Jeney Zoltán Márton (DB7ZTC) OEP 18.csoport 2024.05.07.

# Feladat

Egy-egy elfekből illetve orkokból álló csoport csatázik egymás ellen. A csata addig tart, amíg valamelyik csoport összes tagja meg nem hal. A csata több menetből is állhat. Egy menetben az elfek egymás után megküzdenek egyenként az orkokkal: először az elf támad az orkra, majd fordítva. Amikor valaki támad, akkor az ő támadó értékének és az ellenfél pajzs értékének különbségével csökken az ellenfél életereje. Akinek az életereje elfogy, az meghal, és a kincse a legyőzőjéé lesz.   
  
Az elfek minden küzdelem után gyógyíthatják magukat, ha van életelixírjük, vagy kincsük, amit elixírre tudnak átváltani.

Az elfeknek három típusa van: vakmerő, ügyes és bölcs. Az elfeknek ismerjük a nevét, fajtáját, életerejét (0 .. max közötti egész szám, a max az elf típusától függő konstans), támadó és pajzs értékét (típusától függő konstans természetes számok), életelixírjét és kincsét (természetes számok). Kezdetben nincs kincsük.   
Az orkoknak három típusa van: vérengző, ravasz és óvatos. Az orkoknak ismerjük a nevét, fajtáját, életerejét (0 .. max közötti egész szám, a max az ork típusától függő konstans), támadó és pajzs értékét (típusától függő konstans természetes számok), és kincsét (természetes szám).

**Elfek tulajdonságai:**

A vakmerő maximális életereje 100; az életelixírjéből csak annyit használ fel, hogy életereje ne legyen 30 pont alatt, ha kincshez jut, azt soha nem költi életelixírre. Támadó értéke 30, pajzsa 10.

Az ügyes maximális életereje 80; az életelixírjéből annyit használ fel, hogy életereje ne legyen 50 pont alatt, ha van kincse egy küzdelem után, annak felét életelixírre váltja. Támadó értéke 20, pajzsa 20.

A bölcs maximális életereje 60; minden küzdelem után feljavítja az életerejét amennyire csak lehet, ha kincshez jut, azt mind életelixírre váltja. Támadó értéke 10, pajzsa 10.

**Orkok tulajdonságai**:

A vérengző maximális életereje 100. Támadó értéke 30, pajzsa 5.

A ravasz maximális életereje 90. Támadó értéke 15, pajzsa 20.

Az óvatos maximális életereje 80. Támadó értéke 10, pajzsa 15.   
A kincs, az életelixír, és az életerő egy az egyben váltható át.

Készítsen használati eset diagramot, ahol egy elf és egy ork szempontjából lényeges eseteket, valamint ezek kapcsolatát ábrázolja. Adja meg egy elf és egy ork komunikációjának szekvencia diagramját. Készítse el az osztály diagramot! Használjon sablonfüggvény és látogató tervezési mintákat.

Implementálja a modellt, és oldja meg az alábbi feladatot: Szimuláljunk egy harcot, és minden menet után jelenítsük meg az életben maradt harcosokat a nevük, fajtájuk, életerejük, elixírjük és kincsük kiírásával! A program egy szövegfájlból olvassa be az adatokat! Ennek minden sora egy harcos adatait tartalmazza szóközökkel elválasztva. Egy harcosnál (egy sorban) megadjuk a harcos nevét (sztring), fajtájára utaló két karaktert (e – elf, o – ork, v – vakmerő v. vérszomjas, u – ügyes, b – bölcs, r – ravasz, o – óvatos), orkok esetén a kincsüket (természetes szám). (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet: Falánk o v 7 Sudár e u Köpcös o v 4 Nyúlánk e v Készítsen teszteseteket, és hozzon létre ezek kipróbálására automatikusan tesztkörnyezetet!

