

Feladat leírása

Készíts egy programot, amely egy kétdimenziós mezőn nyulak életét szimulálja. A mezőben véletlenszerűen lehetnek üres helyek és nyulak. A szimuláció során figyelni kell a **születéseket** és a **halálozásokat**, így egy demográfiai vizsgálatot is tudunk végezni.

Követelmények

1. Objektumorientált felépítés

- Hozz létre egy **IÉlőlény** nevű **interfészt**, amely tartalmazza az élőlények alapvető viselkedéseit (pl. él-e még, szaporodik-e). Figyelj rá, hogy az adott folyamat állapotot vagy esetleg viselkedést szemléltet!
- Hozz létre egy **Állat** nevű **absztrakt osztályt**, amely az élőlények közös tulajdonságait és metódusait tartalmazza (pl. életkor, nem, lépés a szimulációban) továbbá megvalósítja az **IÉlőlény** interfészt
- Készíts egy **Nyúl** nevű **konkrét osztályt**, amely az **Állat** osztályból származik. A nyúlnak legyenek például a következő tulajdonságai:
 - maximális élettartam
 - szaporodási esély

2. Kapcsolatok és működés

- A mezőt reprezentálja egy külön **Mező** osztály, amely egy 2D adatszerkezetet tartalmaz. Ez kezeli a nyulak elhelyezését, mozgását, szaporodását és elhalálozását.
- Legyen egy **Szimuláció** osztály, amely lépésenként futtatja a szimulációt, és nyilvántartja a statisztikát (pl. hány nyúl született, hány pusztult el).

3. Futtatás

Készíts egy **Main** (vagy Program) osztályt, amelyben:

- létrehozol egy mezőt adott mérettel,
- véletlenszerűen elhelyezel benne nyulakat és üres helyeket,
- elindítod a szimulációt több lépésen keresztül,
- a végén kiírod a demográfiai statisztikát (születések, halálozások, jelenlegi nyúlpopuláció).

4. Adatszerkezet

A mezőt egy 2D adatszerkezetben (pl. mátrix) modellezzük. Minden mezőben vagy egy nyúl van, vagy üres.

Bővített feladat: Nyúlszimuláció 2.0

Új cél

Egy **ökoszisztéma-szimuláció** létrehozása, ahol nemcsak a nyulak életét és szaporodását vizsgáljuk, hanem megjelennek **ragadozók (pl. rókák)**, valamint a környezeti tényezők (pl. évszakok, élelem mennyisége).

Követelmények

1. Interfész

- **IÉlőlény** (változatlan)

- `el() : bool` – visszaadja, hogy az élőlény életben van-e.
- `szaporodik() : bool` – visszaadja, hogy képes-e szaporodni.

2. Absztrakt osztály

- **Állat** (bővített)
 - Tulajdonságok:
 - `életkor : int`
 - `maxÉlettartam : int`
 - `energia : int` (pl. ha nincs elég élelem, csökken, ha nullára megy → halál)
 - Metódusok:
 - `növelÉletkor()`
 - `változtatEnergia(mennyiség : int)`
 - `lepés()` – alapvető életciklus (öregszik, energiát veszít).

3. Konkrét osztályok

- **Nyúl** (Állatból származik)
 - Saját tulajdonság: `szaporodásiEsély : double`
 - Saját metódus:
 - `táplálkozik(Mező m)` – ha talál fűvet, energiát nyer.
 - `szaporodik()` – bizonyos eséllyel új nyúl születik a közelben.
 - `szimulaciosLepes()` – végrehajtja a szimulációs lépést
- **Róka** (Állatból származik)
 - Saját tulajdonság: `vadászásiEsély : double`
 - Saját metódus:
 - `táplálkozik(Mező m)` – ha talál nyulat, elfogyasztja, energiát nyer.
 - `szaporodik()` – bizonyos feltételek mellett új róka születhet.
 - `szimulaciosLepes()` – végrehajtja a szimulációs lépést

4. Környezet

- **Mező**
 - Tulajdonság: `rács : IÉlőlény[][]`
 - Metódusok:
 - `elhelyez(élőlény : IÉlőlény, x:int, y:int)`
 - `lepés()` – minden élőlény lépése végrehajtódik.
 - `getNépeség(osztály : Class)` – visszaadja az adott típusú állatok számát.
 - `keresSzomszédos(x, y, típus)` – keres szomszédos mezőn adott típust (pl. nyulat a róka számára).
- **Évszak** (új osztály vagy enum)
 - Értékek: `TAVASZ, NYÁR, ŐSZ, TÉL`
 - Hatás:
 - `TÉLEN` kevesebb az élelem (nyulak nehezebben táplálkoznak).
 - `TAVASSZAL` nagyobb a szaporodási esély.

5. Vezérlés

- **Szimuláció**
 - Tulajdonságok:
 - `mező`: `Mező`
 - `születések`: `int`
 - `halálozások`: `int`
 - `aktuálisÉvszak`: `Évszak`
 - Metódusok:
 - `futtat(lépések: int)` – minden lépésben az évszak változhat, a `mező` állapota frissül.
 - `statisztika()` – kiírja a népesség alakulását fajonként.

6. Futtatható osztály

- **Main**
 - Létrehoz egy `Mező`-t, véletlenszerűen elhelyez nyulakat és rókákat.
 - Elindítja a `Szimuláció`-t, pl. 50 lépésig.
 - A végén kiírja a statisztikákat:
 - hány nyúl született,
 - hány nyúl halt meg,
 - hány róka született,
 - hány róka halt meg,
 - aktuális populációk.

Plusz ötletek a kihívás növelésére

- A `Mező` osztályba bevezethető **Fű** mint külön "erőforrás", ami minden körben nőhet. A nyulak csak akkor tudnak enni, ha van fű a cellában.
- A `Róka` energiája gyorsabban fogy, így ha nincs elég nyúl, éhen hal.
- Lehetőség van többféle `Állat`-ra (pl. Sas, ami a rókákat is vadászhatja).
- A szimuláció végeredményét lehet grafikonon is ábrázolni (nyulak és rókák populációja az időben).