

# Objektumorientált programozás

## XI.

### OOP gyakorlás

# Hallgatói tájékoztató

**A jelen bemutatóban található adatok, tudnivalók és információk a számonkérendő anyag vázlatát képezik. Ismeretük szükséges, de nem elégséges feltétele a sikeres zárthelyinek, illetve vizsgának.**

**Sikeres zárthelyihez, illetve vizsgához a jelen bemutató tartalmán felül a kötelező irodalomként megjelölt anyag, a gyakorlatokon szóban, illetve a táblán átadott tudnivalók ismerete, valamint a gyakorlatokon megoldott példák és az otthoni feldolgozás céljából kiadott feladatok önálló megoldásának képessége is szükséges.**

# Gyakorló feladat

Egy számítógép-alkatrész webbolt adatbázisa 100 termékből áll. Minden Terméknek van ára, valamint típusa (pl. proceszor, memória, alaplap) és gyártója. A típust, a gyártót és az árat is véletlenszerűen sorsoljuk ki (az ár 20000-60000 Ft-ig legyen valamekkora).

A feladatunk az adatbázis egyes kereső funkcióinak megvalósítása. Az adatbázist reprezentáljuk osztállyal, és a feladatokat az Adatbázis osztály metódusaival valósítjuk meg. Minden kereső funkciónak egy tömbbel kell visszatérnie, amely a keresésnek megfelelő elemeket tartalmazza.

Lehessen keresni típus alapján.

Lehessen keresni gyártó alapján, mégpedig úgy, hogy több gyártót is lehessen megadni. Több gyártó megadása esetén az összes megadottnak a termékeit kapjuk vissza.

Lehessen keresni egy bizonyos típusból "olcsó", illetve "drága" termékeket. Olcsó az a termék, aminek az ára a típus átlagára alá esik vagy egyenlő azzal, drága, ami az átlagár felett van.

# Segítség a feladathoz

- A típus és akár a gyártó véletlenszerű kisorsolásához is pl. a felsorolás típust alkalmazhatjuk.

```
enum AlkatreszTípus
{
    Processzor,
    Memoria,
    Alaplap
}
```

- A felsorolás minden tagjának megfeleltethetünk egy egész (numerikus) értéket. Ha mást nem adunk meg, akkor alapértelmezés szerint a számozás nullától kezdődik és deklaráció szerinti sorrendben eggyel növekszik.
- Használata:

```
AlkatreszTípus tipus = AlkatreszTípus.Alaplap; //alaplap lesz a típus
if (tipus == AlkatreszTípus.Memoria)
{ ... }
```

- De a számokkal való megfeleltetés miatt akár véletlenszerűen is kisorsolhatjuk:

```
Random rand = new Random();
//véletlenszerű típus:
AlkatreszTípus tipus2 = (AlkatreszTípus)rand.Next(0, 3);
```

**GPS alapú túraútvonal-tervező programot készítünk. A program adatbázisában látványosságok vannak eltárolva. A látványosságok mindegyikének x és y koordinátája van, amelyek -10-10 közöttiek lehetnek (ez a jelenlegi pozíciókhoz viszonyított, km-ben mért helyzetüket jelzi), és van 0-1000 közötti tengerszint feletti magasságuk is.**

**Ha a programnak megadunk egy pontot, ahonnan indulni akarunk (-10,10 közötti koordinátákkal), akkor visszaad egy útvonaljavaslatot, amely az adott kiindulóponthoz 5 km-nél közelebb lévő látványosságokat tartalmazza. Egy útvonaljavaslatot reprezentáljunk Utvonal osztállyal.**

**Az összes, a program által ismert látványosság egy tömbben helyezkedik el, amelyet úgy hozunk létre, hogy 50-100 db véletlenszerű látványosságot tartalmazzon. Az Utvonalba ezek közül vesszük bele a közel lévőket.**



















**Az Utvonal osztály tartalmazza a következő funkciókat:**

- Tavolsag(), amely visszaadja a teljes út légvonalban vett hosszát (a magassággal nem kell számolni).**
- Szintemelkedes(), amely visszaadja a szintemelkedést (a legmagasabb és a legmélyebb pont közötti különbséget).**
- Szures(x), amely egy új Utvonalat készít az aktuális útvonal alapján, de úgy, hogy az új Utvonalban nem szerepelnek az aktuális útvonal x-nél magasabb pontjai.**
- Két útvonalat lehessen összekapcsolni is, azaz előállítani belőlük egy olyan új Utvonalat, amely mindkét útvonal pontjait tartalmazza, de úgy, hogy egy pontot sem tartalmazhat többször.**

