

Programozás I.

Szöveges fájlok kezelése
String műveletek
Feladatok

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.

Programozás I.

Szöveges fájlok kezelése

String műveletek

Feladatok

Szöveges fájlok kezelése

- A szöveges fájlok egyes soraiban stringek találhatóak
- A fájlokat soronként előlről hátrafelé haladva tudjuk bejárni
- A fájl háromféle módon nyithatjuk meg
 - Olvasás: Ilyen esetben soronként ki tudjuk olvasni a fájl tartalmát
 - Írás: Ilyenkor új fájl jön létre, melybe soronként írhatunk, vagy a már létező fájl törlődik és egy ugyanolyan nevű új fájlba írhatunk soronként
 - Hozzáfűzés: Ekkor a fájlunk végére tudunk új sorokat írni
- A fájlkezelés végén a fájlt kötelező bezárnunk
- A fájlműveletek a `System.IO` névtérben találhatóak

Fájl olvasása

- A StreamReader osztályt kell használnunk

```
StreamReader sr = new StreamReader("fajlnev.txt");
```

- Egy sor kiolvasása a fájlból

```
string s = sr.ReadLine();
```

- Fájl végének ellenőrzése

```
sr.EndOfStream
```

- Fájl bezárása

```
sr.Close();
```

```
StreamReader sr = new StreamReader("fajlnev.txt");  
while (!sr.EndOfStream)  
{  
    string s = sr.ReadLine();  
    Console.WriteLine(s);  
}  
sr.Close();
```

Fájl írása

- A StreamWriter osztályt kell használnunk

```
StreamWriter sw = new StreamWriter("fajlnev.txt");
```

- Egy sor kiírása a fájlba

```
sw.WriteLine(s);
```

- Fájl bezárása

```
sw.Close();
```

```
StreamWriter sw = new StreamWriter("fajlnev.txt");  
foreach (string s in sTomb)  
{  
    sw.WriteLine(s);  
}  
sw.Close();
```

Hozzáfüzés és karakterkódolás

- Ha hozzá akarunk fűzni már létező fájlhoz, akkor a fájl megnyitásakor kell ezt megadnunk

```
StreamWriter sw = new StreamWriter("fajlnev.txt", true);
```

- `false` használata felülírást eredményez

```
StreamWriter sw = new StreamWriter("fajlnev.txt", false);
```

- A szövegfájlok karakterkódolása a mentő programtól, beállítástól függhet
- A StreamWriter-t úgy kell megnyitni, hogy a szövegfájlnak megfelelő karakterkódolásban olvasson
 - Alapértelmezetten UTF-8-ban próbál olvasni
- StreamWriter megnyitása karakterkódolás megadásával:

```
StreamWriter sw =  
    new StreamWriter("fajlnev.txt", Encoding.Default);
```


Elérési út megadása

- A fájlnev megadásánál megadhatunk abszolút elérési utat:
`StreamWriter sw = new StreamWriter("c:\\hallgato\\fajlnev.txt");`
- `StreamWriter sw = new StreamWriter(@"c:\\hallgato\\fajlnev.txt");`
- Nem ajánlott, más rendszeren nem lesz pontosan azon a helyen a fájl

- Relatív elérési út megadása:

- `StreamWriter sw = new StreamWriter("fajlnev.txt");`
- `StreamWriter sw = new StreamWriter("szovegek\\fajlnev.txt");`
- Ez a megadott nevű fájlt a munkakönyvtárban (working directory) keresi
- A munkakönyvtár egyszerű esetben az a könyvtár, ahol az exe van

Feladatok

1. Egy string tömbbe olvassa be egy szöveges fájl sorait, majd a sorokat írja ki a konzolra!
2. Határozza meg, hogy hány sora van a beolvasott fájlnek, illetve hány betű található benne.
A fájl végére írjon két új sort, mely ezeket az adatokat tartalmazzák!
3. Olvasson be egy fájlt, majd a beolvasott szöveget alakítsa át úgy, hogy csak a betűket és számokat tartsa meg. Az eredménnyel írja felül az eredeti fájlt!

Programozás I.

