Dokumentáció Nagybeadandóhoz

# Feladat:

A cég telephelyekkel, kamionokkal és sofőrökkel rendelkezik. Egy kamionoknak ismert a rendszáma, a tengelyeire vetített terhelhetősége, fogyasztása (100km-en), tengelyeinek száma (nyergesvontató: 3, fülkésvázas: 2). A kamionok vagy úton vannak, vagy az egyik telephelyen állnak. A sofőrök között vannak kezdők, több éves gyakorlattal rendelkezők, illetve a cég törzsgárdájához tartozók. A telephelyeket a címük azonosítja.

Amikor egy kamion egy fuvarra kap megbízást, akkor meg kell adni a fuvar távolságát, súlyát, díját (amit a cég kap a szállításért), az indulási dátumát, tervezet szállítási időt (órában), és a sofőrt. A fuvar teljesítése után ismert lesz annak érkezési ideje (addig ez egy extremális érték). Új megbízást nemcsak egy telephelyen állomásozó kamion kaphat, hanem egy már úton levő is, ha annak nincs teljesítendő fuvarja.

A sofőrök a teljesített fuvarjaik után kapnak bért, amelynek összege a fuvarral megtett távolságtól, a kamion típusától (nyerges, fülkés), illetve a sofőr besorolásától (kezdő, gyakorlott, törzstag) függ az alábbi módon: bér = távolság ∙ együttható.

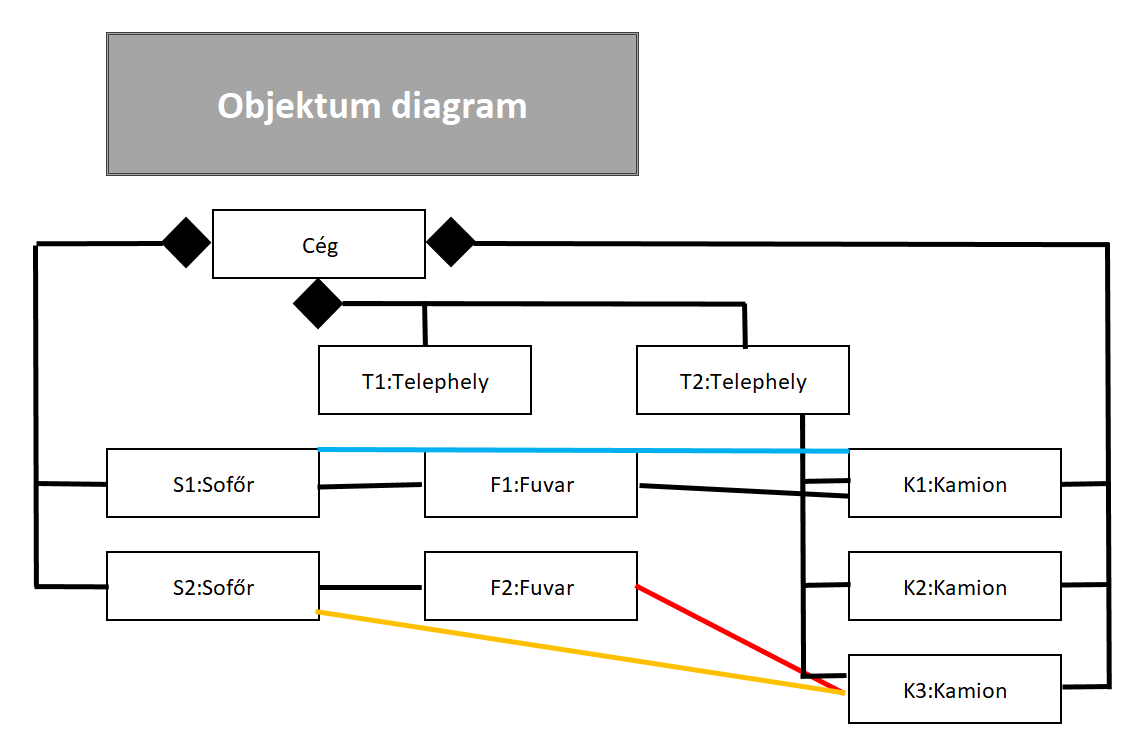
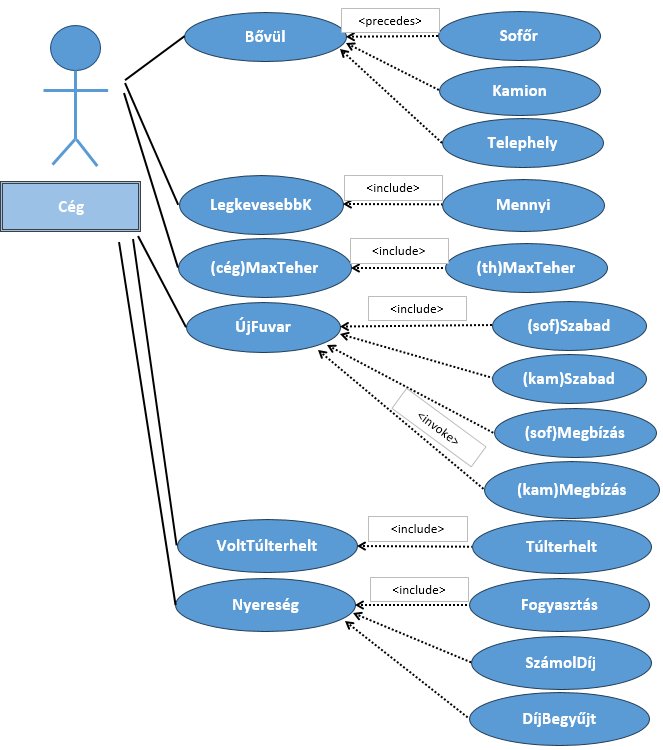
## Együttható táblázat:

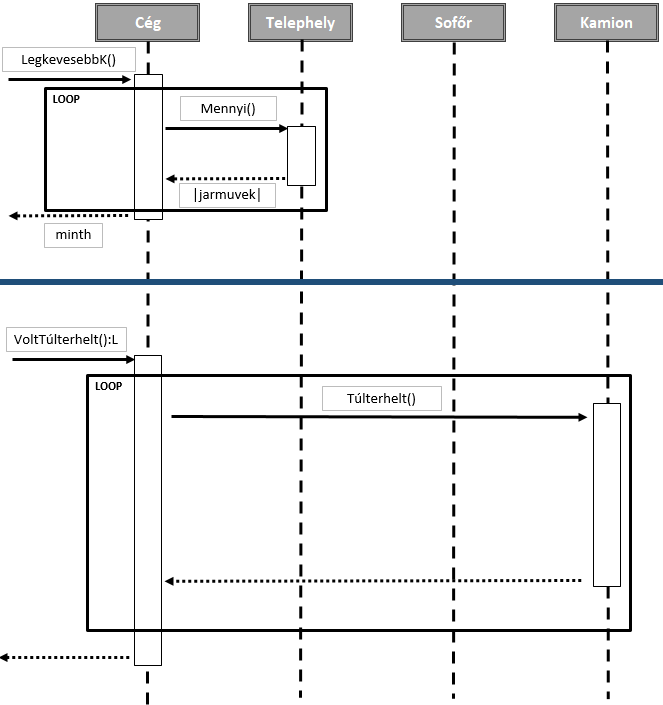
|  |  |  |  |
| --- | --- | --- | --- |
| Együttható | Kezdő | Gyakorlott | Törzstag |
| Nyerges | 25 | 35 | 40 |
| Fülkés | 20 | 30 | 40 |

## Kérdések:

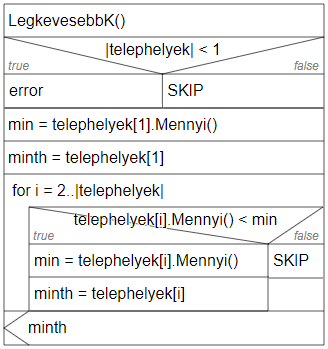
* Melyik telephelyhez tartozik éppen most a legtöbb kamion?
* Mekkora terhet képesek egyszerre elfuvarozni egy adott hely kamionjai?
* Volt-e olyan kamion, amely túl volt terhelve egy fuvarban?
* Mekkora volt a cég nyeresége? (nyereség = díj – bérek – üzemanyag)

# Terv:

A feladat megoldásának központi mozzanata az, hogy a cég kamionjainak – és sofőrjeinek - ki lehessen osztani különböző időpontokra megrendeléseket/fuvarokat, amiket aztán azok ki tudjanak szállítani, ezzel bevételt termelve a cégnek.

