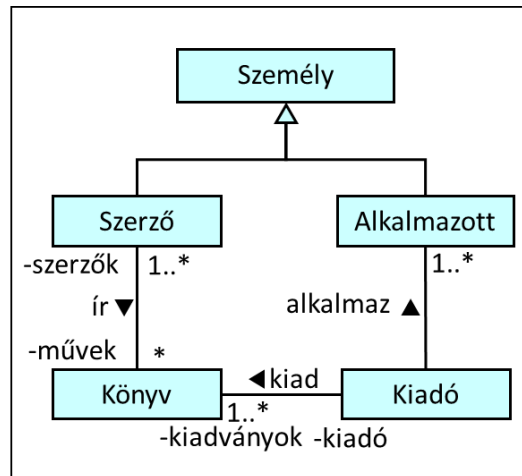


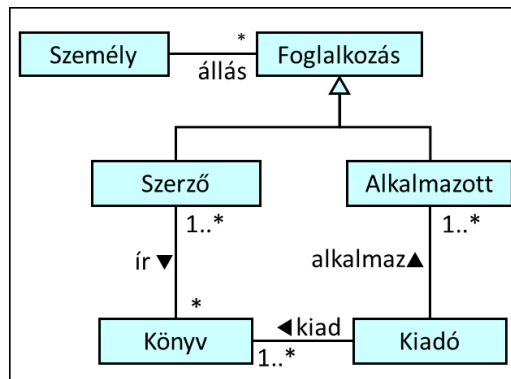
9. Származtatás

Témakör: Származtatás, virtuális metódusok felüldefiniálása, futási idejű polimorfizmus.

1. A könyveket legalább egy szerző írja és pontosan egy kiadó adja ki. Egy kiadó legalább egy könyvet kiad. A kiadó legalább egy alkalmazottat foglalkoztat. Egy alkalmazottat pontosan egy kiadó alkalmaz. Az alkalmazottak és a szerzők is személyek, és nincs olyan szerző, aki kiadói alkalmazott lenne. (Itt már származtatás is kell) Feladatok: Hány könyvet írt egy adott szerző? Ki egy kiadónál legtöbb könyvet írt szerző?



Ebben a modellben éles határ van a szerzők és a kiadói alkalmazottak között, továbbá egy személy legfeljebb egy kiadónál lehet alkalmazott. Változtatni kell a modellen ahhoz, hogy lehessen valaki egyszerre szerző is, és több kiadónak az alkalmazottja is.

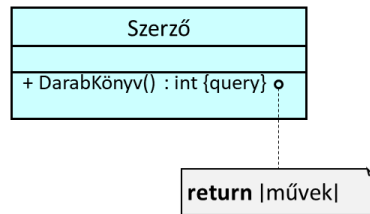


Most pedig oldjuk meg az alábbi feladatokat:

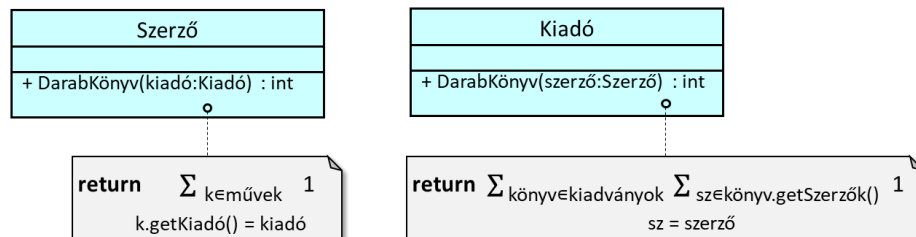
- a) hány könyvet írt egy adott szerző?
- b) hány könyvet írt egy adott szerző egy adott kiadónál?
- c) ki egy kiadó által legtöbbet foglalkoztatott szerző?

Döntsük el, hogy melyik osztályban helyezzük el metódusként a fenti tevékenységeket? Ehhez vizsgáljuk meg:

- melyik objektum felelősségi köréhez kapcsolódik a kérdés?
 - melyik objektumból kiindulva biztosít hatékonyabb elérést a modell a kérdés megválaszolásához?
 - kell-e módosítani a modellen? (például legalább 1 multiplicitás a maximum kiválasztás gyűjteményéhez, közvetlen kapcsolat két osztály objektumai között, unique specifikátor)
- a) Elég a Szerző osztályban definiálni a kérdést megválaszoló DarabKönyv()metódust, és felvenni a szerző könyvre hivatkozó művek szerepnevet privát adattagként:

