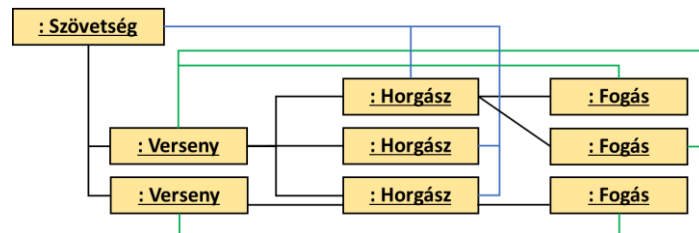


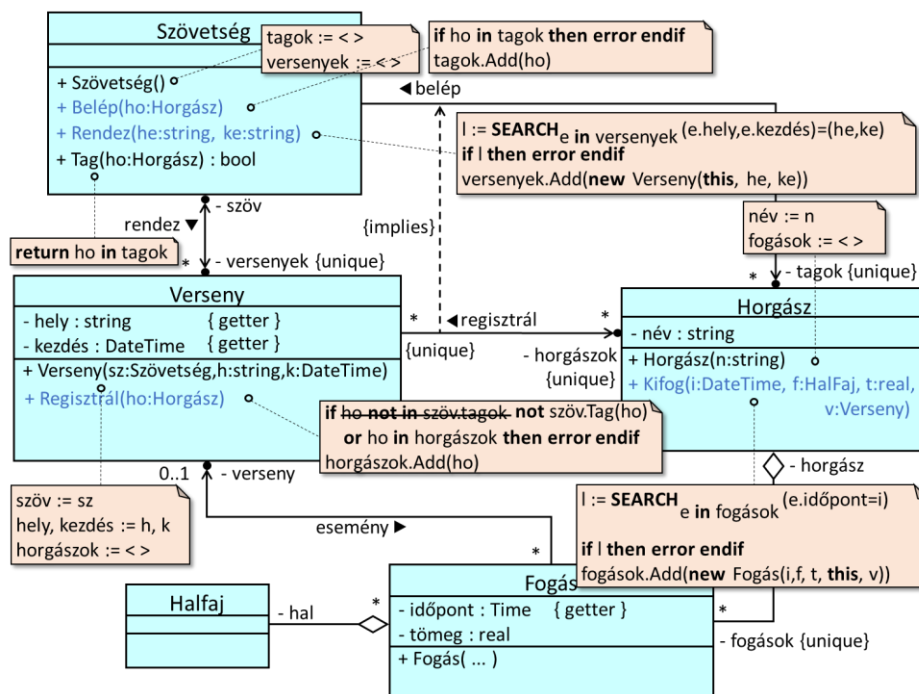
10. Modellezés II.

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.



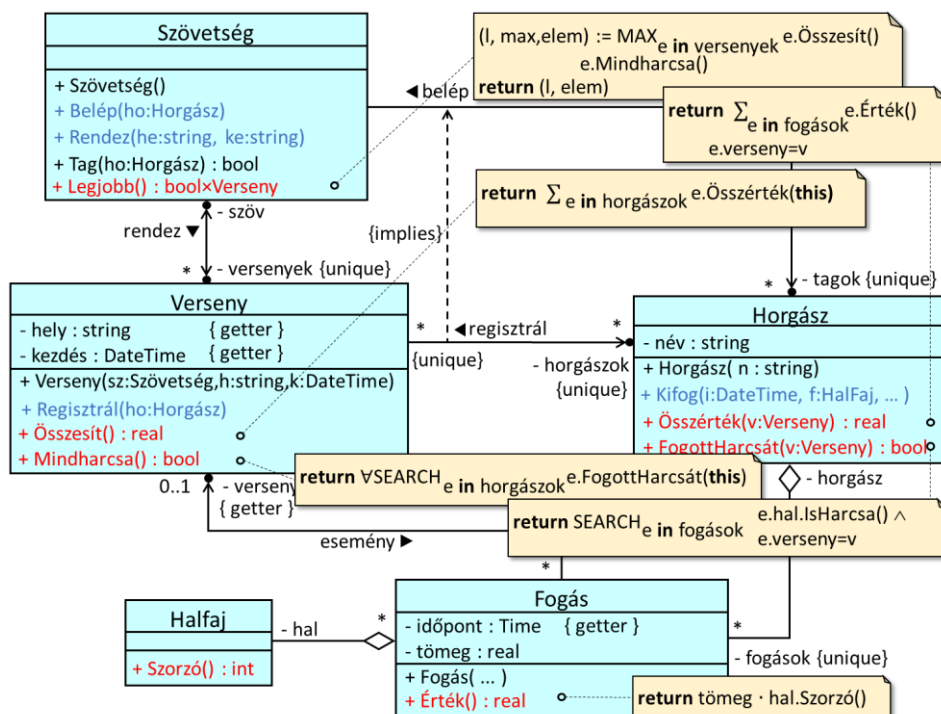
Az asszociációk tervezésével együtt születnek a kapcsolatokat felépítő metódusok.



