

# Objektumorientált Programozás

## VI.

Tömb emlékeztető

Egyszerű programozási tételek

Összetett programozási tételek

# 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.**

# Objektumorientált Programozás

## VI.

Tömb emlékeztető

Egyszerű programozási tételek

Összetett programozási tételek

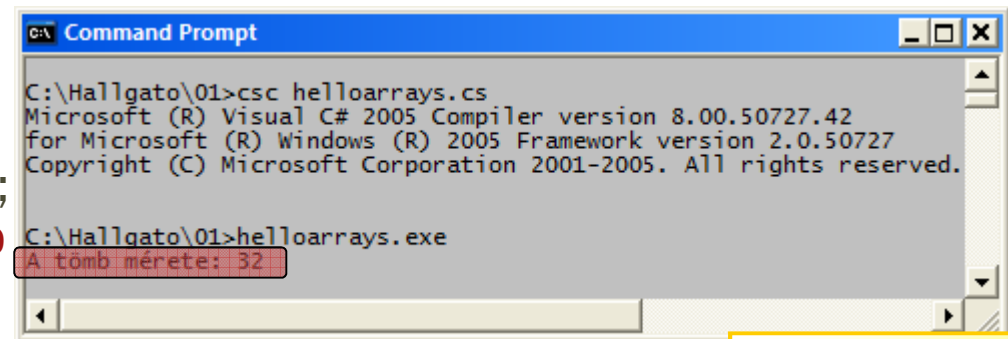
# Egyéb alaptípusok: a tömb

- A tömbök („array”) adattípusa bármilyen beépített típus vagy újonnan definiált saját típus lehet
- A tömbök indexelése 0-tól kezdődik
- A tömbök ún. referenciatípusok

```
class HarmadikProgram
{
    static void Main()
    {
        int[] egészTömb = new int[32];
        int[] ElőreMegadottTömb = {2, 3, 5, 7, 11, 13, 17, 19};

        egészTömb[20] = 1;
        egészTömb[31] = 9;

        int tömbméret = egészTömb.Length;
        System.Console.WriteLine("A tömb mérete: " + tömbméret);
    }
}
```



```
C:\Hallgato\01>csc helloarrays.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\01>helloarrays.exe
A tömb mérete: 32
```

helloarrays.cs

# Feladat

Egy tömbben neveket tárolunk (minden név egy string).  
Mindenkinek egy vezetékneve és egy vagy több keresztnéve lehet. Feltételezzük, hogy legalább egy eleme van a tömbnek, és nincsenek azonos nevek.

A neveket bekérhetjük a konzolról is, de a tesztelés gyorsítása érdekében célszerű fixen megadott adatokkal dolgozni (erre jól használható a tömbök inicializálása).

```
string[] A = new string[] { "Kiss Béla", "Nagy Margit", "Közepes Kálmán",  
                            "Tóth József Béla", "Balogh Márta Erzsébet", "Tóth Péter" };  
int N = A.Length;
```

# Objektumorientált Programozás

## VI.

Tömb emlékeztető

Egyszerű programozási tételek

Összetett programozási tételek

# Feladat (1)

1. Egy megadott nevű ember szerepel-e a listában, és ha igen, akkor hányadik helyen?

(keresés programozási tétel)

## Lineáris keresés

### Bemenet

$A$ : Feldolgozandó tömb  
 $N$ : Tömb elemeinek száma  
 $T$ : Tulajdonság függvény

### Kimenet

$VAN$ : Logikai változó  
 $SORSZ$ : Első  $T$  tulajdonságú elem indexe

## Pszeudokód

```
Eljárás Keresés( $A, N, T, VAN, SORSZ$ )  
   $i := 1$   
  Ciklus amíg ( $i \leq N$ ) és  $\neg(A[i]$  teljesíti  $T$ -t)  
     $i := i + 1$   
  Ciklus vége  
   $VAN := (i \leq N)$   
  Ha  $VAN$  akkor  
     $SORSZ := i$   
Eljárás vége
```



# Feladat (2)

## 2. Van-e megadott vezetéknevű ember a listában? (eldöntés programozási tétel)

### Eldöntés

#### Bemenet

$A$ : Feldolgozandó tömb  
 $N$ : Tömb elemeinek száma  
 $T$ : Tulajdonság függvény

#### Kimenet

$VAN$ : Logikai változó

### Pszudokód

Eljárás  $Eldöntés(A, N, T, VAN)$

$i := 1$

Ciklus amíg  $(i \leq N)$  és  $\neg(A[i]$  teljesíti  $T$ -t)

$i := i + 1$

Ciklus vége

$VAN := (i \leq N)$

Eljárás vége



# Feladat (3)

## 3. Hányadik helyen szerepel a leghosszabb nevű ember a listában?

(maximum kiválasztás programozási tétel)

