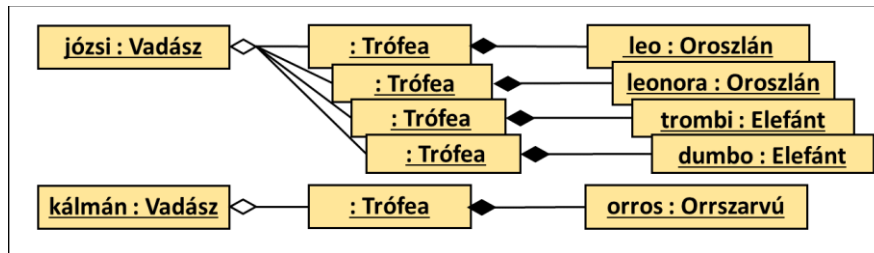


1. Egy vadász (név és az életkor) számos trófeát gyűjtött. Egy trófeán az elejtett vad fajtáját (elefánt, orrszarvú, oroszlán), az elejtés helyét és dátumát, az elejtett vad tömegét (kg-ban), a vad nemét, valamint egy különleges adatot: elefántok esetén agyarainak hosszát külön-külön (cm-ben), orrszarvúaknál a szarvának tömegét értjük.

Adjunk választ egy adott vadásznál az alábbi kérdésekre:

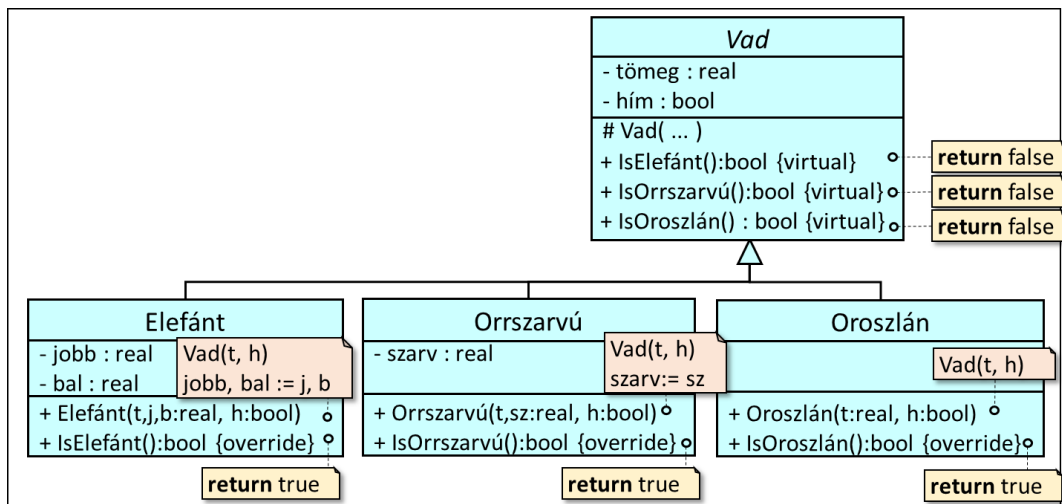
- Hány hímoroszlánt lőtt?
- Melyik a legnagyobb szarv/testtömeg arányú orrszarvú-zsákmánya?
- Van-e olyan elefánt-zsákmánya, ahol az agyarak hossza megegyezik?

Készítsünk először egy objektum diagramot.