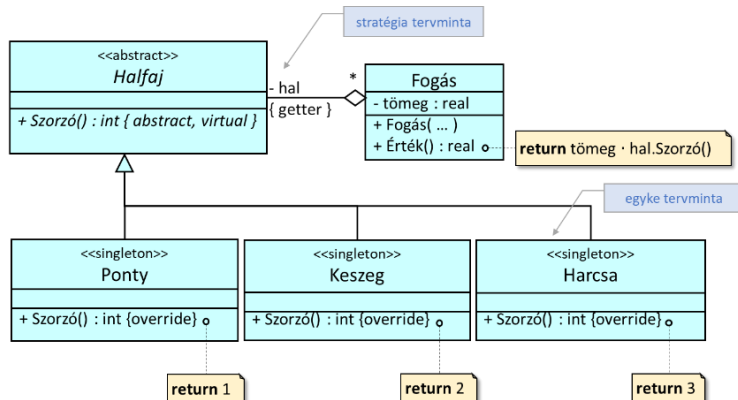
Egy kapcsolat felépítését végző metódust a kapcsolat valamelyik osztályában kell elhelyezni. Bizonyos esetekben a kapcsolat felépítésekor kell példányosítani a kapcsolat másik objektumát (pl. rendez, kifog asszociációknál), máskor már létező objektumok között kell kapcsolatot létrehozni (lásd a belép, regisztrál asszociációkat). Ettől függ az, hogy a gyűjtemények „unique” tulajdonságát milyen ellenőrzéssel kell biztosítani: létező objektumok esetén elég az objektum-hivatkozás szintű ellenőrzés (ld. Belép(), Regisztrál()), máskor tartalom szintű ellenőrzés kell (lásd a Rendez(), Kifog()).

Egy privát adattag „láthatóságát” nemcsak publikus getter, hanem speciális ellenőrző metódus is biztosíthatja (ld. Tag()).

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).

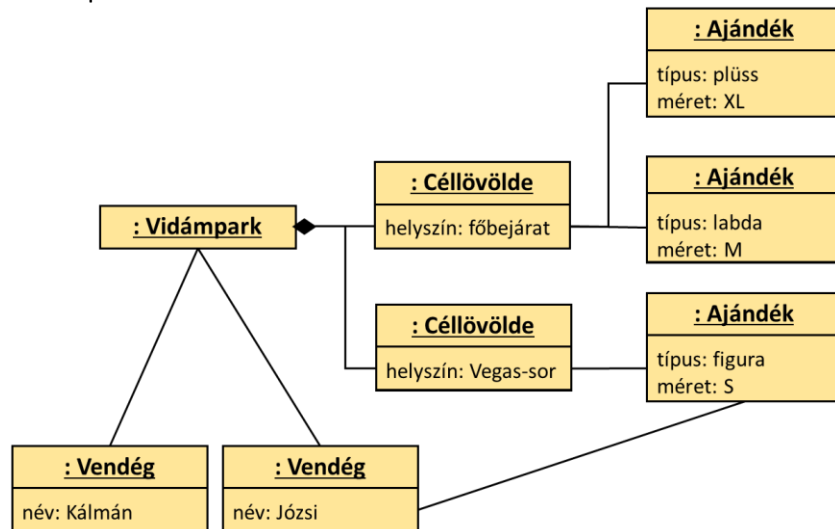


A fogások Érték() metódusa a kifogott hal tömegének és a fajtától függő szorzótényezőnek (Szorzó()) szorzatát számolja ki. Annak eldöntéséhez, hogy egy fogás halfaja harcsa-e, az IsHarcsa() metódusra van szükség. A Szorzó() és az IsHarcsa() metódusoknak a hal fajtától függő viselkedését a Halfaj alosztályai biztosítják majd. Az így felépülő szerkezet a stratégia tervezési mintát idézi.



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.



