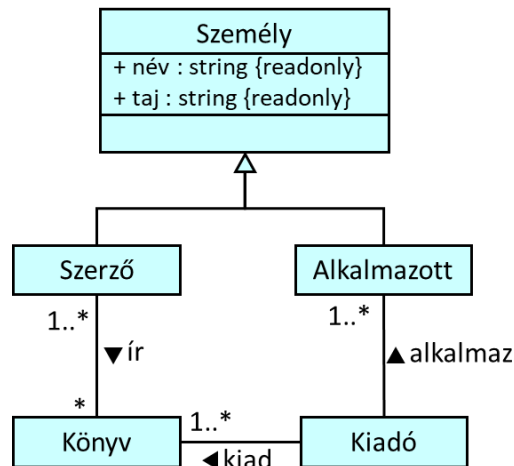


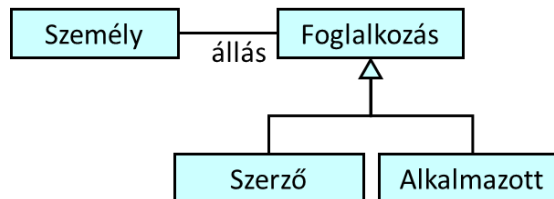
## 8. Származtatás

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. Váloztatni kell a modellen ahhoz, hogy lehessen valaki egyszerre szerző is, és több kiadónak az alkalmazottja is.



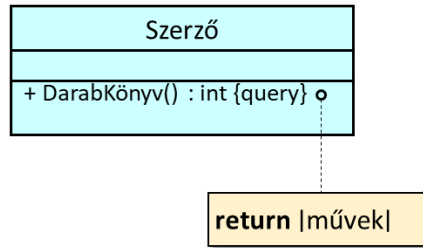
Most pedig oldjuk meg az alábbi feladatokat:

- 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?

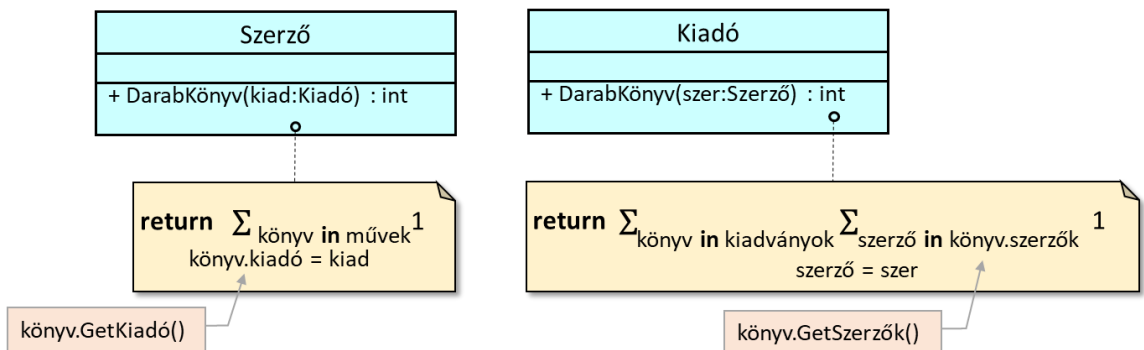
Döntsük el, hogy melyik osztályban helyezzük el metódusként a fenti tevékenységeket? Ehhez vizsgáljuk meg:

- 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)

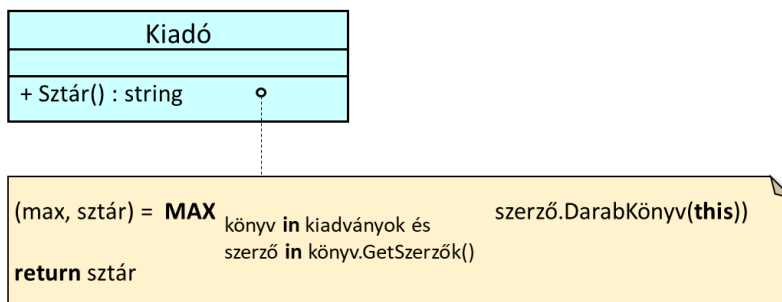
- a) Elég a Szerző osztályban definiálni a kérdést megválaszoló DarabKönyv() metódust feltéve, hogy innen elérjük a szerző műveit. Ehhez vezetjük be a „művek” szerepnevet, amely a Szerző adattagja lesz.



