

OEP

12. táblás gyakorlat

Feladatok

Vadászat

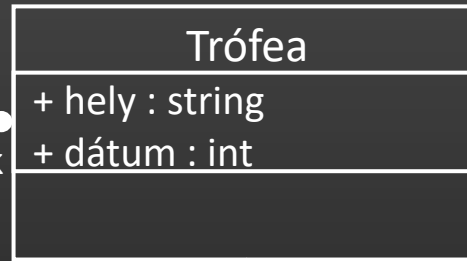
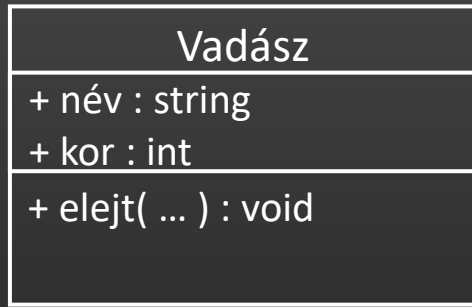
Vidámpark

Horgászverseny

Rally

Obszervatórium

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), valamint egy különleges adatot: elefántok esetén agyarainak hossza külön-külön (cm-ben), orrszarvúaknál az orrszarvú súlya, oroszlánoknál a vad neme (hím vagy nőstény) értjük.

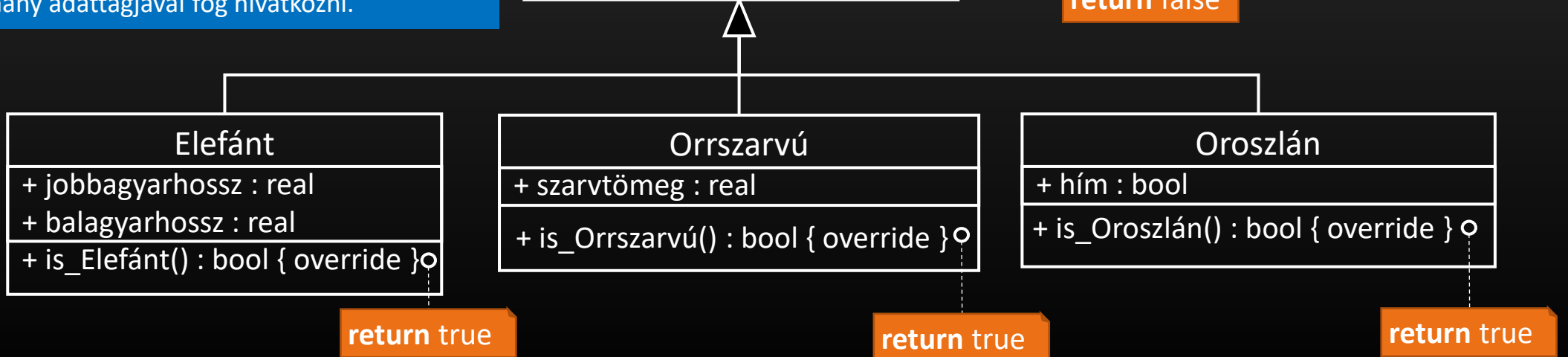


Feladat:
Hány hím oroszlánt lőtt a vadász?

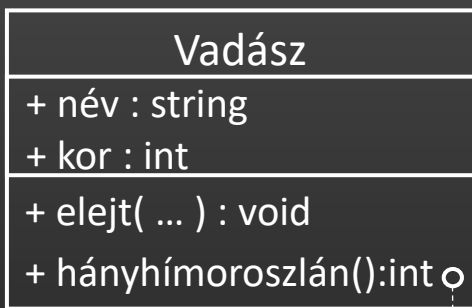
Az `elejt()` metódus a kapcsolatok felépítéséért felel: létrehoz egy trófát és beleteszi annak hivatkozását a trófeák gyűjteménybe. Egy trófea létrehozását olyan konstruktor támogatja, amely átveszi az `elejt()` paramétereit (elejtett vad fajtáját, tulajdonságait, elejtés helyét és idejét), létrehoz egy Vad típusú objektumot, amelyre a zsákmány adattagjával fog hivatkozni.



return false
return false
return false



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), valamint egy különleges adatot: elefántok esetén agyarainak hossza külön-külön (cm-ben), orrszarvúaknál az orrszarvú súlya, oroszlánoknál a vad neme (hím vagy nőstény) értjük.

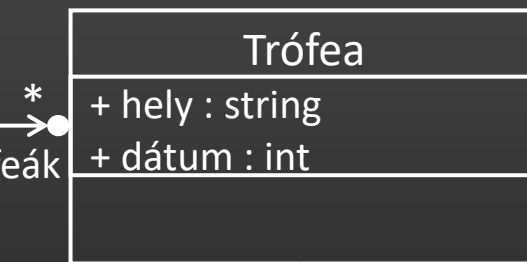


return $\sum_{e \in \text{trófeák}} 1$
 $e.\text{zsákmány}.\text{is_Oroszlán}()$
 $\wedge e.\text{zsákmány}.\text{hím}$

Hány hímoroszlánt lőtt egy vadász?

```

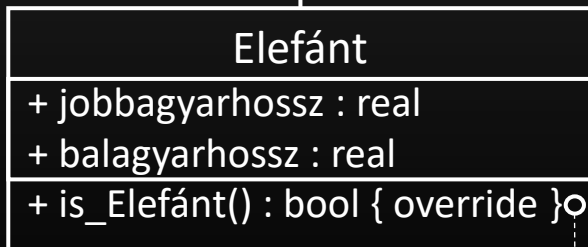
int Hunter::countMaleLions() const{
    int c = 0;
    for( Trophy* e : _trophies ){
        if( e->_animal->is_Lion() &&
            ((Lion*)(e->_animal))->_male ) ++c;
    }
    return c;
}
  
