

Feladat

Különféle élőlények egy túlélési versenyen vesznek részt. A lények egy pályán haladnak végig, ahol váltakozó terep viszonyok vannak. Egy lénynek a terep fajtájától függően változik az életereje, miközben azon keresztül halad, és közben a terep fajtáját is átalakítja. Először az első lény próbál végig jutni a pályán, hacsak el nem fogy az életereje és elpusztul. Egy terep akkor is átalakul, ha azon áthaladva egyébként a lény elpusztul. Ezen az átalakított pályán indul a második lény, majd sorban a többi. Adjuk meg a pályán végig jutó, életben maradt lények neveit!

A pályán három fajta terep fordulhat elő: fű, homok, mocsár. A lények különböző fajokhoz tartoznak. Három féle fajt különböztetünk meg.

Zöldike: fűvön az életereje eggyel nő, homokon kettővel csökken, mocsárban eggyel csökken; a mocsaras terepet fűvé alakítja, a másik két terep fajtát nem változtatja meg.

Buckabogár: fűvön az ereje kettővel csökken, homokon hárommal nő, mocsárban négyvel csökken; a fűvet homokká, a mocsarat fűvé alakítja, de a homokot nem változtatja meg.

Tocsogó: fűvön az életereje kettővel, homokon öttel csökken, mocsárban hattal nő; a fűvet mocsárrá alakítja, a másik két fajta terepet nem változtatja meg.

Minden lénynek van neve (sztring), ismert az aktuális életereje (egész szám) és a fajtája. Amíg az életereje pozitív, addig a lény életben van.

A verseny adatait egy szövegfájlból olvassuk be! A fájl első sora tartalmazza a lények számát, amelyet a lények adatai követnek. Egy karakter azonosítja a lény fajtáját, amit szóköz után a lény neve követ, majd a lény kezdeti életereje. Az azonosítók: Z – zöldike, B – buckabogár, T – tocsogó. A lények után következik a pálya leírása. Egy egész szám adja meg a pálya hosszát, amit pontosan ennyi szóközökkel elválasztott egész szám követ, amelyek a terepek fajtái adják meg. A fajták azonosítói: 0 – homok, 1 – fű, 2 – mocsár. Feltehetjük, hogy a fájl formátuma helyes.

Elemzés¹

A feladat önálló objektumai a különféle tulajdonságokkal rendelkező lények, amelyek három csoportba sorolhatók: zöldikék, a buckabogarak és a tocsogók.

Mindegyiknek van neve és életereje, meg lehet róla kérdezni, hogy hívják, él-e még (az életereje pozitív-e), és meg lehet vizsgálni, mi történik, ha a versenypálya egy bizonyos terepén áthalad. Egy terepen történő áthaladás az alábbi módon hat a lényre illetve a terepre:

Zöldikék esetén, feltéve, hogy az életerő pozitív:

terep	életerő változás	terepváltozás
homok	-2	-
fű	+1	-
mocsár	-1	fű

¹ Ez az elemzés rész a hallgatói beadandó dokumentációjából elhagyható, az átalakítási táblázatokat a tervezés részben elég lehet tüntetni.

Buckabogarak esetén, feltéve, hogy az életerő pozitív:

terep	életerő változás	terepváltozás
homok	+3	-
fű	-2	homok
mocsár	-4	fű

Tocsogók esetén, feltéve, hogy az életerő pozitív:

