

Beadandó I

Magyar Tamás | RNYR2F

(1. Feladat)

Tartalom

Feladat Leírása	2
UML Diagram	3
Metódusok rövid leírása	4
Game::Load() : void.....	4
Game::Play() : Entity	4
Weather::fromChar(c : char) : Weather.....	4
Entity::isAlive() : bool	4
Entity::getMaxWater() : int {abstract}	4
Entity::apply(condition : Weather) {abstract} és (water : int, distance : int): void.....	4
Entity::Create(scanner : Scanner) : Entity {static}.....	4

Feladat Leírása

Egy többnapos versenyen lények vesznek részt. **Ki nyeri a versenyt, azaz melyik lény teszi meg a legnagyobb távolságot úgy, hogy közben életben marad?** Kezdetben minden lény valamennyi vízzel rendelkezik, és a megtett távolsága 0. A verseny során háromféle nap lehetséges: napos, felhős és esős. Ezekre a különböző fajtájú lények eltérő módon reagálnak vízfogyasztás és haladás szempontjából. Minden lény először a rendelkezésére álló víz mennyiségét változtatja meg, ezután ha tud, mozog. Bármely lény elpusztul, ha a vize elfogy (0 lesz az érték), ezután értelemszerűen semmilyen tevékenységre sem képes. Minden lény jellemzői: az egyedi neve (string), a rendelkezésre álló víz mennyisége (egész), a maximálisan tárolható víz mennyisége (egész), hogy él-e (logikai), illetve az eddig megtett távolság (egész). A versenyen részt vevő lények fajtái a következők: homokjáró, szivacs, lépegető.

fajta	Víz változtatás			Távolság			Max. víz
	Napos	Felhős	Esős	Napos	Felhős	esős	
Homokjáró	-1	0	3	3	1	0	8
Szivacs	-4	-1	6	0	1	3	20
Lépegető	-2	-1	3	1	2	1	12

Az egyes lények a vízkészlet megváltoztatása során nem léphetik túl a fajtára jellemző maximális értéket, legfeljebb azt érhetik el.

A program egy szövegfájlból olvassa be a verseny adatait! Az első sorban az induló lények száma szerepel. A következő sorok tartalmazzák a lények adatait szóközzel elválasztva: a lény nevét, a fajtáját és a kezdetben rendelkezésére álló víz mennyiségét. A fajtát egy karakter azonosítja: h – homokjáró, s – szivacs, l – lépegető.

A lényeket leíró részt követő sorban a verseny napjai szerepelnek egy karaktersorozatban. Az egyes jelek értelmezése: n – napos, f – felhős, e – esős.

A program kérje be a fájl nevét, majd jelenítse meg a nyertes nevét! (Feltehetjük, hogy a fájl formátuma helyes.) Egy lehetséges bemenet:

4

Vandor h 4

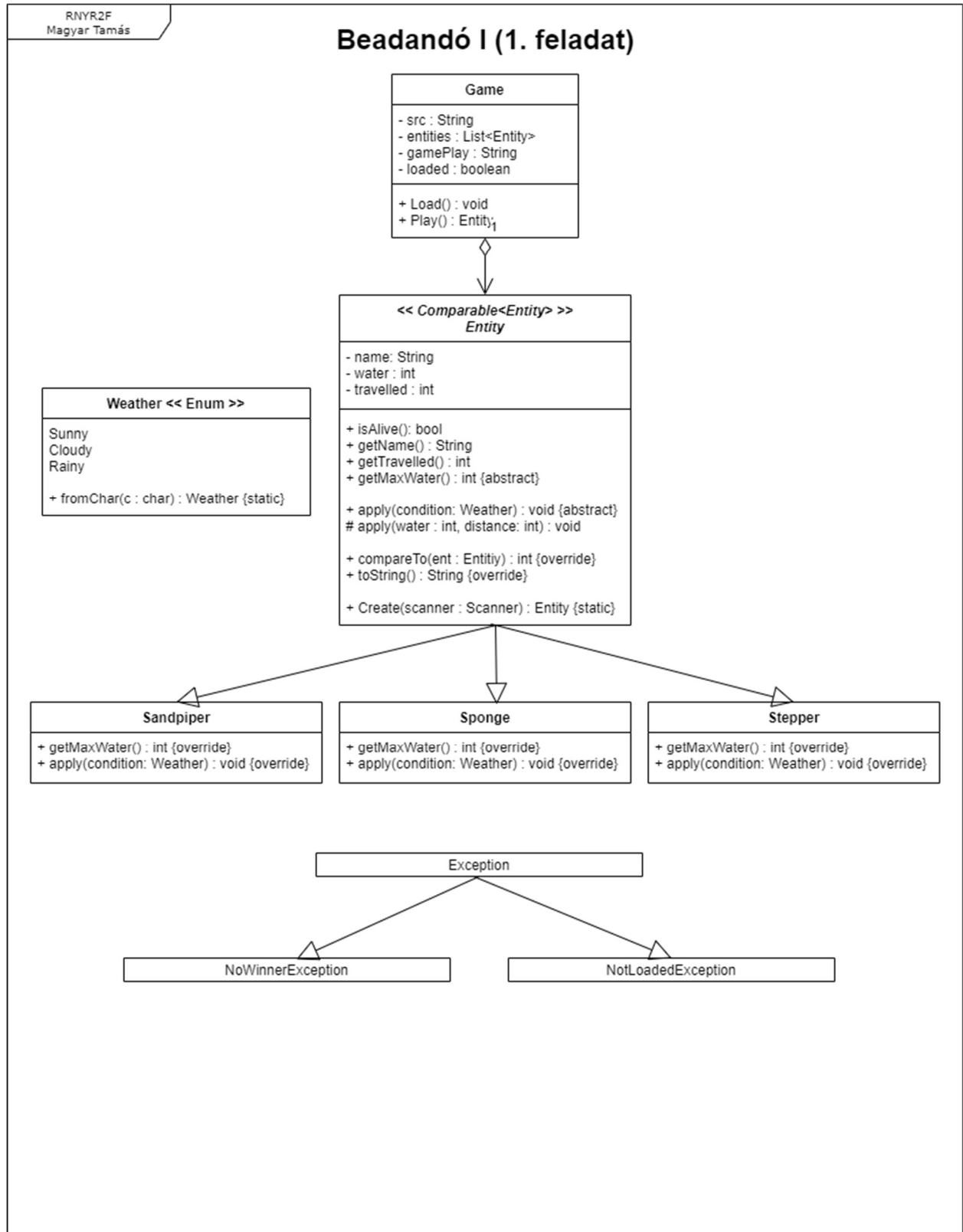
Seta l 7

Csuszo s 12

Siklo s 10

nffeeennf

UML Diagram



Metódusok rövid leírása

`Game::Load() : void`

A konstruktorban megadott fájlnev alapján megpróbálja betölteni a verseny alapállapotát, `IllegalArgumentException` ad ha nem megfelelő a előlány típusa.

`Game::Play() : Entity`

Lejátsza a verényt, majd leszűkíti a entities listát az túlélőkre és azt rendezni a megtett távolság alapján, ennek a listának az első elemét visszaadja ha van. Ha nincs `NoWinnerException` dob. Ha a `gamePlay` (azaz a napok időjártása) nem megfelelő karaktert tartalmaz akkor `IllegalArgumentException` dob.

`Weather::fromChar(c : char) : Weather`

A megadott karakter alapján visszaad egy időjárást. `IllegalArgumentException` dob ha rossz karaktert kap.

`Entity::isAlive() : bool`

`water > 0`

`Entity::getMaxWater() : int {abstract}`

A megvalósító osztályok itt adják vissza a maximális vízt amit tárolni tudnak.

`Entity::apply(condition : Weather) {abstract}` és `(water : int, distance : int): void`

Ez függvénnnyel lehet egy lénynek az állapotát megváltoztatni az időjárás alapján. A metódus túltehret az egyparaméteres változata publikus, a kétparaméteres védett így a leszármaztatottban a metódus a típus alapján beállított értékekkel meghívja és ezt változtatja ténylegesen az objektum állapotát.

`Entity::Create(scanner : Scanner) : Entity {static}`

Ez a metódus gyártja le a sorban következő lényt.

A többi geter funkciót tölt be, illetve a Comparable interface-t megalósító compateTo a travelelld alapján összehasonlít. (és a toString() szebb kiírás végett)