```



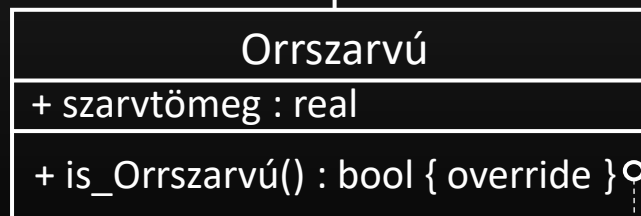
return false

return false

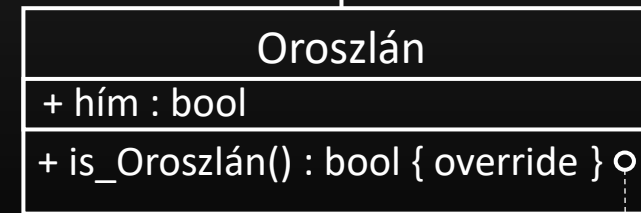
return false



return true



return true



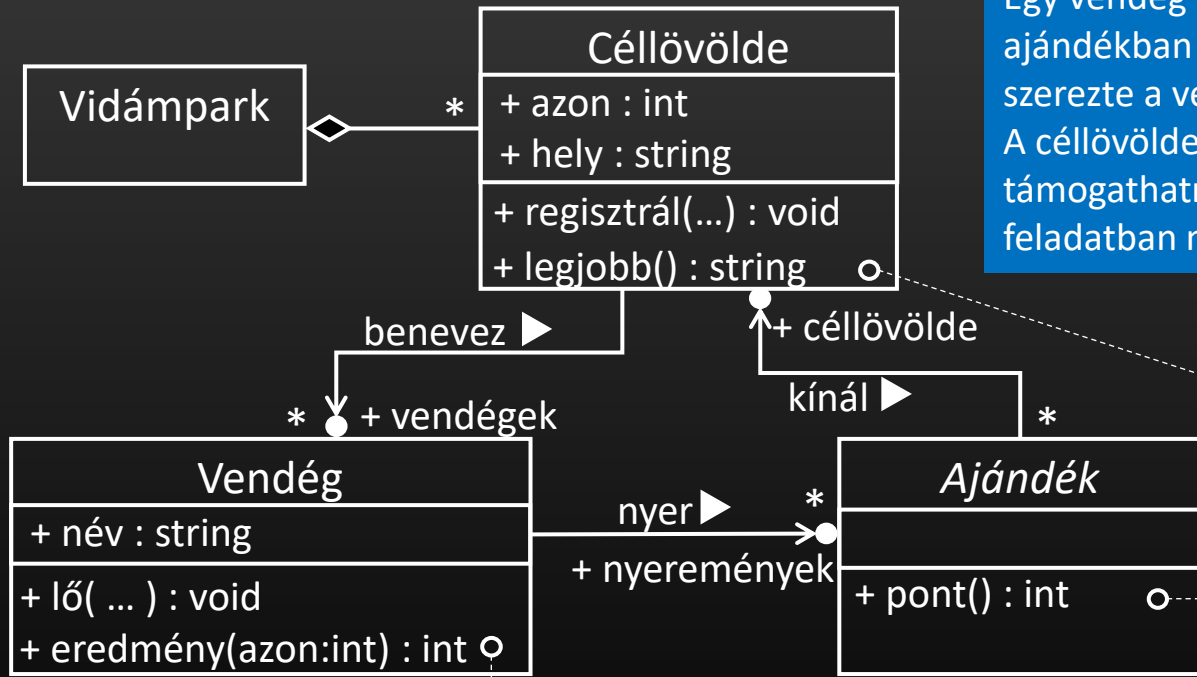
return true

- Egy vidámpark több céllövöldét is üzemeltet, ahol a vendégek szerencsét próbálhatnak. Egy céllövöldében több vendég is lőhet akár többször is.
- A céllövöldék egyedi azonosítóval rendelkeznek, és ismert a helyszínük.
- A vendégek is kapnak egy egyedi azonosítót, valamint ismert, hogy az egyes céllövöldékben melyek voltak a nyereményeik. Egy nyeremény leírja, hogy egy adott azonosítójú céllövöldében elnyert ajándéknak mi a típusa és a mérete (S, M, L, XL). Az ajándék típusa lehet plüss állat, műanyag figura, vagy labda.
- Az S méret 1 pontot, az M 2 pontot, az L 3 pontot, az XL 4 pontot ér. Az ajándék pontszáma plüss állat esetén 3 pont, műanyag figura esetén 2 pont, labda esetén 1 pont.
- Egy nyeremény értékét az ajándék típusából és méretéből származó pontok szorzata adja.

Ki a legjobb céllövő egy adott céllövöldében?

A regisztrál() metódus a „benevez” kapcsolatok felépítését végzi. Már létező objektumokat köt össze úgy, hogy a vendég hivatkozását beteszi az adott céllövölde vendégek gyűjteményébe.

A lő() metódus a „nyer” kapcsolatok felépítéséért felel: beleteszi a lelőtt ajándék hivatkozását a az adott vendég nyeremények gyűjteményébe.

