

TÖMBÖK

C#

készítette: Vastag Attila
2016

Véletlen szám generálás

- C#-ban véletlen számot a Random osztály segítségével generálhatunk
- Első lépésben egy példányt kell készíteni a Random osztályból:

Random rnd = new Random();

- Ezzel rnd néven létrejött egy példány a Random osztályból, ezen keresztül használhatjuk a következő metódusait:
 - **rnd.Next(max);** 0 és max közötti egész számot ad vissza
 - **rnd.Next(min,max);** min és max közötti egész számot ad vissza
 - **Rnd.NextDouble();** 0.0 és 1.0 közötti valós értéket ad eredményül

Véletlen szám generálás

`Random rnd = new Random();`

- `int a= rnd.Next();` → $0 \leq a \leq 2,147,483,647$
- `int b= rnd.Next(1000);` → $0 \leq a < 1000$
- `int c= rnd.Next(10,99);` → $10 \leq a < 99$
- `int d= rnd.NextDouble();` → $0.0 \leq x \leq 1.0$

Tömb

Emlékezz arra a feladatra, mikor 3 számot kellett növekvő sorrendbe tenni! Már ez a három szám sorba rakása is gondot okozott!

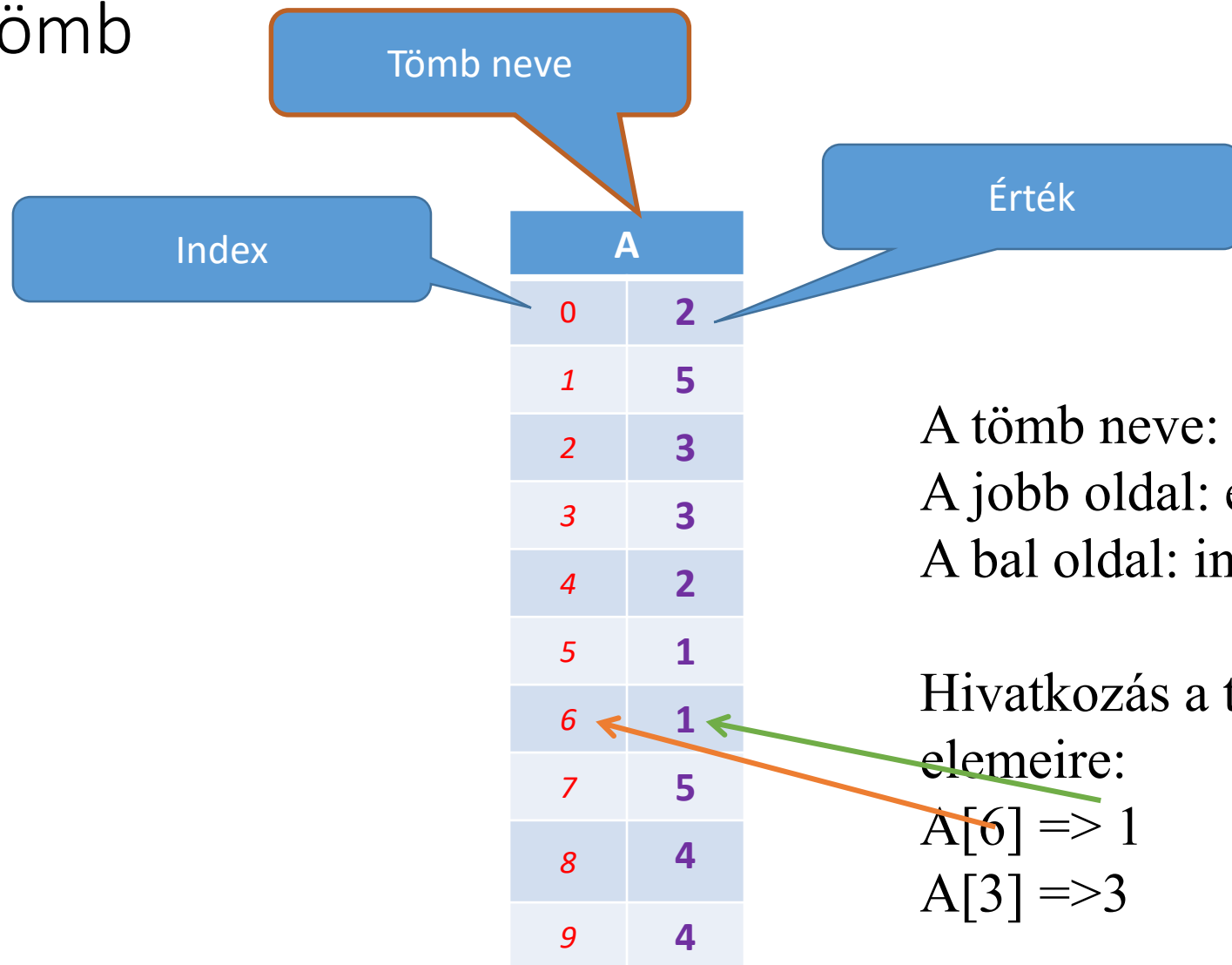
Most megmutatjuk, hogy ezt lehet egyszerűbben is, például tömbök segítségével. De nem csak e probléma megoldására alkalmasak, hanem számos más matematika műveleteknél is alkalmazhatóak.