Szöveges fájlok kezelése

String műveletek

Feladatok

Műveletek karaktersorozatokkal

- A karaktersorozat („string”) karakterek halmazaként is felfogható
- Mivel gyakran használt, igen fontos típusról van szó, rengeteg beépített segédfunkció áll rendelkezésre hozzá
- Néhány kiemelt művelet és segédfunkció:
 - Összekapcsolás (+ operátor)
 - Részszorozat kiválasztása (**Substring** függvény)
 - Részszorozat keresése (**IndexOf /LastIndexOf, Contains**)
 - Konverziók (**változónév.ToString()** és **típusnév.Parse()**)
 - Kis- és nagybetűs formára alakítás (**ToUpper, ToLower**)
 - Formázott megjelenítés (**String.Format**)
 - Karaktersorozat kezelése karakterenként

Műveletek karaktersorozatokkal

- Összekapcsolás

```
class Összekapcsolás
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        string str1 = "Szervusz";
```

```
        string str2 = "C#";
```

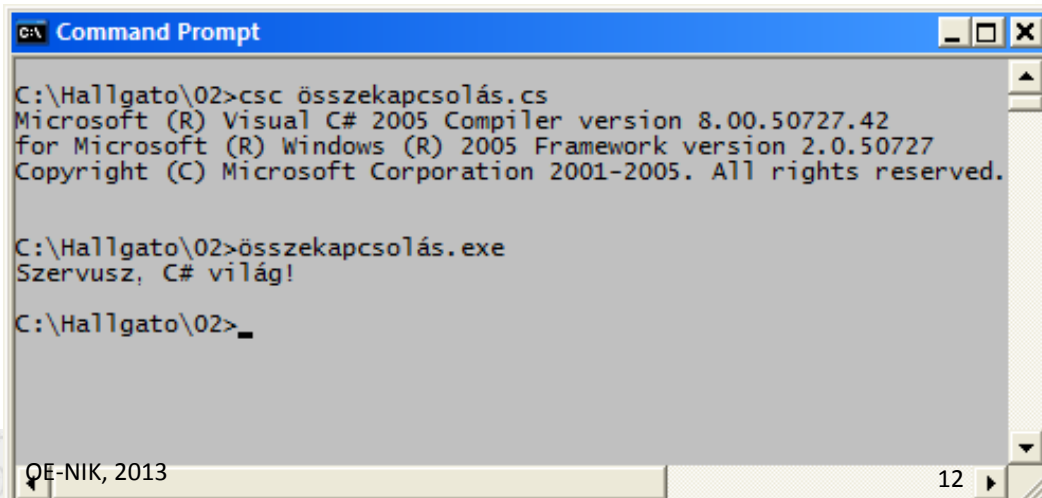
```
        string str3 = "világ!";
```

```
        string str4 = str1 + ", " + str2 + " " + str3;
```

```
        System.Console.WriteLine(str4);
```

```
    }
```

```
}
```



```
C:\> Command Prompt

C:\Hallgato\02>csc összekapcsolás.cs
Microsoft (R) Visual C# 2005 Compiler version 8.00.50727.42
for Microsoft (R) Windows (R) 2005 Framework version 2.0.50727
Copyright (C) Microsoft Corporation 2001-2005. All rights reserved.

C:\Hallgato\02>összekapcsolás.exe
Szervusz, C# világ!

C:\Hallgato\02>
```

Műveletek karaktersorozatokkal

```
int proba = 0;
string be;
do
{
    Console.Write(proba + ". próba: ");
    be = Console.ReadLine();
    proba++;
} while (be == "");

Console.Write(proba + 1 + ". próba: ");
Console.Write("A(z) " + proba + 1 + ". próba: ");
Console.Write("A(z) " + (proba + 1) + ". próba: ");
```

Műveletek karaktersorozatokkal

- Metódusok hívása: `stringváltozó.valami();`
- Előtte inicializálni kell a változót
- A forrásváltozót a metódus hívása NEM módosítja, a művelet eredménye a kimenetben lesz
- `stringváltozó=stringváltozó.valami();` ← visszaírás
- `stringváltozó2=stringváltozó.valami();` ← átírás
- Az első karakter sorszáma: 0

Műveletek karaktersorozatokkal

```
class Részsorozat
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        string s1, s2;
```