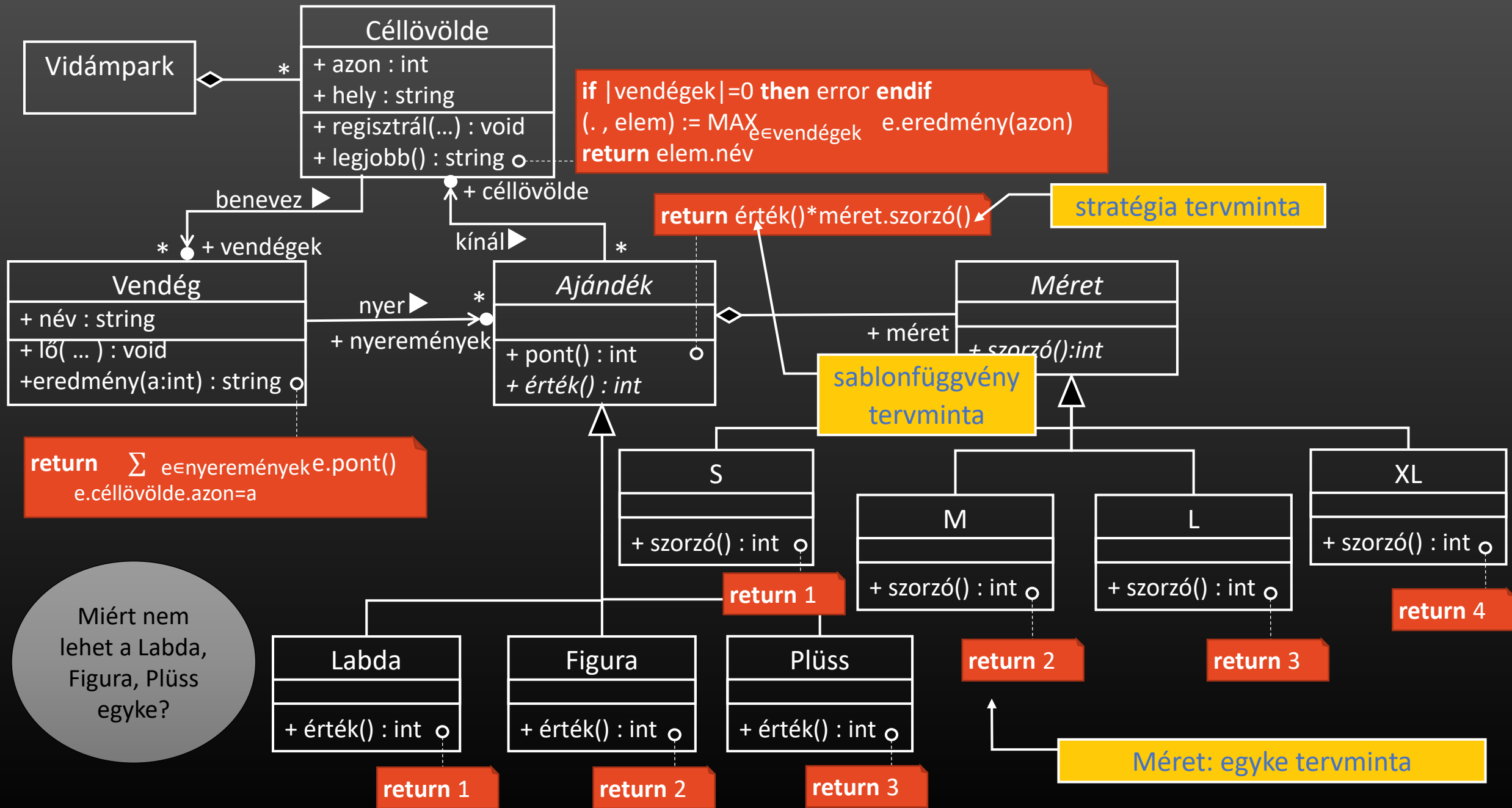


Egy vendég több céllövöldében is nyerhet. Az ajándékban tároljuk, hogy melyik céllövöldében szerezte a vendég. A céllövölde kínálja az ajándékait, ezt is támogathatnánk gyors navigációval, de erre most a feladatban nincs szükség.

maximum kiválasztás a céllövölde vendégeinek eredménye között

nyeremény pontszáma

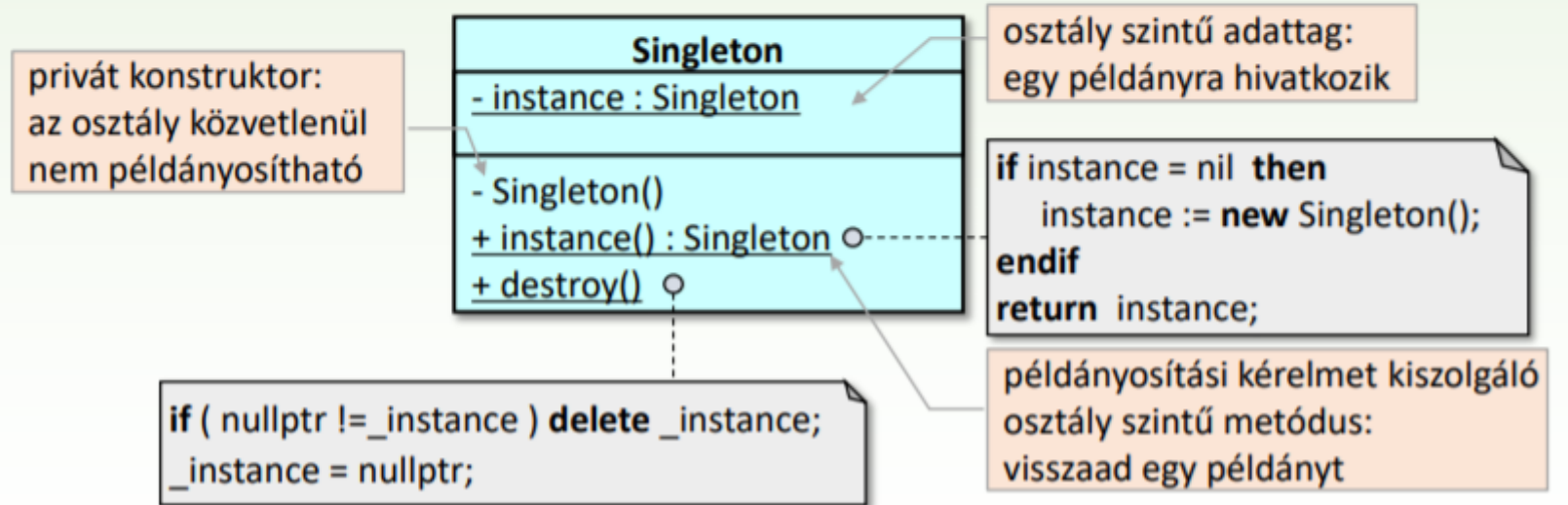
feltételes összegzés az adott céllövölde nyereményeinek pontjaira



