Halmai Tímea 3. beadandó/7.feladat 2021. 05. 12.  
BBNLTP  
[bbnltp@inf.elte.hu](mailto:bbnltp@inf.elte.hu)  
13.csoport

Feladat

Egy bolygón különböző fajtájú növények élnek, minden növény tápanyagot használ. Ha egy növény tápanyaga elfogy (a mennyisége 0 lesz), a növény elpusztul. A bolygón háromféle sugárzást különböztetünk meg: alfa sugárzás, delta sugárzás, nincs sugárzás. A sugárzásra a különböző fajtájú élő növények eltérő módon reagálnak. A reakció tartalmazza a tápanyag változását, illetve a következő napi sugárzás befolyásolását. A másnapi sugárzás alakulása: ha az alfa sugárzásra beérkezett igények összege legalább hárommal meghaladja a delta sugárzás igényeinek összegét, akkor alfa sugárzás lesz; ha a delta sugárzásra igaz ugyanez, akkor delta sugárzás lesz; ha a két igény közti eltérés háromnál kisebb, akkor nincs sugárzás. Az első nap sugárzás nélküli. Minden növény jellemzői: az egyedi neve (sztring), a rendelkezésre álló tápanyag mennyisége (egész), hogy él-e (logikai). A szimulációban részt vevő növények fajtái a következők: puffancs, deltafa, parabokor. A következőkben megadjuk, hogy az egyes fajták miként reagálnak a különböző sugárzásokra. Először a tápanyag változik, és ha a növény ezután él, akkor befolyásolhatja a sugárzást.   
Puffancs: Alfa sugárzás hatására a tápanyag mennyisége kettővel nő, sugárzás mentes napon a tápanyag eggyel csökken, delta sugárzás esetén a tápanyag kettővel csökken. Minden esetben úgy befolyásolja a másnapi sugárzást, hogy az 10 egységgel növeli az alfa sugárzás igényét. Ez a fajta akkor is elpusztul, ha a tápanyag mennyisége 10 fölé emelkedik.   
Deltafa: Alfa sugárzás hatására a tápanyag mennyisége hárommal csökken, sugárzás nélküli napon a tápanyag eggyel csökken, delta sugárzás hatására a tápanyag néggyel nő. Ha a tápanyag mennyisége 5-nél kisebb, akkor 4 egységgel növeli a delta sugárzás igényét, ha 5 és 10 közé esik, akkor 1 értékben növeli a delta sugárzás igényét, ha 10-nél több, akkor nem befolyásolja a másnapi sugárzást. Parabokor: Akár alfa, akár delta sugárzás hatására a tápanyag mennyisége eggyel nő. Sugárzás nélküli napon a tápanyag eggyel csökken. A másnapi sugárzást nem befolyásolja. Szimuláljuk a növények viselkedését és adjuk meg, hogy x nap után melyik életben maradt egyed a legerősebb! Minden lépésben írjuk ki az összes növényt a rájuk jellemző tulajdonságokkal, valamint az aktuális sugárzást! A program egy szövegfájlból olvassa be a szimuláció adatait! Az első sorban a növények száma szerepel. A következő sorok tartalmazzák a növények adatait szóközökkel elválasztva: a növény nevét, a fajtáját és a kezdetben rendelkezésére álló tápanyag mennyiségét. A fajtát egy karakter azonosít: p - puffancs, d - deltafa, b - parabokor. A növényeket leíró részt követő sorban a szimuláció napjainak száma adott egész számként. A program kérje be a fájl nevét, majd jelenítse is meg a tartalmát.