- Többször adódik olyan feladat, amelyben több azonos típusú adatot kell tárolni, majd feldolgozni.
- Pl. Kérjünk be 10 osztályzatot 1-5 között, majd számoljuk ki az átlagot
- Első, nem túl hatékony megoldás az, ha létrehozunk 10 egész változót...☹
- Másik megoldás a tömbök használata

Tömb fogalma

- A tömb homogén adatszerkezet, melynek elemeit a tömbön belül elfoglalt helyük azonosít
 - Homogén: Minden eleme kizárólag azonos típusú lehet (egész, valós, stb...)
 - Tömböm belül elfoglalt helyük azaz az **indexük** segítségével tudjuk azonosítani az egyes tömb elemeink értékét!

Tömb



Tömb deklarációja

- A tömb deklarációjához megadjuk a típusát, majd mögé tesszük szögletes zárójelpárt [], aztán jöhet a tömb neve. Ezzel még a memóriában nem foglalódik hely a tömbnek

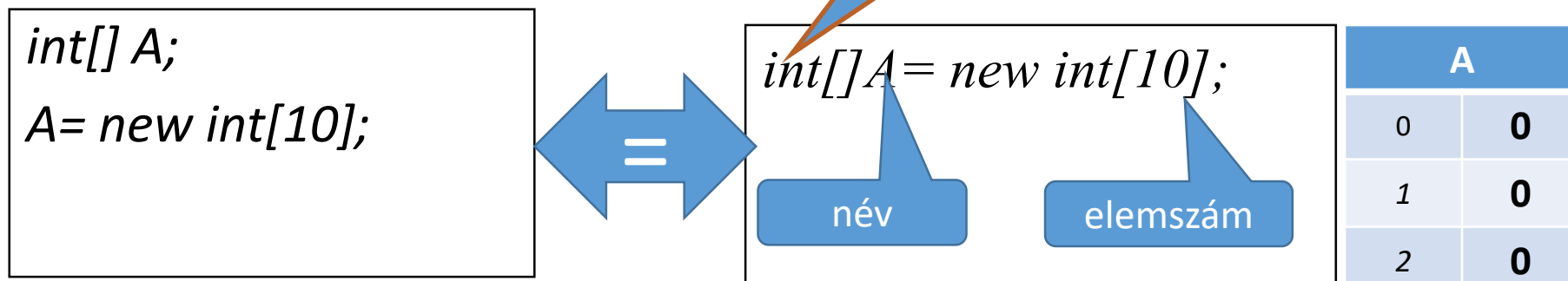
int[] A = null;

- Ahhoz, hogy a memóriában foglalódjon hely a tömb számára a **new** operátor és a tömb elemszámának megadása szükséges

A = new int[10];

int[] A = new int[10];

Tömb deklarációja



A két deklaráció ugyanaz, általában a második, tömörebb megoldást használjuk!