8. előadás

Egyke (singleton) tervezési minta

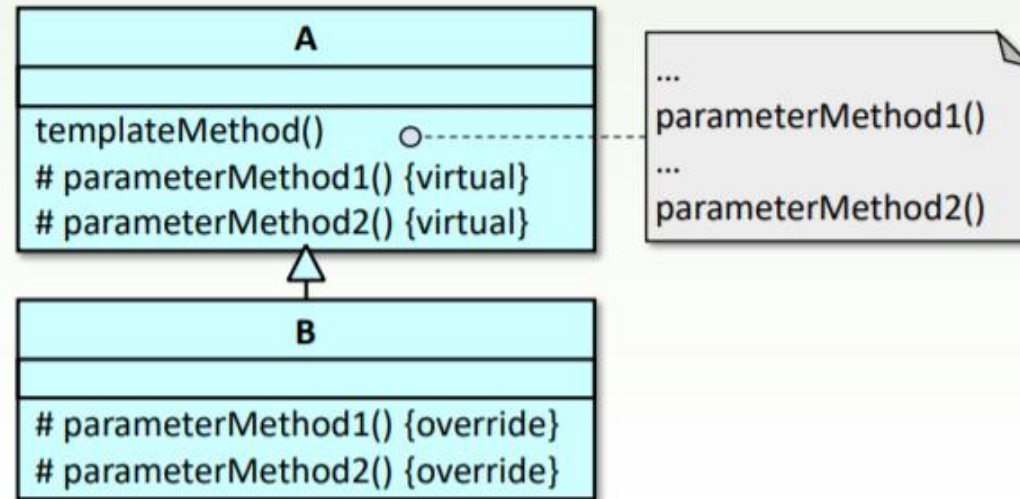
- Egy osztályhoz legfeljebb egy objektumot példányosítsunk függetlenül a példányosítási kérelmek számától.



8. előadás

Sablonfüggvény tervezési minta (template method)

- Egy algoritmust úgy adunk meg egy osztály metódusaként, hogy annak egyes lépéseit az algoritmus szerkezetének változtatása nélkül a futási idejű polimorfizmusra támaszkodva lehessen megváltoztatni.



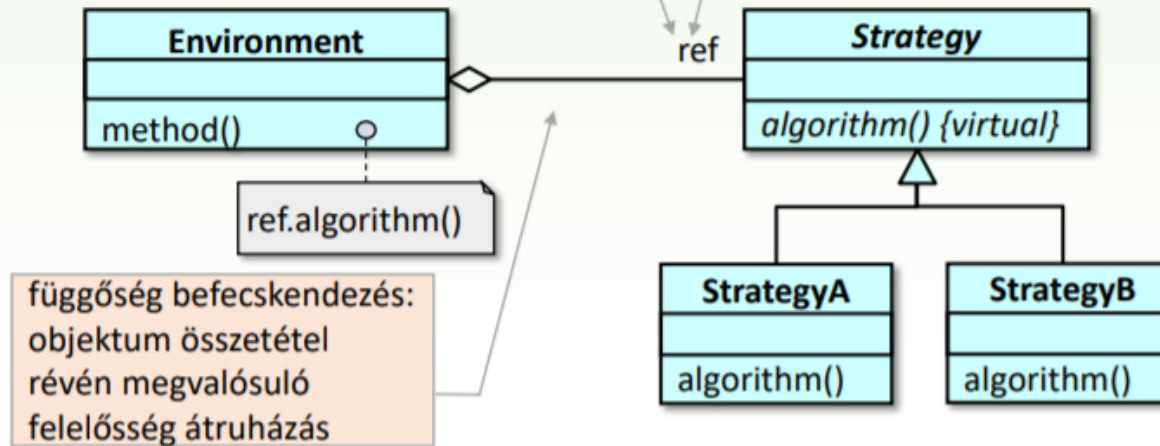
8. előadás

Stratégia (strategy) tervezési minta

- Egy algoritmus-családot definiálunk azért, hogy annak egyik tagját egy másik osztály metódusa használhassa, de hogy melyiket, az nem a felhasználó metódustól függ, hanem attól, hogy metódust tartalmazó osztály egy adattagja az algoritmuscsalád melyik tagját képviseli.

A ref adattag a Strategy bármelyik alosztályának példányára hivatkozhat, és ettől függ a method() működése. Például az Environment konstruktora állíthatja be.

