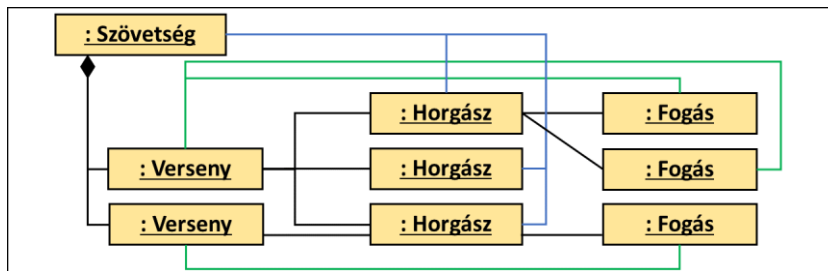


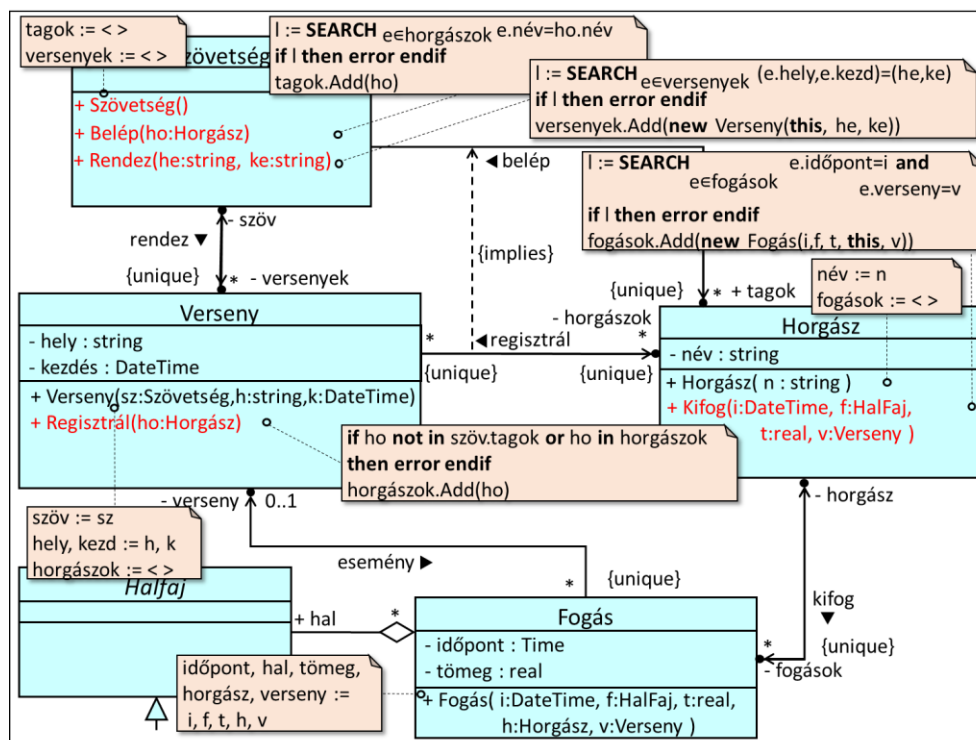
11. Osztály diagram II.

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:3, ponty:2, keszeg:1) szorzata.
 - Melyik a legeredményesebb verseny: ahol a horgászok fogásainak összértéke a legnagyobb és mindenki fogott harcsát?

Első olvasatra négyféle típusú objektummal számolhatunk.