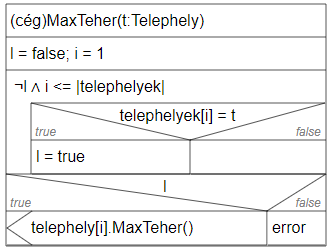
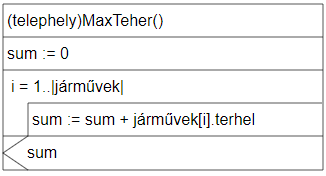
A szimuláció után, avagy alatt pedig a cég le tudja kérdezni a feladat kérdéseire a válaszokat. Például: melyik az a telephely ahol a legkevesebb kamionja állomásozik? Ehhez szüksége van a telephely Mennyi() metódusára, mely visszaadja a saját kamionjai számát. Ezt mindegyik telephelyre meghívva pedig csak egy minimumkeresést kell végrehajtania a cégnek és meg is kapta az eredményt. Egy másik eset pedig, hogy néhány, avagy az összes fuvar kiszállítása után a cég le tudja kérdezni, hogy volt-e túlterhelt kamion valamelyik kiszállítás során. Ehhez meghívja a VoltTúlterhelt() metódust, ami egy logikai értéket fog visszaadni. A metódus továbbá a kamionok Túlterhelt() metódusára hagyatkozik, ami egy kamion esetén összeveti a saját fuvarjaival a teherbírását addig, ameddig nem talál egy olyat, ahol túl volt terhelve.

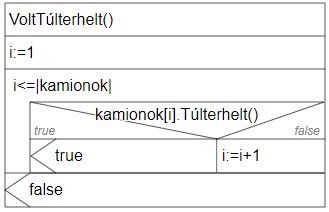
A legkevesebb kamion lekérdezés csak egy minimumkiválasztás a telephelyek „*járművek*” adattagjain, amit az alábbi módon specifikáljuk:

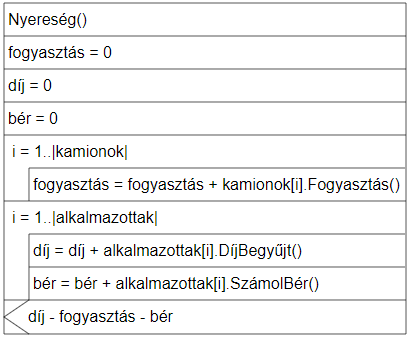
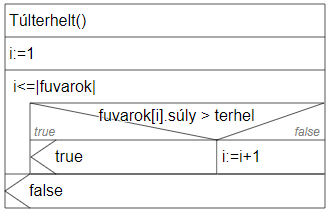
Ahol a Mennyi() metódus csupán lekérdezi a telephely „járművek” adattagjának hosszát.

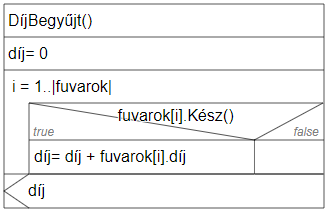
A második feladat arra kérdez rá, hogy egy megadott telephelyen mekkora súlyt tudnak egyszerre elszállítani az ott állomásozó kamionok. Evidens tehát, hogy a telephelyeknek írunk egy metódust, ami végig iterál az összes ott állomásozó kamionon és összegezi a terhelhetőségüket. Ezt az alábbi módon specifikáljuk:

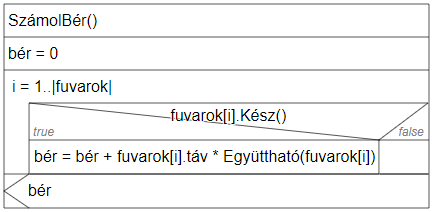
Itt pedig a cég meghívja a saját MaxTeher() metódusát, aminek átadja, melyik telephely esetén kíváncsi az értékre:

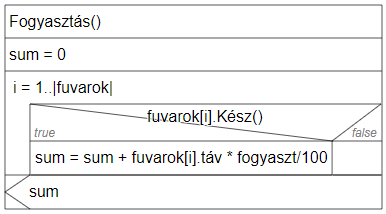
A harmadik feladatunk, hogy valamelyik kamionunk akármelyik fuvarja során túl volt-e terhelve. Ezt egyszerűen megoldhatjuk úgy, hogy az összes kamionra meghívjuk a Túlterhelt() metódusát, ami logikai értékként visszaadja a választ:

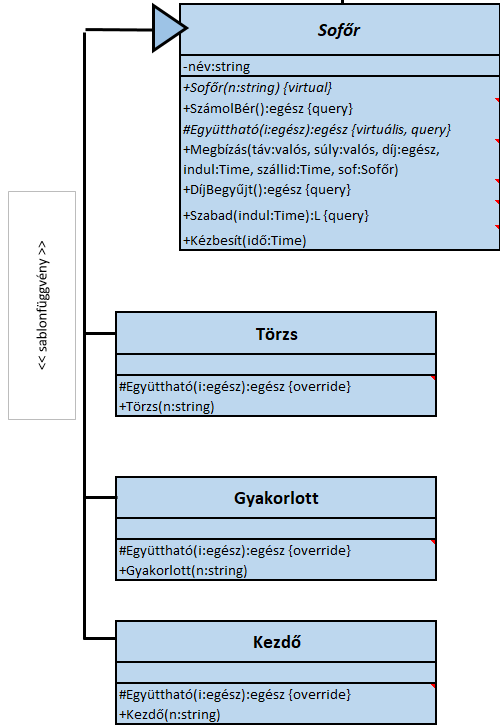
A Túlterhelt() eldönti, hogy túl volt-e terhelve a kamion úgy, hogy megnézi, hogy az egyes fuvarok közül valamelyiknek a súlya több volt-e, mint amennyi a kamionban nyilvántartott terhelhetőség:

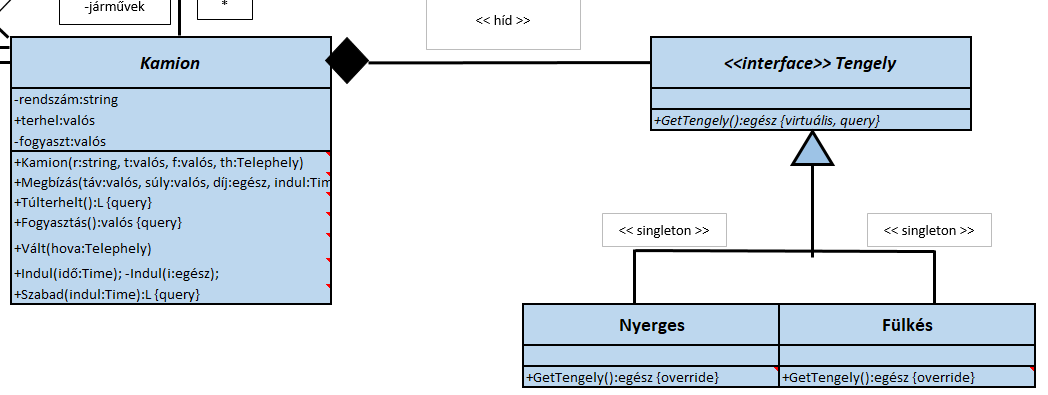
Végül az utolsó feladatunk, hogy kiszámoljuk a cég nyereségét az eddig kiszállított fuvarok alapján. Ehhez szükségünk lesz a kamionokra, hogy kiszámoljuk a fogyasztást valamint a sofőrökre is, hogy kiszámoljuk a béreket. Továbbá valamelyik számolás mellett célszerű a beszedett díjak összegét is számolni, ezt teljesen mindegy melyikük mellett csináljuk, hisz mind a kamionok, mind a sofőrök lefedik az összes beérkezett rendelést. Most a sofőröket választottam:

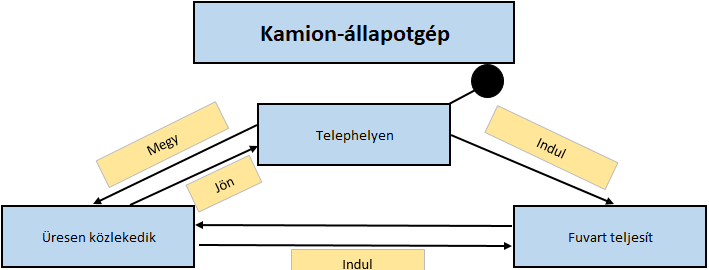
Ehhez három segédmetódust használok és mindhárom ugyanazon az elven működik. Mindhárom végig iterál az objektumának „*fuvarok*” adattagján, a DíjBegyűjt() és a SzámolBér() egy sofőr fuvarjain, a Fogyasztás() pedig egy kamion fuvarjain. A DíjBegyűjt() kiszámolja az egy sofőr által összegyűjtött díjat a kiszállított fuvarokért járó összegek összegzésével:

A SzámolBér() kiszámolja a kiszállított fuvarok alapján, hogy a sofőr tapasztalatától és a szállítás távolságától függően a sofőr milyen bérezésben részesül:

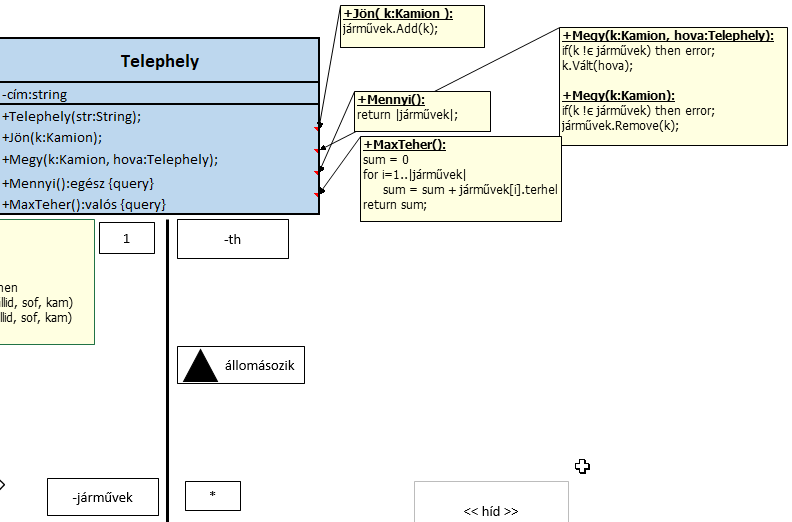
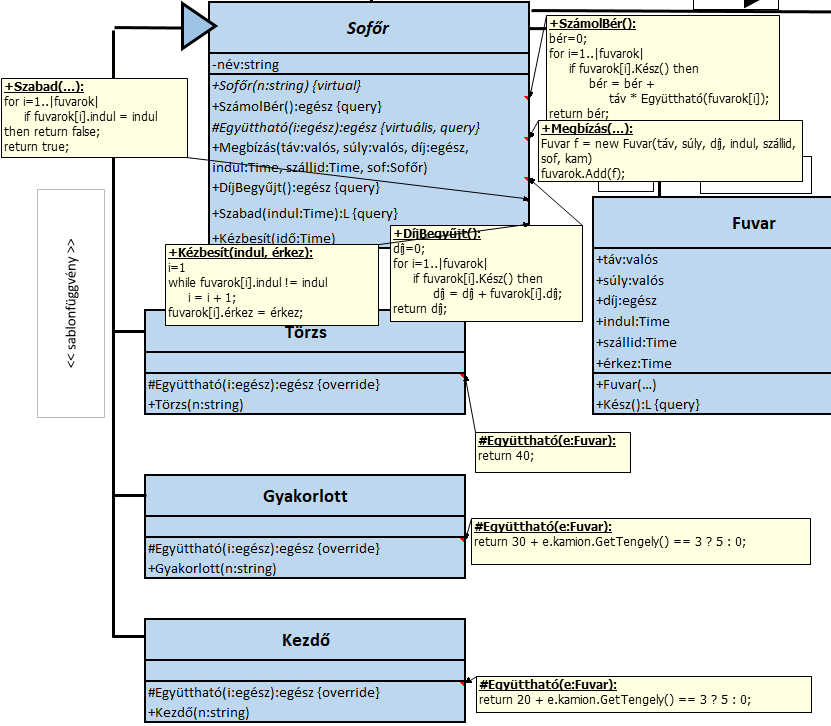
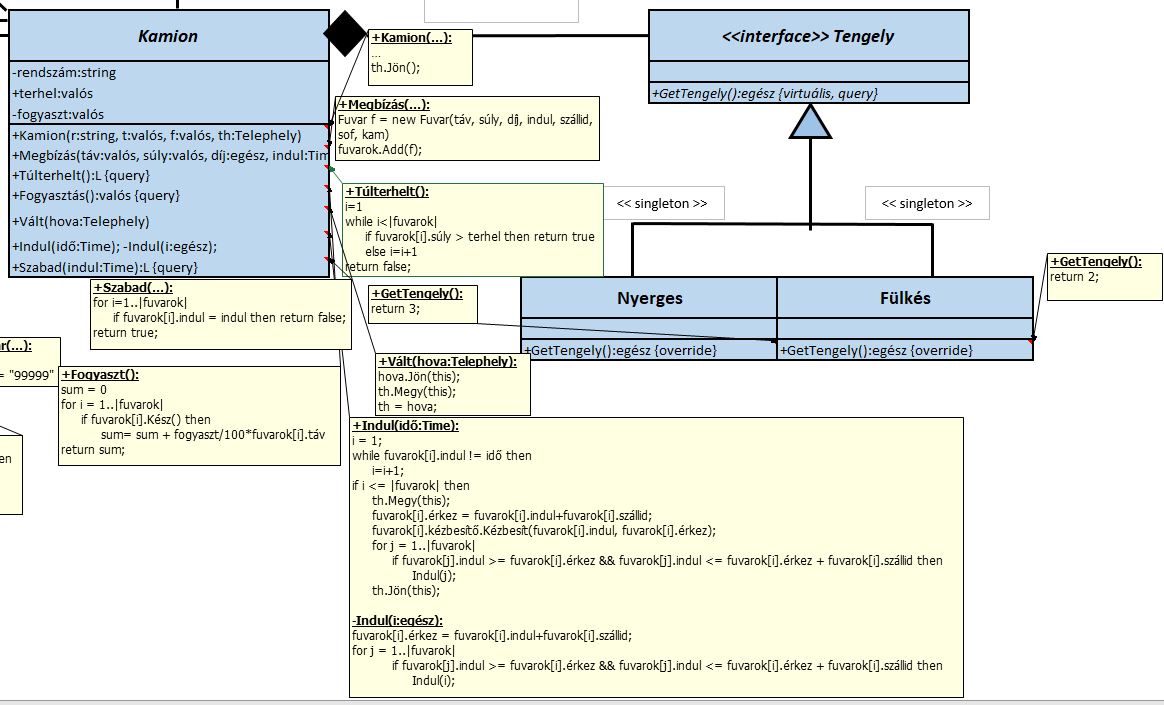