### Maximumkiválasztás

#### Bemenet

*A*: Feldolgozandó tömb  
*N*: Tömb elemeinek száma

#### Kimenet

*MAX*: Maximális elem indexe

### Pszudokód

Eljárás Maximumkiválasztás(*A*, *N*, *MAX*)

*MAX* := 1

Ciklus *i* := 2-től *N*-ig

Ha  $A[i] > A[MAX]$  akkor

*MAX* := *i*

Elágazás vége

Ciklus vége

Eljárás vége

## Feladat (4)

### 4. ABC sorrendben melyik az első név? (minimum kiválasztás programozási tétel)

- Stringek ABC-rend szerinti összehasonlítása a CompareTo() függvénnel:

```
string a = "aaa";  
string b = "ccc";  
if (a.CompareTo(b) < 0)  
    System.Console.WriteLine("a van előbb");
```

- a.CompareTo(b) eredménye:
  - < 0, ha a megelőzi b-t a betűrendben.
  - == 0, ha a ugyanabban a pozícióban van a betűrendben, mint b.
  - > 0, ha a hátrébb van a betűrendben, mint b.

# Feladat (5)

## 5. Hány megadott vezetéknevű van a listában? (megszámlálás programozási tétel)

### Megszámlálás

#### Bemenet

$A$ : Feldolgozandó tömb  
 $N$ : Tömb elemeinek száma  
 $T$ : Tulajdonság függvény

#### Kimenet

$DB$ :  $T$  tulajdonságú elemek száma

### Pszudokód

Eljárás Megszámlálás( $A$ ,  $N$ ,  $T$ ,  $DB$ )

$DB := 0$

Ciklus  $i := 1$ -től  $N$ -ig

Ha ( $A[i]$  teljesíti  $T$ -t) akkor

$DB := DB + 1$

Elágazás vége

Ciklus vége

Eljárás vége

## **Feladat (6)**

**6. Hány olyan ember van a listában, akiknek egyik keresztnévük sem egyenlő egy megadottal?**

**(megszámlálás programozási tétel)**

# Objektumorientált Programozás

## VI.

Tömb emlékeztető

Egyszerű programozási tételek

Összetett programozási tételek

# Feladat (7)

## 7. Egy másik tömbbe másoljuk át a vezetékneveket! (másolás programozási tétel)

### Másolás

#### Bemenet

$X$ : Feldolgozandó tömb  
 $N$ : Tömb elemeinek száma

#### Kimenet

$Y$ : Eredmény tömb

### Pszudokód

Eljárás Másolás( $N$ ,  $X$ ,  $Y$ )  
Ciklus  $i := 1$ -től  $N$ -ig  
     $Y[i] :=$  művelet  $X[i]$   
Ciklus vége  
Eljárás vége





# Feladat (8)

8. Egy másik tömbbe válogassuk át a megadott vezetéknevű emberek neveit!

(kiválogatás programozási tétel)

## Kiválogatás

### Bemenet

$X$ : Feldolgozandó tömb  
 $N$ : Tömb elemeinek száma  
 $T$ : Tulajdonság függvény

### Kimenet

$Y$ : Eredmény tömb  
 $DB$ : Tömb elemeinek száma

### Pszekokód

Eljárás Kiválogatás( $N$ ,  $X$ ,  $DB$ ,  $Y$ )

$DB := 0$

Ciklus  $i := 1$ -től  $N$ -ig

Ha  $T(X[i])$  akkor

$DB := DB + 1$

$Y[DB] := X[i]$

Elágazás vége

Ciklus vége

Eljárás vége



# Feladat (9)

9. Egy tömbben tároljuk az ismerősök neveit. Válogassuk ki azokat, akik szerepelnek az eredeti tömbben is!

(metszet programozási tétel)

## Metszet

### Bemenet

*X*: Egyik feldolgozandó tömb  
*Y*: Másik feldolgozandó tömb  
*M*: *X* elemeinek száma  
*N*: *Y* elemeinek száma

### Kimenet

*Z*: Eredmény tömb  
*DB*: *Z* elemeinek száma

## Pszeudokód

```
Eljárás Metszet(M, X, N, Y, DB, Z)
  DB := 0
  Ciklus i := 1-től M-ig
    j := 1
    Ciklus amíg j ≤ N és X[i] ≠ Y[j]
      j := j + 1
    Ciklus vége
    Ha j ≤ N akkor
      DB := DB + 1
      Z[DB] := X[i]
    Elágazás vége
  Ciklus vége
Eljárás vége
```

# Objektumorientált Programozás

## VI.

- ✓ Tömb emlékeztető
- ✓ Egyszerű programozási tételek
- ✓ Összetett programozási tételek

# 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>**