Az alábbi táblázat összefoglalja a lények típusait és azok tulajdonságait.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Elfek** | **Maximális életerő** | **Elixir felhasználáséhoz életerő kritérium** | **Kincs költési mennyiség** | **Támadó érték** | **Pajzs érték** |
| Vakmerő | 100 | 30 | 0 | 30 | 10 |
| Ügyes | 80 | 50 | 50% | 20 | 20 |
| Bölcs | 60 | 60 (100%) | 100% | 10 | 10 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Orkok** | **Maximális életerő** | **Támadó érték** | **Pajzs** |
| Vérengző | 100 | 30 | 5 |
| Ravasz | 90 | 15 | 20 |
| Óvatos | 80 | 10 | 15 |

Az alábbi táblázatok pedig összefoglalják a látogató tervmintához felhasznált életerő csökkenés értékeket, az összes lehetséges párosításnál:  
  
**Orkok támadása esetén:**

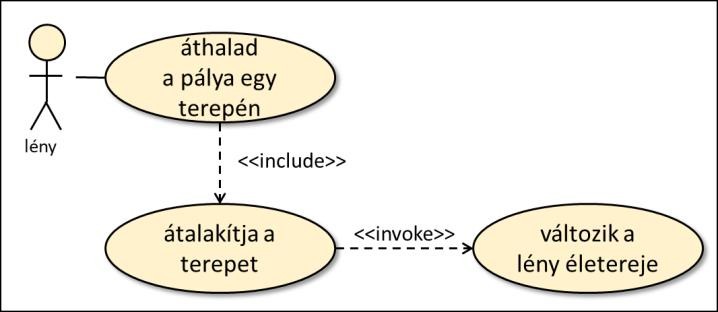
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ravasz- >Bölcs | 5 | Ravasz->Ügyes | 0 | Ravasz->Vakmerő | 5 |
| Óvatos-> Bölcs | 0 | Óvatos-> Ügyes | 0 | Óvatos-> Vakmerő | 0 |
| Vérengző-> Bölcs | 20 | Vérengző-> Ügyes | 10 | Vérengző-> Vakmerő | 20 |

**Elfek támadása esetén:**

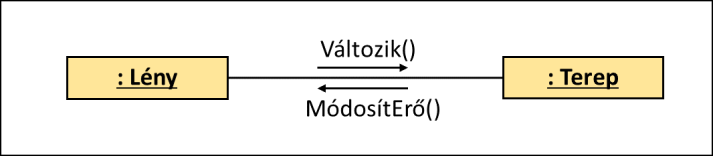
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Bölcs->Ravasz | 0 | Bölcs->Óvatos | 0 | Bölcs->Vérengző | 5 |
| Ügyes-> Ravasz | 0 | Ügyes-> Óvatos | 5 | Ügyes-> Vérengző | 15 |
| Vakmerő-> Ravasz | 10 | Vakmerő-> Óvatos | 15 | Vakmerő-> Vérengző | 25 |

