

11. OEP Gyakorlat

Horgászverseny

1. Egy horgászszövetség több horgászversenyt is rendez, amelyekre csak a szövetség tagjai nevezhetnek be; ugyanaz a horgász több versenyen is részt vehet.

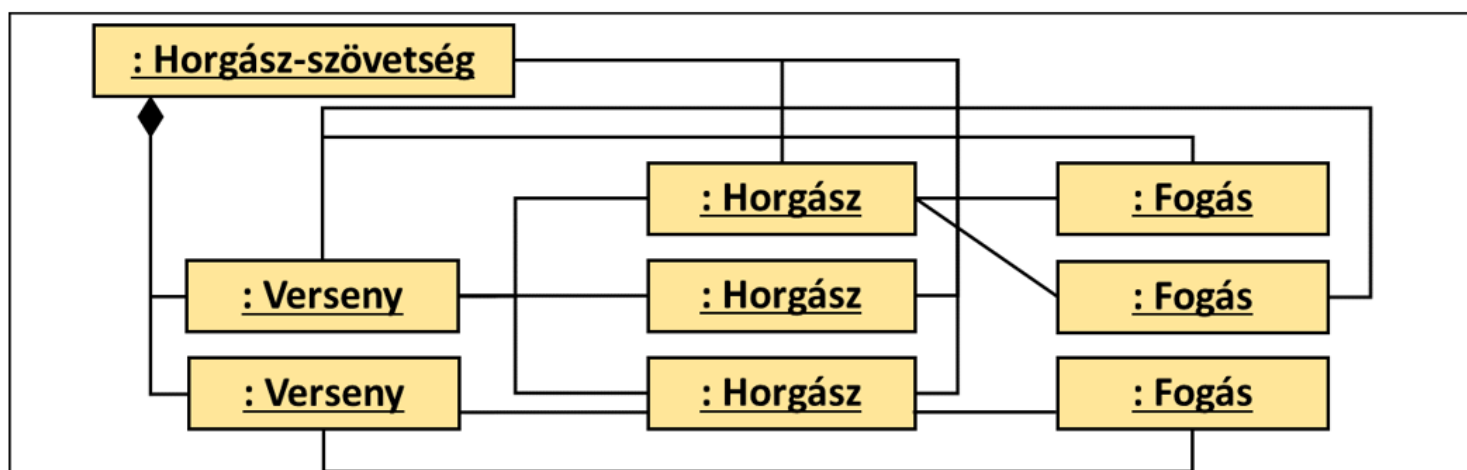
A versenyeknek ismert a helyszíne. A horgászoknak ismerjük a nevét, tudjuk, hogy milyen fogásaik voltak az egyes versenyeken.