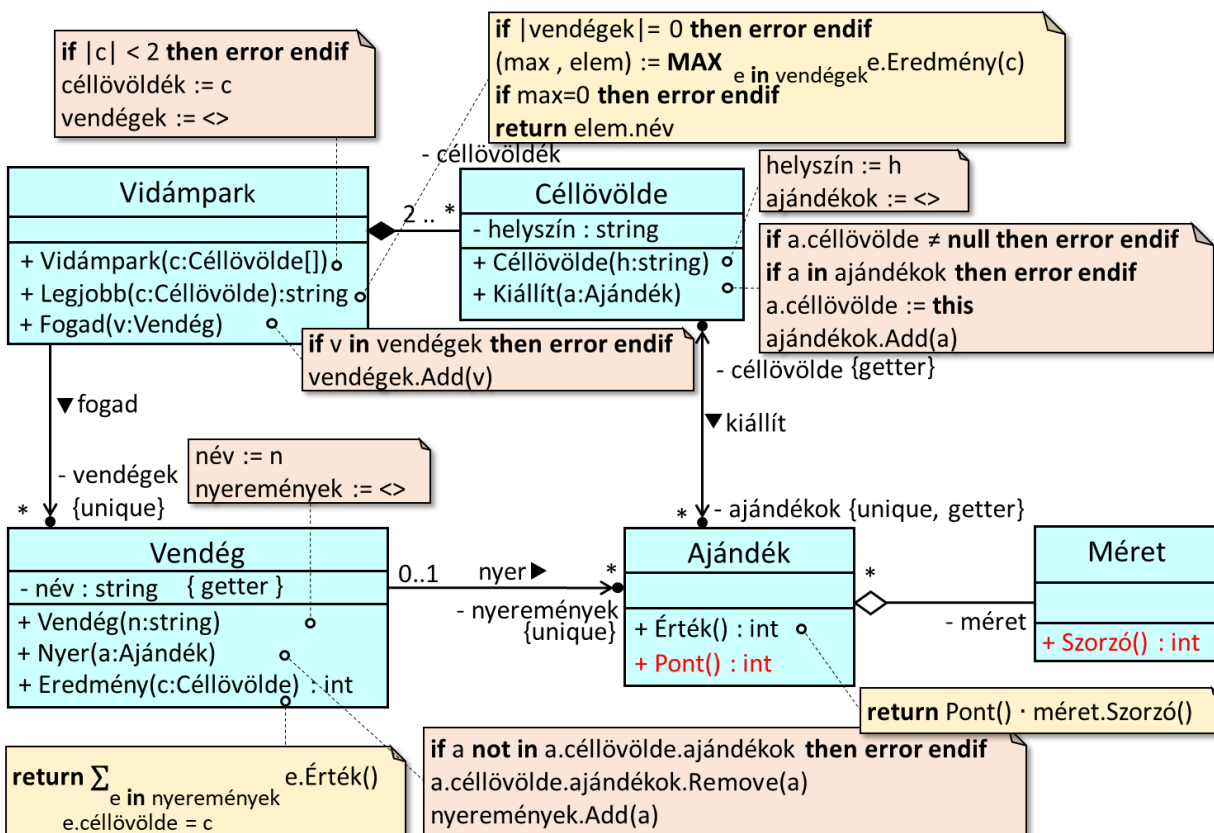
A megfelelő osztály diagram a kapcsolatokat kiépítő metódusokkal:

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. lőtték. 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 **Céllövölde** **Kiállít()** metódusa egy új ajándékot vesz fel a céllövölde ajándékai közé (ez az ajándékok nevű gyűjtemény), amelyen feltünteti a céllövöldét (lásd céllövölde szerepnevet), hogy lehessen látni az ajándékról, hogy melyik céllövöldében lehet (vagy lehetett) elnyerni. A már elnyert ajándékról is leolvasható majd, hogy melyik céllövöldében nyerték, annak ellenére, hogy már nem lesz benne az ajándékok gyűjteményben. A **Vendég** **Nyer()** metódusa írja le azt, amikor egy céllövöldében egy ajándékot nyer a vendég. Ekkor az ajándék kikerül a céllövölde ajándékai közül, és bekerül a vendég nyereményei közé, miközben az ajándék „őrzi annak emlékét”, hogy melyik céllövöldében lőtték.

Egy céllövölde legjobb vendégét megkereső **maximum** kiválasztás (ez a **Céllövölde** 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.