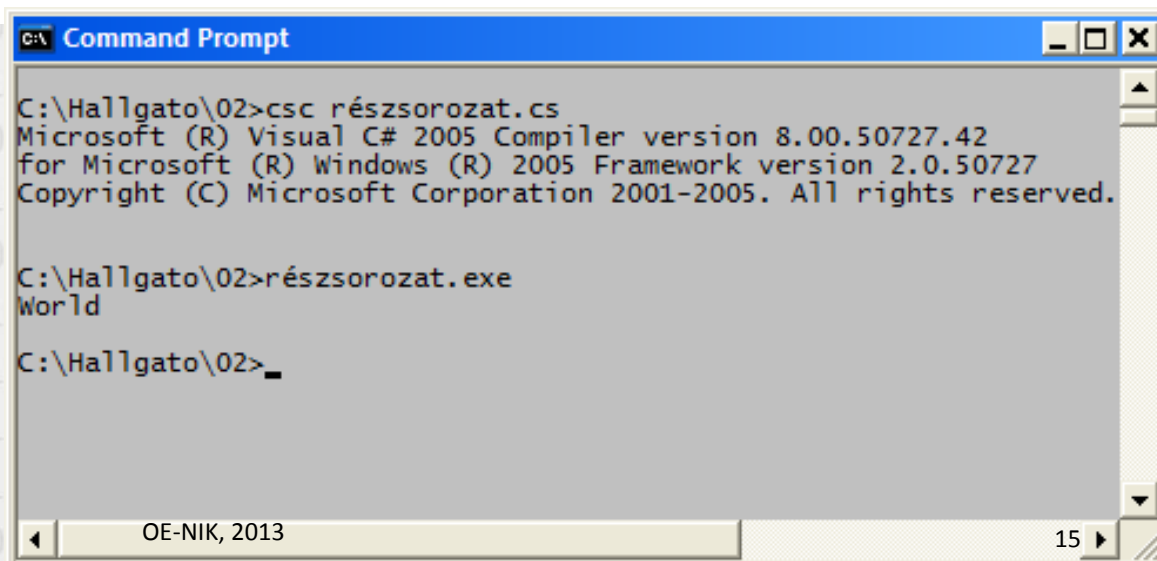
```
        s1 = "Hello, World";
```

```
        s2 = s1.Substring(7, 5); // Kezdő index: 0
```

```
        System.Console.WriteLine(s2);
```

```
    }
```

```
}
```



```
C:\> Command Prompt

C:\Hallgato\02>csc részsorozat.cs
Microsoft (R) Visual C# 2005 Compiler version 8.00.50727.42
for Microsoft (R) Windows (R) 2005 Framework version 2.0.50727
Copyright (C) Microsoft Corporation 2001-2005. All rights reserved.

C:\Hallgato\02>részsorozat.exe
World

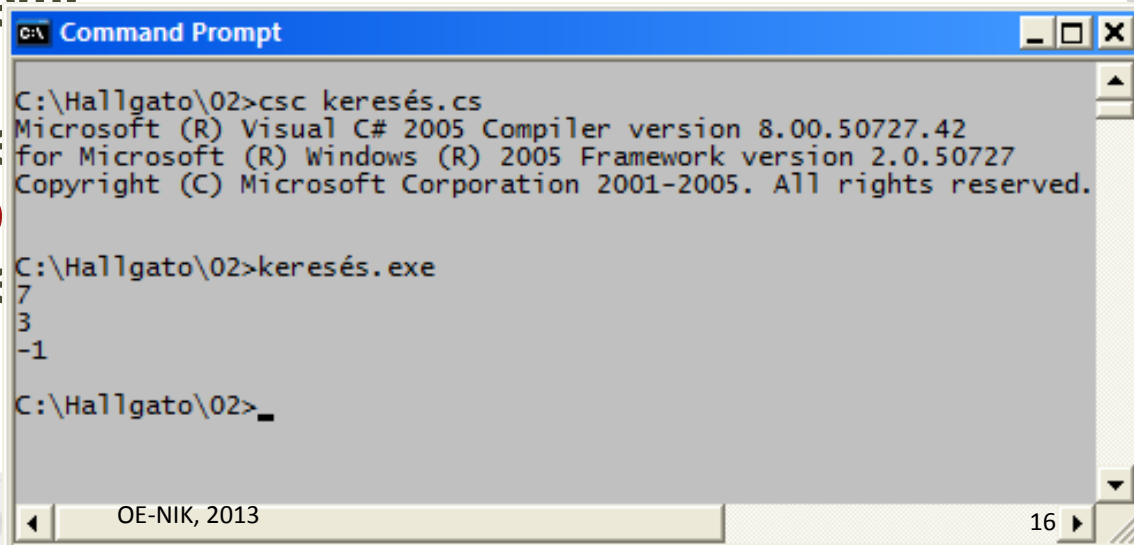
C:\Hallgato\02>
```