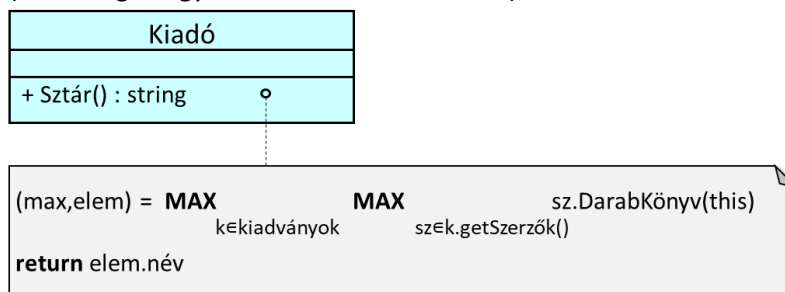


- b) Két lehetőség is van: vagy a szerző, vagy a kiadó metódusaként vezetjük be ezt a tevékenységet. Be kell vezetni egy kiadó által kiadott könyvekre hivatkozó szerepnevet (kiadványok), egy könyv kiadóját azonosító szerepnevet (kiadó), illetve a szerzőire mutató szerepnevet (szerzők), ez utóbbi kettőhöz publikus getter-t is biztosítani kell.



A baloldali a hatékonyabb.

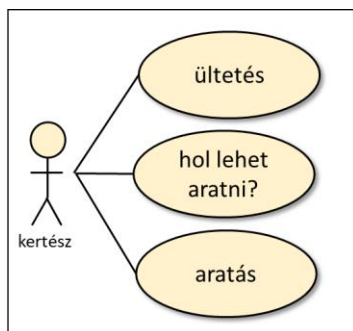
- c) Ez a kiadó metódusa kell legyen, és megoldja egy
- maximum kiválasztás, amely felsorolja a kiadó által kiadott könyveket, azon belül a könyv szerzőit (ennél fogva ugyanazt a szerző többször is), és veszi szerzők által írt könyvek számát;



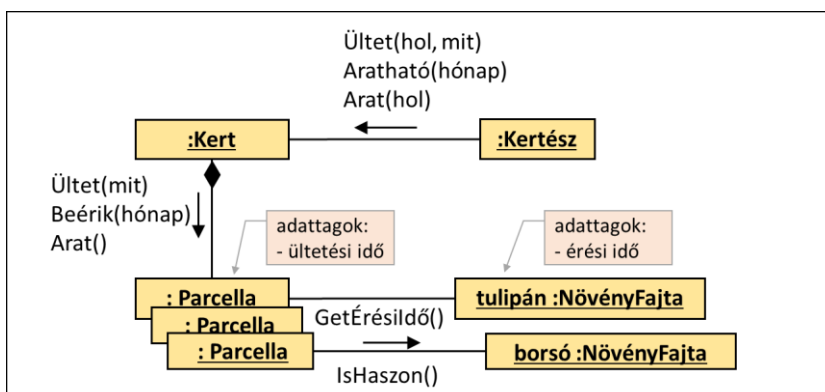
- zsákszerű gyűjteménybe helyezni az egyes könyvek szerzőit, majd lekérni leggyakoribb szerzőt;
- közvetlen kapcsolatot létrehozni a kiadók és a szerzőik között;

2. Egy kertet egy kertész gondoz. A kert parcellákból áll, minden parcellába egyféle növény ültethető. Az ültetés idejét eltároljuk (hónapban). A növények lehetnek haszonnövények, mint burgonya, borsó, paprika; vagy virágok, mint tulipán, szegfű, rózsza. A növényeknek ismerjük az érési idejét (hónapban). Listázza ki a kertész azokat a parcellákat, ahol az adott hónapban haszonnövények fognak beérni!

A kertész háromféle tevékenységet végezhet: növényt ültet egy adott parcellába, megkérdezi, hogy mely parcellákban értek meg haszonnövények, és learatja a termését egy adott parcellának.



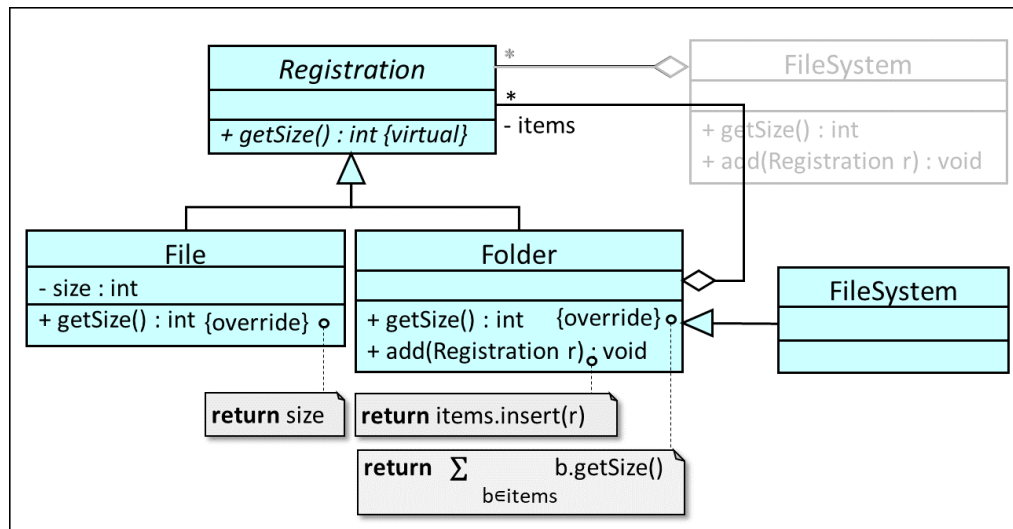
A feladat modelljében négyféle objektummal kell számolni: kertész, kert, parcella, növény. A kerthez több parcella tartozhat.