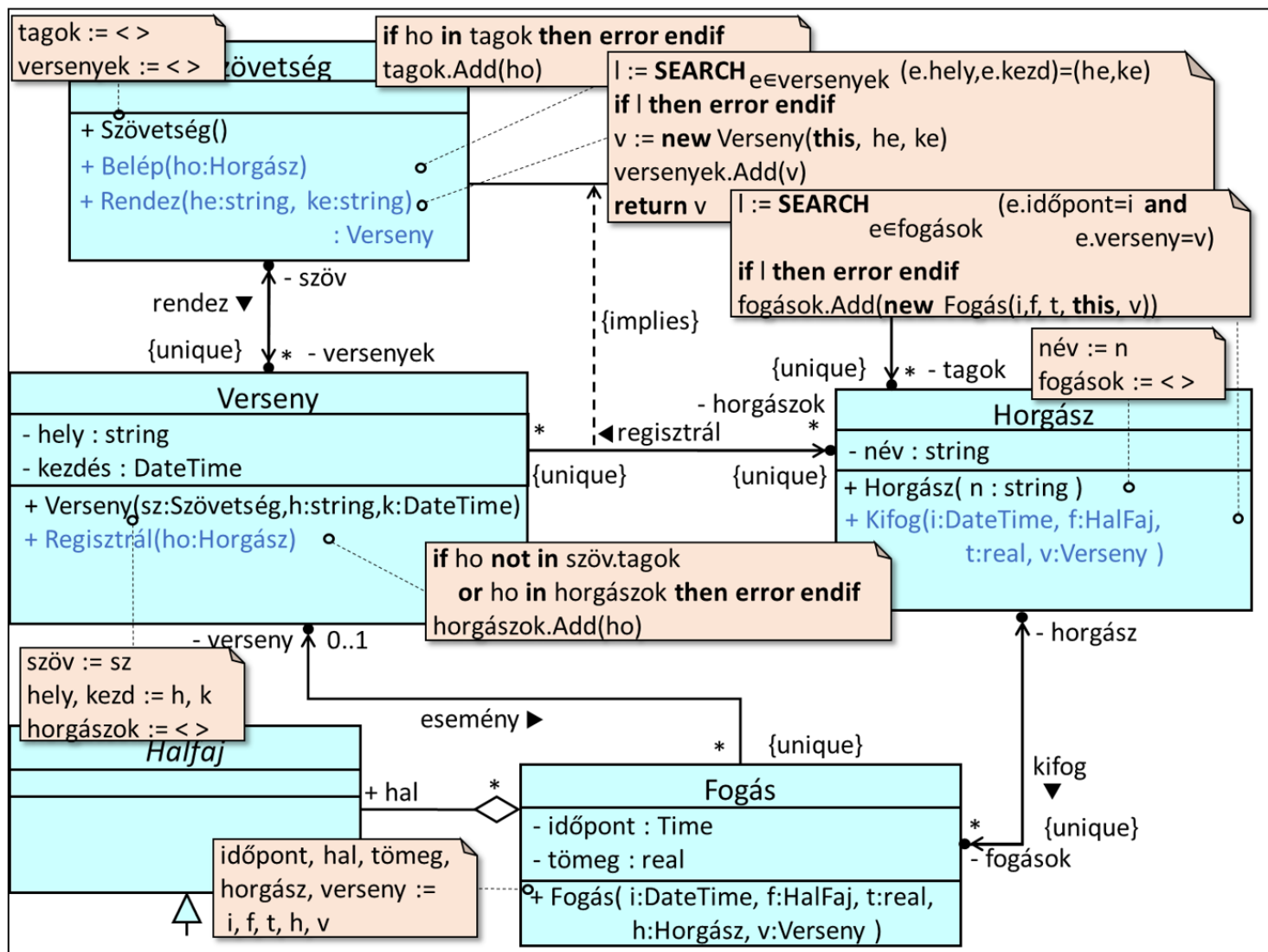
Egy fogás leírja, hogy melyik versenyen fogták, ki volt a horgász, mi a kifogott hal fajtája és a tömege (kg-ban).

A halak fajtája lehet ponty, keszeg, vagy harcsa.

A hal értéke a hal tömegének és a halfajta szorzójának (harcsa:4, ponty:2, keszeg:1) szorzata.

– Melyik a legeredményesebb verseny: ahol a horgászok fogásainak összértéke a legnagyobb és mindenki fogott harcsát?





Bizonyos esetekben a kapcsolat felépítésekor kell példányosítani a kapcsolat egyik objektumát (pl. rendez, kifog), máskor már létező objektumok között kell kapcsolatot létrehozni (belép, regisztrál). Külön figyelmet érdemel a gyűjtemények unique tulajdonságának biztosítása.