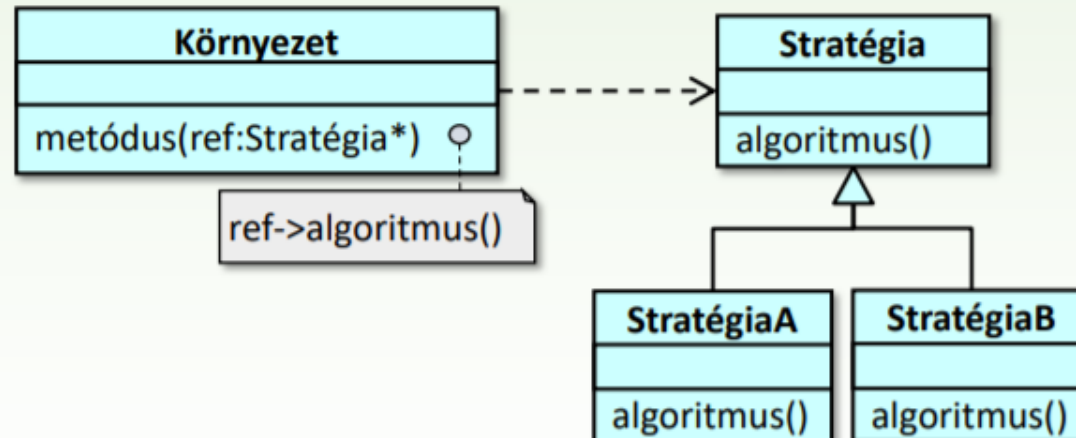
Ennek a szerepének **hivatkozásként** vagy **pointerként** kell az ellentétes oldali objektumban szerepelni, hogy a futásidejű polimorfizmus működjön.



Stratégia tervezési minta 2.

- A stratégia minta gyenge változata: a felhasználó metódus paraméterként kapja meg az algoritmus-család egyik osztályának példányát.