# Programozási tételek összeépítése (1)

- **Feladat - mintaadatok**

Állatkertünkben háromféle állat tárolunk különböző méretű ketrecekben az alábbi mintaadatok szerint

1. ketrec	2. ketrec	3. ketrec	4. ketrec
<b>Kormos</b> Hím 20 kg 		<b>Morgó</b> Hím 310 kg 	<b>Nindzsa</b> Hím 2 kg 
<b>Mici</b> Nőstény 320 kg 		<b>Morcos</b> Hím 320 kg 	<b>Nyami</b> Nőstény 12 kg 
<b>Kajás</b> Hím 40 kg 	<b>Nóri</b> Hím 4 kg 	<b>Kolbász</b> Hím 40 kg 	<b>Kicsi</b> Nőstény 10 kg 
<b>Killer</b> Hím 3 kg 	<b>Marcsi</b> Nőstény 320 kg 	<b>Kati</b> Nőstény 13 kg 	<b>Norbi</b> Hím 5 kg 
<b>Kaller</b> Hím 5 kg 	<b>Nándi</b> Hím 4 kg 	<b>Karez</b> Hím 15 kg 	<b>Krumpli</b> Hím 10 kg 

# Osztályok, típusok

## AllatFaj

Enum

Kutya

Nyul

Medve

## Allat

Class

### Fields

- faj
- himmemu
- nev
- tomeg

### Properties

- Faj
- Himmemu
- Nev
- Tomeg

### Methods

- Allat

## Ketrec

Class

### Fields

- aktualis
- allatok

### Properties

- Allatok

### Methods

- Ketrec
- push

# Programozási tételek összeépítése (2)

- **Feladat - kérdések**

Válaszoljuk meg az alábbi kérdéseket az előző oldalon látható mintaadatokból felépített modell alapján.

Az összetett feladatok során próbáljuk a már megismert programozási tételek (illetve az előzőleg elkészült részfeladatok) segítségével elkészíteni a megoldást. Tételek összeépítésénél használható mindhárom előadáson megismert összeépítési technika.

- **Egyszerű programozási tételek ( $N \rightarrow 1$ )**

- Megadott ketrecben hány darab megadott fajú állat található?

*int FajDarab(Allat[ ] A, AllatFaj faj)*

- Megadott ketrecben van-e megadott fajú és nemű állat?

*bool FajEsNemVanE(Allat[ ] A, AllatFaj faj, bool himnemu)*

- **Egyszerű programozási tételek ( $N \rightarrow N$ )**

- Megadott ketrecben melyek a megadott fajú állatok?

*Allat[ ] FajAllatok(Allat[ ] A, AllatFaj faj)*



# Programozási tételek összeépítése (3)

- **Programozási tételek összeépítése**

- Megadott ketrecben mennyi a megadott fajú állatok átlagos tömege?

*float AtlagFajTomeg(Allat[ ] A, AllatFaj faj)*

- Megadott ketrecben melyik a legnehezebb megadott fajú állat?

*Allat FajLegnehezebb(Allat[ ] A, AllatFaj faj)*

- Megadott ketrecben hány (a ketrecen belül a saját fajára számított) átlagosnál nehezebb állat van?

*int AtlagnalNehezebbDarab(Allat[ ] A)*

- Melyik ketrecben van a legtöbb (a ketrecen belül a saját fajára számított) átlagosnál nehezebb állat?

*int LegtobbAtlagnalNehezebb(Ketrec[] K)*

- Hány olyan ketrec van, ahol az előzőleg kiszámolt számú átlagosnál nehezebb állat található?

*int LegtobbAtlagnalNehezebbDarab(Ketrec[] K)*

# Programozási tételek összeépítése (4)

- **Programozási tételek összeépítése**

- Melyik ketrecben található a legtöbb megadott fajú állat?

*int LegtobbFaj(Ketrec[] K, AllatFaj faj)*

- Megadott ketrecben van-e legalább egy azonos fajú, de ellenkező nemű egyedekből álló páros?

*bool AzonosFajEllenkezoNemVanE(Allat[] A)*

- Megadott ketrecben tartozik-e mindenkihez legalább egy azonos fajú, de ellenkező nemű állat? (ahhoz nem ragaszkodunk, hogy mindenkihez egy kizárólagos pár tartozzon)

*bool AzonosFajEllenkezoNemMindenkinek(Allat[] A)*

- Hány olyan ketrec van, ahol van legalább egy azonos fajú, de ellenkező nemű tagokból álló páros?