A deklarációval létrejön egy 10 elemű egészeket tartalmazó tömb, amelyben minden elem értéke 0.

A	
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0

Tömb deklarációja

- A tömb első elemének indexe 0, a másodiké 1 stb...
- A tömb utolsó elemének indexe egyel kevesebb, mint az elemszáma
- A tömb egyes elemeit a tömb neve mellett elhelyezkedő szögletes zárójelbe tett index segítségével érhetjük el.

Tömbelemek elérése

- Értékadás tömbelemnek:

$A[4] = 10;$

- Tömbelem értékének kiolvasása:

$int\ b = A[7]; \rightarrow b = 6$

A	
0	-5
1	-3
2	1
3	8
4	10
5	5
6	1
7	6
8	7
9	12

Tömbelemek ellátása kezdőértékkel

- A tömbelemekhez a deklarációnál rendelhetünk kezdőértéket (kezdőértékkel való ellátás: inicializálás)

```
int[] A = new int[10]  
{  
    3,4,53,23,102,43,0,53,33,87  
};
```

A	
0	3
1	4
2	53
3	23
4	102
5	43
6	0
7	53
8	33
9	87

Tömb elemszáma

- A tömb elemszámát nem kötelező deklarációnál megadni:

int[] A = new int[] {3,4,53,23,0,53,33,87};

- A tömb elemeinek számát a tömb **Length** tulajdonságából olvashatjuk ki:

int elemszam = A.Length;

(elemszam = 8)

Tömb feldolgozása

- A tömb feldolgozása azt jelenti, hogy végiglépkedünk a tömb elemein és különböző műveleteket hajtunk végre rajtuk.
- A tömb elemeit általában for ciklus segítségével dolgozzuk fel

```
for (int i=0; i < A.Lenght; i++)  
{  
  
    A[i] elem feldolgozása  
  
}
```

- ***A[i]***-t a ciklusmagban ***aktuális tömbelemnek*** nevezzük!

Tömb feltöltése véletlen számokkal

```
int[] A = new int[10];  
Random rnd = new Random();  
for (int i=0; i < A.Length; i++)  
{  
    A[i] = rnd.Next(100);  
}
```

1. *Létrehozunk egy példányt a Random osztályból*
2. *A for ciklus kezdőértéke 0*
3. *Ha az i (ciklusváltozó) értéke nagyobb mint a tömb elemszáma akkor a ciklus után folytatjuk a végrehajtást, különben*
4. *Végrehajtjuk a ciklusmagot, amelyben az aktuális tömbbelem A[i] –nek adunk egy véletlen értéket 0-100 között*

Tömb feltöltése felhasználói bevitellel

```
int[] A = new int[10];  
for (int i=0; i < A.Length; i++)  
{  
    Console.Write($"Kérem az {i + 1} .-ik számot: ");  
    A[i] = Int.Parse(Console.ReadLine());  
}
```

1. *A for ciklus kezdőértéke 0*
2. *Ha az i (ciklusváltozó) értéke nagyobb mint a tömb elemszáma akkor a ciklus után folytatjuk a végrehajtást, különben*
3. *Végrehajtjuk a ciklusmagot, amelyben kiírjuk, hogy kérjük az aktuális (i) tömbelemet, majd az A[i] -ik elemét beolvassuk a Console.ReadLine() -al.*
4. *Visszaugrunk a 3-as lépésre*

Ezzel a tömbelemek nem lesznek kiírva a képernyőre!

Tömb elemeinek kiírása

```
int[] A = new int[] {3,4,53,23,0,53,33,87};  
for (int i=0; i < A.Length; i++)  
{  
    Console.WriteLine($"A[{i}] = {A[i]} ");  
}
```

1. *A for ciklus kezdőértéke 0*
2. *Ha az i (ciklusváltozó) értéke nagyobb mint a tömb elemszáma akkor a ciklus után folytatjuk a végrehajtást, különben*
3. *Végrehajtjuk a ciklusmagot, amelyben a ciklus minden egyes körében kiírjuk az aktuális (i) tömbelemet, az A[i] -ik elemet a következő kimenetet kapva:*
4. $A[0] = 3$
5. $A[1] = 4$
6. $A[2] = 53$

Foreach ciklus

- A ciklusok negyedik fajtája C#-ban a foreach ciklus, amelynek segítségével egy gyűjtemény elemein (tömb) tudunk végighaladni:

foreach (*típus* ciklusváltozó **in** tömbnév)

```
{  
    változónévben tárolt aktuális tömbelem feldolgozása  
}
```

A *foreach* végiglépked a tömb összes elemén indexelés nélkül. Minden egyes körben az *ciklusváltozó* értéke a tömb egyik elemének fog megfelelni.

