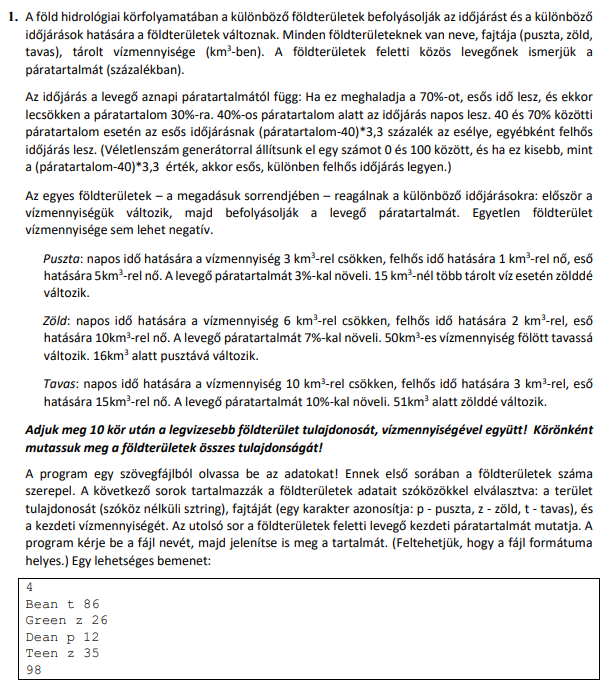
**Tóth Botond**  **3.beadandó/1.feladat** 2021.05.23.

MQH41V

[mqh41v@inf.elte.hu](mailto:mqh41v@inf.elte.hu) vagy tothbotond00@gmail.com

10. csoport

**Feladat**



**Elemzés**

A feladat önálló objektumai az egymást követő földterületek lesznek. Meg tudunk különböztetni pusztát, zöldet és tavas jellegű földterületet. Egy földterületnek tudjuk a nevét, a fajtáját, hogy hány km3 víz található benne és azt is tudjuk róluk, hogy mi történik velük különböző időjárás hatására, plusz azt is, hogy az időjárásra ők milyen hatással vannak.

Az alábbi táblázatban látszik, hogy a földre milyen hatással van az időjárás.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Napos | Felhős | Esős |
| Puszta | -3 km3 | +1 km3 | +5 km3 |
| Zöld | -6 km3 | +2 km3 | +10 km3 |
| Tavas | -10 km3 | +3 km3 | +15 km3 |

Az egyes területek típusa, a vízmennyiségük alapján határozható meg:

|  |  |
| --- | --- |
| 0-15 km3 | Puszta |
| 16-50 km3 | Zöld |
| 51+ km3 | Tavas |