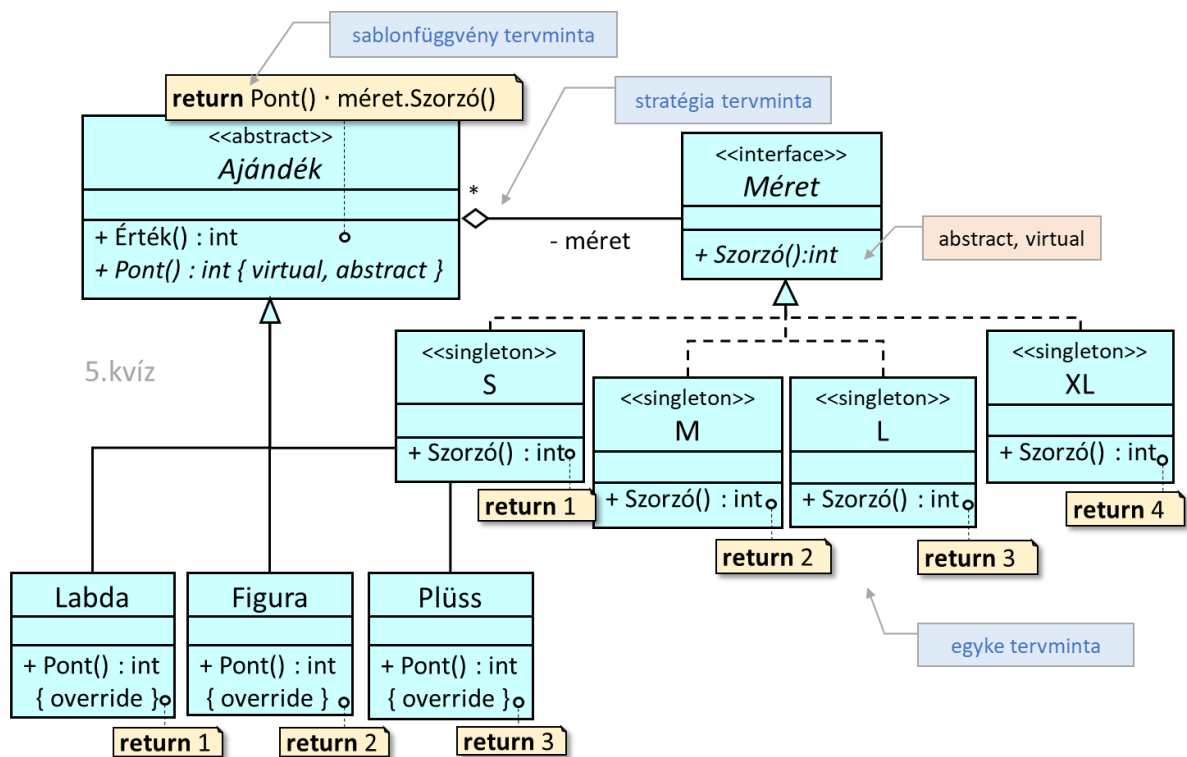


Az Ajándék osztály Érték() metódusának kialakítását több nevezetes tervezési minta támogathatja.

Az Érték() metódus legyen egy sablonfüggvény, amely törzsében használt Pont() függvény absztrakt, és ezt az Ajándék osztályból származtatott konkrét ajándékfajták osztályai (Labda, Figura, Plüss) írják felül. Egy ajándék értékének kiszámolásához az ajándék méretétől függő szorzótényezőre is szükség van, amelyet stratégia tervmintával adunk meg. Bevezetjük a Méret osztályt, amely a Szorzó() metódust specifikálja. Mivel többféle méret is van, a Méret egy tisztán absztrakt ősz osztály, egy ún. interfész lesz, és ezt a konkrét méreteket leíró (S, M, L, XL) osztályok különféle módon implementálják. Az Ajándék osztályba egy konkrét Méret típusú objektumot kell aggregálnunk ahhoz, hogy hozzájussunk a mérethe jellemző szorzótényezőhöz.

A konkrét méret osztályok egykeként implementálják a Méret interfészt, hiszen egy méretből elég egy példány, amelyet minden olyan ajándék meghívkozhat majd, amelynek ez a mérete.

A sablonfüggvény- és a stratégia tervminta a rugalmas bővítés lehetőségét, az Open-Close szoftvertervezési elvet támogatja, míg az egyke tervminta a memória spórolás eszköze.