# Terv

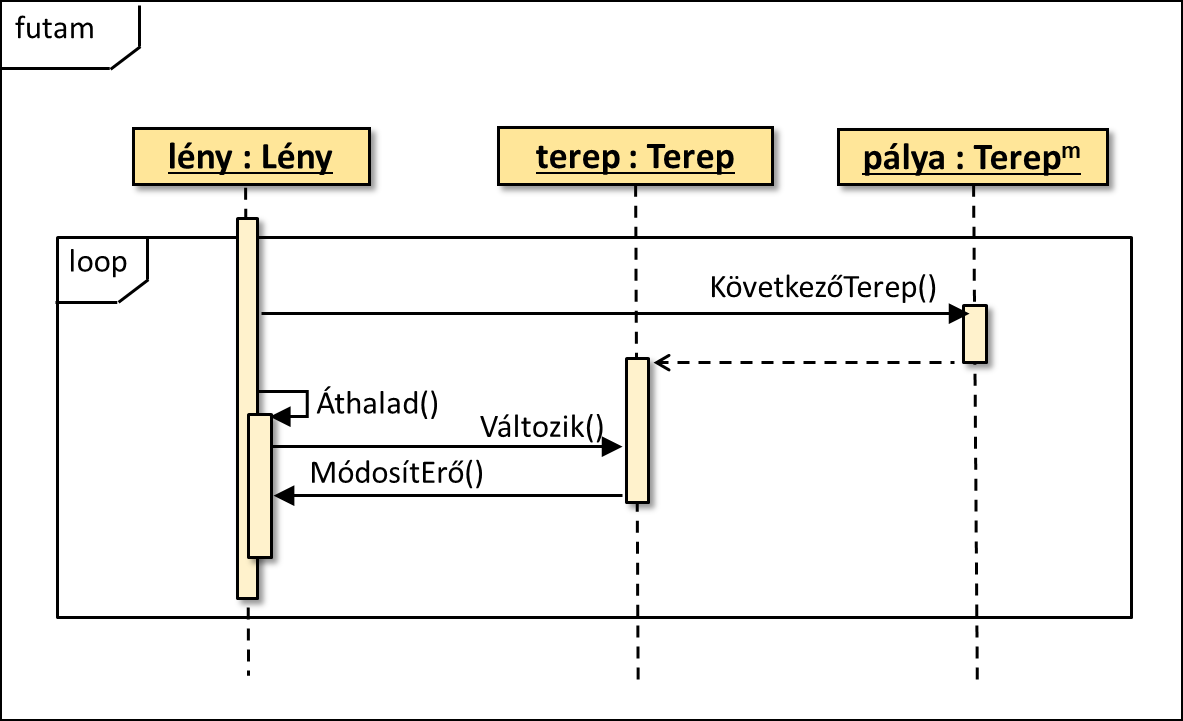
A feladat megoldásának központi mozzanata az, amikor egy lény a futama során áthalad a pálya egy terepén, ezzel megváltoztathatja a terepet, és közben a lény (életereje) is módosulhat a tereptől függően.



Egy lénynek a pálya terepein való áthaladása során a lény és egy terep között zajlik kommunikáció.



A lény meghívja a terep *Változik()* metódusát átadva neki önmagát (a lényt), amely megváltoztathatja a terepet, és a lény életerejét is módosíthatja a lény a *MódosítErő()* metódusának hívásával. Ezt a folyamatot a lény *Áthalad()* metódushívása indítja el, amelyet ciklikusan kell majd meghívni. Ezt írja le az alábbi szekvencia diagram.



Egy lény teljes futama (*Futam()*) során a pálya egymást követő szakaszaira kell az *Áthalad(terep)*

metódust meghívni, feltéve, hogy a lény él. Egy lény futamát az alábbi módon specifikáljuk:

*A* = ( pálya: Terepm, lény: Lény )

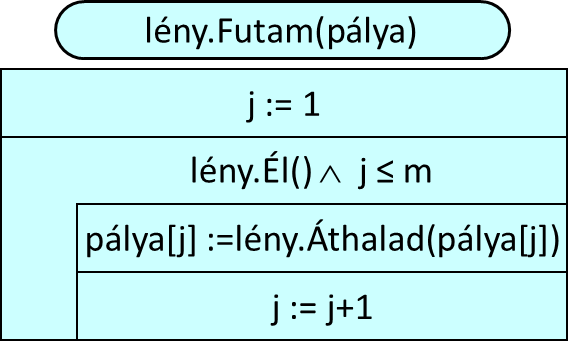
*Ef* = ( pálya = pálya’  lény = lény0 )