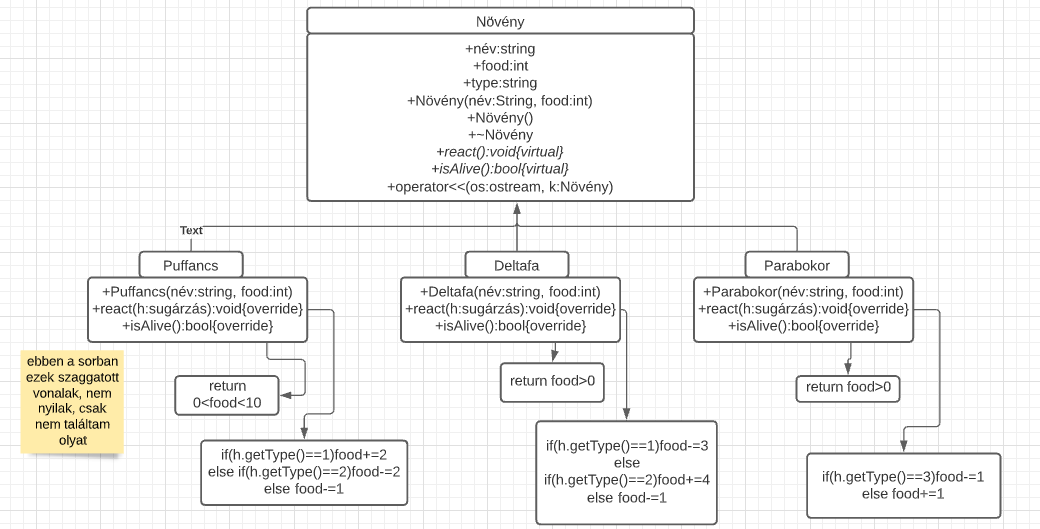
Terv:

A növények leírásához bevezetünk négy osztályt: a növények általános tulajdonságait megadó Növények ősosztályt (Plants), és ebből származtatjuk a konkrét típusú növények, a Puffancs, Deltafa és a Parabokor osztályait. Attól függetlenül, hogy a növények milyen fajtájúak, számos közös tulajdonsággal rendelkeznek. Mindegyiknek van neve (name), kajája (food), típusa (type), meg lehet róla kérdezni, hogy él-e (isAlive()), emellett hogy hogy reagál bizonyos sugárzásokra (react()), mely művelet módosítja a tápanyaga mennyiségét. A két metódus csak a konkrét osztályok szintjén implementálható, hiszen hatásük függ attól, hogy egy növény milyen fajtájú. Ennél fogva az általános kalandozó típusát leíró osztály absztrakt lesz, hiszen a react() és isAlive() metódus absztrakt, másrészt úgyse akarunk ilyen objektumot létrehozni. A különböző növény osztályok konstruktorai az ősosztály szintjén inicializálják a nevet és a típust, és egyedi módom felüldefiniálják az isAlive(), react() metódusokat. A react() metódusban elágazásokat kell elhelyezni, melynek feltételei a paraméterként megadott sugárzás típusát vizsgálják. Az elágazások létrehozása azonban nem szerencsés, ha fel kell készítenünk a programunkat arra, hogy később új sugárzás fajtákkal is bővülhet a feladat. Ekkor ugyanis az összes konkrét növény react() metódusát módosítani kell. Ezt elkerülentő a „látogató” tervmintát alkalmazzuk úgy, hogy a sugárzások leírására szolgáló osztályok töltik majd be a látogató szerepét. A sugárzások általános tulajdonságait a Sugarzas ősosztály írja le, és ebből származtatjuk a konkrét sugárzásokat, az Alfa, Delta és semmilyen (None) sugrárzások osztályait. Minden sugárzás rendelkezik típussal (type), egy értékkel, mely befolyásolja a másnapi sugárzást (need), emellett meg lehet róla kérdezni, hogy milyen a típusa (getType), melyet az ősosztály szintjén lehet implementálni.   
A különböző növény osztályok konstruktorai az ősosztály konstruktorával inicializálják a nevet és a típus, egyedi módon felüldefiniálják a konkrét isAlive() és react() metódusokat. Az inicializáláshoz és a felüldefiniáláshoz a feladat szövegéből kinyert táblázatok adnak útmutatást:

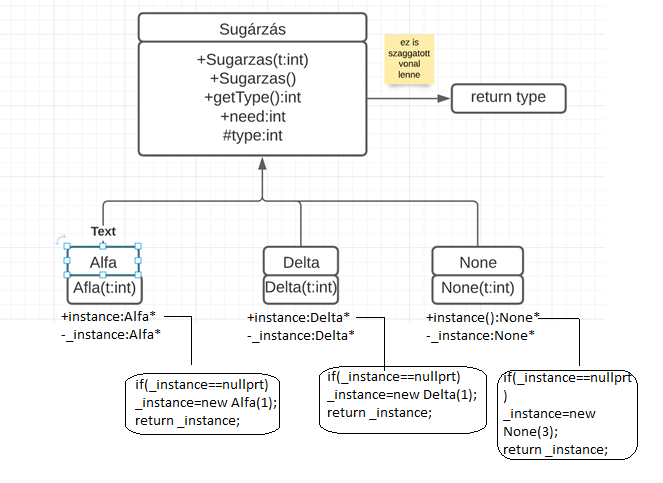
|  |  |  |  |
| --- | --- | --- | --- |
| Sugárzás hatása az ételnél | Puffancs | Deltafa | Parabokor |
| Alfa | +4/növény | -3/növény | +1/növény |
| Delta | -2/növény | +4/növény | +1/növény |
| Semmilyen | -1/növény | -1/növény | -1/növény |

A táblázat szerint a react() metódusban elágazásokat kell elhelyezni, amelynek feltételei a paraméterként megadott sugárzás típusát vizsgálják.

|  |  |
| --- | --- |
| Másnapi sugárzás befolyásolása |  |
| Puffancs | +10 alfa |
| Deltafa | ha food<5: +4 delta ha food<10:  +1 delta |