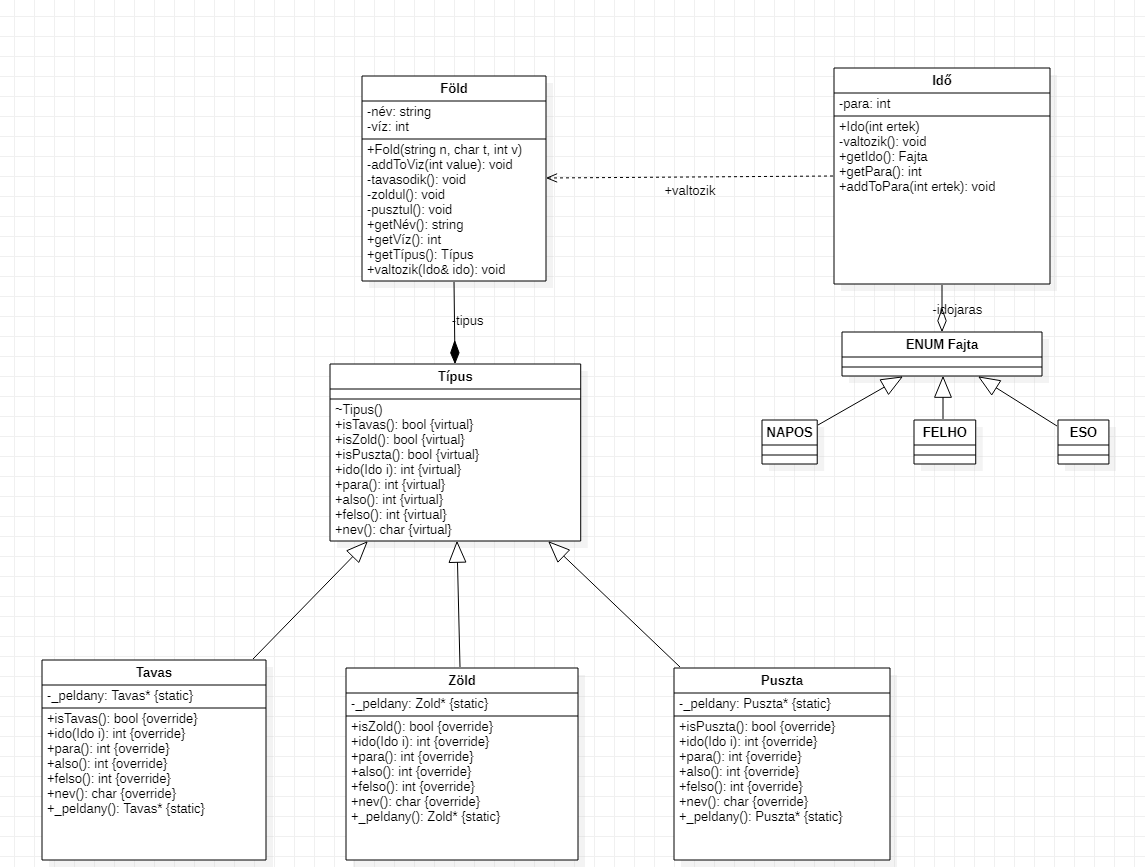
Emellett lesz egy időjárás objektumunk is. Ennek egy folyamatosan változó páratartalma lesz, amire különböző hatással vannak a földterületek. Emellett az időjárás típusa is meghatározható a páratartalom függvényében:

|  |  |  |  |
| --- | --- | --- | --- |
| Puszta | +3% | Napos | 0-40% |
| Zöld | +7% | Felhős | 40+% esély |
| Tavas | +10% | Esős | 40+% esély, 70+% fix |

**Tervezés**

A földek leírásához, szükségünk van egy Föld nevű osztályra amiben tárolni fogjuk a földterület nevét és annak a vízmennyiségét. Ezeket az adattagokat majd getterekkel tudjuk elérni és a konstruktorral fogjuk inicializálni. Minden föld területnek lesz egy típusa, a típus egy virtuális osztály lesz és abban fogjuk tárolni, hogy a földtípus, hogy reagál majd az időjárásra és az hogy változtatja meg az időjárást. Ennek jelenleg 3 alosztálya lesz, de így könnyen tudjuk majd bővíteni a későbbiekben a földtípusok fajtáit. Ehhez látogató tervmintát használunk majd. Valamint fontos megjegyezni, hogy egy típus objektumot, elég egyszer létrehoznunk, ezért emellett a singleton/egyke tervezési mintát is kihasználhatjuk a feladat megoldásánál.

Lesz még egy Időjárás objektumunk is, aminek lesz egy páratartalma és az időjárás típusát eldöntő metódusa. Az időjárás fajtáira elég egy felsorolót létrehoznunk, mert ezeknek nem lesz semmi fontos információja, hiszen a földtípus fog a fajtára reagálni, nem pedig a fajta a földtípusra.

**UML ábrák és főbb metódusok**

Az itt látható UML ábra írja le a főbb függvényeket és a programban megvalósítandó osztályokat.

Főbb tervezési minták: Egyke tervminta, Látogató tervminta

Az egyes osztályok függvényeit és metódusait a következő oldalakon fogom kifejteni.