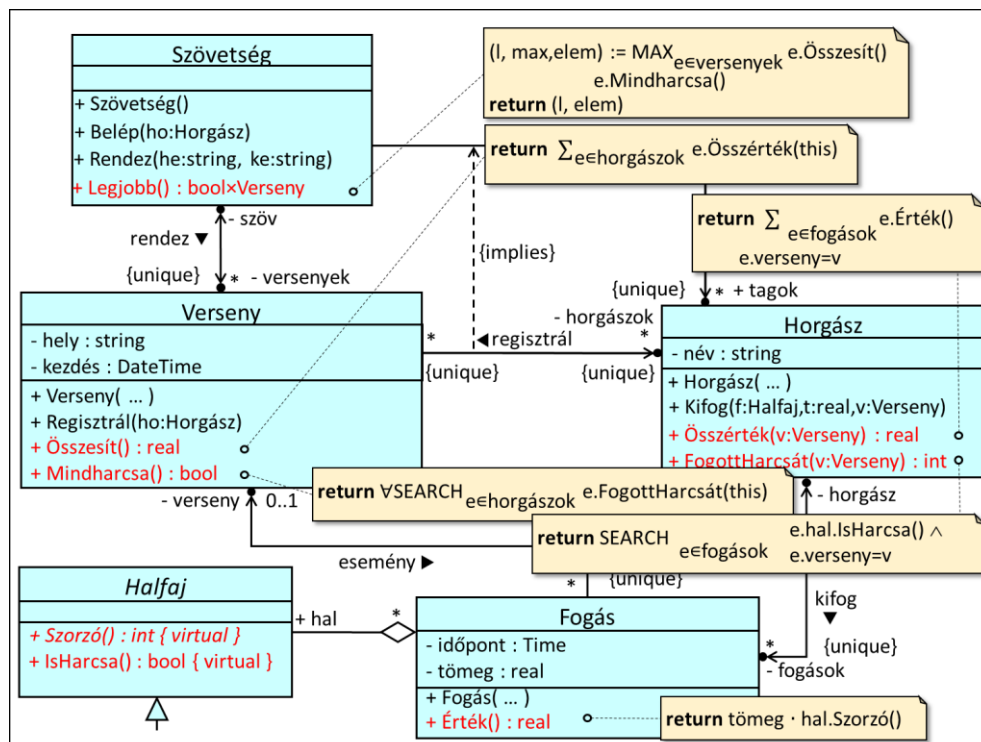


Az asszociációk tervezésével együtt születnek a kapcsolatokat felépítő metódusok.

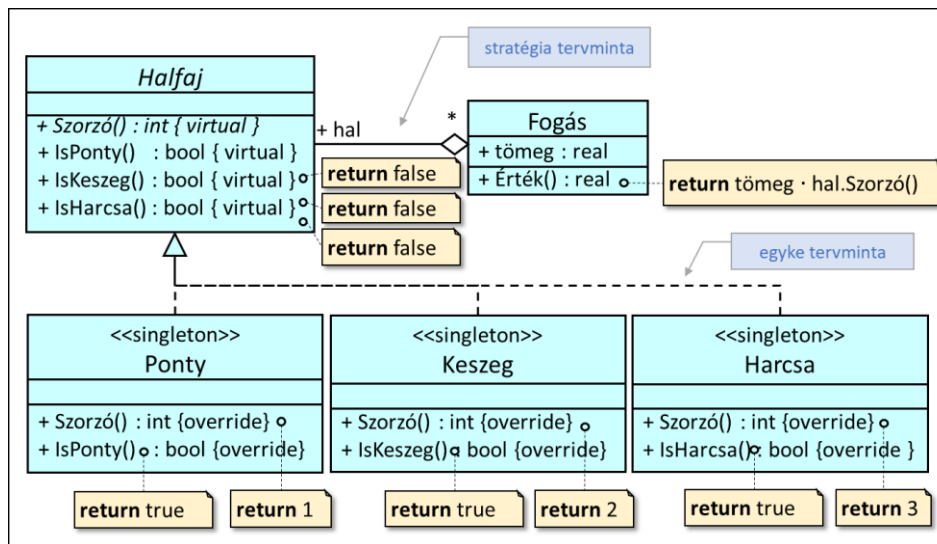


Bizonyos esetekben a kapcsolat felépítésekor kell példányosítani a kapcsolat egyik objektumát (pl. rendez, kifog), máskor már létező objektumok között kell kapcsolatot létrehozni (Belép(), Kifog()). Külön figyelmet érdemel a gyűjtemények unique tulajdonságának biztosítása. Ehhez néha elég egy objektum-hivatkozás szintű ellenőrzés (Regisztrál()), de többnyire tartalmi ellenőrzés is kell (Belép(), Rendez(), Fogás()).