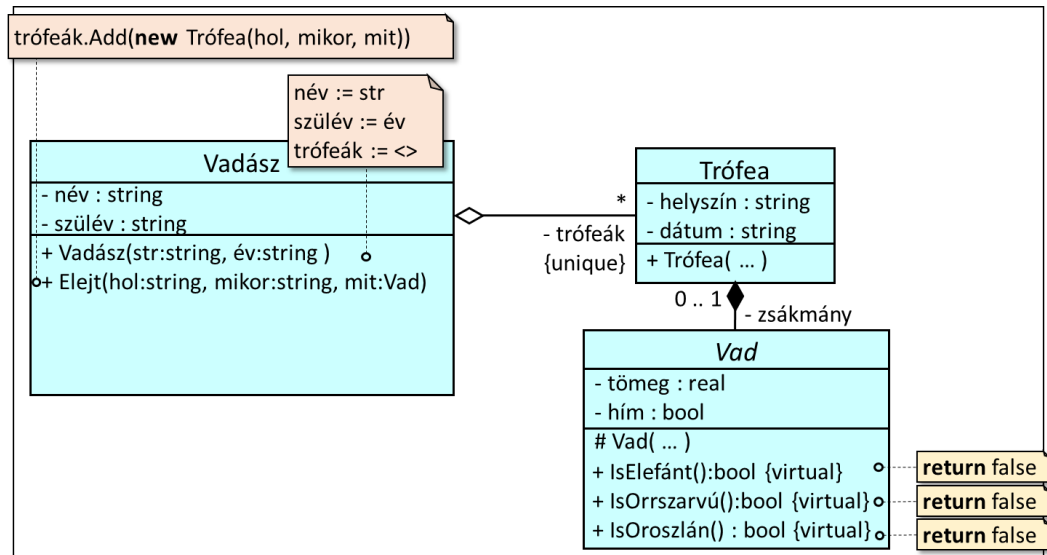


Jelenleg három féle vaddal számolunk. A tömeg, és a vad nemének megadása általános, mindenféle vadra érvényes adatok, ezért az őssztályban jelennek meg; az agyarak viszont az elefántok specifikus adata, a szarv pedig az orrszarvúaké.

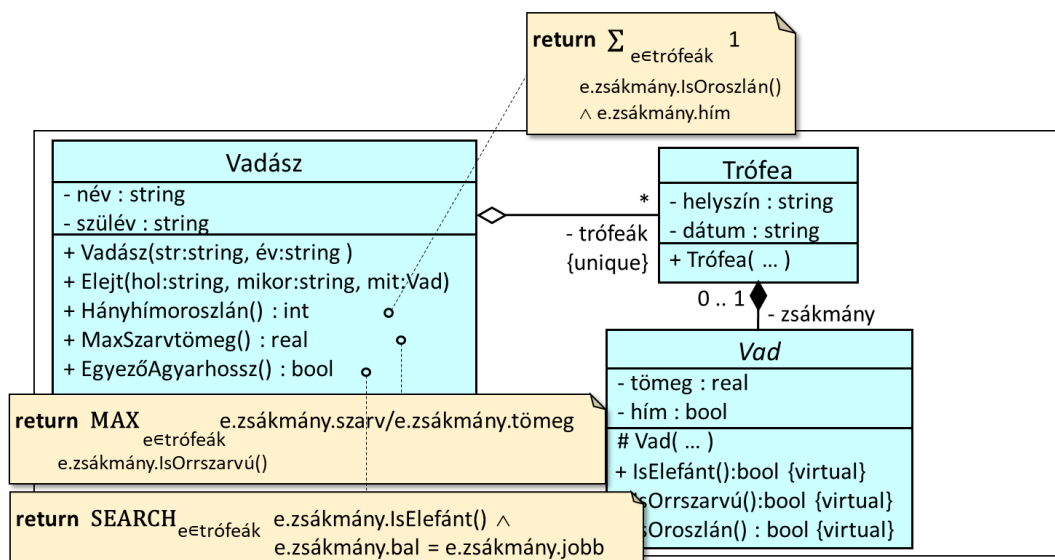
Alkalmazunk az osztálydiagram ábrázolásánál néhány egyszerűsítést. Ha egy modellben egy privát (vagy védett) adattagra az osztályán kívül hivatkozunk, akkor feltételezzük, hogy majd a megvalósítás során egy publikus getter is készül, amit itt a modellben nem kell még jelölnünk. Egy konstruktor paraméterlistájában használhatjuk a (...) jelölést, ha ott fel kellene sorolni az összes adattag számára megadott inicializáló paramétereket. Ilyenkor a konstruktor törzse is elhagyható. Például a Vad(...) jelöli a Vad(tömeg:real, hím:bool) { **this**.tömeg:=tömeg; **this**.hím := hím } definíciót.



A vadász birtokolja a trófeáit, egy trófea egy vadat tartalmaz, de vannak olyan vadak, amelyek nem részei egyetlen trófeának sem. A Trófea és Vad közti kapcsolatokat a Trófea konstruktora hozza létre, a Vadász és Trófea közti kapcsolat létrehozásáért az Elejt(mit:Vad, hol:string, mikor:string) metódus felel. Ez utóbbi példányosítja a trófeát (helyszín, dátum, vad), és elhelyezi annak hivatkozását a trófeák gyűjteménybe. A Vadász konstruktora a név és kor adatágok kitöltéséért felel.



