény a tesztelés szempontjából fontos. A test_run tulajdonképpen ugyan azt csinálja, mint a simulate_whole, annyi különbséggel, hogy ez nem egyszer fut végig a vektoron, hanem addig szimulál, míg minden típus egyenlő nem lesz.

Az isempty függvény szintén a tesztelésnél fontos, viszont ezt a főprogramban is felhasználjuk. A függvény annyit néz, hogy a megnyitott file üres-e vagy sem. **deleteVector és deleteTajak függvények**

Ez a két függvény a memória felszabadítására szolgál. Az előbbi törli a tajak vektor elemeit, a második pedig törli a létrehozott tájak megmaradt példányait. A: (tajak: Taj* vektor)

∀i ∈ [1..size(tajak)] : delete tajak[i]

Főprogram

A főprogramon belül hozzuk létre a vektorokat illetve szükséges változókat, mint pl. a para nevű egész változó, illetve a file nevét eltároló string.

Ezen kívül itt hívjuk meg a szimulációs függvényeket, itt írjuk ki szimulációnként az adatokat, és itt hívjuk meg a törlő metódusokat.

A tesztelést már tartalmazza a program, több bemenetre is.