Műveletek karaktersorozatokkal

- Részszorozat keresése: `IndexOf(substr)` / `Contains(substr)`

```
class Keresés
```

```
{  
    static void Main()  
    {  
        int i;  
        string s1;  
        s1 = "Ez egy karaktersorozat";  
        i = s1.IndexOf("karakter");  
        System.Console.WriteLine(i);  
        i = s1.IndexOf("egy");  
        System.Console.WriteLine(i);  
        i = s1.IndexOf("ez nincs benne");  
        System.Console.WriteLine(i);  
    }  
}
```



The screenshot shows a Windows Command Prompt window with the following text:

```
C:\Hallgato\02>csc keresés.cs  
Microsoft (R) Visual C# 2005 Compiler version 8.00.50727.42  
for Microsoft (R) Windows (R) 2005 Framework version 2.0.50727  
Copyright (C) Microsoft Corporation 2001-2005. All rights reserved.  
  
C:\Hallgato\02>keresés.exe  
7  
3  
-1  
  
C:\Hallgato\02>
```

Műveletek karaktersorozatokkal

- Konverziók
- A stringgé történő konverzió a C# nyelven MINDEN változónál ugyanúgy történik:

byte b=250;

float f=3.14f;

string s1=b.ToString();

string s2=f.ToString();

- Stringből számmá tudunk konvertálni:

string s="123";

string s2="123,456";

byte b=**byte**.Parse(s);

float f=**float**.Parse(s2);

Műveletek karaktersorozatokkal

- Kis- és nagybetűs formára alakítás

```
class CsupaKisÉsNagybetű
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        int i;
```

```
        string s;
```

```
        i = 1982;
```

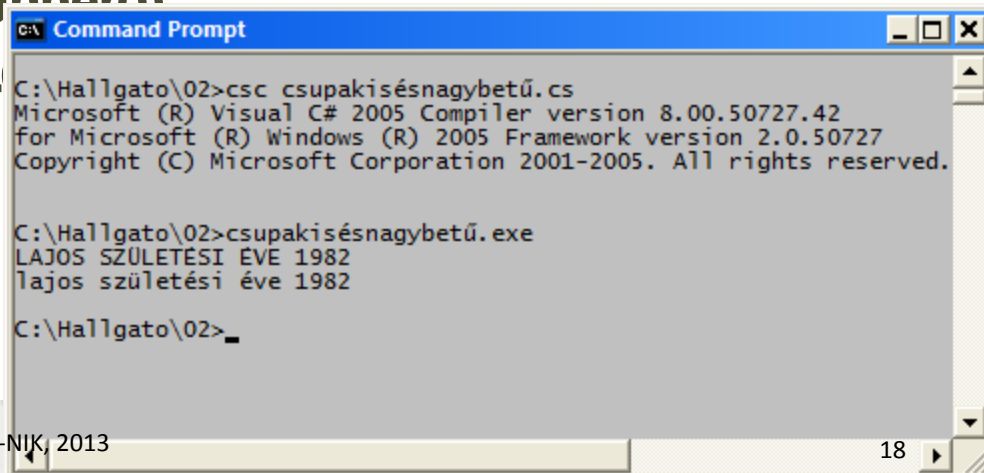
```
        s = "Lajos születési éve " + i;
```

```
        System.Console.WriteLine(s.ToUpper());
```

```
        System.Console.WriteLine(s.ToLower());
```

```
    }
```

```
}
```



```
C:\> Command Prompt

C:\Hallgato\02>csc csupakisésnagybetű.cs
Microsoft (R) Visual C# 2005 Compiler version 8.00.50727.42
for Microsoft (R) Windows (R) 2005 Framework version 2.0.50727
Copyright (C) Microsoft Corporation 2001-2005. All rights reserved.

C:\Hallgato\02>csupakisésnagybetű.exe
LAJOS SZÜLETÉSI EVE 1982
lajos születési éve 1982

C:\Hallgato\02>
```

Műveletek karaktersorozatokkal

- Formázott megjelenítés

```
string mi = "árvíztűrő tükörfúrógép";
```

```
string milyen = "legjobb";
```

```
int db = 123;
```

```
float ar = 2.5f;
```

```
string kimenet = string.Format("Ha nekem {0, 5}  
darab {1}em lenne {2:F} forintért, az lenne a {3}",  
db, mi, ar, milyen);  
Console.WriteLine(kimenet);
```

➔ Ha nekem 123 darab árvíztűrő tükörfúrógépem lenne 2,50 forintért, az lenne a legjobb