A feladat kérdései mind egy adott vadászra vonatkoznak, ezért az azokra válaszoló metódusokat a Vadász osztályban vezetjük be. Mindhárom kérdést olyan algoritmus mintára vezethetjük vissza, ahol az adott vadász trófeáit kell felsorolni. Az első kérdést egy számlálás, a másodikat egy feltételes maximum keresés, a harmadikat egy lineáris keresés oldja meg.

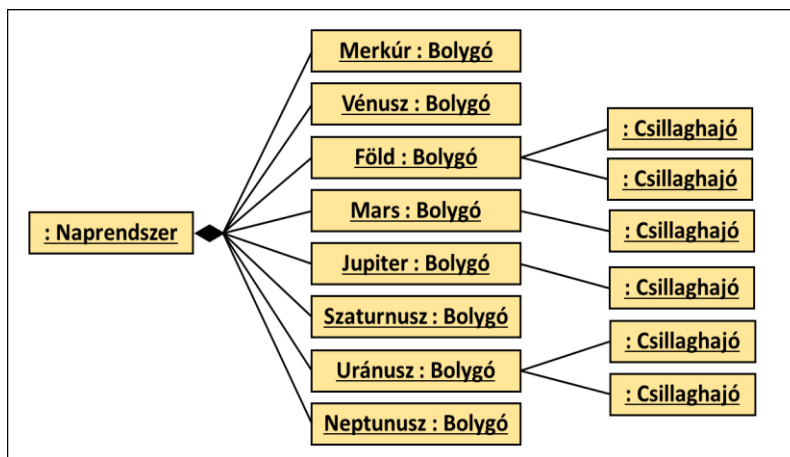


2. A bukott légiók útban vannak a Birodalom központi naprendszere felé Horus Lupercal vezetésével, hogy megdöntsék a Császár hatalmát. Ahogy az áruló sereg közeleg, a birodalomhoz hűséges bolygók egymás után borulnak sötétségbe a naprendszer körül. Az utolsó reményt azok a hűséges légiók jelentik, amelyek a naprendszer bolygóinál gyülekeznek.

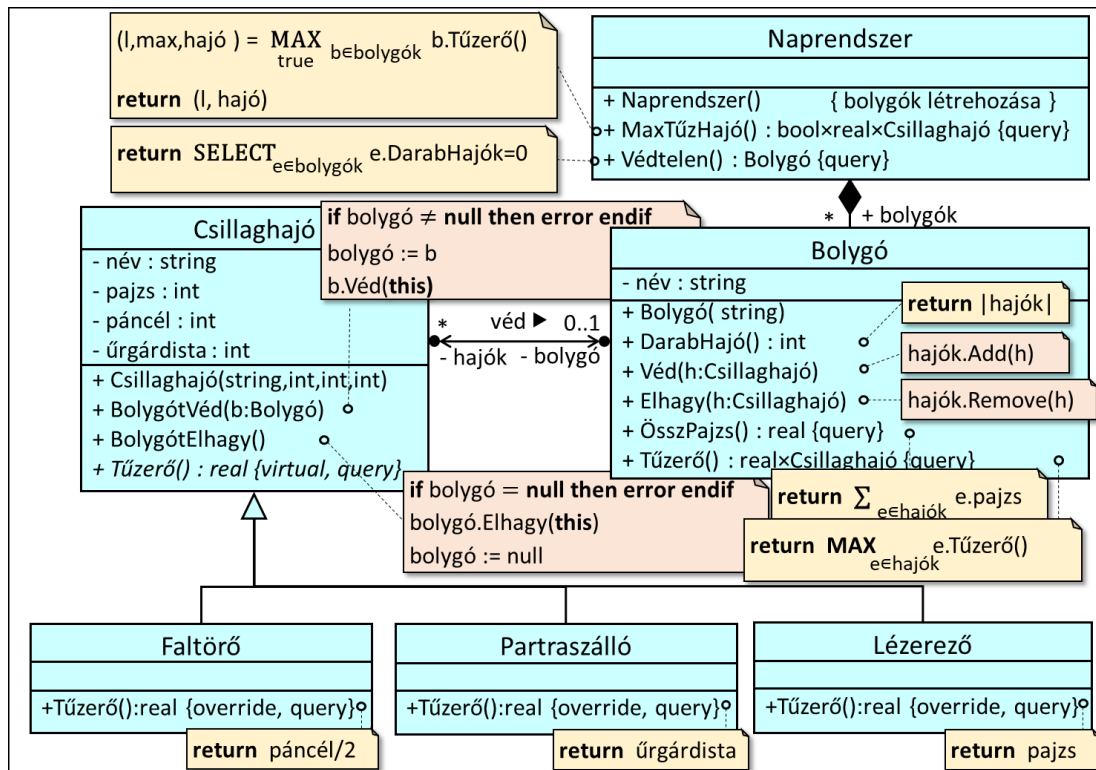
A csillaghajóknak van neve, tudjuk, melyik bolygót védik, ismert a páncélozottságuk (egész szám), pajzserősségük (egész szám), és hogy hány űrgárdista teljesít szolgálatot rajtuk.