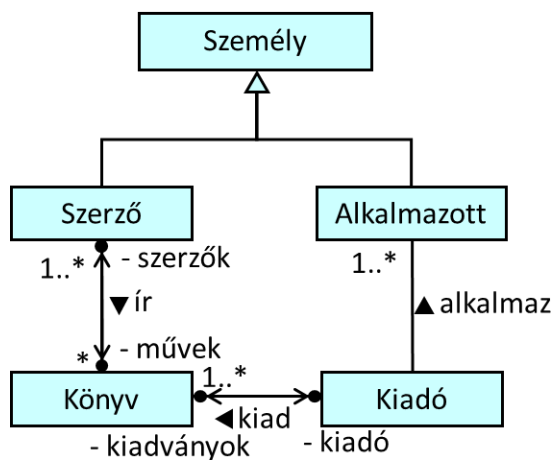
- b) Két megoldás is van: vagy a Szerző, vagy a Kiadó osztályba kell ezt a metódust felvenni. Az első megoldásnál szükség van arra, hogy egy könyv kiadóját lekérdezhessük. Erre szolgál a „kiadó” szerepnév, amelyet a Könyv osztály adattagjaként veszünk fel. Célszerű, hogy ez privát legyen, de ekkor egy getter-re is szükségünk van a kiolvasásához (`getKiadó()`). A második megoldásnál hozzá kell férnünk egy kiadó által kiadott összes könyvhöz, amelyet a „kiadványok” szerepnév biztosít, ami a Kiadó osztály adattagja lesz. Ezen túlmenően egy kiadvány szerzőit is el kell tudnunk érni. Bevezetjük ezért a „szerzők” szerepnevet is, amely szintén a Könyv osztály adattagja lesz, amelynek, mivel privát, egy `getSzerzők()` publikus metódussal biztosítjuk a lekérdezését.



- c) Ez lehet a kiadó módszere, amely egy olyan maximum kiválasztás, amelyhez fel kell sorolni a kiadó által kiadott könyvek szerzőit (egymásba ágyazott felsorolás), hogy az adott kiadónál általuk írt könyvek számát. Észrevehetjük, hogy így ugyanazt a szerzőt többször is érintheti a felsorolás.

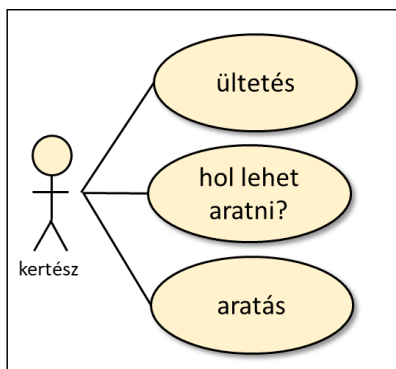


Az eddigiek alapján módosított osztály diagram (a metódusok nélkül):

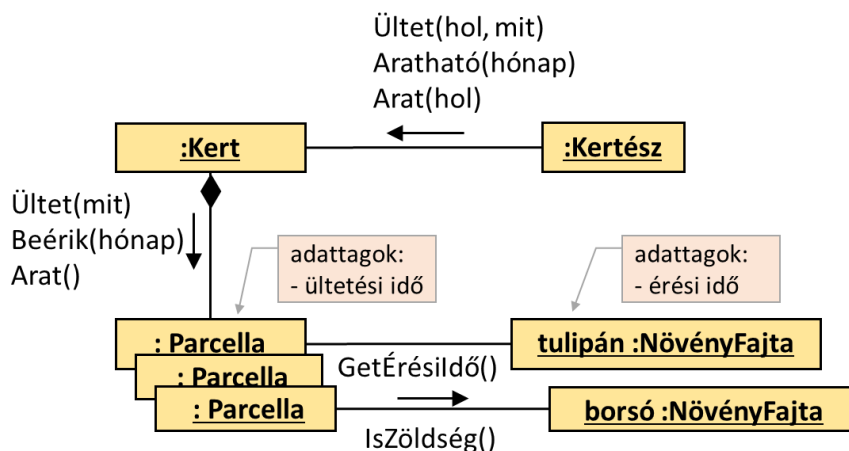


2. Egy kertet egy kertész gondoz. A kert parcellákból áll, minden parcellába egyféle növény ültethető. Az ültetés idejét eltároljuk (hónapban). A növények lehetnek zöldségek, mint burgonya, borsó, paprika; vagy virágok, mint tulipán, szegfű, rózsza. A növényeknek ismerjük az érési idejét (hónapban).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

