

Feladatleírás

- Adott az állatvilág fajainak rendszertani besorolásának hierarchiája egy fájlban
 - A fájl egy-egy sora ír le vagy egy rendszertani kategóriát (taxont)
 - A rendszertani kategóriák az alábbiak (az általánostól a konkrétabbig haladva):
 - Phylum (törzs)
 - Classis (osztály)
 - Ordo (rend)
 - Familia (család)
 - Genus (nem)
 - Species (faj)
 - Minden sor egy betűvel kezdődik, amit egy szóköz követ
 - A betű a fenti taxonok egyikét jelenti (P - törzs, C - osztály, O - rend, stb)
 - Utána következik az adott taxon latin neve, ami biztosan egy szó
 - A fajok esetében ezek után következik a faj teljes magyar neve, ami legalább egy szó, de lehet több is
 - Utána vagy következik egy függőleges vonal vagy nem
 - A függőleges vonal után vesszővel elválasztva soroljuk fel azon állatokat, amelyekre az adott faj vadászik (ha nincs vonal, itt persze nincs semmi)
 - Mint a lenti példából is láthatjuk, itt szerepelhet konkrét faj neve (erről a szabályt lásd később), vagy egy magasabb szintű taxon a kategória-azonosítójával együtt
 - Utóbbi úgy értendő hogy az adott taxon minden faja az épp feldolgozás alatt álló ragadozó zsákmányállata lehet
 - Azt feltehetjük, hogy a fájlban csak olyan taxonok vagy konkrét fajok jelennek meg ebben a részben, amelyeknek minden faja már korábbi sorokban felsorolásra került
 - Végül két szám következik
 - Az első a faj egyedszáma a vizsgált időszak kezdetén (természetes szám)
 - A második pedig a faj szaporodási potenciálja, ami itt most jelentse azt, hogy ennyi naponként születik egy-egy újabb példány minden meglevő párra (azaz, ha a legutóbbi nyusziáldás óta eltelt 3 nap, és most 10 nyúlunk van, akkor ma 5 fog születni, majd ha senki nem bántja őket, újabb 3 nap múlva még további 7)
 - Az egymást követő sorok közötti viszony attól függ, mi az adott két sor kezdőbetűje:
 - Ha az i. sor kezdőbetűje a taxonok sorrendjében éppen eggyel megelőzi az i+1. sorét, akkor az i+1. taxon a felette levő sor kategóriája alá tartozik. A mellékelt bemenetben például láthatjuk,

hogy a kutyafélék (Canidae) családja a ragadozók (Carnivora) rendjébe tartozik

- Ha a két kezdőbetű azonos, akkor ők a faszerkezetben "testvérek". Pl. ha a G Vulpes alatt többféle S kategóriájú rókát is felsoroltam volna
- Ha a későbbi sor kezdőbetűje a hierarchiában feljebb van, az azt jelenti, hogy annak a szülője valahol a korábbi sor előtt van. Például az "S Europaeus" sor után a Rodentia (rágcsálók) rendje jön, ami azt jelenti, hogy annak a sornak az őse nem a mezei nyúl lesz, hanem a mezei nyúl azon őse, ami a rend felett van a sorrendben, azaz az emlősök (Mammalia) osztálya
- Más esetre nem kell felkészülni - azaz bár előfordulhat, hogy nem megy egy leszármazási lánc végig, de olyan nincs, hogy egy szint kimarad
- Egy faj teljes tudományos nevét úgy adhatjuk meg, hogy a nem és a faj nevét írjuk le egymás mellé - utóbbit kis betűvel. Tehát a mezei nyúl neve "Lepus europaeus", a házi egéré "Mus musculus", míg a vörös rókéé "Vulpes vulpes"
- Az egy szinten levő taxonok között nem kell névütközésre számítanunk, kivéve a faj esetében, viszont természetesen egy nem alatt csak különböző nevű fajok szerepelnek, így végül is a faj teljes tudományos neve (azaz a nem és faj) már egyedi

Példabemenet

```
P Chordata
C Mammalia
O Lagomorpha
F Leporidae
G Lepus
S Europaeus mezei nyul 5 3
O Rodentia
F Muridae
G Mus
S Musculus hazi eger 6 4
O Carnivora
F Canidae
G Vulpes
S Vulpes voros roka | Lepus europaeus, F Muridae 2 6
```