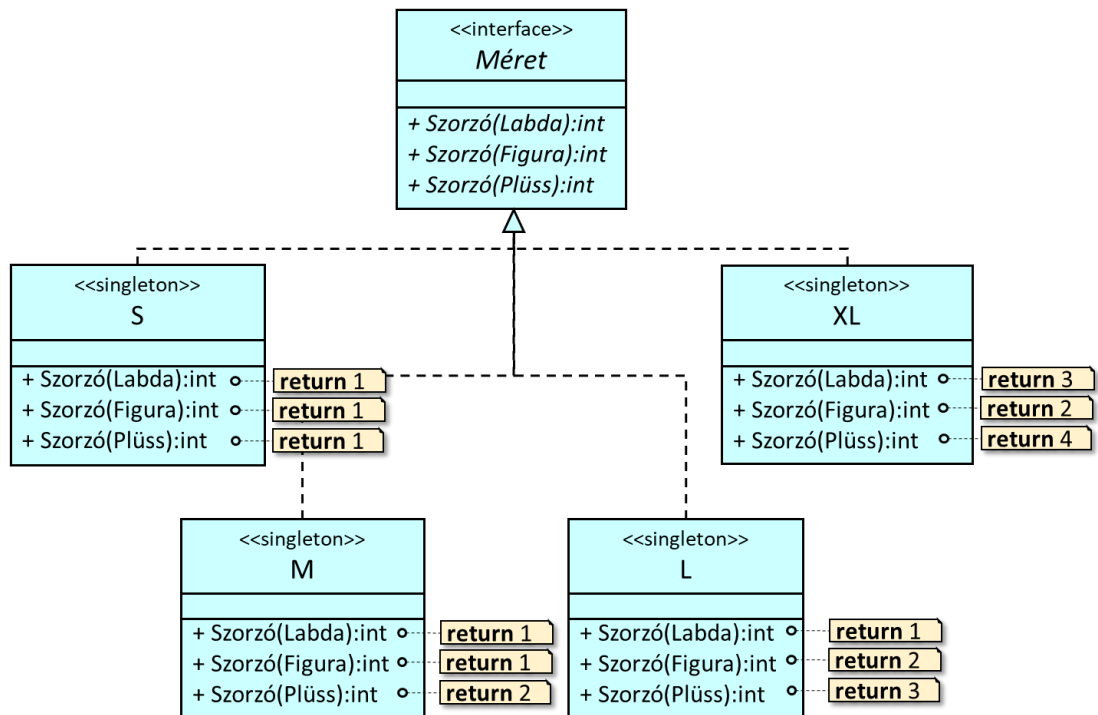


* * *

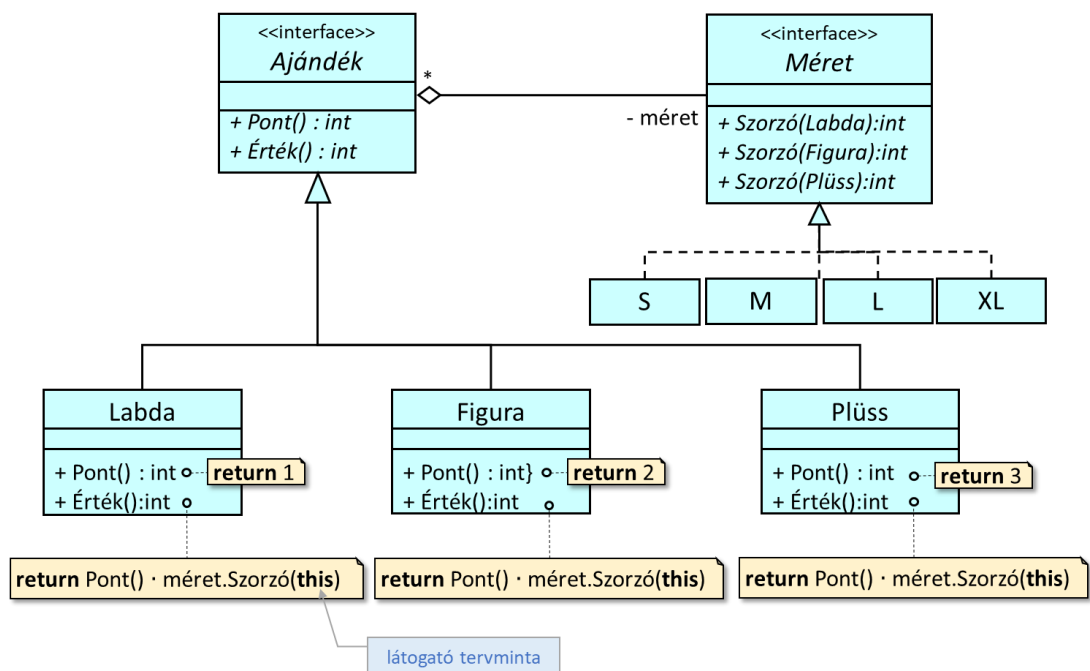
Alkalmazhatnánk a tervben látogató tervmintát is. Ez különösen akkor lenne hasznos, ha az ajándékok értékének kiszámolásánál az ajándék alapárát egy olyan a szorzótényezővel kellene megszorozni, amely egyszerre függ a mérettől is és az ajándék fajtájától is. Egy ilyen szorzótényezőket tartalmazó táblázatot láthatunk itt:

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

Ilyenkor Méret osztálynak és leszármazottjainak nemcsak egy-egy Szorzó() metódusa lesz, hanem annyi, ahányféle ajándék van (esetünkben három-három), amelyeket az különböztet meg, hogy paraméterként egy Ajándék típusú objektumot várnak.

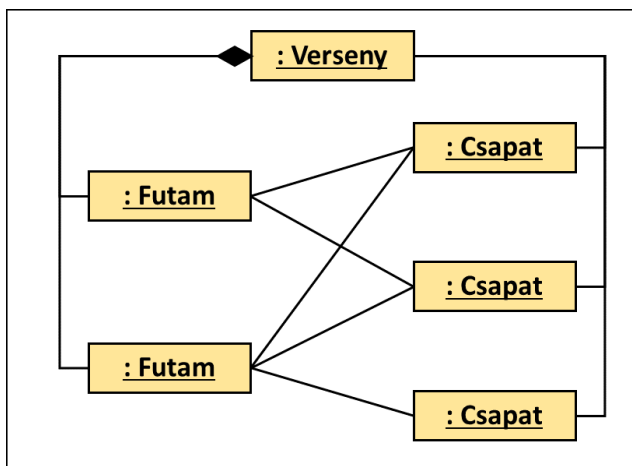


Egy Ajándék típusú objektum a stratégia tervmintához hasonlóan kap (akár aggregációval, akár az Érték() metódusuknak átadott paraméterrel) egy Méret típusú objektumot, amelynek Szorzó() metódusát használja az érték kiszámolásához, és a Szorzó() metódus paramétere az aktuális Ajándék típusú objektum lesz. Az ajándék „elmegy látogatóba” a Szorzó() metódushoz azért, hogy az ne csak a mérettől, hanem az ajándék fajtájától is függő szorzótényezőt állíthasson elő.