Tömb elemeinek kiírása II.

```
int[] A = new int[4]  
{  
    3,4,53,23  
};
```

```
foreach (int x in A)  
{  
    Console.WriteLine($"{{x}}");  
}
```

Kimenet:

3
4
53
23

Feladatok I.

1. Töltsünk fel egy 10 elemű tömböt random számokkal:
 - a) Írassuk ki a tartalmát fordított sorrendben
 - b) Számítsuk ki az elemek összegét
 - c) Átlagoljuk a tömbelemeket
 - d) Írassuk ki a páros elemeket
 - e) Számoljuk meg, hogy hány két jegyű szám van a tömbben
 - f) Írassuk ki az egyjegyű számokat
 - g) Számítsuk ki a páratlan számok összegét
 - h) Számoljuk meg hány nullára végződő szám van a tömbben
 - i) Rakjuk sorba a tömböt növekvő/csökkenő sorrendbe

Feladatok II.

2. Töltsünk fel egy tömböt véletlenszerű 3 jegyű egész számokkal

- a) Számítsuk ki az elemek összegét
- b) Számítsuk ki az elemek átlagát
- c) Számoljuk meg hány szám van 500 felett!

3. Dobjunk egy dobókockával egymás után 7-szer, az eredményeket tároljuk el!

- a) Számoljuk ki a dobások átlagát
- b) Számoljuk meg hány hatos dobás történt
- c) Számoljuk meg hány dobás volt páratlan

Feladatok III.

4. Töltsünk fel egy tíz elemű *char* típusú tömböt random számokkal (0 – 255 közt), majd határozzuk meg hány szám, betű és szimbólum van a tömbben.

Feladatok IV.

- 5 Kérjük be egy dolgozó bruttó fizetését 12 hónapra (minden hónapra külön-külön).
- Számoljuk ki mennyi SZJA –t fizet (az évi fizetés összegének 33,5%-a)
- Számoljuk ki mennyi egészségügyi hozzájárulást (SZJA 45%-a) és nyugdíjalapot fizetett (SZJA 55%-a)

5 Lottósorsolás. Generáljunk 7 random számot (1 - 45 közt), de ezek nem ismétlődhetnek a tömbben (do-while!). Majd kérjünk be a felhasználótól 7 számot, melyek szintén nem ismétlődhetnek és tároljuk el egy tömbben. Határozzuk meg a találatok számát. A tippek után írjuk ki a kisorsolt számokat és a tippelt számokat is.

Feladatok V.

6. Töltsünk fel a felhasználó által definiált hosszúságú tömböt random számokkal (-100 és 100 közt), majd határozzuk meg:

- a) Pozitív és negatív számok számát a tömbben
- b) A legkisebb számot és annak indexét
- c) Keressük ki a legnagyobb számot
- d) Hozzunk létre egy új tömböt, amely az eredeti fordítottja
- e) Számoljuk meg hány olyan elem van, mely nagyobb a tömb átlagától
- f) Rakjuk sorba a tömböt csökkenő sorrendbe
- g) Írjuk ki a tömb átlagánál kisebb számokat
- h) A páros vagy a páratlan számok száma a nagyobb
- i) Van e olyan szám amelyik 5-re végződik

Feladatok VI.

8. Program bekér egy mondatot, kiírja hogy 'a' vagy 'e' betűből tartalmaz-e többet.
9. Program bekér egy mondatot, a szóköz karaktereket lecseréli #-re.
10. Program bekér egy mondatot, kiírja a szavak számát.

Feladatok VII.