A kert metódusai biztosítják a kertész tevékenységeit (Ültet(hol, mit), Aratható(), Arat(hol)). Ezek a parcellák tevékenységeitől függenek. Amíg az Ültet(hol, mit) és az Arat(hol) egyszerűen csak meghívják az adott parcella hasonló tevékenységét (hol.Ültet(mit), hol.Arat()), addig az Aratható() metódushoz fel kell sorolni az összes parcellát, és azoknál vizsgálni, hogy beért-e rajta haszonnövény (Beérik()). Ez utóbbinál szükség lesz arra, hogy megkapjuk egy növény érési idejét (GetÉrésildő()), és azt a tulajdonságát, hogy haszonnövény-e (IsHaszon()).

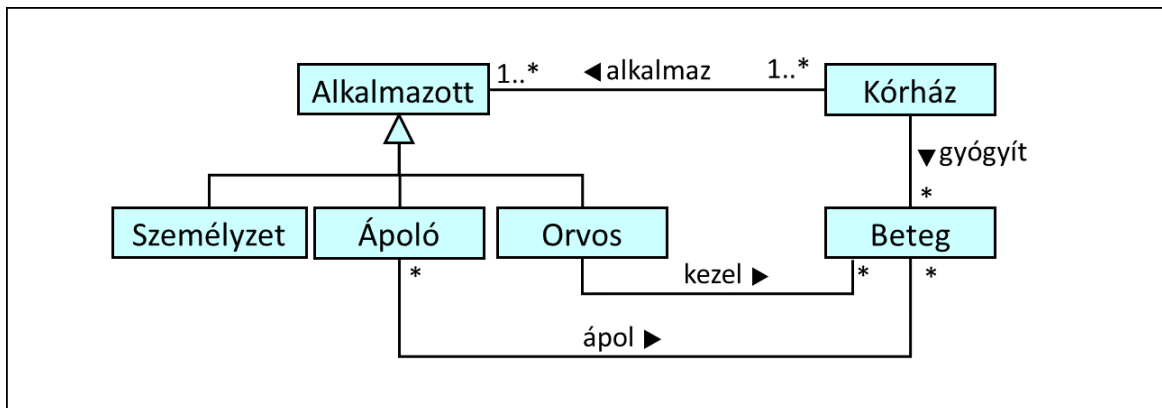
3. Egy számítógépes fájlrendszerben a fájlokat könyvtárakba szervezzük. Minden könyvtár tetszőleges számú fájlt vagy könyvtárat tartalmazhat. A fájlrendszerben a fájlok lehetnek közvetlen a fájlrendszerhez kötve (root), vagy valamelyik könyvtárban is elhelyezkedhetnek. Mennyi memóriát foglal egy adott könyvtár?

Ennek a feladatnak a modellezése lehetőséget ad az „összetétel” (composite) tervminta beépítésére. Valójában ez a feladat egy klasszikus alkalmazása ennek a tervmintának, amely az objektumoknak olyan fa hierarchiában elrendezett kapcsolatrendszerét képes leírni, ahol a fa leveleiben illetve a fa belső pontjaiban eltérő típusú objektumok vannak. A FileSystem osztály kétféleképpen is definiálható.