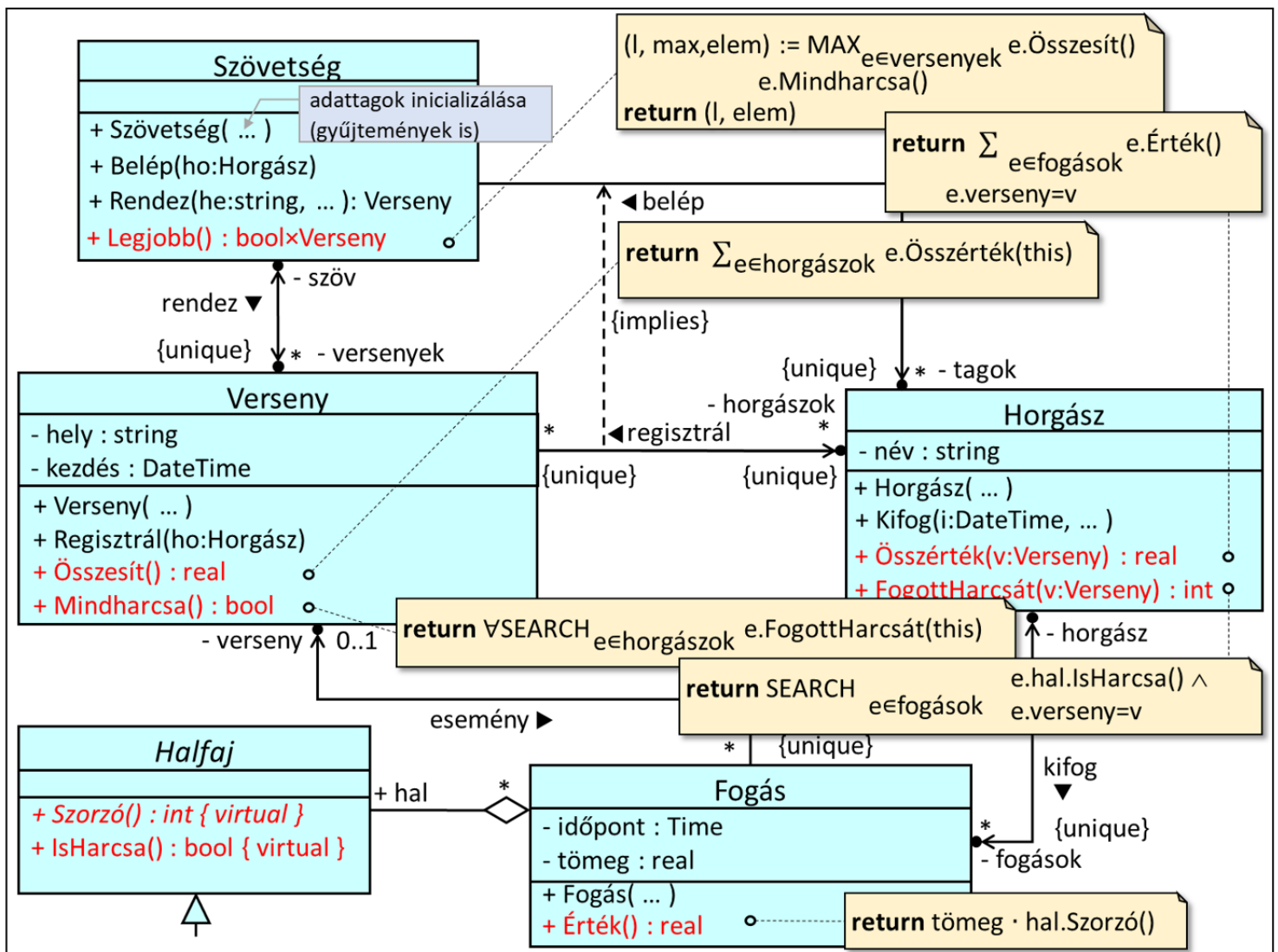
A feladat kérdésére választ adó Legjobb() metódust a Horgász-szövetség osztályában helyezük el.

Ez a versenyek felsorolására épített feltételes maximumkeresés, amelynek feltételét és értékét a Verseny osztályba telepített metódusok szolgáltatják.

Ezek a metódusok a verseny horgászainak felsorolására épülnek: az Összesít() egy összegzés, a Mindharcsa() egy optimista lineáris keresés.

Az Összesít() a Horgász osztály Összérték() metódusa által szolgáltatott értékeket adja össze; a Mindharcsa() a Horgász osztály FogottHarcsát() metódusát használja a keresés feltételeként.