Készítsen programot egy család heti kiadásainak nyilvántartásához!
Program, kérje be hétfőtől vasárnapig a család kiadásait napi bontásban!
(A napok nevét konstans tömbben tároljuk!)

Írja ki:

1. A heti összkiadást!
2. Melyik napon volt a legkisebb a kiadás és mennyi?
3. Volt-e 10000 Ft-os kiadás?

Egy jégkorong csapat tagjainak száma 9 fő.

Program kérje be a csapat tagjainak nevét és a bajnokságban szerzett góljainak számát!

Írja ki:

1. Listázza ki az átlag alatt teljesítők nevét és szerzett góljainak számát!
2. Hány versenyző teljesített átlag felett?
3. Ki szerezte a legtöbb gólt és mennyit?

Feladatok VII.

Program kérjen be egy kézilabda a csapat tagjainak nevét (6) és a bajnokságban szerzett góljainak számát számát!

Írja ki:

1. Hány versenyző teljesített átlag alatt?
2. Listázza ki az átlag felett teljesítők nevét és szerzett góljainak számát!
3. Ki szerezte a legkevesebb gólt és mennyit?

Írjon programot, amely 10 tanuló nevét és megtakarított pénzét!

1. Írja ki a megtakarítások összegét!
2. Írja ki az átlagos megtakarítást!
3. Kinek van a legnagyobb megtakarítása és mennyi?
4. Kinek van a legkisebb megtakarítása és mennyi?
5. Hány főnek van 2000 Ft felletti megtakarítása?
6. Van-e olyan tanuló akinek nincs megtakarítása?
7. Listázza ki azon tanulók nevét, akiknek a megtakarítása átlag alatti!

Feladatok VII.

Egy forgalom ellenőrző ponton 10 autó adatait vizsgálják. A program kérje be az autók rendszámát és sebességét (ranom számmal 30 és 200 közt)!

1 - Írja ki azon autók rendszámát, sebességét és büntetését, amelyek 90 km/h felett közlekedtek.

- Büntetési tételek
- 91-100 km/h 10 ezer FT
- 101-110 km/h 20 ezer FT
- 110 km/h felett 30 ezer FT

2 - Írja ki a hatóság összesen mekkora büntetést szabott ki?

3 - Írja ki a leggyorsabb autó rendszámát és sebességét!

4 - Írja ki hány autó közlekedett szabályosan és ez az összes áthaladó autó hány százalékát tette ki?

5 - Írja ki, hogy volt-e 60 km/h-val közlekedő autó?

Feladatok VII.

Egy üzemanyag kútnál 10 autó tankolt. Kérje be az autók rendszámát és a tankolt mennyiséget!

- 1 - Írja ki, azon autók rendszámát és a tankolt mennyiséget amelyek 40 liter felett tankoltak!
- 2 - Írja ki, az autók összesen hány liter üzemanyagot tankoltak?
- 3 - Írja ki, melyik autó tankolt a legtöbbet és mennyit!
- 4 - Írja ki, melyik autó tankolt a legkevesebbet és mennyit!
- 5 - Írja ki, hány autó tankolt 30 liter alatt?
- 6 - Írja ki, volt-e olyan autó, amely pontosan 50 litert tankolt?

Feladatok VII.

Program gyümölcsraktár készletének nyilvántartásához.

Kérje be 10 gyümölcs nevét, mennyiségét és egységárát!

- Írja ki összesen hány kg gyümölcs van készleten!
- Írja ki gyümölcs fajtánként a készletértéket (mennyiség egységár szorzata)!
- Írja ki a teljes készletértéket!
- Írja ki a legdrágább gyümölcs nevét és árát!
- Melyik gyümölcsből van a legkevesebb a készleten és mennyi?
- Listázza ki azon gyümölcsök adatait, ahol a mennyiség 100 kg alatti!
- Van-e olyan gyümölcs a nyilvántartásban, amelynek egységára 1500 Ft feletti?

Feladatok VII.