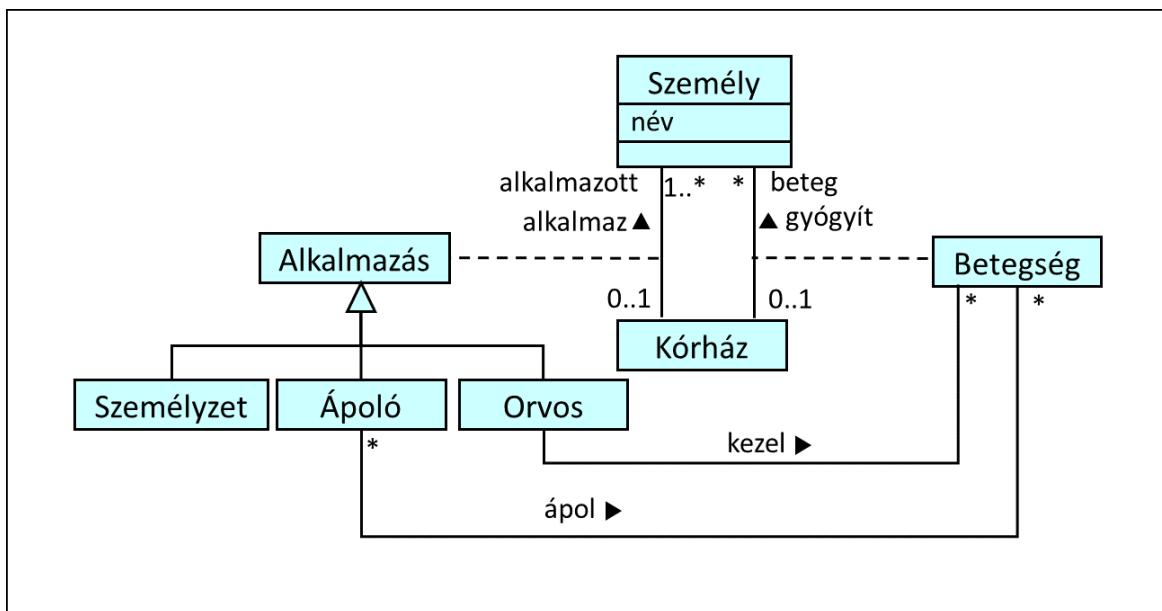


Ezt a szerkezetet ügyesen használja ki a `getSize()` metódus, amely logikai értelemben egy rekurzívan hívható függvénynek tűnhet, habár nyelvi szempontból itt nincs rekurzív hívás, hiszen itt az egyik objektum `getSize()` metódusa egy másik objektum `getSize()` metódusát hívja.

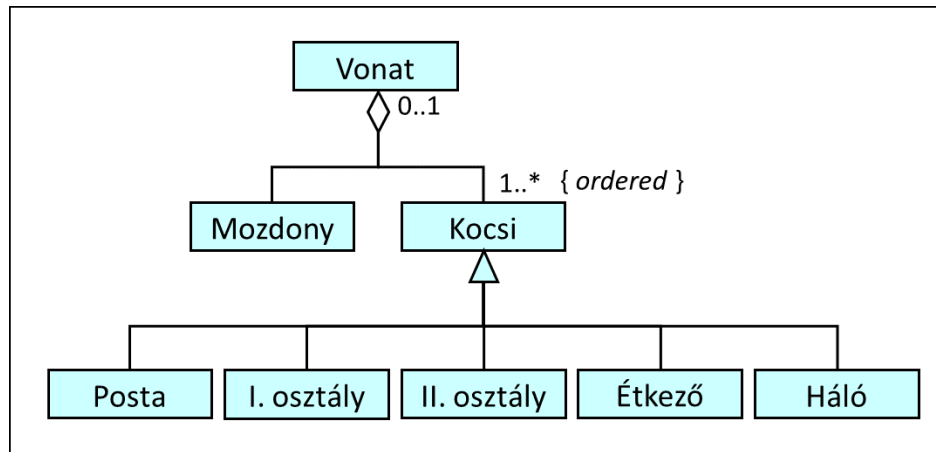
4. A kórházban legalább egy alkalmazott dolgozik, aki lehet orvos, ápoló vagy a személyzethez tartozhat. A kórházban betegek vannak, akiket orvosok kezelnek és ápolók ápolnak. Egy beteget pontosan egy orvos kezel, és tetszőleges számú ápoló ápol. Egy orvos tetszőleges számú beteget kezelhet, egy ápoló tetszőleges számú beteget ápolhat.



Vegyük figyelembe, hogy alkalmazott is lehet beteg!



5. Egy vonatszerelvény egy mozdonyból és legalább egy kocsiból áll. A kocsikat a mozdony után adott sorrend szerint kapcsolják össze. A vonatot különböző típusú kocsikból állíthatják össze. A lehetséges típusok: első osztályú, másodosztályú, posta, étkező, háló.



Egészítsük ki a fenti feladatot az alábbiakkal. Engedjük meg, hogy legyen tehervagon és platóvagon is. Fontos adat, hogy a kocsiknak van hosszúsága is. Egy szerelvény vagy személyszállító (nincs tehervagon, sem platóvagon), vagy teherszállító (csak tehervagon vagy platóvagon van). Egy mozdony, illetve kocsi egy időben csak egy vonathoz tartozhat. Hogyan lehet egy adott szerelvény hosszát kiszámolni?