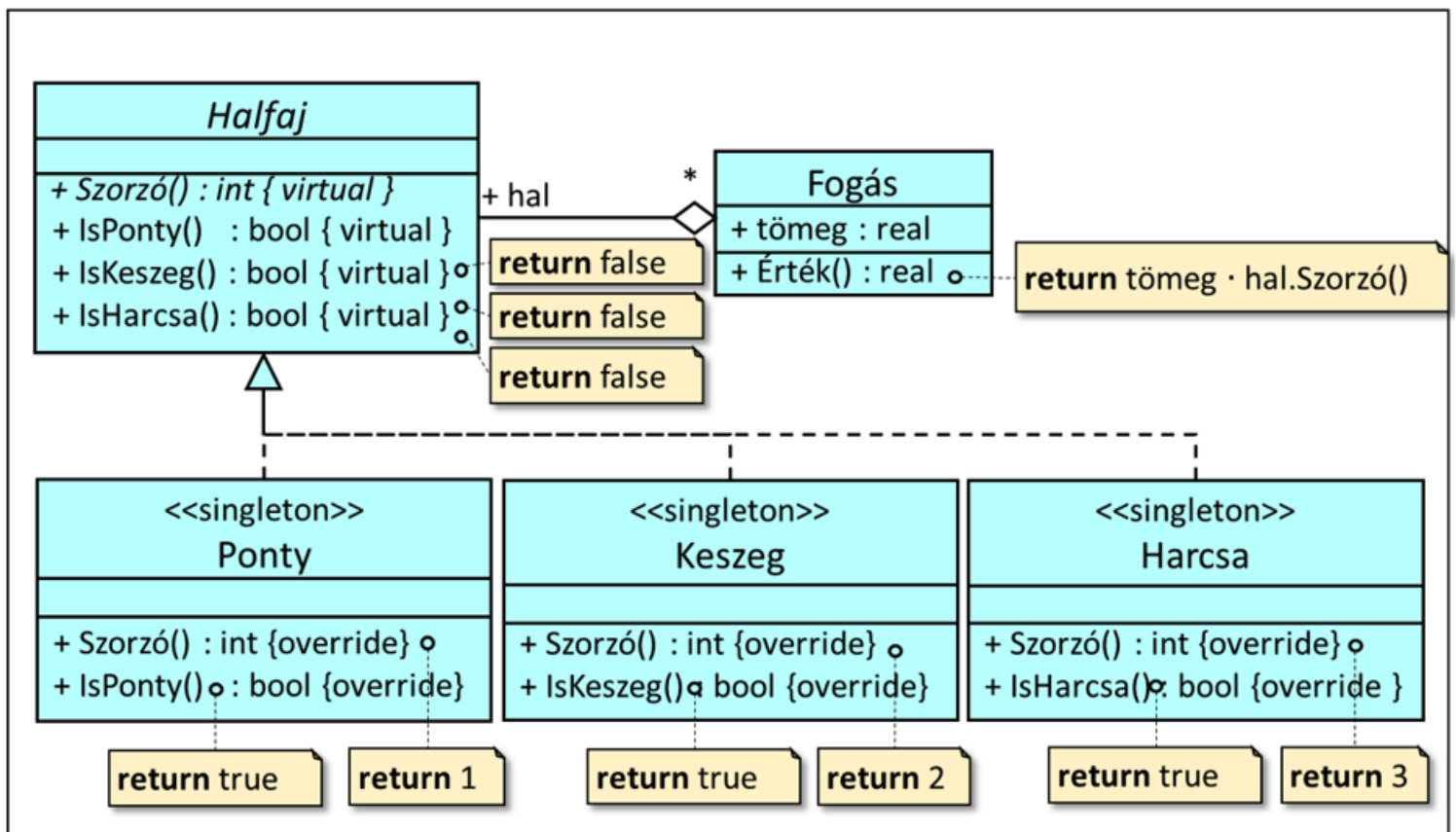
Az újabb metódusok (Összérték(), FogottHarcsát()) egy horgász fogásait sorolják fel: az egyik kiszámolja az adott versenyen fogott halak értékét (összegzés), a másik azt vizsgálja, hogy fogott-e harcsát a horgász egy adott versenyen (lineáris keresés).