Program kérje be a kosárlabda csapat tagjainak (5 játékos) nevét és a bajnokságban szerzett pontjainak számát, írja ki:

- a csapat a szezonban összesen hány pontot szerzett,
- ki szerezte a legtöbb pontot és mennyit,
- ki szerezte a legkevesebbet és mennyit,
- hány versenyző teljesített átlag alatt,
- listázza ki az átlag felett teljesítők nevét és pontszámát.

Feladatok VII.

Program kérje be egy kézilabda a csapat tagjainak nevét (6 játékos) és a bajnokságban szerzett góljainak számát számát!

Írja ki:

1. Hány versenyző teljesített átlag alatt?
2. Listázza ki az átlag felett teljesítők nevét és szerzett góljainak számát!
3. Ki szerezte a legkevesebb gólt és mennyit?

Írjon programot, amely bekéri 10 tanuló nevét és megtakarított pénzét!

1. Írja ki a megtakarítások összegét!
2. Írja ki az átlagos megtakarítást!
3. Kinek van a legnagyobb megtakarítása és mennyi?
4. Kinek van a legkisebb megtakarítása és mennyi?
5. Hány főnek van 2000 Ft fellelti megtakarítása?
6. Van-e olyan tanuló akinek nincs megtakarítása?
7. Listázza ki azon tanulók nevét, akiknek a megtakarítása átlag alatti!

Kamion mérőállomáson a szállított árú tömegét ellenőrzik (5 kamion).

A mérés után feljegyzik a kamion rendszámát és az össz tömegét.

A felhasználótól kérjük be a kamion rendszámát, random szám segítségével állítsuk elő a tömeget tonnában értelmezve (3.5 - 40).

0 - Irja ki az adatokat.

1 - Mekkora volt az össz tömeg?

2 - Mekkora volt az átlag tömeg?

3 - Melyik kamionon volt a legnehezebb?

4 - Volt-e 10t-s mérés?

5 - Írjuk ki a 10t-nal nehezebb kamionok rendszámát.

6 - Hanyadiknak mertek meg a legkisebb kamiont?

Feladatok VII.

Kérje be 10 egy zenei nagylemez zeneszámjainak sorszámát, címét és hosszát másodpercekben.

A sorszámot automatikusan kell, hogy a program hozzárendelje, kezdve az 1 számtól.

A hossz min 20s és max 500s lehet, ezt random számmal állítjuk elő!

A cím nem lehet kevesebb mint 1 karakter.

Készítse el az adatok tárolására alkalmas adatszerkezetet. E öszetett adatszerkezetnek definiálja felül a *ToString()* függvényét is, hogy a következő mintának megfelelően írja ki az adatszerkezetet:

{sorszám} – {cím} ({min:sec}) -> pl. 1 - Love is all (3:46)

- a) Milyen hosszú a lemez percekben feltüntetve?
- b) Milyen hosszúak az albumon szereplő zeneszámok átlagban?
- c) Melyik a legrövidebb zeneszám?
- d) Van e 4min-től hosszabb zeneszám?
- e) Hanyadik zeneszám a leghosszabb a lemezen?
- f) Van-e két egyforma hosszúságú zeneszám a lemezen? Írjuk ki őket ha van!

Feladatok VII.

Kérje be egy 6 tagú röplabdás csapat adatait: név, magasság, évi össz pontszám.

A magasság min 150cm, még az évi össz pontszám nem lehet negatív szám.

- a) Keresse ki azon játékosokat, akik alacsonyabbak a csapat átlagánál.
- b) Hány százaléka magasabb a csapatnak az átlagnál.
- c) Hány pontot szerzett összesen ez a csapat a szezonban?
- d) Ki szerezte a legtöbb pontot?
- e) Ki szerezte a legkevesebb pontot?
- f) Készítsen egy adatszerkezetet, melyben el tudja tárolni a legalacsonyabb és legmagasabb játékost, valamint a magasság különbségét közöttük. E összetett adatszerkezetnek definiálja felül a *ToString()* függvényét is, hogy a következő mintának megfelelően írja ki az adatszerkezetet:

A legmagasabb játékos {játékos neve}({magasság}).

A legalacsonyabb játékos {játékos neve} ({magasság}).

A különbség {különbség}cm.