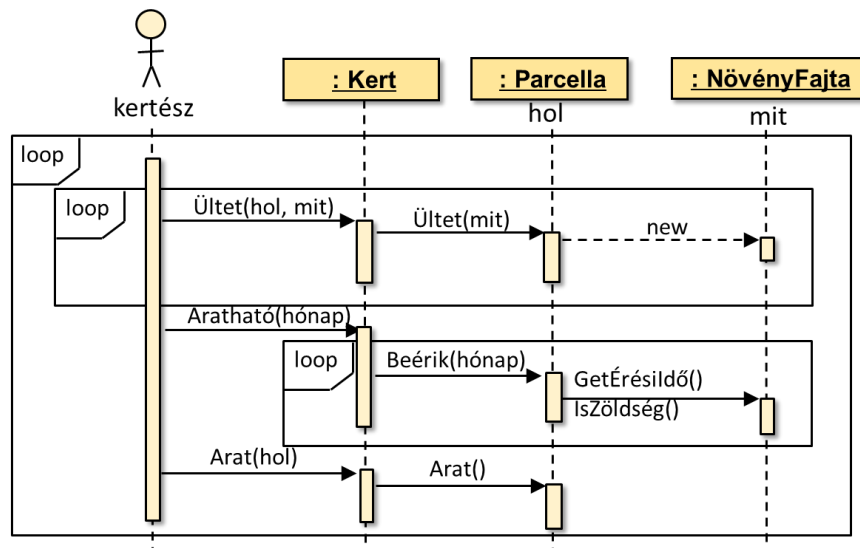
A kertész háromféle tevékenységet végezhet: növényt ültet egy adott parcellába, megkérdezi, hogy mely parcellákban értek meg zöldségek, és learatja a termését egy adott parcellának.



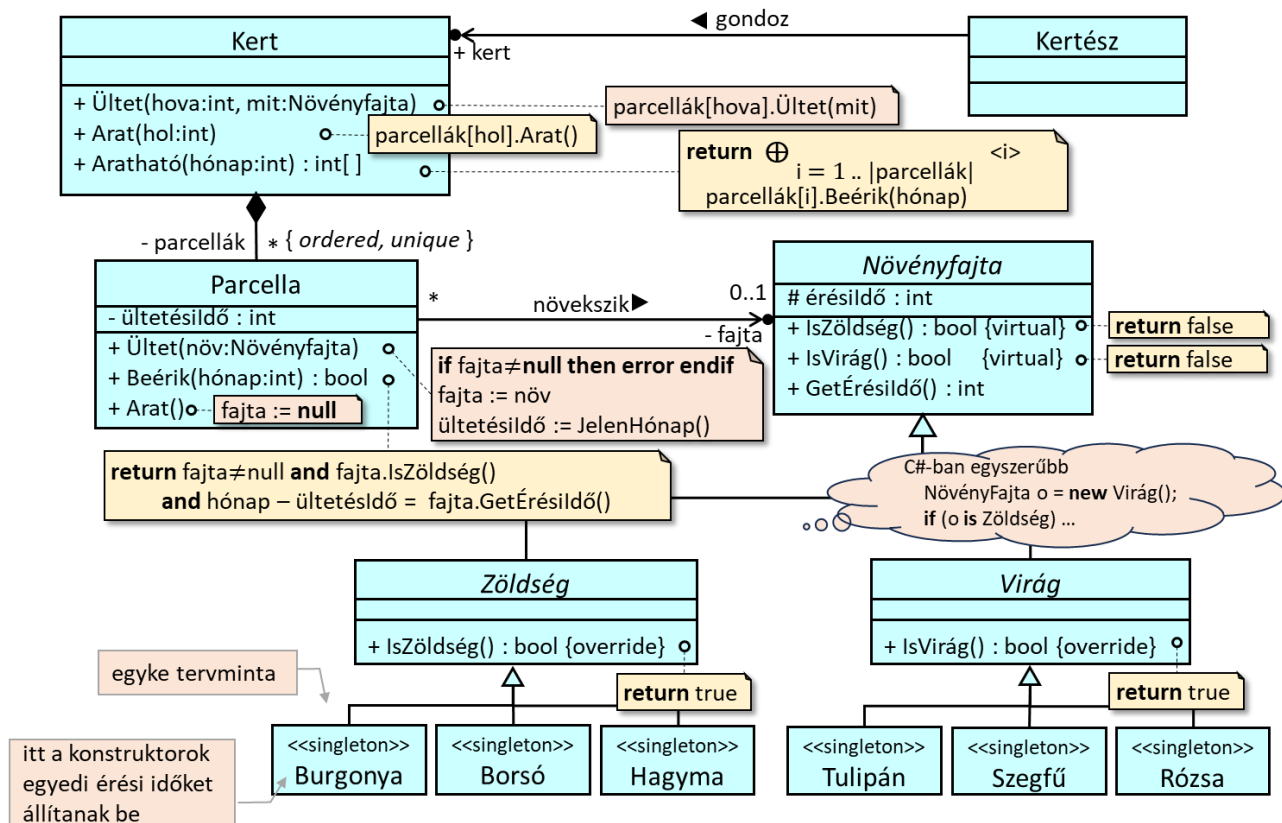
A feladat modelljében négyféle objektummal kell számolni: kertész, kert, parcella, növény. A kerthez több parcella tartozhat.