A kérdésre választ adó Legjobb() metódust a Horgász-szövetség osztályában helyezük el. Ez egy versenyek felsorolására épített feltételes maximumkeresés, amelynek feltételét és értékét a Verseny osztályba telepített metódusok szolgáltatják: az Összesít() a horgászok felsorolására épülő összegzés, amely az Összérték() metódusra támaszkodik; a feltételt kiszámító Mindharcsa() metódus a horgászok felsorolására épülő optimista lineáris keresés, amely a FogottHarcsát() metódust használja. Az újabb metódusok (Összérték(), FogottHarcsát()) egy horgász fogásait sorolják fel: az egyik összegzi az adott versenyen fogott halak értékét, a másik azt vizsgálja, hogy fogott-e harcsát a horgász egy adott versenyen.

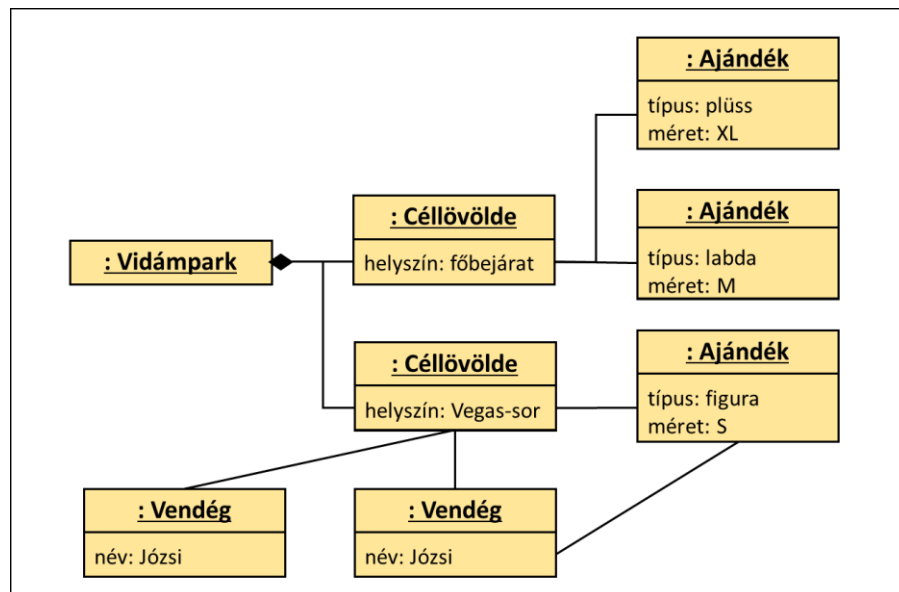


Egy fogás Érték() metódusához a kifogott hal tömege és fajtájából adódó szorzótényező (Szorzó()) szükséges; annak eldöntéséhez pedig, hogy egy fogás halfaja harcsa-e, az IsHarcsa() metódusra van szükség. Ez utóbbi két metódust a Halfaj osztály, illetve az abból származtatott osztályok biztosítják.