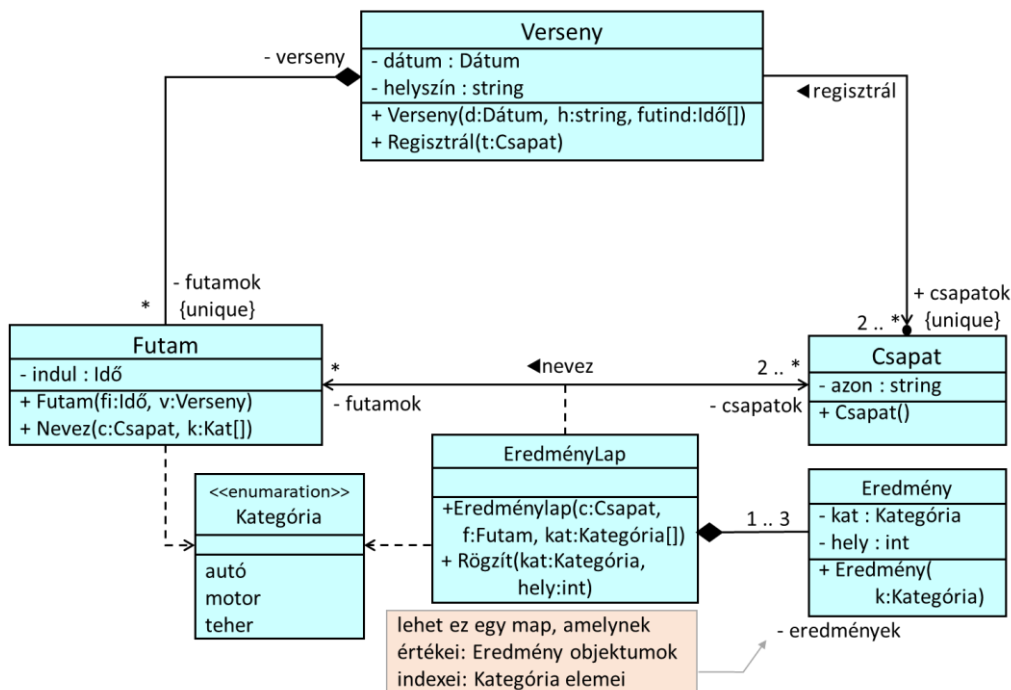


3. Egy országos rally autóversenyre történő regisztrációkor a csapatok egyedi azonosítót kapnak, amellyel az egyes futamokra benevezhetnek. Egy verseny több (legalább egy) futamból áll, egy futamon több (legalább kettő) csapat is indul legalább egy kategóriában (sportautó, teherautó, motor). A futamok megadott időben indulnak. Egy csapatnak egy futamon elért eredménye az egyes kategóriákban elért helyezéseitől függ. Ehhez mindazon kategóriákban, ahol indított versenyzőt egy csapat, a *(futamon induló csapatok száma+1-helyezés)*-tényező képlet értékét kell kiszámolni, és ezen értékeket összeadni. A képletbeli tényező kategóriánként eltérő: ez a motoroknál 1, a sportautóknál 3, a teherautóknál 4.

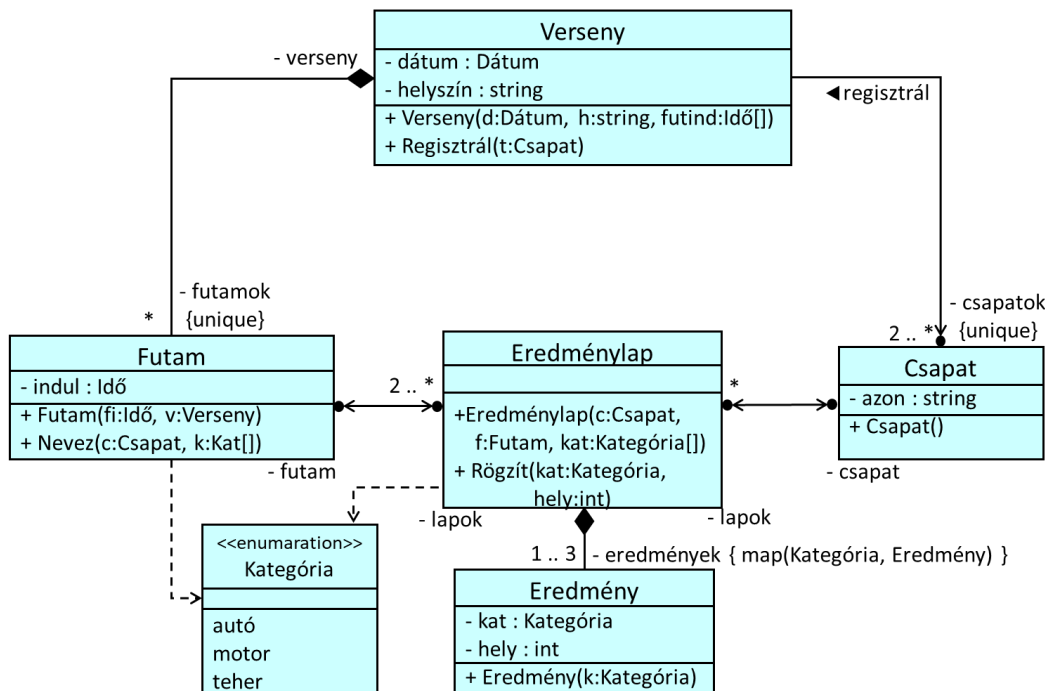
Melyik csapat a nyertese a teljes versenynek a futamokon elért eredményeik értékének összege alapján?