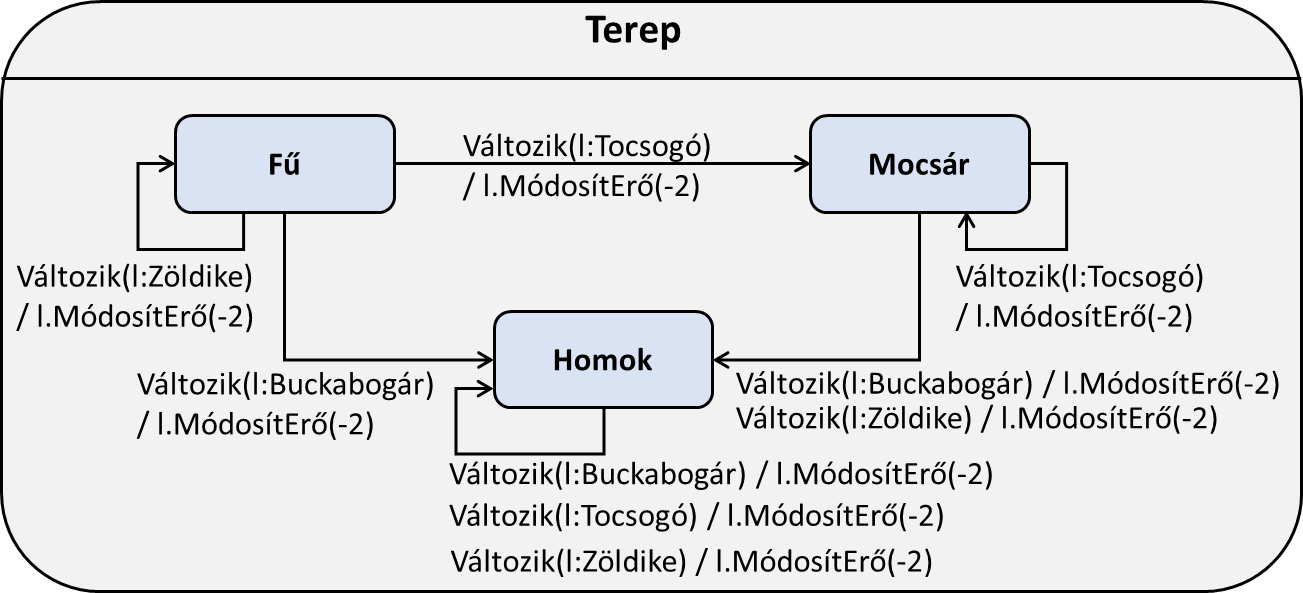
*Uf* = ( lény = lénym  j[1..m]: Él(lényj-1) → (lényj, pálya[j]) = Áthalad(lényj-1, pálya’[j])

 Él(lényj-1) → (lényj, pálya[j]) = (lényj-1, pálya[j]) )

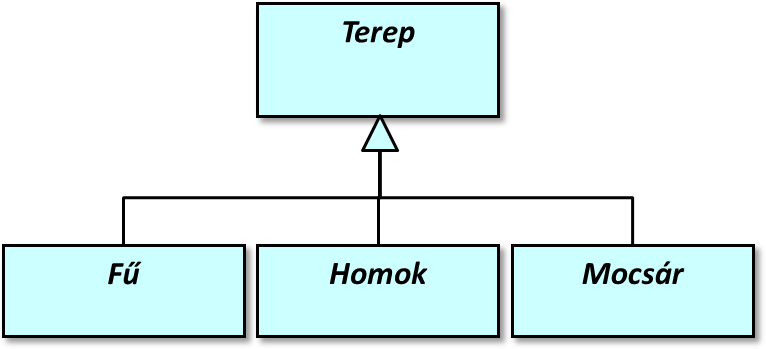
A specifikációban az alábbi jelöléseket használjuk: pálya’ ~ a pálya állapota a lény futama előtt lény0 ~ a lény állapota a lény futama előtt

lényj-1 ~ a lény a *j*-dik terepen való áthaladás előtt lényj ~ a lény a *j*-dik terepen való áthaladás után