Az állományban az eredeti UML terv is elérhető **beadando3.mdj** néven.

Típus:

A típus szinte összes metódusa virtuális, éppen ezért inkább a belőle származott osztályok fontosabbak.

isTavas()/isZold()/isPuszta : **return false**

ido(Ido i)/also() / felso() / para() : **return 0**

nev() : **return ’a’**

A típusból származtatott osztályok:

|  |  |  |
| --- | --- | --- |
| Tavas  isTavas() : **return true**  ido(Ido i):  if i.getIdo() == ESO **return 15**  else if i.getIdo() ==FELHO **return 3**  else **return -10**  also() : **return 51**  felso() : **return ∞**  para() : **return 10**  nev() : **return ’t’** | Zöld  isZold() : **return true**  ido(Ido i):  if i.getIdo() == ESO **return 10**  else if i.getIdo() ==FELHO **return 2**  else **return -6**  also() : **return 16**  felso() : **return 50**  para() : **return 7**  nev() : **return ’z’** | Puszta  isPuszta() : **return true**  ido(Ido i):  if i.getIdo() == ESO **return 5**  else if i.getIdo() ==FELHO **return 1**  else **return -3**  also() : **return 0**  felso() : **return 15**  para() : **return 3**  nev() : **return ’p’** |

Emelett az egyke tervezési minta miatt minden osztálynak lesz egy \_peldany tagja amiben egy sajátmagára mutató pointert fog tárolni. Ha szeretnénk létrehozni egy ilyen osztályt akkor azt a \_peldany() metódussal tehető meg ami mind a 3 osztály estében így néz ki:

if (NULL = \_peldany ) **return new Tipus()** /itt a Típus vagy Tavas vagy Zold vagy Puszta

else return **\_peldany**

Ezzel a módszerrel biztos, hogy egy objektumot nem hozunk létre kétszer, ami a Föld osztályon belül igen gyakori eset lenne ha nem használnánk ezt a tervezési mintát.

Idő:

Az Idő osztályunk legfőbb adattagja egy felsorolás lesz, amiben a jelenlegi időjárás lesz megtalálható: ez lehet ESŐ, NAPOS, vagy FELHŐ. A látogató tervmintának köszönhetően a Föld osztály minden időjárásra másképp fog reagálni (pontosabb a Föld osztály típusa fog reagálni az időjárásra). Emellett lesz az időjárásnak egy páratartalma, ami ha változik, akkor az időjárás is változni fog(ami tudja változtatni a páratartalmat ha ESŐ )

Az adattagokon kívül az alábbi metódusok vannak definiálva az osztályon belül:

getIdo() : **return idojaras**

getPara() : **return para**

valtozik() : (ez egy privát függvény amit a konstruktor és az addToPara() hív meg)

if para < 40 **idojaras = NAPOS**

else if para > 70 **idojaras = ESO para = 30**

else random = new random(0..100-ig)

if (para-40) \*3.3 > random **idojaras = ESO para = 30**

else **idojaras = FELHO**

A random számot a C++-ban a srand és a rand() segítségével tudjuk majd megoldani. Itt csak jelzem, hogy a random szám 0-100-ig vehet majd fel értéket.

addToPara(int ertek):

**para = para + ertek valtozik()**

Konstruktor(int ertek):

**para = ertek valtozik()**

Fontos, hogy a valtozik() metódus megváltoztathatja a párát. Ez ezért nem probléma, mert a Föld osztály csak az időjárásra kíváncsi.

Föld:

A Föld osztálynak 3 privát adattagja lesz: a neve, a vízmennyiség ami benne található és a típusa amit egy Típus\*-ben fog tárolni.

Amikor létrehozzuk az objektumot akkor az alábbi értékek kerülnek beállításra:

Konstruktor(string n, char t, int v):

**nev = n víz = v**

if t ==’t’ **tipus = Tavas.peldany()**

else if t ==’z’ **tipus = Zold.peldany()**

else **tipus =Puszta.peldany()**

Az első két értéket inicializációs listával megoldhatjuk, a típusát pedig ellenőriznünk kell.