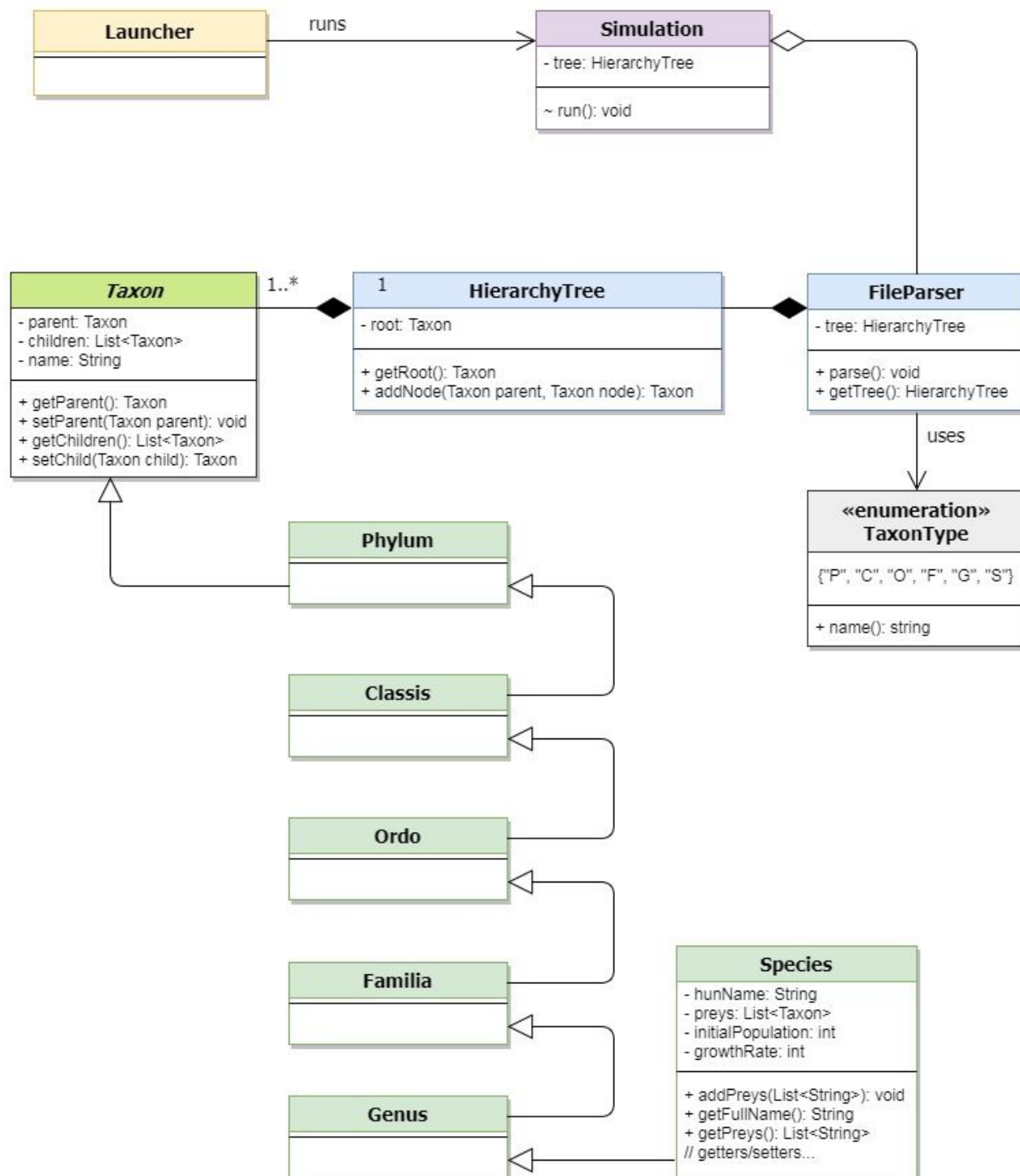
Konkrét feladat

16. Adjunk meg konzolról egy természetes számot. Állapítsuk meg, melyik fajból mennyi fog élni a megadott szám szerinti nap múlva. A kezdeti populációt és a szaporodási rátát ismerjük, a ragadozók táplálkozási szokásait még nem: amelyik fajnál van zsákmányállat, annak a fajnak minden példánya minden nap elfogyaszt egyet az adott zsákmányból, különben elpusztul. A precedenciája a megadott sorrendet követi, tehát amíg vannak nyulak, a rókák azokat fogják enni.