A fogások `Érték()` metódusa a kifogott hal tömegének és a fajától függő szorzótényezőnek (`Szorzó()`) szorzatát számolja ki.

Annak eldöntéséhez, hogy egy fogás halfaja harcsa-e, az `IsHarcsa()` metódusra van szükség.

A `Szorzó()` és az `IsHarcsa()` metódusoknak a hal fajától függő viselkedését a `Halfaj` alosztályai biztosítják.



Céllövöldék

Egy vidámparkban a vendégek több céllövöldét is kipróbálhatnak.

A céllövöldéknek ismert a helyszínük.

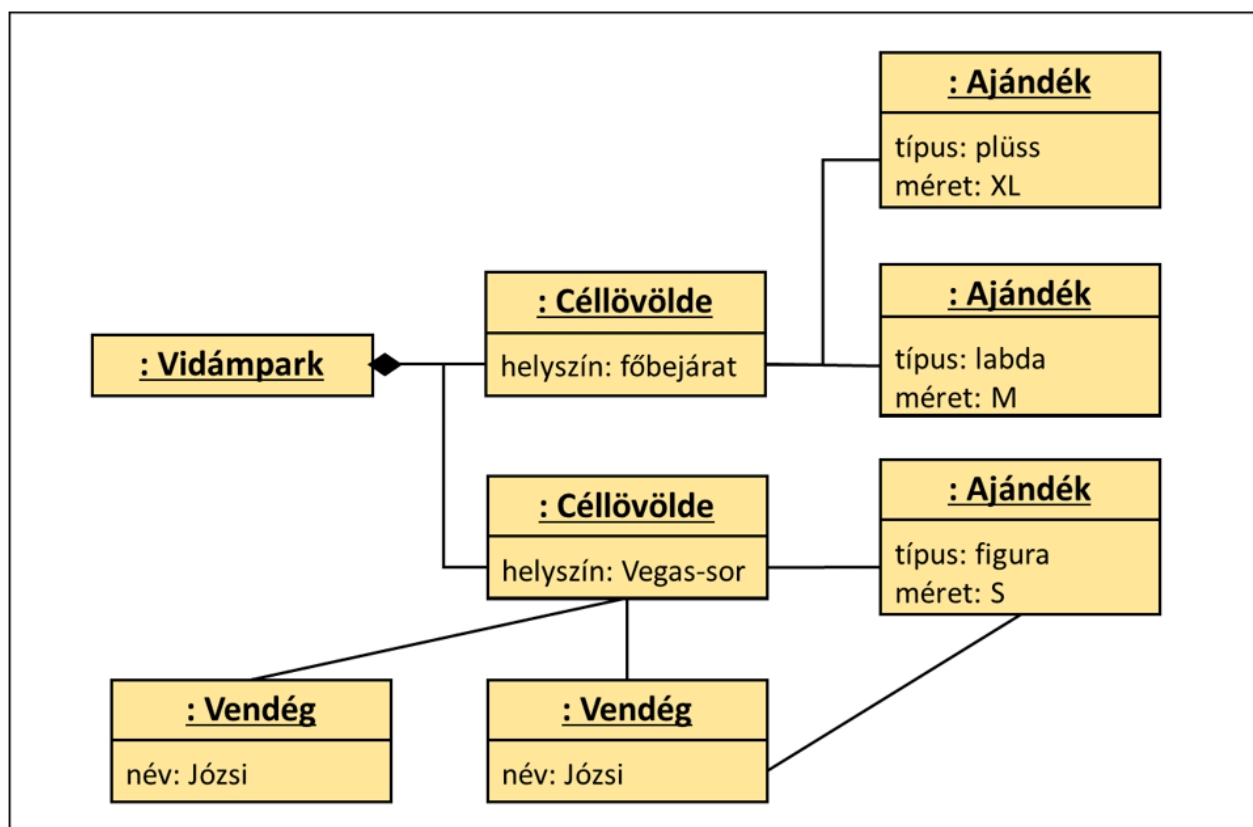
Egy céllövöldében egy vendég többször is lőhet, és sikeres találat esetén ajándékot nyer.