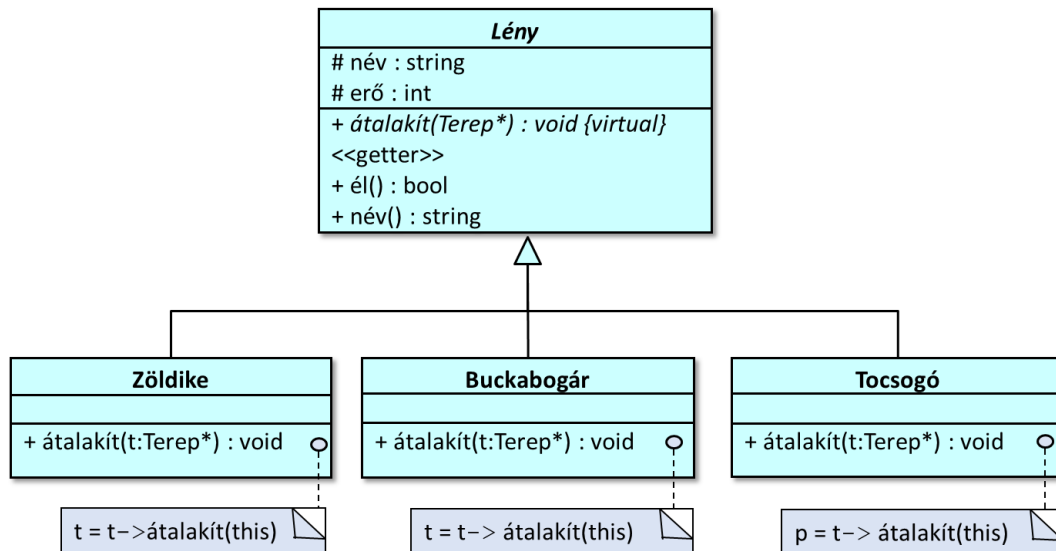
terep	életerő változás	terepváltozás
homok	-5	-
fű	-2	mocsár
mocsár	+6	-

Terv²

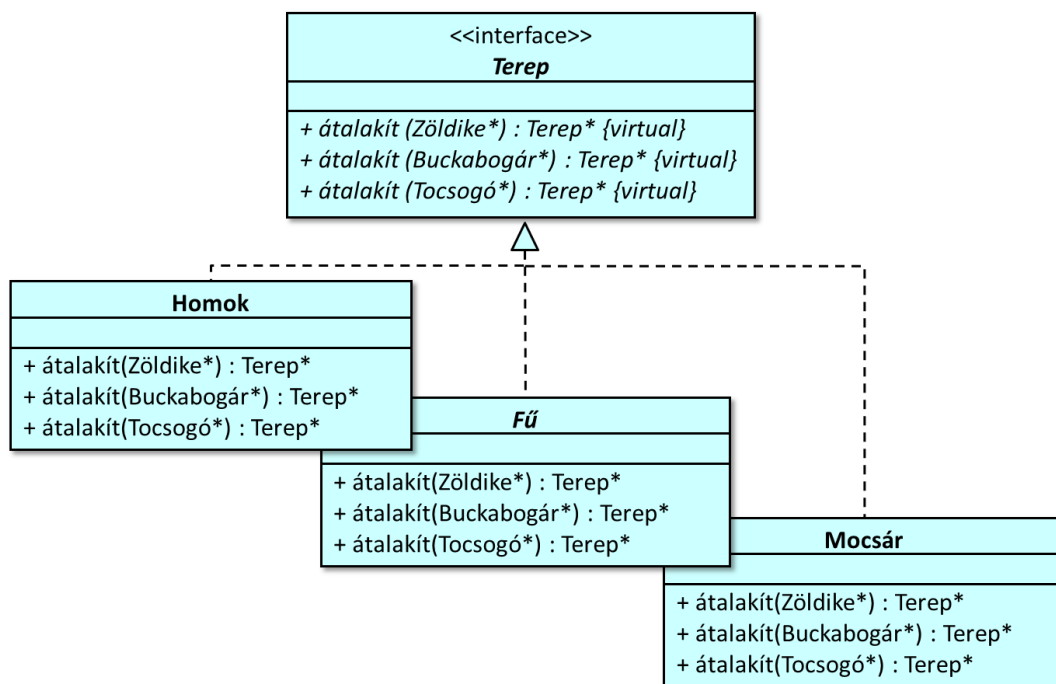
A lények leírásához bevezetünk négy osztályt: a lények általános tulajdonságait megadó őosztályt (*Lény*), és ebből származtatjuk a konkrét fajú lények, a zöldikék, a buckabogarak és a tocsogók osztályait. Attól függetlenül, hogy egy lény konkrétan kicsoda vagy mi a fajtája, számos közös tulajdonsággal rendelkezik. Mindegyiknek van neve (*_név*) és életereje (*_erő*), meg lehet róla kérdezni, hogy hívják (*név()*), él-e (*él()*) még (az életereje pozitív-e), és meg lehet vizsgálni, mi történik, ha a versenypálya egy bizonyos terepén áthalad. Ez utóbbi művelet (*átalakít()*) módosítja a lény életerejét, és átalakítja a neki átadott terepet. Az *él()* és *név()* metódusok már az őosztály szintjén implementálhatók, de az *átalakít()* csak a konkrét osztályok szintjén. Ennek hatása ugyanis attól függ, hogy egy lény milyen fajú. Ennél fogva az általános lény típusát leíró osztály absztrakt lesz, hiszen az *átalakít()* metódus absztrakt, másrészt úgysem akarunk ilyen objektumot létrehozni. A terepek általános tulajdonságait egy őosztály (*Terep*) írja le, és ebből származtatjuk a konkrét terepek, a homok, a fű, a mocsár osztályait. Minden konkrét terep három-három metódussal bír: ezek rendre azt mutatják meg, hogyan változik az adott terepen egy zöldike, egy tocsogó, illetve egy buckabogár, és hogyan alakul át eközben a terep. Az objektumokra pointerekkel hivatkozunk.