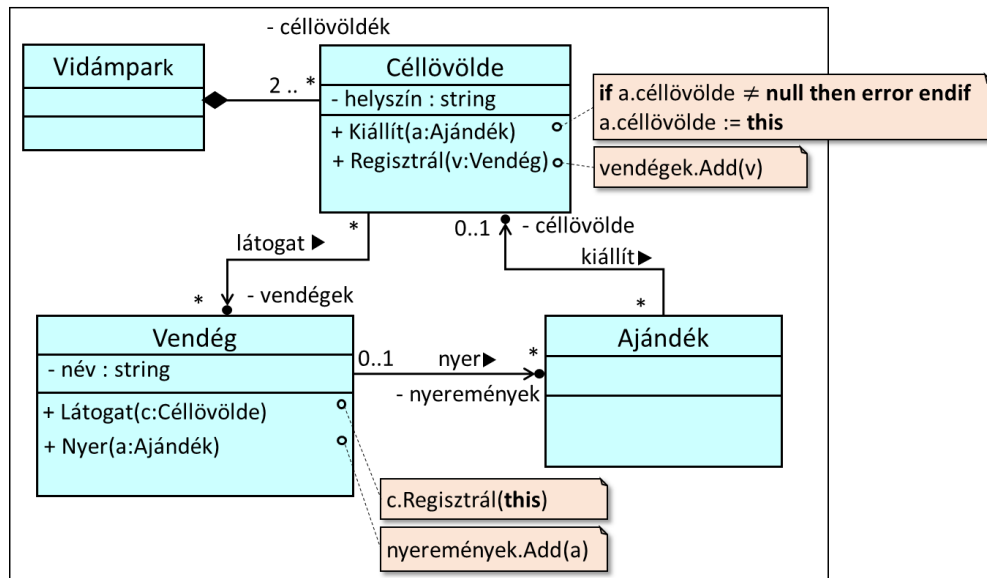


2. Egy vidámparkban a vendégek több céllövöldét is kipróbálhatnak. A céllövöldéknek ismert a helyszínük. Egy céllövöldében egy vendég többször is lőhet, és sikeres találat esetén ajándékot nyer. Egy ajándékról tudjuk, hogy melyik céllövöldében nyerték, mi a típusa (labda, műanyag figura, plüss állat) 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.



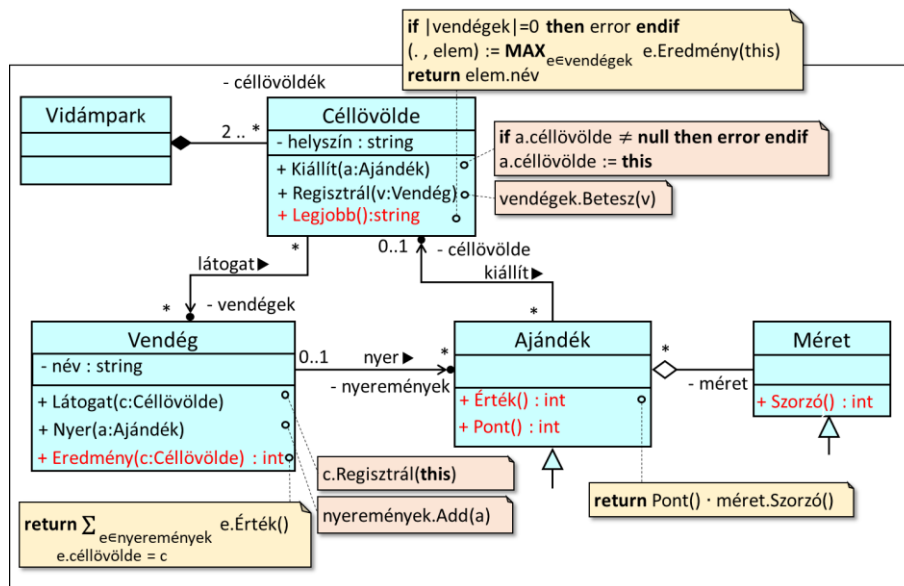
Érdekes része a diagramnak a három különböző típusú objektum kört alkotó kapcsolata. A publikus adattagok a végső megoldásban legyenek mind privátok, és publikus getterek biztosítsák az elérésüket.



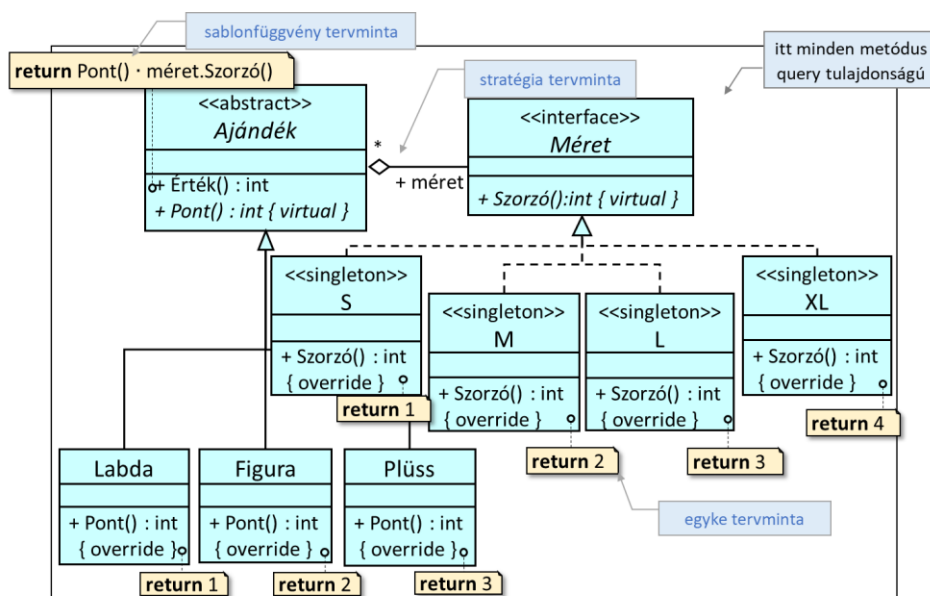
Az asszociációk navigálási iránya már a megoldandó részfeladatokra vannak tekintettel. Nincs szükség a gyűjtemények „unique” tulajdonságát megkövetelni. (Egy vendég többször ellátogathat egy céllövöldébe; ugyanolyan ajándékból több is lehet.)

