Budapesti Műszaki és Gazdaságtudományi Egyetem
Közlekedés- és Járműirányítási Tanszék

Számítástechnika I.

BMEKOKAA152

BMEKOKAA119

Infokommunikáció I.

BMEKOKAA606

Dr. Bécsi Tamás

4. előadás

Véletlen számok generálása a Random osztály

- System.Random

Metódus	Típus	Funkció
Next()	int	Egy véletlen nem-negatív int értékkel tér vissza.
Next(int maxvalue)	int	Egy véletlen nem-negatív int értékkel tér vissza, amely kisebb a megadott maximumnál (maxvalue).
Next(int minvalue, int maxvalue)	int	Egy véletlen int értékkel tér vissza, amelyre igaz, hogy minvalue<=value<maxvalue
NextDouble()	double	Egy véletlen double értékkel tér vissza, amelyre igaz, hogy 0.0<=value<1.0

A Math osztály (System.Math)

Metódus	Művelet
Math.Sin(x)	sin(x), ahol az x szög értékét radiánban kell megadni
Math.Cos(x)	cos(x)
Math.Tan(x)	tg(x)
Math.Exp(x)	e^x
Math.Log(x)	ln(x)
Math.Sqrt(x)	x négyzetgyöke
Math.Abs(x)	x abszolút értéke
Math.Round(x)	kerekítés a matematikai szabályok szerint
Math.Ceiling(x)	felfelé kerekítés
Math.Floor(x)	lefelé kerekítés
Math.Pow(x,y)	hatványozás, x^y
Math.PI	a PI konstans (3.14159265358979323846)
Math.E	az e konstans (2.71828182845904523 54)

Tömbök (Arrays)

- A tömb meghatározott számú, azonos típusú elemek halmaza.
- Minden elemre egyértelműen mutat egy index(egész szám).
- A tömbök referenciatípusok.
- A C# mindig folytonos memóriablokkokban helyezi el egy tömb elemeit.
- Definíció: `típus[] név; (pld. int[] szamok; )`
- Inicializálás:
  - `int[] szamok= new int[4];`
  - `int[] szamok= {1,2,4,3,5};`
  - `int[] a= new int[] {21,32,43,54,65,76,76,32};`

a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]
21	32	43	54	65	76	76	32

Tömbök (Arrays)

- Elemeke elérése: `szamok[0]=1;`
  - Az egyes elemekre az indexelő operátorral (szögletes zárójelek: **[]**) és az elem indexével (sorszámával) hivatkozunk.
  - A számozás mindig nullától kezdődik, így a legutolsó elem indexe: az elemek száma mínusz egy.
- Mivel a tömbök referenciatípusok, ezért példányosítani kell őket a `new` kulcsszó (operátor) segítségével
- Hossz lekérése, a **Length** tulajdonság:

```
int[] szamok= new int[4]{1,2,3,4};
int sum=0, i;
for(i = 0; i < szamok.Length; i++)
   sum += szamok[i];
```

Többdimenziós tömbök

- Definíció: típus[,] név; (pld. int[,] matrix;)

```
int[,] numbers = new int[3, 2];
int[,] numbers = new int[3, 2] {{1,2}, {3,4}, {5,6}};
int[,] numbers = new int[,] {{1,2}, {3,4}, {5,6}};
int[,] numbers= {{1,2}, {3,4}, {5,6}};
```

- A **Length** tulajdonság a teljes elemszámot adja vissza (6 jelen esetben)
- A **Rank** tulajdonság a mátrix dimenzióját adja vissza.
- A **GetLength(int i)** metódus adja vissza az adott dimenzió hosszát

Tömbök tömbje


```
int[][] haromszog= new int[5][];
for(int i = 0; i < haromszog.Length; i++)
{
  haromszog[i] = new int[i+1];
  for(int j = 0; j < haromszog[i].Length; j++)
  haromszog[i][j] = j;
}
```

```
haromszog[0]: { 0}
haromszog[1]: { 0 , 1 }
haromszog[2]: { 0 , 1 , 2 }
haromszog[3]: { 0 , 1 , 2 , 3 }
haromszog[4]: { 0 , 1 , 2 , 3 , 4 }
```