A speciális lény osztályok konstruktorai az őosztály konstruktorával inicializálják a nevet és az életerőt, és egyedi módon felüldefiniáljuk az *átalakít()* metódust. Az inicializáláshoz és a felüldefiniáláshoz a feladat szövegéből kinyert táblázatok (lásd elemzés) ad útmutatást. A táblázatok szerint az *átalakít()* metódusban elágazásokat kell elhelyezni, amelynek feltételei a paraméterként megadott terep típusát (fajtáját) vizsgálják. Az elágazások létrehozása azonban nem szerencsés, ha fel kell készítenünk a programunkat arra, hogy később új terep fajtákkal is bővíthet a feladat. Ekkor ugyanis az összes konkrét lény *átalakít()* metódusát módosítani kell. Ezt elkerülendő a „látogató” tervmintát alkalmazzuk úgy, hogy a terepek leírására szolgáló osztályok töltsék majd be a látogató szerepét.

² A szöveges magyarázatra a hallgatói beadandók dokumentumaiban nincs szükség.



A konkrét lények *átalakít()* metódusa paraméterként vár egy terep objektumot, mint látogatót, és ennek a konkrét lény típusához illeszkedő metódusát hívja vissza.



A terepek összes osztályát az „egyke” tervmintának megfelelően valósítjuk meg, hiszen elég mindegyik terepfajta-hoz egy-egy objektumot létrehozni.

A feladat specifikációjában szükségünk lesz a pályának az $n+1$ darab különböző változatával számolni, hiszen mindegyik lény átalakítja a pályát (a nulladik változat a kezdeti pálya). Egy-egy lény áthaladását az $\text{áthalad} : \text{Lény} \times \text{Terep}^m \rightarrow \text{Lény} \times \text{Terep}^m$ függvénnyel jelöljük, amely megadja a megváltozott lényt és a megváltozott pályát. A pálya i -edik változatát a pálya_i -vel jelöljük, de ez a programban már nem jelenik meg, csak valamikor értéke lesz a pálya változónak.

Állapottér: $pálya: Terep^m, lény: Lény^n, túlélők: String^*$
 Előfeltétel: $lény = lény_0 \wedge pálya = pálya_0$
 Utófeltétel: $pálya = pálya_n \wedge$
 $\forall i \in [1..n]: lény[i], pálya_i = áthalad(lény_0[i], pálya_{i-1}) \wedge$
 $túlélők = \bigoplus_{i=1}^n \langle lény[i].név \rangle$
 $lény[i].él()$

A lények (áthaladás utáni) új állapotainak összefűzése és a pálya fokozatos (a lények egymás utáni áthaladása általi) átalakítása egy-egy összegzésre vezethető vissza, akárcsak a túlélők kiválogatása. Mivel ezek mindegyike ugyanarra a felsorolóra ($i = 1 .. n$) épül, egy közös ciklusba összevonhatóak.