Egy ajándékról tudjuk, hogy melyik céllövöldében nyerték, mi a típusa (labda, műanyag figura, plüss állat) és mekkora a mérete (S, M, L, XL).

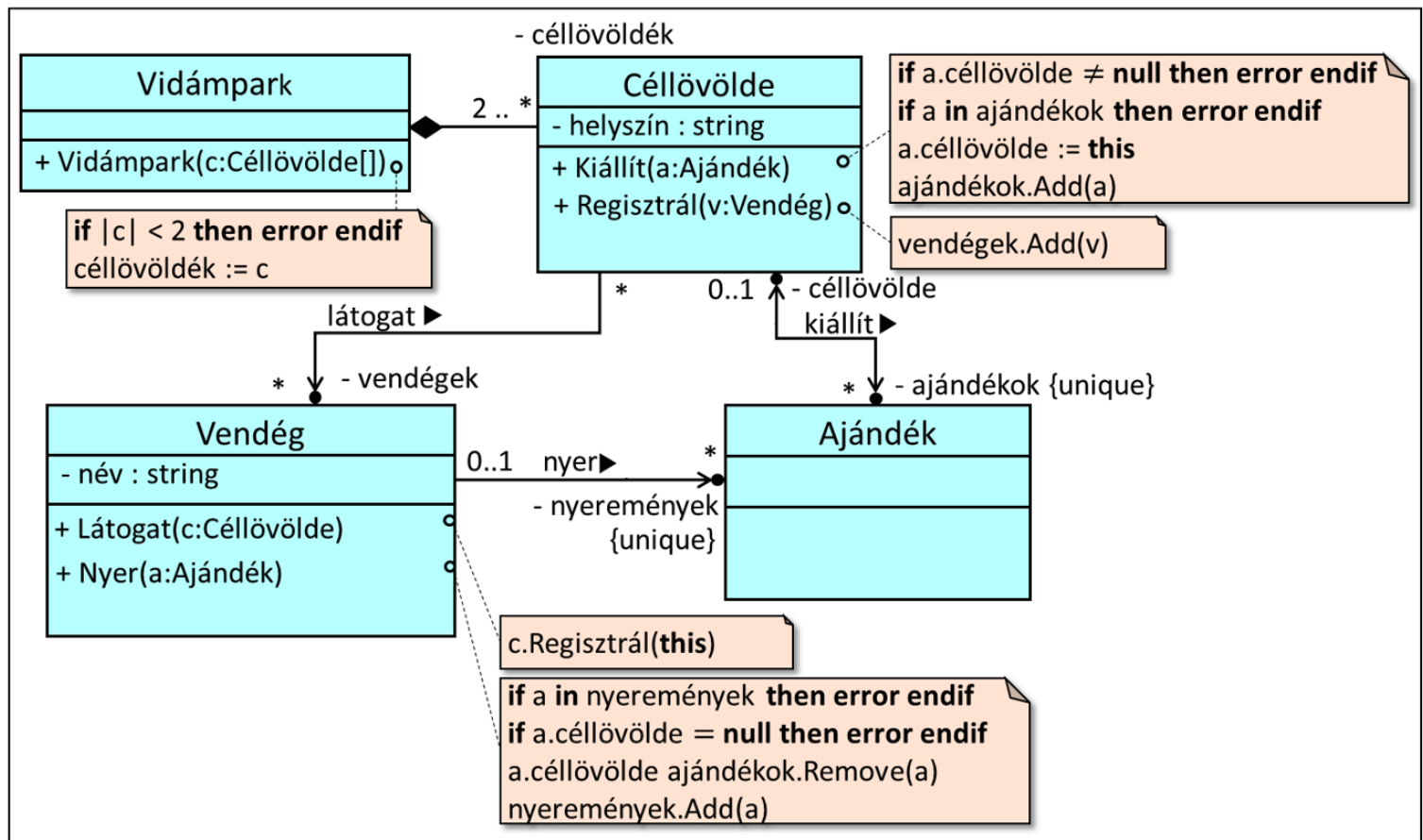
Az ajándék értékét úgy számítjuk ki, hogy a típusa után járó pontszámot (plüss állatra 3 pont, műanyag figurára 2 pont, labdára 1 pont) megszorozzuk a mérete után járó szorzóval (az S méret 1 pont, az M 2 pont, az L 3 pont, az XL 4 pont).