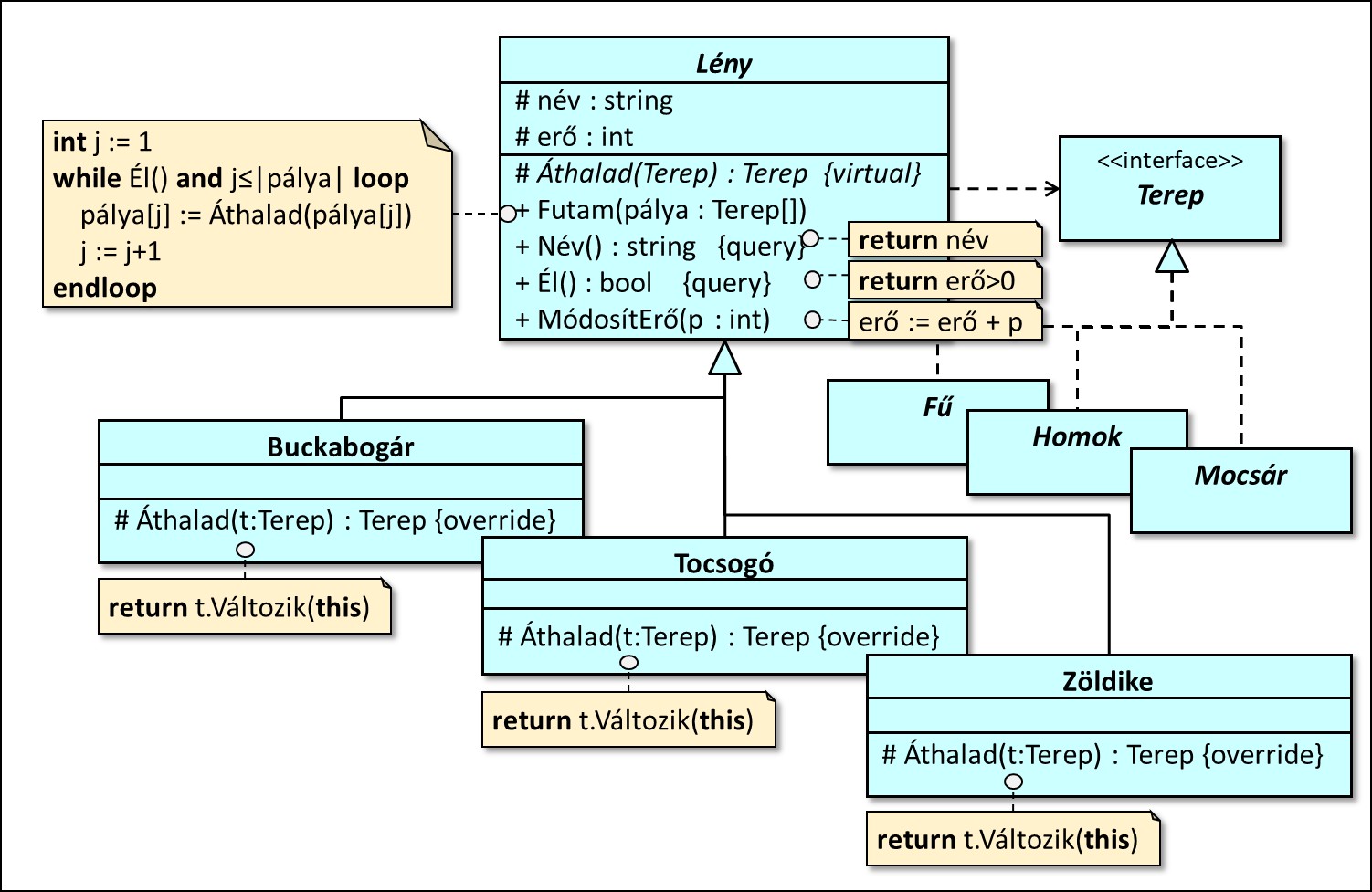
A feladat visszavezethető egy feltételig tartó (amíg a lény él) dupla összegzésre. Az egyik összegzés lépésről lépésre változtatja meg a lényt, a másik a pálya terepeit. Ez első összegzés műveleténél (neve: „teljes csere”, jele: ⊜) – amely egy lényt a megváltozott lényre cserél le, azaz két elem közül mindig a másodikat adja vissza – minden elem egyben neutrális elem is. A másik összegzés egy összefűzés, amely új pályát épít a megváltozott terepek egymás után fűzésével, de ez implementálható úgy is, hogy a terepek tömbjének (ez a pálya) az elemeit írjuk felül. Az algoritmus objektumorientált stílusban:

Egy terepnek három állapota van: fű, homok, mocsár. Ez az állapot attól függően változik, hogy milyen fajtájú lény jelenik meg rajta. Ezt a változást a terep *Változik()* metódusa végzi, amely hatással van a terepen megjelenő lény életerejére is, amelyet annak *MódosítErő()* metódusának meghívásával tud befolyásolni.