enor(E)	$i = 1 .. n$
f(e)	$áthalad(lény[i], pálya)_1$
s	lény
H, +, 0	$Lény^*, \oplus, \langle \rangle$

$áthalad()$ függvény értékének első komponense

enor(E)	$i = 1 .. n$
f(e)	$áthalad(lény[i], pálya)_2$
s	pálya
H, +, 0	$Lény^*, \ominus, pálya$

$áthalad()$ függvény értékének második komponense

$a \ominus b ::= b$

enor(E)	$i = 1 .. n$
f(e)	$\langle lény[i] \rangle$ ha $lény[i].él()$
s	túlélők
H, +, 0	$Lény^*, \oplus, \langle \rangle$

Összevonva egy közös ciklusba a fentieket megkapjuk a megoldást:

$túlélők := \langle \rangle$	
$i = 1 .. n$	
$lény[i], pálya := áthalad(lény[i], pálya)$	
$lény[i].él()$	
$túlélők := túlélők \oplus lény[i].név()$	SKIP

Az i -edik lénynek ($lény[i]$) a pályán való áthaladása során $m+1$ darab különböző állapota lesz, miközben a *pálya* lényegében újra épül (A $pálya_{i-1}$ -ből $pálya_i$ lesz). A $lény[i]$ 0-dik állapota a kezdetben megadott lény ($lény[i]_0$), a végállapota pedig az m -dik pályaszakaszon való áthaladás után előállt állapot. Az i -edik lénynek a pályán történő j -edik lépése előtti állapotát a $lény[i]_{j-1}$ jelöli, amelyből az *átalakít()* metódus állítja elő a lény következő állapotát ($lény[i]_j$), miközben a $pálya_{i-1}[j]$ -ből $pálya_i[j]$ lesz. Tehát a megoldandó feladat:

$$\forall j \in [1..m]: lény[i]_j, pálya_i[j] = átalakít(lény[i]_{j-1}, pálya_{i-1}[j]) \wedge lény[i] = lény[i]_m$$

A lény áthaladása a pályán a lény állapotának fokozatos átalakítását jelenti, amellyel együtt a megváltozott pályaszakaszok összefűzésére is sor kerül. Mindkettő egy összegzés, amelynek ugyanaz a felsorolója ($j=1 .. m$):

enor(E)	$j = 1 .. m$	
$f(e)$	$átalakít(lény[i], pálya)_1$	$átalakít()$ függvény értékének első komponense
s	$lény[i]$	
H, +, 0	$Lény^*, \ominus, lény[i]$	$a \ominus b ::= b$

enor(E)	$i = 1 .. n$	
$f(e)$	$átalakít(lény[i], pálya)_2$	$átalakít()$ függvény értékének második komponense
s	$pálya$	
H, +, 0	$Lény^*, \oplus, <>$	

Tehát összevonva egy közös ciklusba:

$lény[i], pálya := áthalad(lény[i], pálya)$

$j = 1 .. m$
$lény[i], pálya[j] := átalakít(lény[i], pálya[j])$

Ha még azt is észre vesszük, hogy sem a $lény[i]$, sem a további terepek nem változnak azután, hogy a $lény[i]$ elpusztul, javíthatjuk a fenti algoritmus hatékonyságát:

$lény[i], pálya := áthalad(lény[i], pálya)$

$j := 1$
$lény[i].él() \wedge j \leq m$
$lény[i], pálya[j] := átalakít(lény[i], pálya[j])$
$j := j + 1$

Tesztelési terv

Szürke doboz tesztesetek:

Külső ciklus (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

Belső ciklus (összegzés)

1. felsorolás hossza szerint:
 - egyetlen lény nulla hosszú pályán
 - egyetlen lény egy hosszú pályán megfelelően változik
 - egyetlen lény hosszabb pályán (túléli, illetve elpusztul)
2. felsorolás eleje, illetve vége:
 - a pálya legelső mezője megfelelően változik egy lény hatására
 - a pálya legutolsó mezője megfelelően változik egy lény hatására

Az átalakít() függvény vizsgálata

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