Osztályszerkezet

2017/18 - űsz

Programozási Technológia I. - 1. Beadandó

Fejér Tamás
EAGFMQ

Megvalósítás

- A feladatot egy Java 8 nyelven írt, objektumorientált szemléletet követő programmal oldottam meg.
- A program belépési pontja a *Launcher* osztályban található. Ez az osztály létrehoz egy szimulációt (*Simulation* osztály), majd meghívja a szimuláció futtatást végző műveletét - *run()*.
- A szimuláció végzi a feladat megoldását, azaz a *FileParser* osztály segítségével egy fa adatszerkezetet (*HierarchyTree* osztály) hoz létre a szöveges fájl tartalma alapján, ezek után pedig kiírja a képernyőre a fájlban szereplő állatfajok népességét a szimulációs adatoknak megfelelően (azaz a billentyűzetről megadott napok számának letelte utáni népességeket).
- A *FileParser* osztály nem létező fájl esetén *FileNotFoundException* kivételt dob.
- A fa adatszerkezet létrehozásához a *FileParser* osztály a *TaxonType* osztályt hívja segítségül, amely a kategória típusok sorrendjét tartalmazza, ezzel lehetséges az egyes kategóriák szülő-gyermek kapcsolatának megalkotása a fában.
- A fájlban megadott kategóriákat a *Taxon* osztály, mint absztrakt osztály reprezentálja, ebből származnak a konkrét kategória típusokat megvalósító osztályok (*Phylum*, *Classis*, *Ordo*, *Familia*, *Genus*, *Species*). A *Taxon* osztály tartalmazza a kategóriák közös tulajdonságait, azaz a szülőt, gyerekeket és nevet.
- Ezen felül egyedül a *Species* osztály rendelkezik egyedi tulajdonságokkal, melyek a faj magyar neve, prédáinak listája, kezdeti népessége és szaporodási rátája.

Tesztelés

A programhoz egyetlen, a nem létező fájl ellenőrző érvénytelen tesztet készítettem:

- [nosuchfile.txt] - *kivételt dob*

A programot tesztelő érvényes tesztesetek:

- Üres fájl: [empty.txt] - *a fa létrejön egyetlen gyökér csúccsal*
- Egyetlen sorból álló fájl [SingleLine.txt] - *a fa létrejön, a gyökér egyetlen gyerekével*
- Egyetlen, préda nélküli fajt tartalmazó fájl [SingleSpeciesNoPrey.txt] - *a fajt reprezentáló csúcs létrejön, minden adata helyes*
- Egyetlen, prédával is rendelkező fajt tartalmazó fájl [SingleSpeciesWithPrey.txt] - *a fajt, és a prédáit reprezentáló csúcsok megfelelően létrejönnek, a faj prédáinak listájában helyesen szerepel a préda állat*
- Egy teljes láncot tartalmazó fájl [SingleChain.txt] - *a fa létrejön, a kategóriák helyes sorrendben kerülnek a fába*
- Két, közös őstől származó, prédával nem rendelkező fajt tartalmazó fájl [TwoSpeciesNoPrey.txt] - *a fa létrejön, a faj csúcsok megfelelő helyre kerülnek be a fába*
- Három, közös őstől származó faj, ebből kettő a harmadik faj prédája (teljes névvel, és kategória azonosítóval megadott préda is szerepel a fájlban) [ThreeSpeciesWithPrey.txt] - *a fajok csúcsai megfelelő helyre kerülnek a fában, a prédák helyesen szerepelnek a ragadozó préda listájában*

- Közös szülőve rendelkező fajokat (testvéreket) tartalmazó fájl [Siblings.txt] - *a fajok helyesen ugyanahhoz a szülő csúcshoz kerülnek gyerekként*
- A szaporodást ellenőrző tesztfájl [Reproduction.txt] - *a megadott napok elteltével a népesség az elvártnak megfelelő*
- A prédák sorrendjét ellenőrző tesztfájl [PreyPrecedence.txt] - *a ragadozó a prédáira a megadott sorrendben vadászik, csak amikor már az első fajból nem maradt példány, akkor kezd el a következő fajra vadászni*