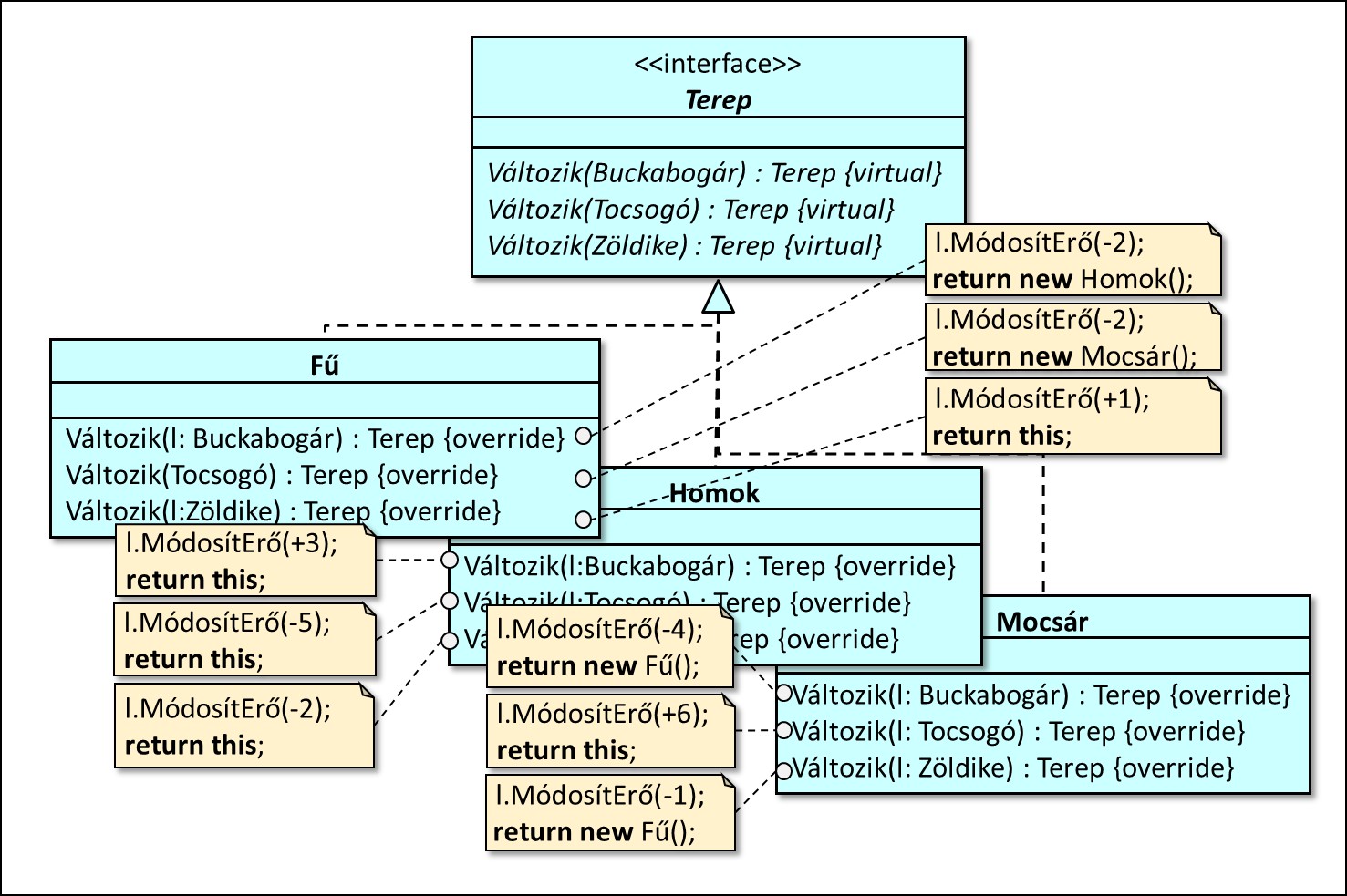
A tereptípusokat alosztályok bevezetésével adhatjuk meg.