Három féle csillaghajó van: faltörő, partraszálló, és lézerező. Ezek tűzerejét különbözőképpen lehet meghatározni. A faltörő tűzereje a páncélozottságának a fele. A partraszállóból az űrgárdisták torpedókból ülvé jutnak át az ellenséges csillaghajók fedélzetére, így a tűzereje az űrgárdisták száma. A lézerező a pajzsokról irányítja át az energiát a lézerfegyvereibe, a tűzereje így megegyezik a pajzserejével.

- Indítunk egy megelőző csapást, amit a legnagyobb tűzerejű csillaghajó vezet majd. Keressük meg ezt a csillaghajót!
- Tudomásunkra jutott, hogy vannak bolygók, amit védtelenül hagytunk, ahol nem állomásozik csillaghajó. Listázzuk ki ezeket a bolygókat!
- Szeretnénk megtudni, hogy egy adott bolygónál állomásozó csillaghajóknak mennyi az összes pajzsereje!



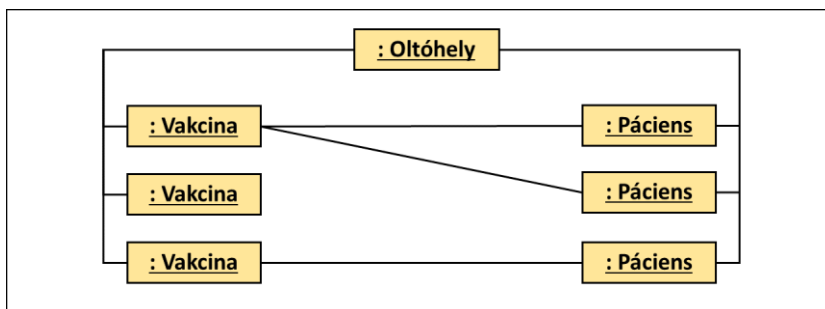
Az adattagok általában „readonly” tulajdonságúak (nem publikusak, de olvashatók). Ebben a feladatban azonban érzékeny (katonailag titkos) adatok is vannak, amelyek olvashatóságát is tiltani kell (páncél, űrgárdista). A Bolygó osztály Véd() és Elhagy() metódusa korlátozott lehetőséget biztosít egy bolygót védő hajók gyűjteményének módosítására. Szükség lesz egy bolygót védő csillaghajók számának lekérdezésére is.



3. Ha egy állampolgár oltakozni akar a COVID-19 ellen, akkor először regisztráltatnia kell magát egy oltóhelyen, majd ott egy adott időpontban felvenni az oltást, feltéve, hogy a páciens által választott oltóanyag rendelkezésre áll. A páciens többféle vakcinával többször is oltathatja magát. Az oltóhelynek ismert a helyszíne, továbbá az, hogy melyik fajta oltóanyagból (ez lehet: pfizer, modena, astrazeneca de később bővíülhet még a lajstrom) hány felhasználható adag van, és egy adott vakcinánál hány nappal később lehet beadni a második oltást. Nyilvántartjuk, hogy a beoltott páciensek mikor milyen vakcinát kaptak.