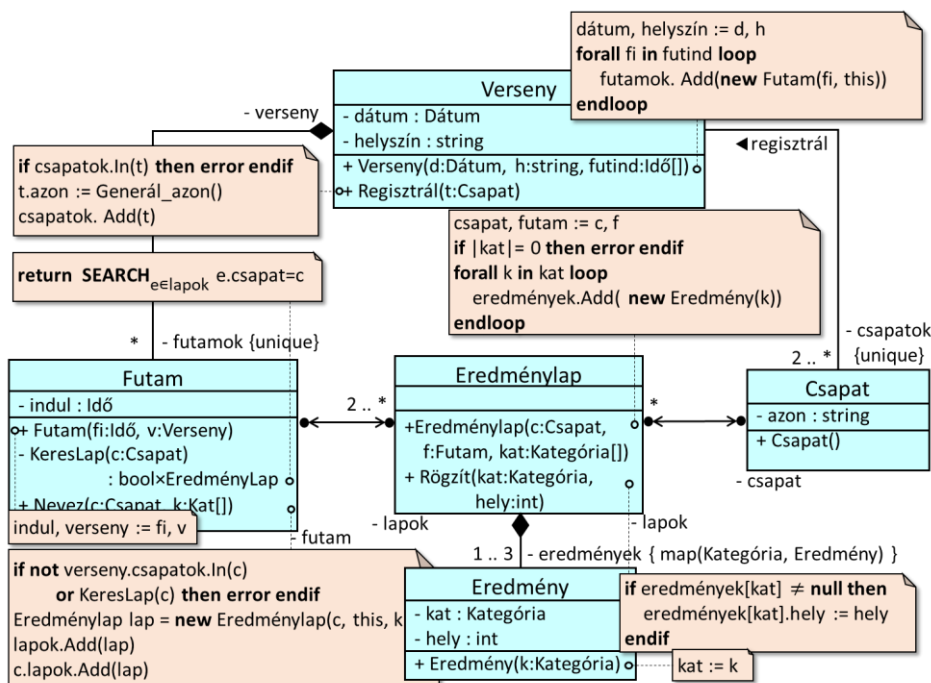
Az objektum diagram futam-csapat kapcsolatai mutatják meg, hogy milyen eredményt ért el egy csapat egy futamon az egyes kategóriákban. Ezt a kapcsolatrendszert az osztálydiagramban az Eredménylap asszociációs osztály segítségével jellemezzük.



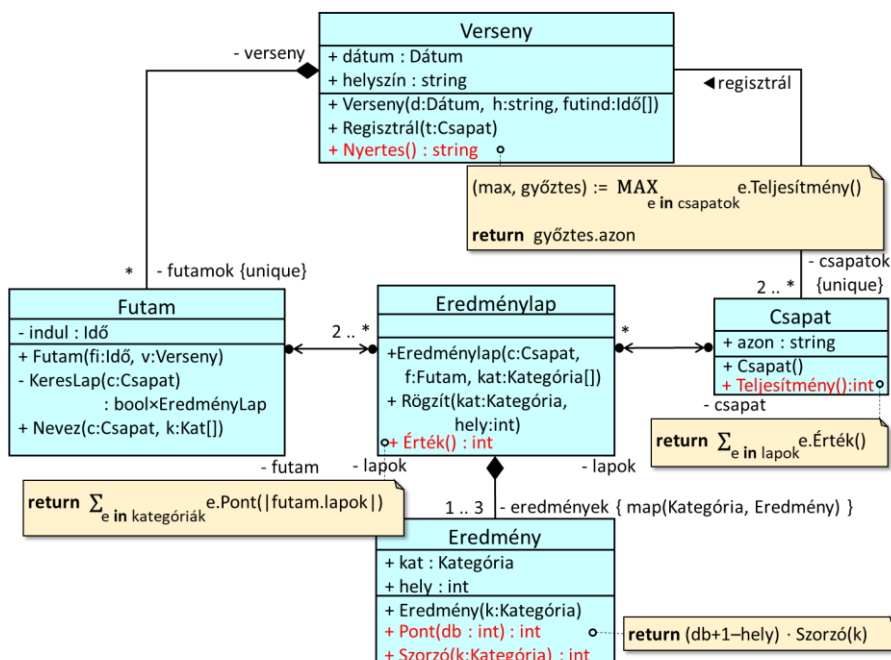
Az Eredménylap osztályba komponált eredmények gyűjteményt egy olyan asszociatív tárolónak (map) képzeljük el, amely legalább egy, legfeljebb három elemű, és az elemeket a kategória nevekkal (autó, motor, teher) lehet indexelni. Ezután elimináljuk az asszociációs osztályt:



A kompozíciós kapcsolatokat a tartalmazó osztályok konstruktorai hozzák létre, az asszociációs kapcsolatokat speciális metódusok alakítják ki.



A többi metódus a kérdésre adott válaszáért felelős: ez a csapatok felsorolására épülő maximum kiválasztásba ágyazott dubla összegzés, ahol az összegzések felsorolják egy csapat összes futamának teljesítményét, azon belül az összes kategóriában elért eredményt.



A kategóriánkénti eredmény szorzótényezőjét származtatás segítségével specializáljuk (sablonfüggvény tervminta).