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 zöldség (Beérik()). Ez utóbbinál szükség lesz arra, hogy megkapjuk egy növény érési idejét (GetÉrésIdő()), és azt a tulajdonságát, hogy zöldség-e (IsZöldség()).

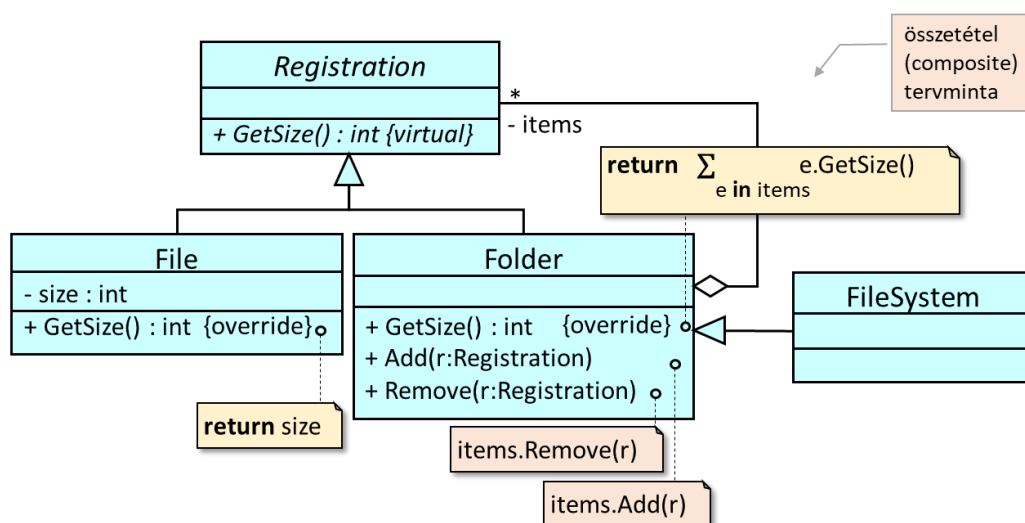


Az elemzés után adjuk meg az osztály diagramot. Az Aratható() metódus paraméterként kapja meg azt a hónapot, amelyről meg akarjuk tudni, hogy mely parcellákban érik be akkor zöldség. A növényfajtaikat származtatással definiáljuk. Egy konkrét növényfajta objektum típusának megállapításához kihasználjuk a dinamikus altípusos polimorfizmust, és alkalmazzuk az egyke tervmintát is.



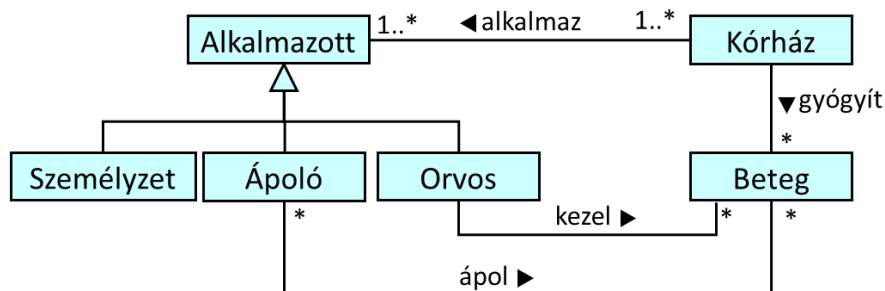
3. Egy számítógépes fájlrendszerben a fájlokat könyvtárakba szervezzük. Fájlok tartozhatnak közvetlenül is a fájlrendszerhez, de annak valamelyik könyvtárban is elhelyezkedhetnek. Minden könyvtár tetszőleges számú fájlt vagy könyvtárat tartalmazhat. Egy fájról ismert, hogy mennyi tárhelyet foglal el; egy könyvtár tárfoglalása a benne levő fájlok és könyvtárak tárfoglalásainak összege. Mennyi tárhelyet 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.