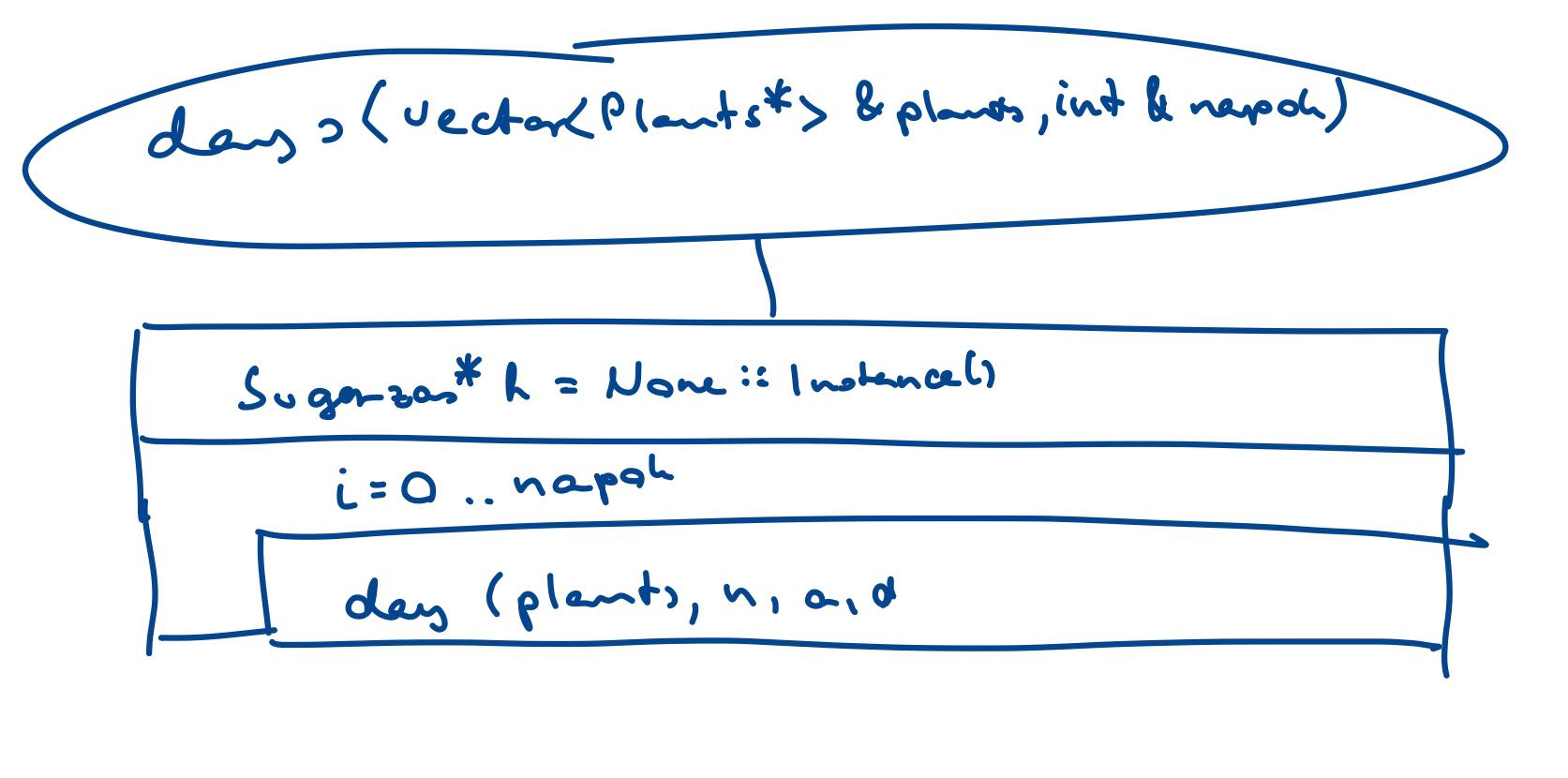
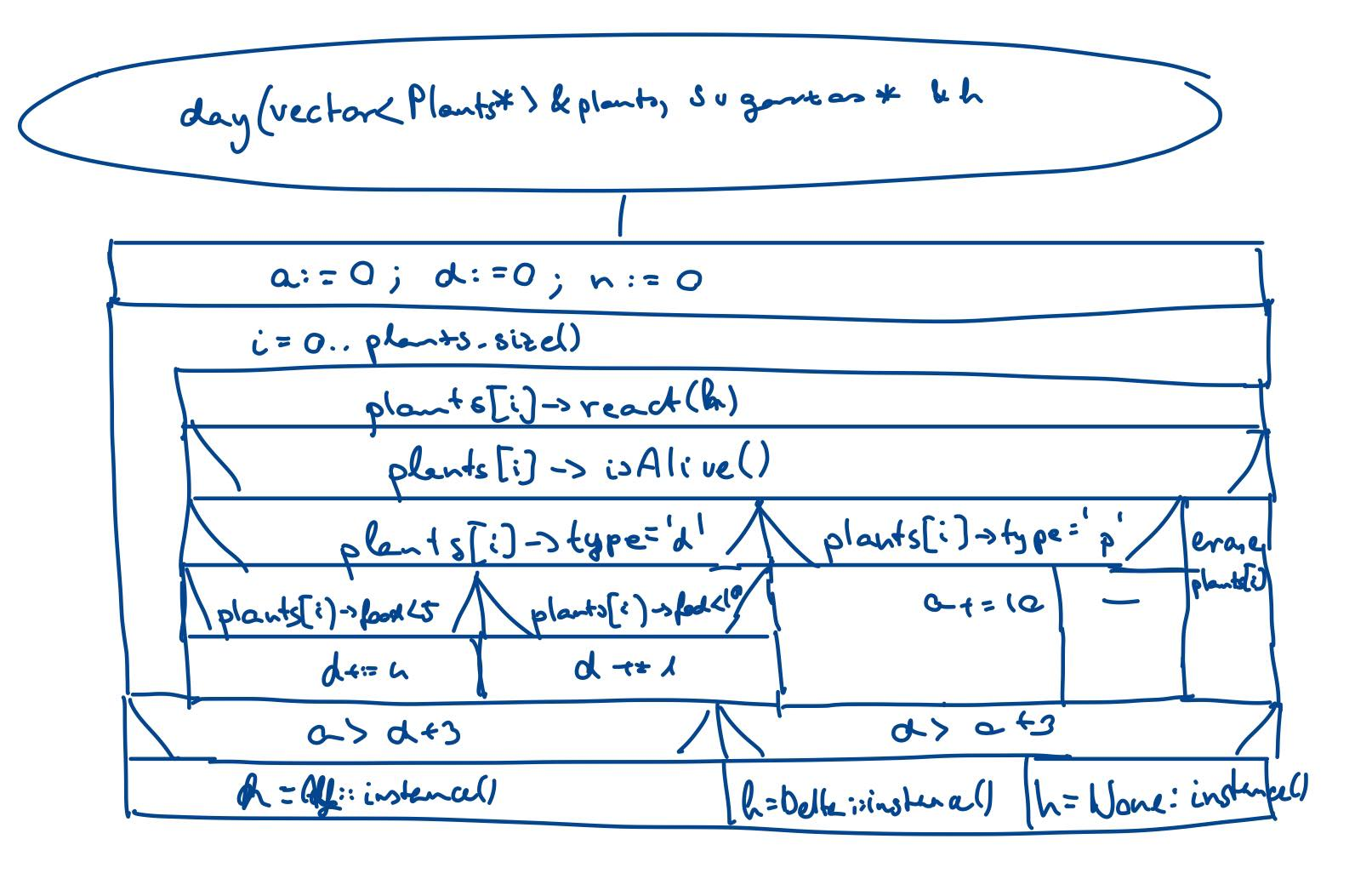
A sugárzások összes osztályát az „egyke” tervmintának megfelelően valósítjuk meg, hiszen elég mindegyik sugárzásfajtához egy-egy objektumot létrehozni.