*int AzonosFajEllenkezoNemDarab(Ketrec[] K)*

- Hányas számú ketrecekben nincs egy azonos fajú, de ellenkező nemű egyedből álló pár se?

*int[] AzonosFajEllenkezoNemNincs(Ketrec[] K)*

# Összetett feladatok megoldása (1)

## • Feladat - mintaadatok

### Receptkönyv

besamel

**Liszt**

50 g



**Vaj**

50 g



**Tej**

50 g



**Tej**

100 g



tejberizs

**Rizs**

100 g



**Tej**

100 g



**Tej**

200 g



**Vaj**

10 g



kijeve

**Vaj**

10 g



**Hús**

100 g



**Vaj**

50 g



**Hagyma**

10 g



rostélyos

**Vaj**

10 g



**Hús**

100 g



**Liszt**

10 g



**Hagyma**

30 g



rántott

**Hús**

100 g



**Liszt**

10 g



**Tojás**

10 g



### Raktár

**Tej**

300 g



**Vaj**

300 g



**Liszt**

300 g



**Rizs**

300 g



**Hús**

50 g



**Teri**

szereti

**Hús**



**Feri**

szereti

**Vaj**



**Liszt**



**Rizs**



**Mari**

szereti

allergiás

**Vaj**



**Liszt**



**Hús**



**Tojás**



### Vendégek

# Osztályok, típusok

**Alapanyag**  
Enum

- Liszt
- Vaj
- Tej
- Rizs
- Hus
- Hagyma
- Tojas

**Hozzavalo**  
Class

- Fields
  - alapanyag
  - mennyiseg
- Properties
  - Alapanyag
  - Mennyiseg
- Methods
  - Adatok
  - Hozzavalo

**Recept**  
Class

- Fields
  - hozzavalok
  - nev
- Properties
  - Hozzavalok
  - Nev
- Methods
  - HozzavalokOssze...
  - KulonbozoAlapan...
  - MegvalosithatoH...
  - MennyireSikeres
  - Recept
  - ReceptOsszMenn...
  - TartalmazAlapan...

**ReceptKonyv**  
Class

- Fields
  - receptek
- Methods
  - AlapanyagotTart...
  - HanyfeleReceptV...
  - HanyReceptNemt...
  - Legbonyolultabb...
  - LegsikerebbRe...
  - MelyikReceptekV...
  - ReceptKivalasztas
  - ReceptKonyv
  - Szemelyreszabot...

**Vendeg**  
Class

- Fields
  - allergias
  - szereti
- Methods
  - AllergiasRa
  - Ertekel
  - Vendeg

# Összetett feladatok megoldása (2)

- **Feladat - kérdések**

Válaszoljuk meg az alábbi kérdéseket az előző oldalon látható mintaadatokból felépített modell alapján (zárójelben megadott osztályban).

- **Egyszerű segéd metódusok**

- Recept tartalmaz megadott alapanyagot? (Recept)  
*bool TartalmazAlapanyagot(Alapanyag alapanyag)*
- Mennyi egy recept hozzávalóinak összesített mennyisége? (Recept)  
*public float ReceptOsszMennyisege( )*
- Receptkönyv egy megadott nevű receptjének a kiválasztása (ReceptKonyv)  
*Recept ReceptKivalasztas(string nev)*
- Hány recept nem tartalmaz egy megadott alapanyagot? (ReceptKonyv)  
*int HanyReceptNemtartalmazAlapanyagot(Alapanyag alapanyag)*
- Melyik receptek tartalmazznak egy megadott alapanyagot? (ReceptKonyv)  
*Recept[ ] AlapanyagotTartalmazoReceptek(Alapanyag alapanyag)*
- Megadott vendég allergiás-e egy megadott alapanyagra? (AllergiasVendeg)  
*bool AllergiasRa(Alapanyag alapanyag)*

# Összetett feladatok megoldása (3)

- **Receptek hozzávalóival kapcsolatos kérdések**

- Recept hányféle egymástól különböző alapanyagot tartalmaz? (Recept)  
*int KulonbozoAlapanyagokSzama( )*

- Melyik a legbonyolultabb recept (amelyik a legtöbb különböző alapanyagot tartalmazza)? (ReceptKonyv)  
*public Recept LegbonyolultabbRecept( )*

- Az egyes hozzávalókból összesítve mennyire van szükség a recept elkészítéséhez? (Recept)  
*public Hozzavalo[ ] HozzavalokOsszesitve( )*

- Megadott hozzávalókból elkészíthető-e a recept? (Recept)  
*bool MegvalosithatoHozzavalokbol(Hozzavalo[ ] raktar)*