Műveletek karaktersorozatokkal

- Formázott megjelenítés vezérlőkarakterei

Kód	Számtípus	Magyarázat	Példa
C	Egész és valós	Helyi pénznem formázási szabályai szerinti kijelzés	1 435,5 Ft (Magyarország) \$1435.5 (USA)
D	Csak egész	Általános egész szám	1435
E	Egész és valós	Tudományos jelölésmód	1,4355E+003 (Magyarország) 1.4355E+003 (USA)
F	Egész és valós	Fixpontos decimális számkijelzés	1435,50 (Magyarország) 1435.50 (USA)
G	Egész és valós	Általános számkijelzés	1435,5 (Magyarország) 1435.5 (USA)
N	Egész és valós	Helyi területi beállítások szerinti számkijelzés	1 435,500 (Magyarország) 1,435.500 (USA)
P	Egész és valós	Százalékos formátum	143 550,00 %
X	Csak egész	Hexadecimális formátum	59B

Műveletek karaktersorozatokkal

<u>Név</u>	<u>Feladat</u>	<u>Paraméterek</u>
Length	String hossza	NEM ELJÁRÁS → int adat
StartsWith(), EndsWith()	String elejének / végének ellenőrzése	substring → bool visszatérési érték
PadLeft(), PadRight()	String feltöltése extra karakterekkel	width / width, paddingChar
Trim(), TrimStart(), TrimEnd()	Whitespace eltávolítása	trimChars
Remove()	Részsorozat eltávolítása	index / index, count
Replace()	Részsorozat cseréje	string, string / char, char

Műveletek karaktersorozatokkal

- Karaktersorozat kezelése karakterenként (substring helyett)

```
class Karakterenként
```

```
{
```

```
    static void Main()
```

```
    {
```

```
        int i;
```

```
        string s = "Karaktersorozat";
```

```
        i = 1;
```

```
        while (i < s.Length)
```

```
        {
```

```
            System.Console.WriteLine(s[i]);
```

```
            i++;
```

```
        }
```

```
    }
```

```
}
```


Gyakorló feladatok

**Készítsen programot, amely egy stringben megkeresi egy adott karakter valamennyi előfordulását!
(a feladatot az `.IndexOf()` nélkül valósítsuk meg!)**

Készítsen programot, amely egy stringben kicserél minden *A* karaktert *B*-re (*A* és *B* legyen tetszőleges)!

Készítsen programot, amely egy adott karaktersorozatot (pl. „Amelyik kutya ugat, az a kutya nem harap”) minden adott karaktersorozatát (pl. „kutya”) egy adott karaktersorozatra (pl. „macska”) cseréli!

Programozás I.

Szöveges fájlok kezelése

String műveletek

Feladatok

Feladatok

- 1. Egy szöveges fájlt alakítson át úgy, hogy minden sor középre legyen rendezve!**
- 2. Egy szöveges fájlt alakítson át úgy, hogy minden sor sorkizárt legyen!**
- 3. Határozza meg, hogy egy szöveges fájlban melyik a leghosszabb szó, illetve melyik szó fordul elő leggyakrabban!**

Feladatok

4. Írjon menüvezérelt programot, amely lehetővé teszi egy futó napi edzéseinek eltárolását

1. Új edzés felvitele

- Dátum YYYYMMDD formátumban
 - Táv #,# km formában
 - Idő HH:MM:SS formátumban
- Az új edzés bekerül a futónapló fájl végére

2. Edzések listázása

A konzolra kilistázza az edzéseket

3. Statisztikák

A konzolra és egy statisztika fájlba kiírja az edzések számát, valamint a távok és idők összegét és átlagát

Feladatok

5. Készítsen szótárprogramot, amely szópárokat tárol, illetve ki is kérdezi a szavakat!

1. Lehessen új szópárokat felvenni, de csak akkor, ha még nincsenek bent a szótárban!
2. Lehessen kilistázni a szótár tartalmát!
3. Tudja kikérdezni a szavakat mindkét nyelv szavai alapján. A kikérdezés végén adja meg, hány helyes válasz érkezett!

Irodalom, feladatok

- **Kotsis-Légrádi-Nagy-Szénási: Többnyelvű programozástechnika, PANEM, Budapest, 2007**
- **Faraz Rasheed: C# School, Synchron Data, 2006**
<http://www.programmersheaven.com/2/CSharpBook>
- **Reiter István: C# jegyzet, DevPortal, 2010,**
<http://devportal.hu/content/CSharpjegyzet.aspx>