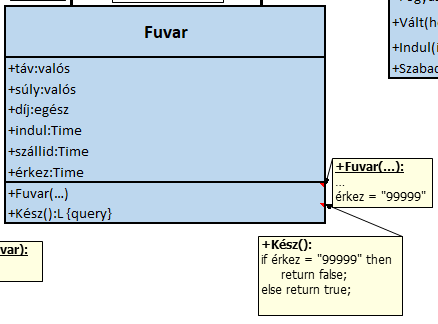
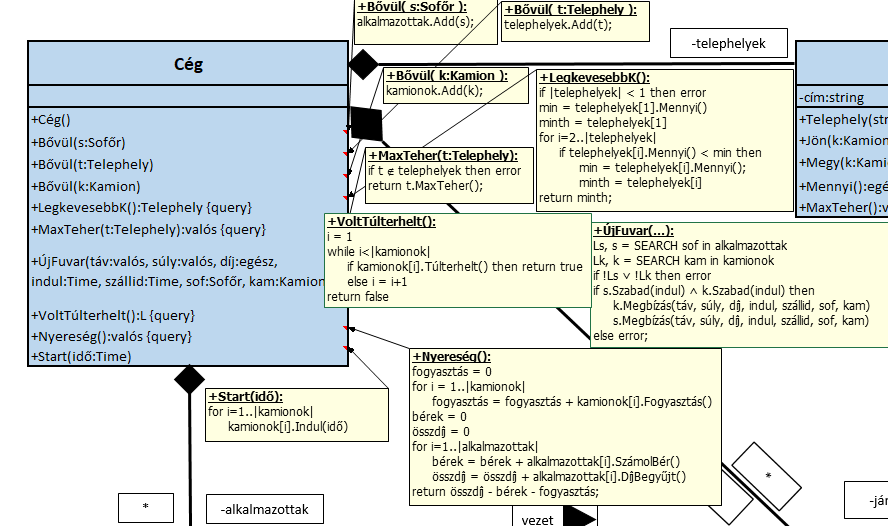
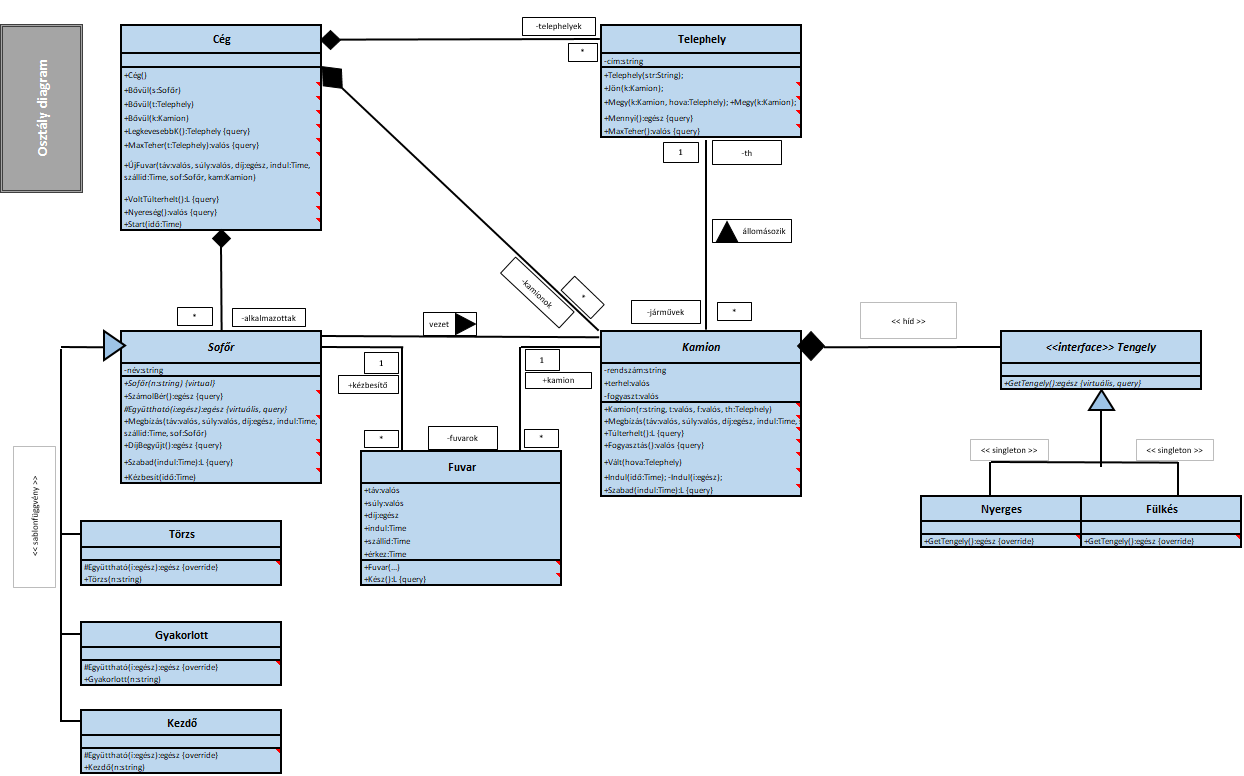
Végül a Fogyasztás() kiszámolja egy Kamion üzemanyagfogyasztását a kiszállított fuvarok távolsága és a 100km-re megadott üzemanyag fogyasztás alapján:

A SzámolBér()-nél a sofőrök tapasztalati besorolását alosztályokkal valósítjuk meg, és alkalmazzuk a sablonfüggvény tervezési mintát:

Egyúttal a kamionok típusait – nyergesvontató illetve fülkésvázas –híd és singleton tervezési minták kombinálásával valósítjuk meg:

A kamionoknak továbbá három állapotuk valósulhat meg szimuláció során:

# A teljes osztálydiagram:



# Tesztelési terv:

## LegkevesebbK() tesztelése:

* 0 telephely
* 1 telephely
* Több telephely

## MaxTeher() tesztelése:

* t nincs a telephelyek között
* t a telephelyek között van, nincs kamionja
* t a telephelyek között van, van kamionja
* t a telephelyek között van, a telephelyek között több Telephely is van, t-nek van több kamionja

## VoltTúlterhelt() tesztelése:

* Nem volt fuvar a kamionoknál egyáltalán – hamis
* Volt fuvar, de nem volt kiszállítva – hamis
* Volt fuvar kiszállítva, de nem volt túlterhelés - hamis
* Volt fuvar, volt túlterhelés

## 4.)Nyereség() tesztelése:

* Nem volt fuvar = 0
* Volt fuvar, de nem kiszállítva = 0
* Volt fuvar, kiszállították = x