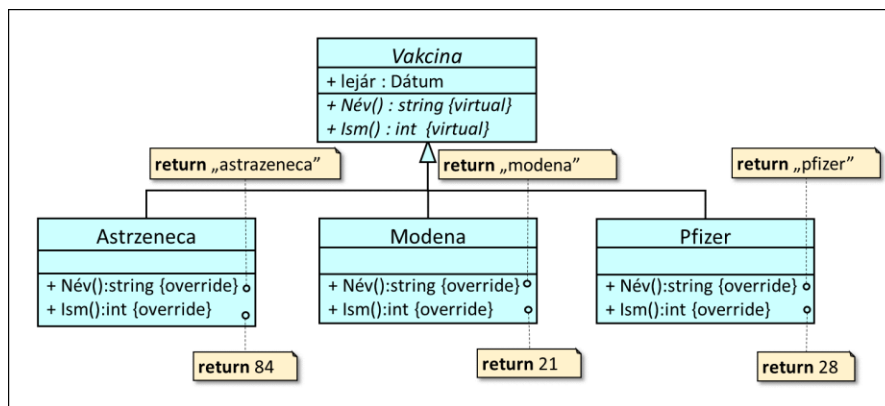
Tegyük lehetővé, hogy

- egy páciens regisztrálhasson az oltóhelyen
- az oltóhely bevételezhessen vakcinát
- rögzíthető legyen az, amikor egy regisztrált páciens egy általa választott vakcinával beoltatja magát
- lekérdezhető legyen, hány olyan páciens van, aki a második oltást is megkapta.

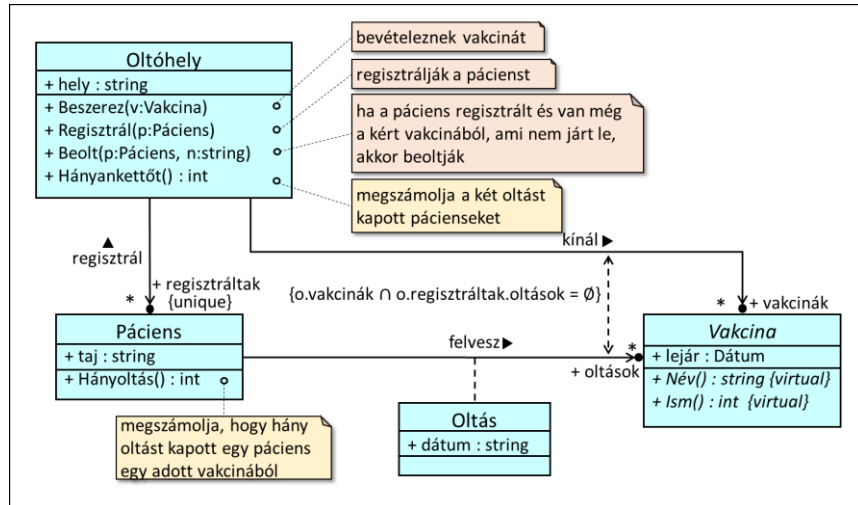


Az objektum diagram pontatlan: egy vakcina nem lehet egyszerre egy oltóhelyen kínálatában is, és az oltóhelyen már felvett vakcina is. Ezt majd az osztálydiagramban jeleznünk kell.

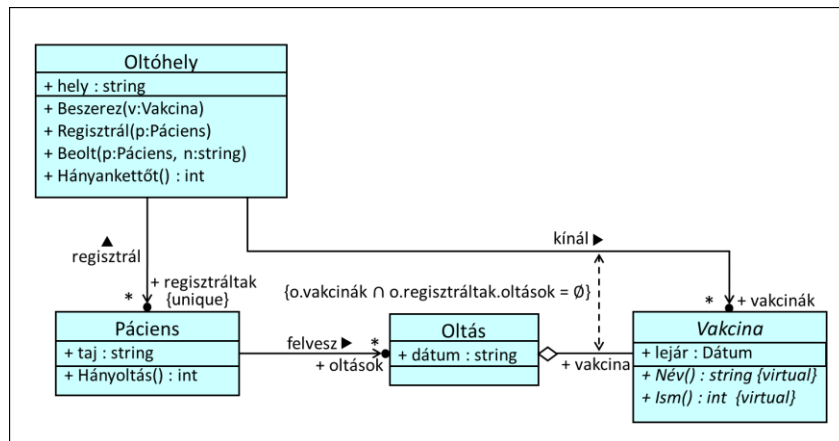
Az oltóanyag fajtákat származtatással definiáljuk.



A modell gerincét az Oltóhely, Vakcina, Páciens osztályok kötötti kapcsolatrendszer adja. Az oltások a páciensek és a felvett vakcinák közötti kapcsolatban jelennek meg: ez tehát egy asszociációs osztály.



Elimináljuk az asszociációs osztályt.



Végül készítsük el a metódusok törzsét.