A lények fajait is származtatással adjuk meg,



Mivel az *Áthalad()* metódus egyszerre függ az adott lény fajától és a terep típusától, ezért alkalmazzuk a látógató tervezési mintát: egy lénynek egy terep objektum a „látogatója”. Minden alosztály annyi Változik metódust tartalmaz, ahányféle lény van.



A tereptípus objektumok nem tartalmaznak adattagokat, így egy tereptípusból elég egyetlen objektumot példányosítani: legyenek a tereptípusok osztályai az „egyke” tervmintának megfelelőek.

A teljes verseny specifikációja:

*A* = ( lények: Lényn, pálya: Terepm, túlélők: String\*)

*Ef* = ( lények = lények0  pálya = pálya0 )

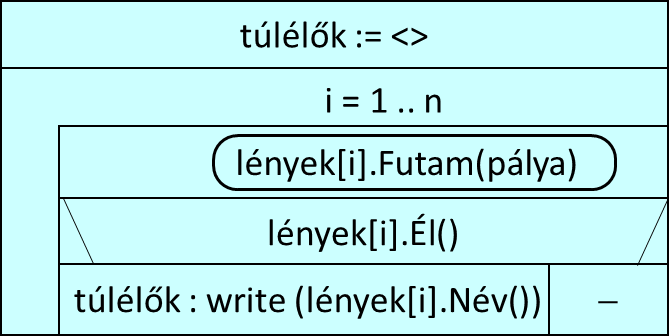
*Uf* = ( pálya = pályan  i[1..n]: (lények[i], pályai) = futam(lények0[i], pályai-1)

 túlélők = ⊕i=1..n <lények[i].név()> )

lények[i].él()

A specifikációban az alábbi jelöléseket használjuk: lények0 ~ a lények állapota a verseny előtt pálya0 ~ a pálya állapota a verseny előtt

pályai-1 ~ a pálya az *i*-dik lény áthaladása előtt pályai ~ a pálya az *i*-dik lény áthaladása után

A feladat visszavezethető egy dupla összegzésre és egy kiválogatásra (ami szintén egy összegzés), amelyeket közös ciklusba vonunk össze. A dupla összegzés egyik komponense a lényeket változtatja meg (ez felfogható egy összefűzésnek, azaz a megváltozott lények újra fűzésének, noha technikailag a lények sorozatának egy elemeit változtatjuk meg sorban egymás után). A másik komponens a pálya újra és újra megismételt megváltoztatása, azaz teljes cseréje. Az algoritmus:

# Tesztelési terv

*Verseny (3 darab összegzés)*

1. felsorolás hossza szerint:
   * nulla lény
   * egy lény
   * több lény
2. felsorolás eleje, illetve vége:
   * első lény túléli, illetve nem éli túl a versenyt
   * utolsó lény túléli, illetve nem éli túl a versenyt
3. A túlélők kiválogatása feltételének vizsgálata (legyen túlélő és elpusztuló lény is)

*Futam (2 darab összegzés)*

1. felsorolás hossza szerint:
   * nulla hosszú pálya
   * egy hosszú pálya (olyan lénnyel, aki túléli a versenyt)
   * hosszabb pálya (olyan lénnyel, aki túléli a versenyt; illetve olyannal, aki nem)
2. felsorolás eleje, illetve vége:
   * a pálya legelső mezője megfelelően változik egy lény hatására
   * a pálya legutolsó mezője megfelelően változik egy lény hatására
3. Ellenőrizzük a pálya változását olyan lényre, aki túléli a versenyt (végigmegy), és olyanra is, aki nem.

*Az Áthalad() függvény vizsgálata*

Kilenc különböző esete van a résztvevő lény és a pályaszakasz függvényében.