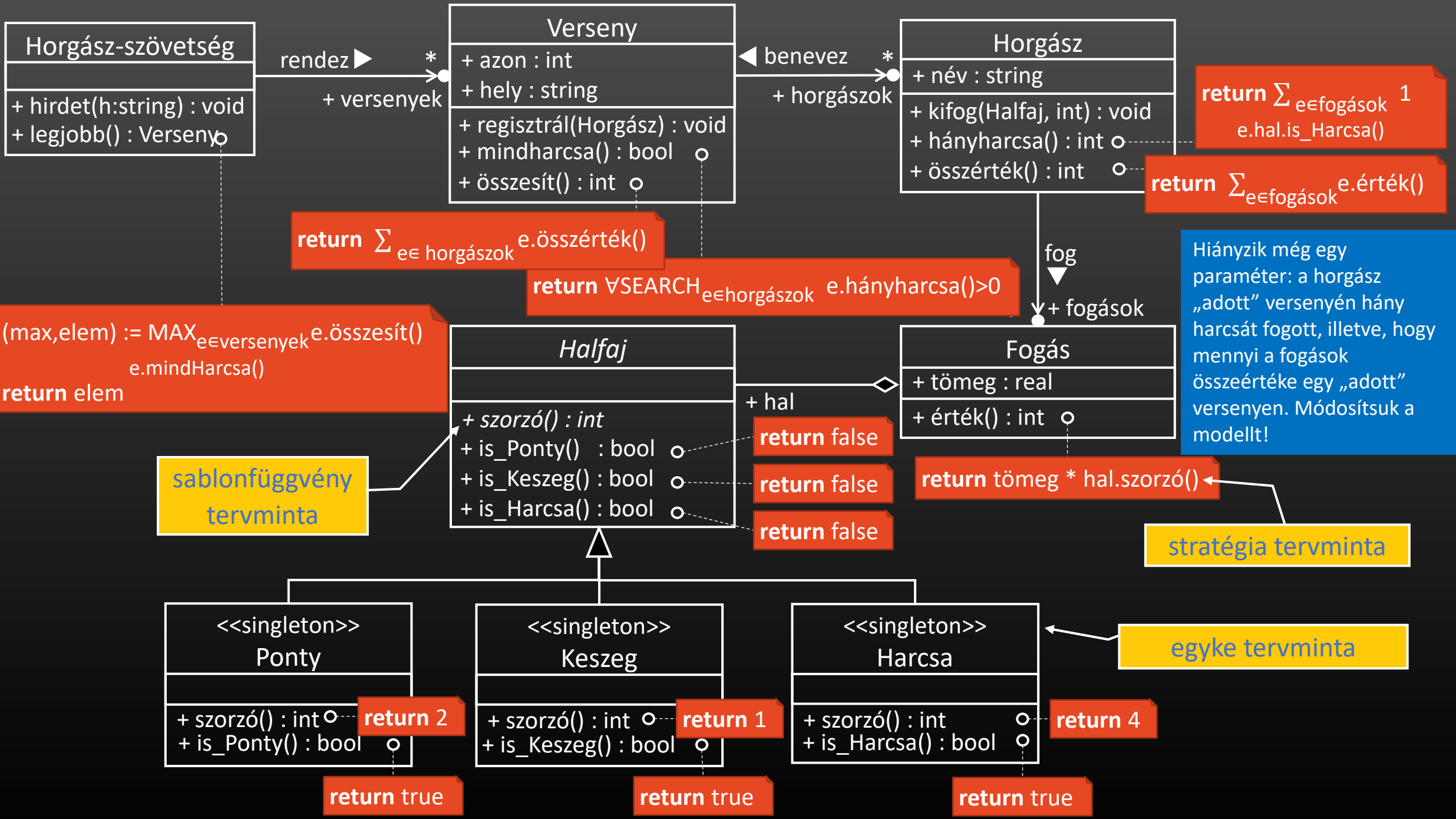
8. előadás

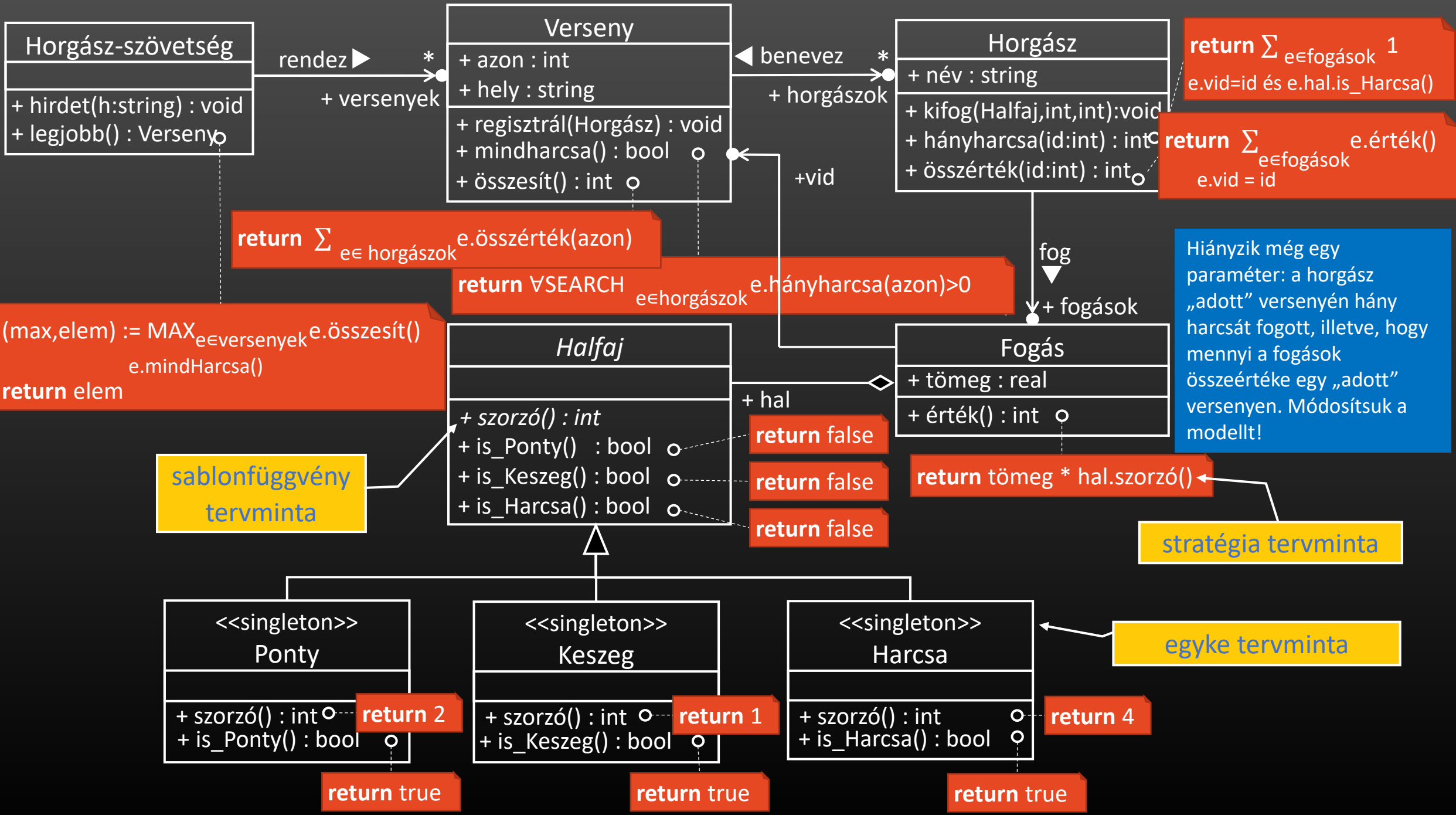


3. Egy horgásztársaság több horgászversenyt rendez, amelyekre horgászok neveznek be; ugyanaz a horgász több versenyen is részt vehet. A versenyeknek van egy azonosítója, és ismert a helyszíne. A horgászoknak ismerjük a nevét, tudjuk, hogy milyen fogásaik voltak az egyes versenyeken. Egy fogás leírja, hogy melyik versenyen fogták, ki volt a horgász, mi a kifogott hal fajtája és a tömege (kg-ban). A halak fajtája lehet ponty, keszeg, vagy harcsa.

- Hogyan határozható meg egy hal értéke, ha ez a hal tömegétől, és a halfajta szorzójától (harcsa:4, ponty:2, keszeg:1) függ?
- Hogyan határozható meg egy horgász hány harcsát fogott egy adott versenyen, illetve hogy mennyi a kifogott halainak értéke?
- Hogyan határozható meg az a legeredményesebb olyan verseny, ahol mindenki fogott harcsát?



