

Órai feladat – Öröklés

A projektben már meglévő osztályok leírása

Terkep osztály:

Eltárolja egy **meretX**×**meretY** méretű térkép magassági adatait a **magassag** nevű tömbben (a magassági adatok egynél kisebbek lehetnek csak, 0 vagy az alatti értékeket víznek tekintjük). Emellett rendelkezik az alábbi publikus metódusokkal:

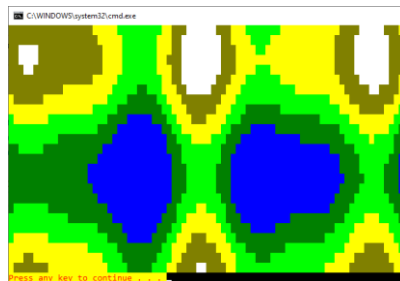
- **TerkepenBeluliPozicio(x,y)** – megadja, hogy az átadott x,y koordináták a térképen belül vannak-e
- **Magassag(x,y)** – megadja egy lebegőpontos számokkal megadott pont magasságát
- **VeletlenFeltoltes()** – feltölti a magasság adatokat „véletlenszerűen”

TerkepRajzolo osztály:

A konstruktorban paraméterként átadott térképet kirajzolja a konzolra a megadott magassági színek segítségével. Az alábbi metódusokat tartalmazza:

- **MiVanItt(x,y)** – visszaad egy karaktert, hogy megadott helyre mit kell rajzolni. Ez egy üres térkép esetén mindig szóköz, de később felül fogjuk írni
- **Kirajzol()** – törli a konzolt és kirajzolja a térképet a magassági adatok és a **MiVanItt** által visszaadott értékek alapján

Teszt: Egy új térképre használva a kirajzolást, hasonlót kapunk (kék: víz, sötét/világos zöld – sárga – barna – fehér: fokozatosan emelkedő terület)



Feladatok

Alapvető járművek elkészítése

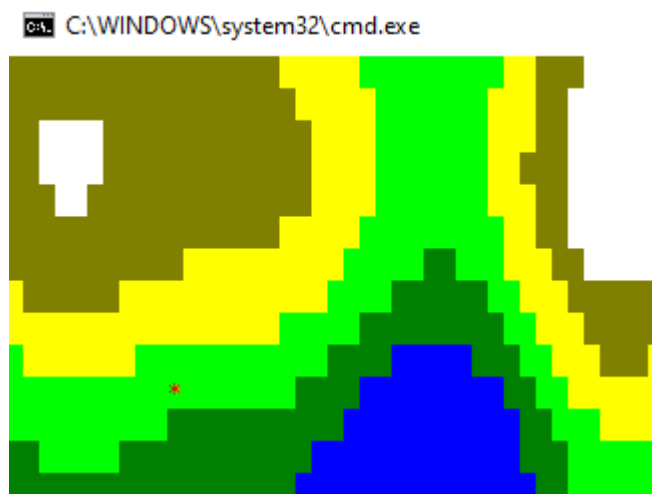
Készíts egy **Jarmu** osztályt, ami az alábbiakkal rendelkezik:

- **azonosito** – karakter, ez fog majd megjelenni a képernyőn
- **x, y** – két float koordináta, hogy hol van a jármű
- **terkep** – egy referencia egy térkép objektumra, ahol a jármű mozogni fog
- egy konstruktor, ami beállítja a fentieket
- csak olvasható tulajdonságok az **azonosító**, **x**, **y** mezőkhöz
- **IdeLephet(x,y)** – virtuális metódus, amely csak akkor adon igaz értéket vissza, ha a paraméterként átadott x,y koordináta a térképen belül van

Készíts egy **TerkepEsJarmuRajzolo** osztályt, ami a **TerkepRajzolo** leszármazottja és azt kibővíti az alábbiakkal:

- **jarmuvek** – tömb, amiben **Jarmu** objektumokat lehet tárolni. Ennek méretét a konstruktorban lehessen megadni
- **jarmuvekN** – azt számolja, hogy hány tényleges jármű van a tömbben
- **JarmuFelvetel(jarmu)** – felvesz egy új járművet a tömbbe
- a konstruktor paraméterként egy térképet várjon (ezt továbbítsa az ős konstruktornak) és egy maximális jármű számot, ami alapján létrehozza a tömböt
- **MiVanItt(x,y)** – az örökölt **MiVanItt** metódust írja felül úgy, hogy ha a paraméterként megadott x,y koordinátákon (egész számra kerekítve) található egy jármű a tömbben, akkor a függvény visszatérési értéke ennek az azonosítója legyen (ha több is van, akkor bármelyiké). Ellenkező esetben adja vissza az ősosztály metódusának visszatérési értékét.