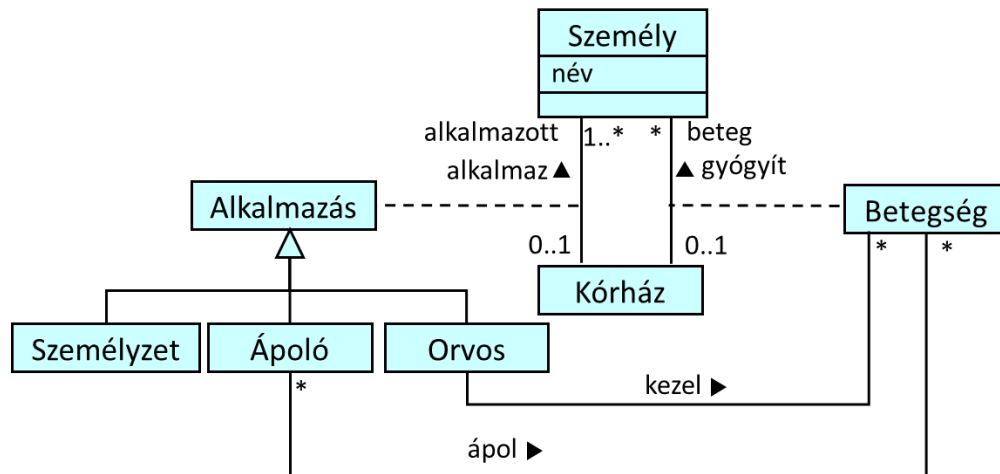


Ezt a szerkezetet ügyesen használja ki a `GetSize()` metódus, amely logikai értelemben egy rekurzívan hívható függvénynek tűnhet, habár nyelvi szempontból itt nincs rekurzív hívás, hiszen itt az egyik objektum `GetSize()` metódusa egy másik objektum `GetSize()` metódusát hívja.

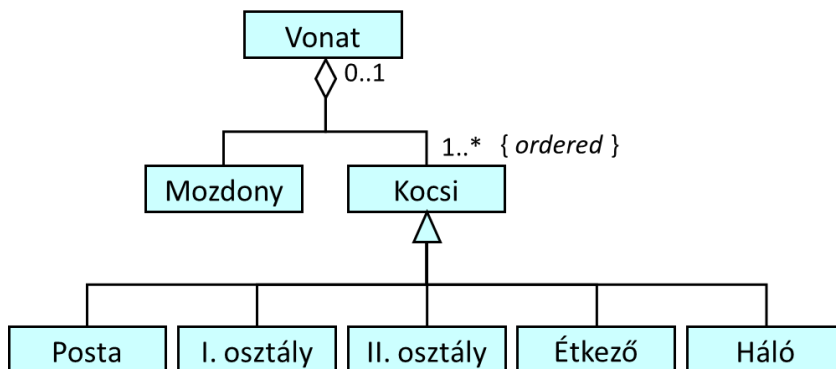
4. A kórházban legalább egy alkalmazott dolgozik, aki lehet orvos, ápoló vagy a személyzethez tartozhat. A kórházban betegek vannak, akiket orvosok kezelnek és ápolók ápolnak. Egy beteget pontosan egy orvos kezel, és tetszőleges számú ápoló ápol. Egy orvos tetszőleges számú beteget kezelhet, egy ápoló 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 kocsiból állíthatják össze. A lehetséges típusok: első osztályú, másodosztályú, posta, étkező, háló.



A feladat kiegészíthető az alábbi módosításokkal. Egy mozdony, illetve kocsi egy időben csak egy vonathoz tartozhat. Megengedjük tehervagonok és platóvagonok használatát, nyilvántartjuk egy kocsinak a hosszúságát, bevezetjük a személyszállító vonat (sem tehervagont, sem platóvagont nem tartalmazó szerelvény) és a teherszállító vonat (csak tehervagonból vagy platóvagonból álló szerelvény) fogalmait. Egy szerelvény vagy személyszállító, vagy teherszállító lehet. Hogyan számolható ki egy adott szerelvény hossza?