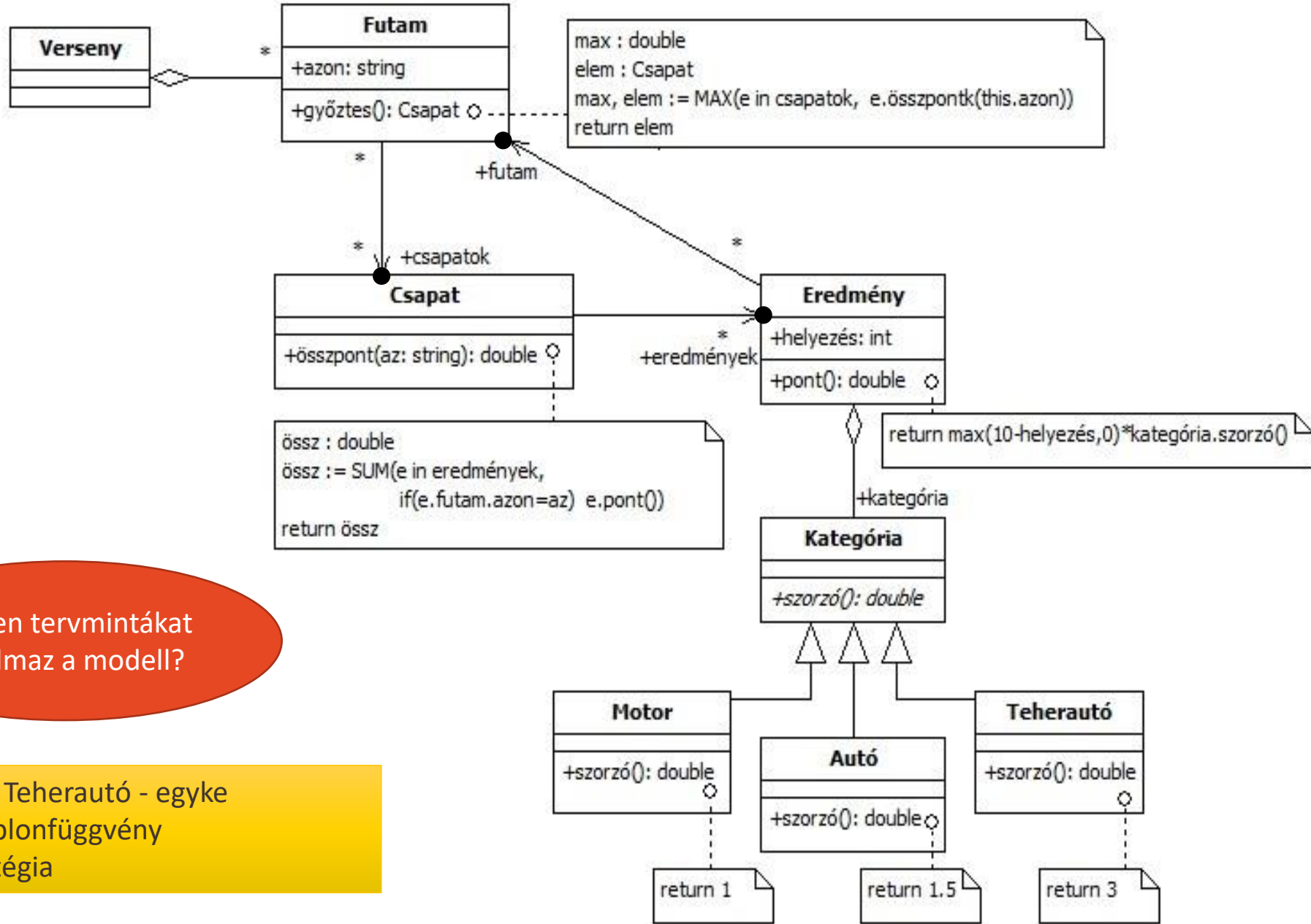




4. Rally

- Egy országos rally autóverseny több futamból áll, a futamokon több csapat indul. Ugyanaz a csapat több futamon is részt vehet. A futamok egyedi azonosítóval rendelkeznek, és ismert a helyszínük.
 - A csapatok több kategóriában versenyezhetnek. A csapatok is kapnak egy egyedi azonosítót, és ismert, hogy az egyes futamokon milyen eredményt értek el.
 - Egy eredmény leírja, hogy egy adott azonosítójú futamon az melyik kategóriában hányadik helyezés volt.
 - A kategória lehet sportautó, teherautó, motor.
 - Egy eredmény értéke függ az elért helyezéstől, valamint egy módosító tényezőtől, amivel a helyezést kell megszorozni. Ez a módosító tényező a motoroknál 1, sportautóknál 1.5, teherautóknál 3
-
- Ki nyerte az adott futamot?

Rally



Milyen tervmintákat alkalmaz a modell?