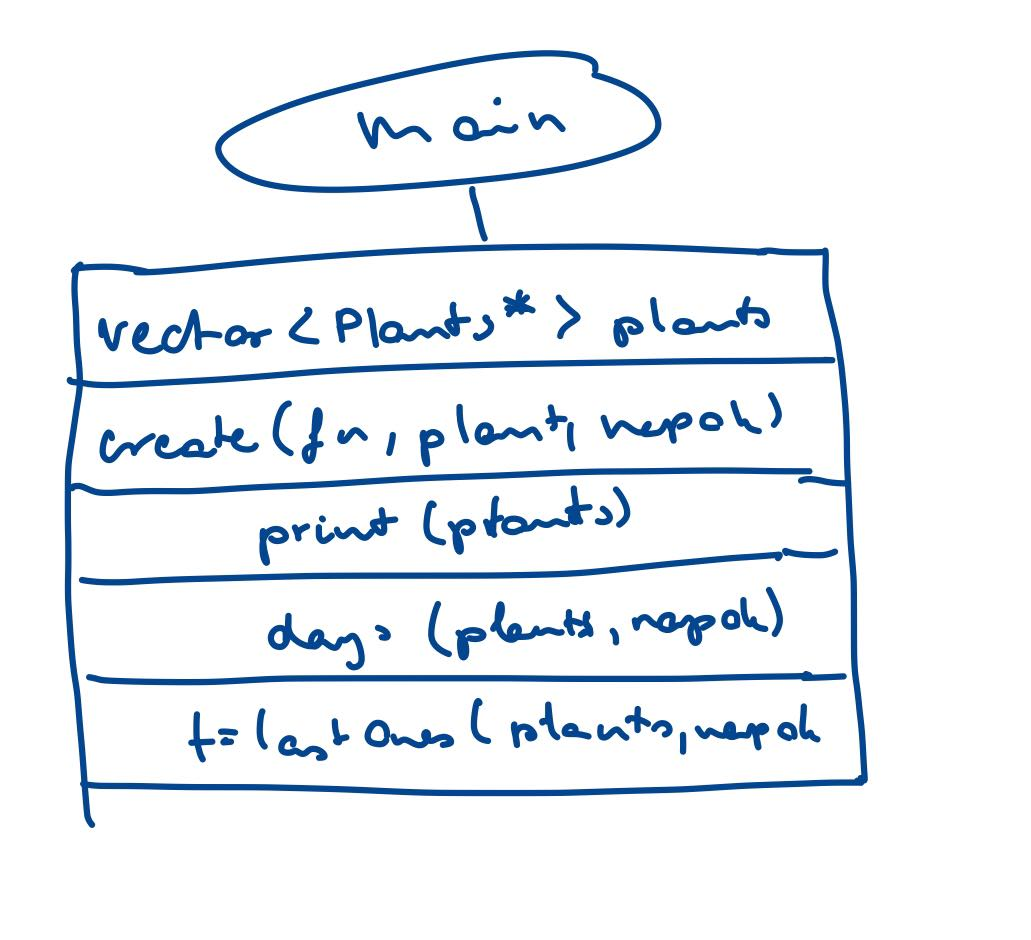


A sugárzás növényekre való hatását a days függvénnyel jelöljük, amely megadja az életben maradt növényeket.

A=(plants:Növények\*n, napok:Int, t:Int)

Előfeltétel=(plants.size()<=plants0.size(), napok=napok0)





Tesztelési terv:

1. 2 növény, 2 nap
2. nincs megadva növény
3. több növény, több nap