Nevezzük meg egy céllövölde legjobb céllövőjét!

Kezdjük most is egy objektum diagrammal.



A publikus adattagok a végső megoldásban legyenek mind privátok, és publikus getterek biztosítsák az elérésüket.



Az asszociációk navigálási iránya már a megoldandó részfeladatokra vannak tekintettel. Az ajándékok gyűjtemény „unique” tulajdonságát a Kiállítja() metódus biztosítja, a nyeremények hasonló tulajdonságát a Nyer() metódus.

(Egy vendég többször is ellátogathat egy céllövöldébe, ezért a vendégek gyűjtemény nem unique.)

Az osztály diagramnak fontos részei a kapcsolatok létrehozásáért felelős metódusok.

A Vidámpark konstruktora gondoskodik arról, hogy legalább két céllövöldéje legyen, de nem részletezzük, hogy ezeket honnan teremti elő, hiszen a feladat kérdéseinek megválaszolásához valójában nincs szükség a vidámpark objektumra.

A Céliövölde Kiállít() metódusa a céllövölde és egy ott kiállított ajándék közötti kapcsolatot építi fel úgy, hogy egy ajándékon feltünteti (céllövölde szerepnév), hogy melyik céllövöldében lehet (vagy lehetett) elnyerni. (Emiatt nem aggregáció ez a kapcsolat.)

A `Vendég Látogat()` metódusa kezdeményezi a vendég regisztrálását egy adott céllövöldében, de mivel nem fér hozzá a céllövölde vendégek gyűjteményéhez (hiszen az privát),