- Hányféle receptet lehet elkészíteni megadott hozzávalókból? (ReceptKonyv)  
*int HanyfeleReceptValosithatoMegHozzavalokbol(Hozzavalo[ ] raktar)*

- Melyik recepteket lehet elkészíteni megadott hozzávalókból? (ReceptKonyv)  
*Recept[ ] MelyikReceptekValosithatokMegHozzavalokbol(Hozzavalo[ ] raktar)*

# Összetett feladatok megoldása (4)

- **Vendégekkel kapcsolatos kérdések**

- Mennyire jónak értékel egy vendég egy receptet (az általa szeretett hozzávalók mennyiségének az összege)? (Vendeg)

*virtual float Ertekel(Recept recept)*

- Valósítsuk meg az értékelést az allergiás vendégek esetén is (ha allergiás bármelyik hozzávalóra, akkor 0, egyébként a szokásos! (AllergiasVendeg)

*override float Ertekel(Recept recept)*

- Megadott vendégnek melyik receptet ajánljuk? (ReceptKonyv)

*Recept SzemelyreszabottAjanelat(Vendeg vendeg)*

- Mennyire jó egy recept egy társaság számára (a megadott vendégek értékelésének átlaga)? (Recept)

*float MennyireSikeres(Vendeg[ ] vendegek)*

- Megadott társaság számára melyik a legsikeresebb recept? (ReceptKonyv)

*Recept LegsikeresebbRecept(Vendeg[ ] vendegek)*

# Programozási tételek – házi feladat

## • 1. Feladat

Egy 364 napos tömbben tároljuk az egy év (52 hét) alatt mért napi átlaghőmérsékleteket. Ezek ismeretében válaszoljunk az alábbi kérdésekre:

- Hány napon érte el a hőmérséklet az éves maximum 90%-át?
- Melyik volt az év legmelegebb hete (maximális heti átlaghőmérséklet)?
- Hány olyan hét volt, amikor legalább egyszer fagyott?
- Hány olyan hét volt, amikor minden nap fagyott?
- Melyik napon volt a legnagyobb lehülés az előző naphoz képest?
- Hány olyan nap volt, amikor előző és következő nap fagyott, de aznap nem?
- Mikor volt a leghosszabb időszak, amikor folyamatosan esett a hőmérséklet?
- Hányszor volt az évben kánikula? (legalább 3 napig 35°C feletti hőmérséklet)
- Volt-e olyan min. 5 napos időszak, amely értékei megismétlődtek később?
- Milyen hosszú volt az a leghosszabb időszak, amikor egyszer se fagyott?
- Milyen hosszú volt az a leghosszabb időszak, amelyen belül egyszer se fagyott egymást követő 5 napon át?



# Összetett feladatok megoldása – házi feladat

## • 2. Feladat

Tervezzük meg és implementáljuk egy egyszerű szerszámkölcsönző cég rendszerét, amely megvalósítja az alábbi funkciókat:

- Tárolja az aktuális időpontot (az év hányadik napja), ami menüből léptethető
- Tárolja a kölcsönözhető eszközök adatait (típus {porszívó, fűrógép, lángvágó}, állapot {0..1}, alapidj mértéke 10 napra, késedelmi díj mértéke ezt követően naponta)
- Tárolja a kölcsönző személyek adatait (név, aktuális kölcsönzési adatok – eszköz, kölcsönzés ideje {max. 5 db}, előjegyzési lista {max. 5 db})
- Kölcsönzéskor a megadott típus alapján automatikusan válasszuk ki a legjobb állapotú, legdrágább eszközt és rögzítsük a kölcsönzést. Legyen lehetőség listázni az összes kölcsönzést
- Ha nincs elérhető eszköz, vegyünk fel előjegyzést a legkorábban lejáró eszközre. Legyen lehetőség listázni az összes előjegyzést
- Legyen lehetőség visszaadni egy eszközt, a program számolja ki az alapidj és a késedelmi díj alapján a fizetendő összeget (illetve csökkentse az eszköz állapotát nap/100-al). Ha volt előjegyzés az eszközre, akkor automatikusan kerüljön át az új személyhez.
- Legyen lehetőség meghosszabbítani egy kölcsönzést, de csak akkor, ha arra az eszközre még nincs előjegyzés
- Selejtezéskor töröljük az eszközt és helyettesítsük a rá vonatkozó előjegyzéseket egy hasonlóval
- Listázzuk eszköztípusonként, hogy aktuális napon mennyi bevételt hoznak a cégnek

Töltsük fel a rendszert adatokkal, és menüvezérelt módon legyen lehetőség elérni a fenti funkciókat!