Teszt: A meglévő térkép mellé készíts egy járművet és ezeket add át egy **TerkepEsJarmuRajzolonak**. Látnod kell, hogy megjelent a jármű azonosítója a megfelelő helyen (a képen '*' a 10,10 koordinátákon)



Szimuláció elkészítése

Készíts egy **MozgoJarmu** absztrakt osztályt, ami a **Jarmu** leszármazotta, és azt az alábbiak szerint egészíti ki:

- **iranyX, irányY** – két float érték, az irányvektort tartalmazzák
- **UjIrányVektor(iranyX, irányY)** – ezen keresztül lehet értéket adni az előzőknek
- konstruktorainak paraméterei ugyanazok mint az ősnek, azokat csak továbbítja
- **Mozog()** – absztrakt, visszatérési érték nélküli metódus. Ezt fogják majd a leszármazottak megvalósítani a mozgási stratégiájuknak megfelelően.

Készíts egy **Helikopter** nevű osztályt, ami a **MozgoJarmu** leszármazottja, és

- **sebeség** – float mező, kezdőértéke legyen 1
- **Gyorsit()** – növeli a sebességet 0.1-el
- **Lassit()** – csökkenti a sebességet 0.1-el (negatív nem lehet)
- **Mozog()** – megvalósítja úgy a mozgás metódust, hogy az **x** és **y** koordináták irányába is ellép az adott irányvektorkomponens * sebesség értékkel. De csak akkor, ha az **IdleLephet** metódus visszatérési értéke igaz erre a helyre, különben nem csinál semmit
- konstruktora paraméterként csak egy térképet és a koordinátákat várja, az ős konstruktorának mindig 'H'-t ad át azonosítóként

Készíts egy **Szimulacio** osztályt, ami a **TerkepEsJarmuRajzolo** leszármazottja, és azt kiegészíti:

- **EgyIdoEgysegEltelt()** – visszatérési érték nélküli metódus, ami végignézi a járműveket tartalmazó tömböt, és ha valamelyik elem egyben **MozgoJarmu** is, akkor annak meghívja a **Mozog** metódusát
- **Fut()** – egy végtelen ciklusban futtatja a szimulációt: meghívja az **EgyIdoEgysegEltelt** metódust, ezt követően a kirajzolást, majd pedig vár fél másodpercet (*System.Threading.Thread.Sleep(500)*)

Teszt: Létrehozva egy vagy több **Helikopter** objektumot és ezeket elhelyezve egy **Szimulacio** objektumban, a **Fut** metódus meghívását követően azt kell látnunk, hogy azok mozognak a térképen.

További leszármazottak készítése

Készíts egy **Auto** osztályt, ami a **MozgoJarmu** leszármazottja az alábbiak szerint:

- **IdeLephet(x,y)** – felülírja úgy az örökölt *idelephet* metódust, hogy az ő által visszaadott értéken felül még azt is megvizsgálja, hogy nincs-e víz a megadott x,y koordinátákban. Ha igen, akkor oda nem léphet
- **Mozog()** – felülírja úgy az örökölt *Mozog* metódust, hogy figyelembe vegye a magasságkülönbségeket. Minden továbblépés előtt nézze meg az aktuális x,y koordináta és az irányvektor által mutatott következő hely közötti magasságkülönbséget. Az elmozdulás során ezt vegye figyelembe, hogy ha növekszik (felfelé megy) akkor lassabban haladjon, ha csökken (lefelé halad) akkor pedig nagyobb lépjen

Készíts egy **Tank** osztályt, ami az **Auto** leszármazottja:

- **IdeLephet(x,y)** – a tank egy nagyszerű jármű, bárhova léphet, mindig adjon vissza igazat
- **uzemanyag** – cserébe viszont sokat fogyaszt, ezért egy külön mezőben tároljuk, hogy mennyi üzemanyag áll rendelkezésre. Ez a konstruktorban kapjon kezdőértéket
- **Mozog()** – módosítsuk úgy a mozgást, hogy csak akkor tudjon a tank mozogni, ha van még üzemanyag. Ilyenkor a mozgás után csökkentjük ennek értékét 10-el
- Legyen lezárta az osztály, tehát nem lehetnek leszármazottjai

Teszt: létrehozva több **Auto** és **Tank** objektumot, ezek is megfelelően működjenek. Az autón látni kell, hogy változik a sebessége (pl. ha elindítunk mellette egy Helikoptert, akkor azt néha megelőzi, néha lemarad). A Tanknak pedig át kell mennie a vízen, de egy idő után le kell állnia magától.