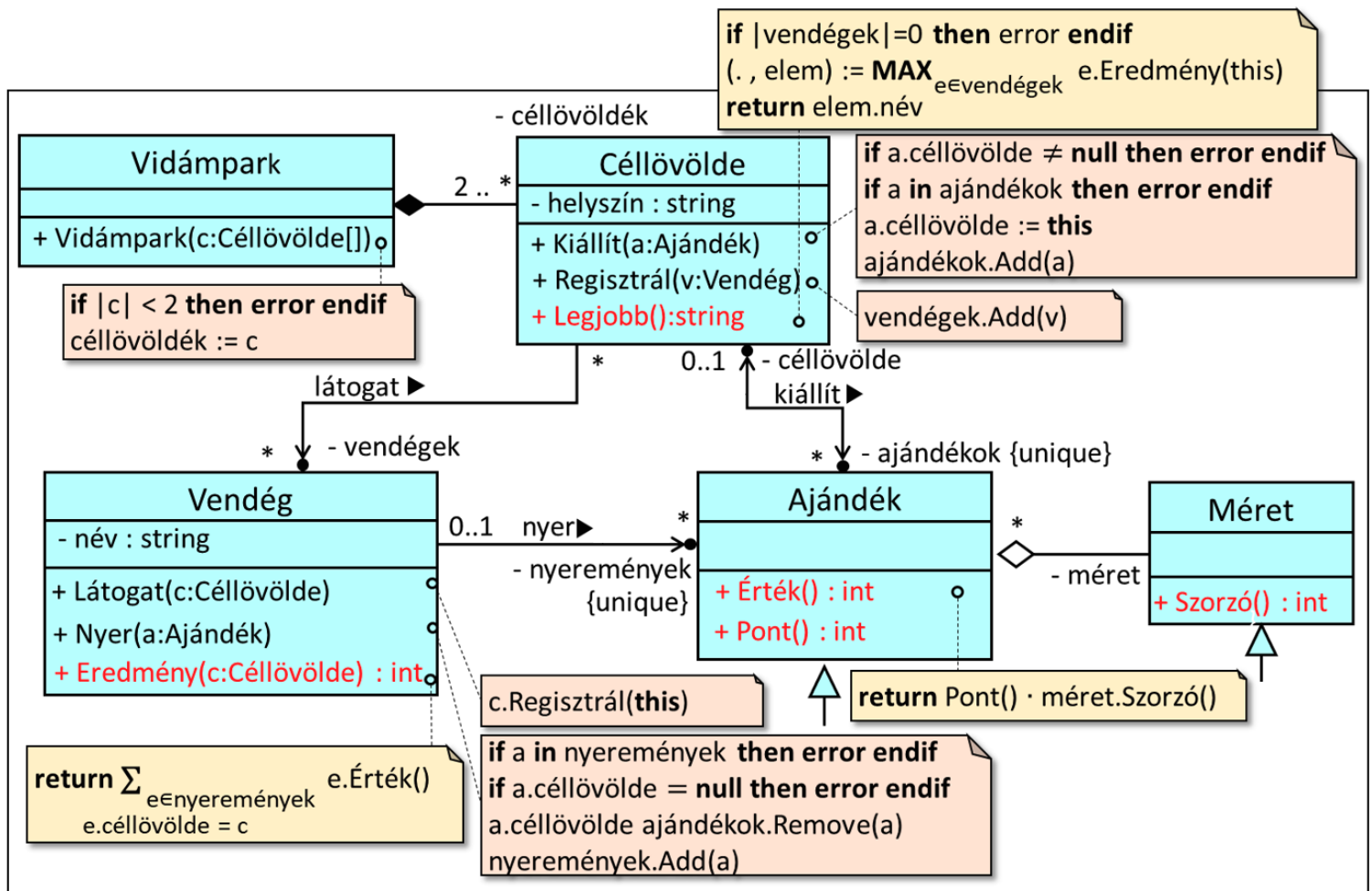
a `Céllövölde Regisztrál()` metódusát hívja, amelyik kiépíti a kapcsolatot a vendég és a céllövölde között úgy, hogy a vendég bekerül a céllövöldénél nyilvántartott vendégek gyűjteményébe.

A `Vendég Nyer()` metódusa helyez el egy elnyert ajándékot a vendég nyereményeit tároló gyűjteménybe.

Egy céllövölde legjobb vendégét megkereső maximum kiválasztás (ez a `Céllövölde osztály Legjobb()` metódusa) számára fel kell sorolni a céllövöldénél regisztrált vendégeket, és ki kell számolni az adott céllövöldében szerzett nyereményeiknek összértékét.

Ez utóbbit a `Vendég Eredmény()` metódusa végezi, amely egy feltételes összegzés, ahol egy vendég adott céllövöldében nyert nyereményeit kell felsorolni, és azok értékét összeadni.

Egy nyeremény (azaz ajándék) értéke (`Érték()`) az ajándék fajtájának pontszámától (`Pont()`) és a méretének szorzótényezőjétől (`Szorzó()`) függ.



Az Ajándék osztályból konkrét ajándékfajták osztályait (Labda, Figura, Plüss) kell származtatnunk, amelyekben felülírjuk a Pont() metódust. Egy ajándék értékének kiszámolásához szükséges szorzótényezőt a stratégia tervmintával adjuk meg.

Bevezetjük a Méret osztályt, amely a Szorzó() metódust definiálja.

Az Ajándék osztályba egy ilyen Méret típusú objektumot kell aggregálnunk, hogy hozzájussunk a méretre jellemző szorzótényezőhöz.

Mivel többféle méret is van, így a Méret egy tisztán absztrakt őssztály, egy ún. interfész lesz, amelyet a konkrét méret osztályok egykeként implementálnak majd.

(Egy méretből elég egy példány, amelyet minden olyan ajándék meghivatkozik, amelynek ez a mérete). Ezek az implementáció definiálják az adott méretre jellemző Szorzó() metódus törzsét.