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 feltünteteti (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 Céliövölde 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 Céliövölde osztály Legjobb() metódusa) számára fel kell sorolni a céllövöldénél regisztrált vendégeket (ugyanazt a vendéget esetleg többször is), 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.



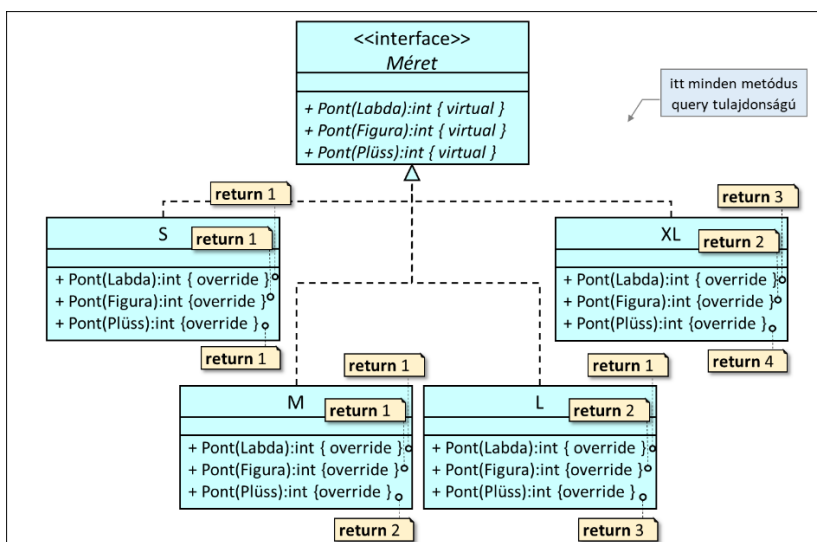
A Pont() és a Szorzó() tevékenységeinek tervezését több nevezetes tervezési minta támogathatja. A Pont() az Érték() metódus törzsében egy sablonfüggvény, amit a konkrét Ajándék objektumok Pont() függvénye helyettesít. Ehhez az Ajándék osztályból konkrét ajándékfajták osztályait (Labda, Figura, Plüss) 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 mérethe 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 mérethe jellemző Szorzó() metódus törzsét. A sablonfüggvény- és a stratégia tervminta a rugalmas bővítés lehetőségét, az Open-Close szoftvertervezési elvet támogatja, míg az egyke tervminta a memória spórolás eszköze.



Alkalmazhatnánk látogató tervmintát is. Ez különösen akkor hasznos, ha egy ajándék értékét nem lehet egy közös képlettel kiszámolni, hanem az egyedi módon függ az ajándék típusától és a méretétől:

	S	M	L	XL
Labda	1	1	1	3
Figura	1	1	2	2
Plüss	1	2	3	4

Legyen a Méret egy objektuma a látogató, amelyet az Ajándék objektuma fog használni az ajándék értékének kiszámolásához. Elvileg lehetne fordítva is (az ajándék a látogató), de egyrészt a feladatban egy Ajándék objektumnak kell kiszámolni az értékét a méret függvényében, és nem fordítva; másrészt így a Méret osztályban és alosztályaiban (ez összesen 5 osztály) csak 3-3 metódust kell bevezetnünk szemben a másik megoldással, ahol 4 osztályban 4-4 metódusra lenne szükség. A látogatót leíró Méret osztály és alosztályai az ajándékok ponttáblázatát tükrözik vissza.



Az Ajándék osztály és alosztályai a látogatóként kapott mérettől függően fogják az ajándék értékét kiszámolni.