Privát metódusok:

pusztul() : **tipus = Puszta.peldany()**

zoldul() : **tipus = Zold.peldany()**

tavasodik() : **tipus = Tavas.peldany()**

addToViz(int value):

**víz = víz + value**

if víz < 0 **víz = 0**

if tipus->also() > víz

if tipus->isTavas() **zoldul()**

else **pusztul()**

else if tipus->felso < víz

if tipus->isPuszta() **zoldul()**

else **tavasodik()**

Ebben a metódusban a változott víz mellet el tudjuk dönteni, hogy a föld típusa romlott-e vagy éppen javult.

Publikus függvények:

getNev() : **return nev**

getViz() : **return víz**

getTipus() : **return tipus**

valtozik(Ido& i):

**addToViz(tipus->ido(i)) i.addToPara(tipus->para())**

A valtozik() függvény az egyik legfontosabb függvénye a feladatnak. Ebbe lép interakcióba a földterület az időjárással és itt változtatják meg egymást kölcsönösen az objektumok.

**Fő függvény**

A Fő függvényünkben már csak 10-szer le kell játszanunk a köröket és ezek után végrehajtani egy maximum kiválasztást. Feltételezzük, hogy a függvénybe már be van olvasva a földterületek száma, az összes földterület egy vektorba van eltárolva és egy Idő is létre van hozva egy kezdeti páratartalommal.

A = (kornyezet: Föld\*n , idojaras: Idő)

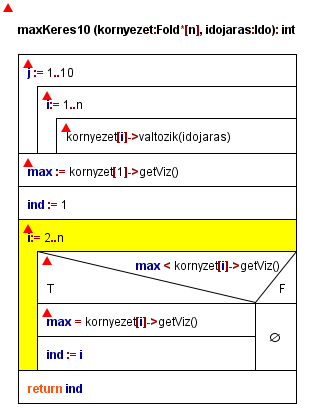
Ef = (n > 0 ∧ kornyezet = kornyezet’∧ idojaras = idojaras’)

Uf = ( Ef ∧ ∧ ∧ max,ind =𝑀𝐴𝑋𝑖=1..𝑛(kornyezet[i]->getViz() )

Visszavezetés:

m,n,max,ind ~ 1,n,kornyezet[i]->getViz(),ind

H ~ Z

Algoritmus:

**Tesztelés**

Osztályok tesztesetei:

1. Ido osztály tesztelése
   1. addToPara() függvény tesztelése(negatív pára, pára hatására változik-e)
2. Fold osztály tesztelese
   1. pusztul() /zoldul() / tavasodik() függvények tesztelése (változik-e a típus)
   2. addToViz() függvény tesztelése (víz 0 alá esik / víz normálisan változik-e típustól függően)
   3. A valtozik() függvény tesztelve lett a fenti függvények segítségével
3. A Tipus-ból származtatott osztályok tesztelve lettek a fenti függvények használatával

Main tesztesetei:

1. Beolvasás tesztelése
   1. üres fájl esetén {test1}
   2. egy elem esetén {test2}
   3. több elem esetén (pl. 10) {test3}
   4. Hibás fájl esetén – a víz értéke 0 alatt van {test4}
2. Körök lejátszásának tesztelése
   1. ennek tesztelése a főprogramban valósul meg
3. Maximum kiválasztás tesztelése
   1. Intervallum hossza szerint
      1. Üres állomány
      2. Egy elemet tartalmazó állomány
      3. Több elemet tartalmazó állomány
   2. Maximum helye szerint
      1. Első helyen van a maximum
      2. Végén van a maximum
      3. Random helyen van a maximum
   3. Maximum tétel alapján
      1. Csak egy maximum van
      2. Több maximum is van (hamarabb beolvasott elem lesz a maximum)