Motor, Autó, Teherautó - egyke szorzó() - sablonfüggvény
pont() - stratégia

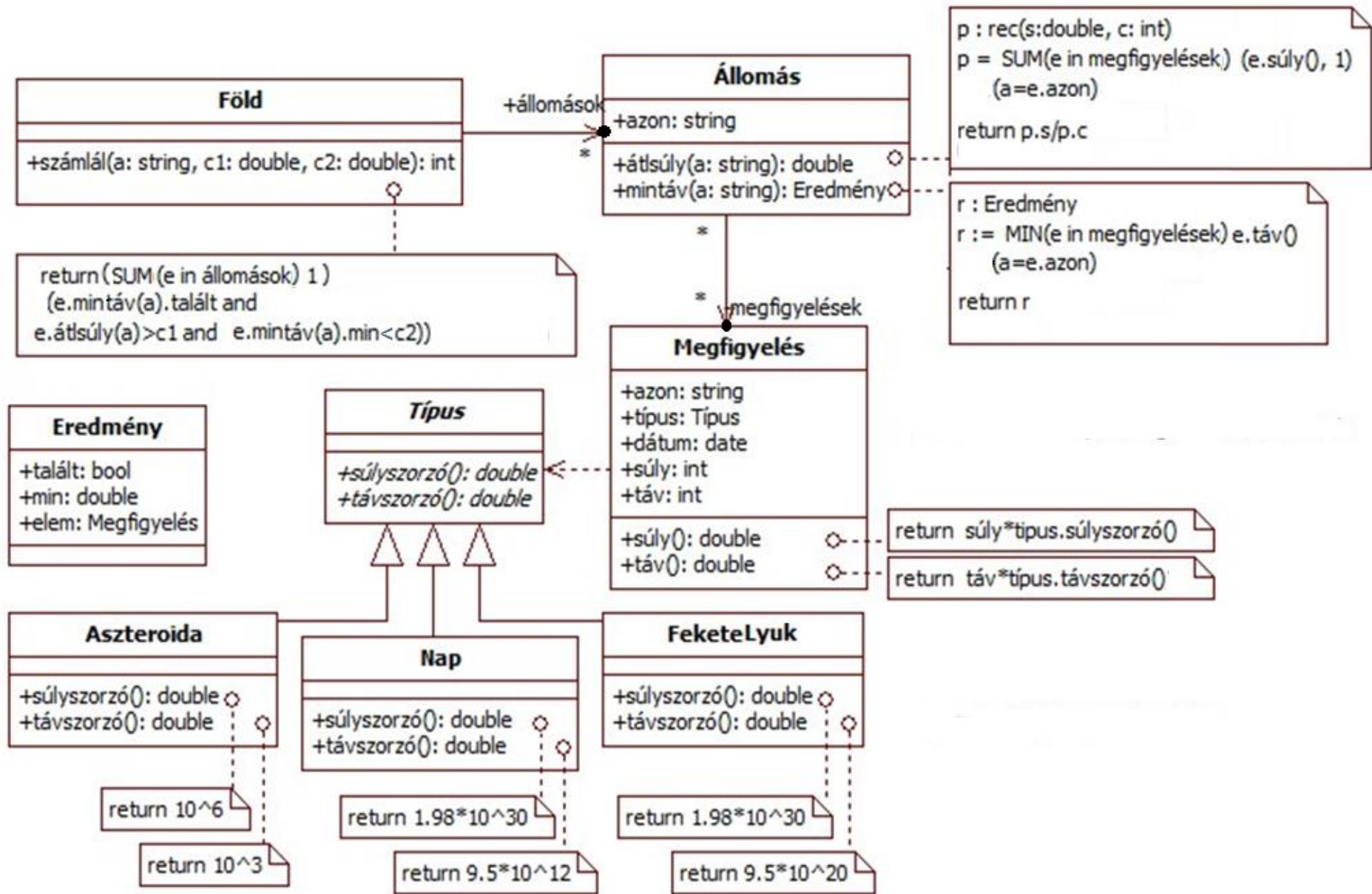
5. Obszervatórium

- Egy A Föld megfigyelőállomásai (obszervatóriumok, műholdak) rendszeresen kémlelik a világűrt, és feljegyzik az általuk észlelt űrbeli objektumok adatait, amely objektumok lehetnek:
 - aszteroidák,
 - fekete lyukak,
 - vagy napok.
- Mindhárom esetben rögzítik az objektum azonosítóját, típusát (aszteroida, fekete lyuk, nap), a megfigyelés dátumát, a becsült tömegét és a Földtől vett távolságát.
 - A tömeget aszteroidák esetében ezer tonnában (10^6 kg),
 - fekete lyukak és napok esetében naptömegben ($1,98 \cdot 10^{30}$ kg) tárolják.
 - A távolságot az aszteroidáknál ezer kilométerben (10^3 km),
 - napoknál fényévben ($9,5 \cdot 10^{12}$ km),
 - fekete lyukaknál százmillió fényévben ($9,5 \cdot 10^{20}$ km) adják meg.
- Ugyanazon objektum több, akár ugyanazon megfigyelőállomás különböző megfigyeléseiben is szerepelhet. Természetesen előfordul, hogy ugyanazon objektum különböző megfigyelései eltérő tömeget és távolságot becsülnek egy adott objektumra.

Számolja meg, hogy egy adott azonosítójú objektumot hány megfigyelőállomás tart veszélyesnek! Egy objektum akkor veszélyes egy állomás szerint, ha az állomáson végzett megfigyelések átlagos tömege nagyobb, mint c_1 kg, és az objektumnak az állomáson mért legközelebbi távolsága kisebb, mint c_2 km. (A c_1 és c_2 kívülről megadott paraméterek.)

Obszervatórium

- Tervminták:
- Egyke:
Aszteroida, Nap,
FeketeLyuk
 - Sablonfüggvény:
súlyszorzó()
távszorzó()
 - Stratégia:
súly()
táv()



3. Beadandó (szimulációs feladat)

- Tervmintákat használó, többféle objektummal működő alkalmazás.
- A feladatok megoldásához több olyan osztályt kell használni, amelyek egy közös ősosztályból származnak és felüldefiniálják az ősosztály virtuális metódusait.
- Ezen osztályok objektumait egy gyűjteménybe kell elhelyezni, majd ezt a gyűjteményt kell bejárni, a benne levő objektumok megfelelő metódusait meghívni.
- Ez a bejárás a futásidejű polimorfizmusra támaszkodik. Használjon legalább két tervezési mintát is.
- A gyűjtemények bejárását továbbra is a tanult programozási tételek alapján végezze.
- A tesztkörnyezet biztosítson egységenkénti (osztály, metódus) tesztelési eseteket is a végfelhasználói teszteseteken kívül.

Látogató (visitor) tervezési minta

- Amikor egy metódus működése attól függ, hogy egy objektum-készlet melyik objektumát kapja meg éppen paraméterként, de nem akarunk ezen készlet számától függő elágazást használni a metódus leírásához.

8. előadás

