

# **BEVEZETÉS A PROGRAMOZÁSBA**

---

## **C# NYELVEN**

# Integrált fejlesztőkörnyezetek

---

## Integrated Development Environment (IDE)

**Microsoft Visual Studio Express 2013 for Windows Desktop**

Legújabb verzió: Visual Studio Community 2015

### **Integrált funkciók:**

- Kódszerkesztő
- Formázott kód megjelenítés
- Automatikus kódkiegészítés
- Fordítás
- Futtatás
- Hibajelzés, hibakeresés
- Stb.

# PROGRAM

---

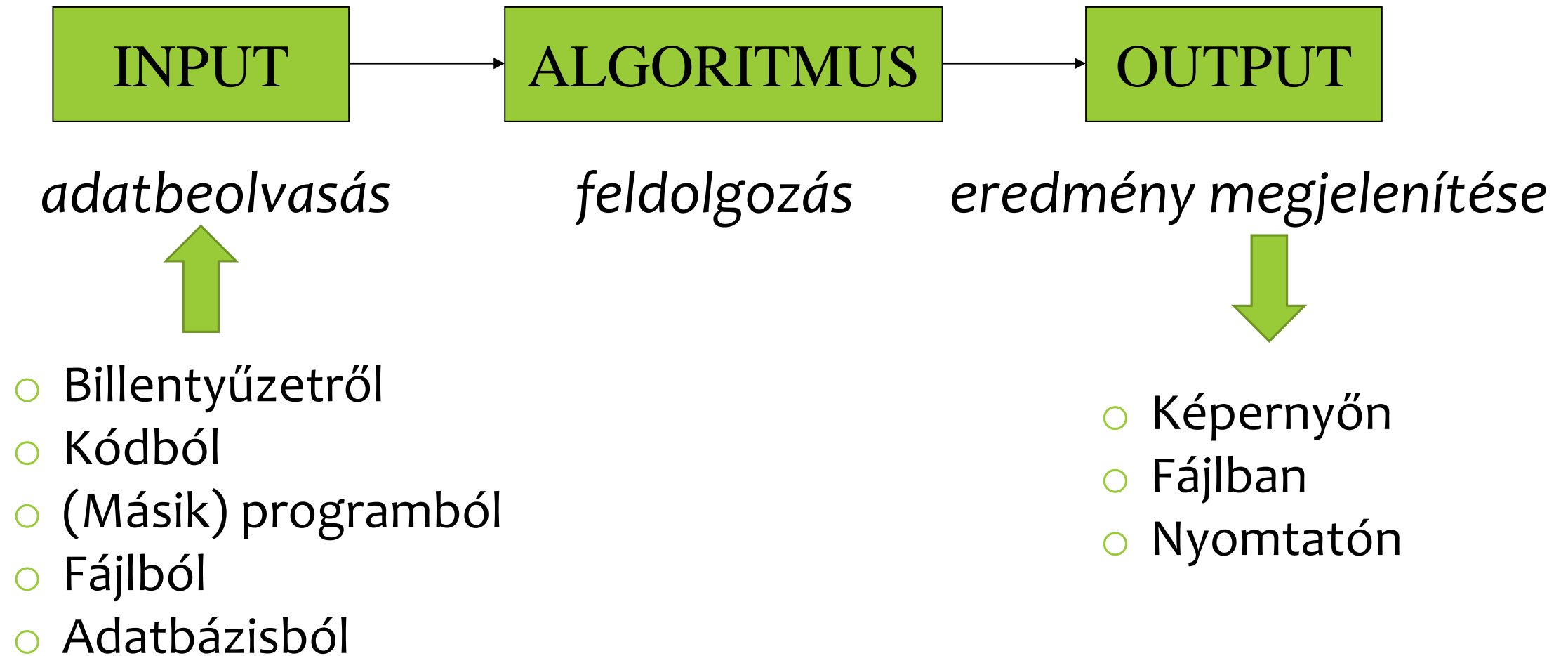
- Adatokon végzett műveletek, utasítások sorozata,
- amely egy feladat, probléma megoldására szolgál.
- Az utasítássorozat a számítógép számára értelmezhető nyelven van megfogalmazva.
- Az utasítássorozatot szoktuk algoritmusnak nevezni.

**program = adatok + algoritmusok**

*(Nicolas Wirth)*

# PROGRAM SZERKEZETE

---



# C# - általános jellemzők

---

- Különbséget tesz a kis és a nagy betűk között.
- Az utasításokat ; -vel zárjuk le.
- A program bizonyos egységeit ún. blokkokba helyezzük, a blokk jele a { }
- Megjegyzések: olyan sorok a kódban, amiket a fordító figyelmen kívül hagy
  - //ez most egy egysoros megjegyzés
  - /\* ez most
  - több soros
  - megjegyzés \*/

# Kiírás a képernyőre

`Console.Write("szöveg");`

Az idézőjelben levő szöveget szó szerint kiírja.

`Console.WriteLine(változó);`

A változóban levő értéket írja ki.

`Console.WriteLine("szöveg "+változó);`

Az idézőjelben levő szöveg mellé a változóban levő értéket írja.

`Console.Write("X értéke: {0}, Y értéke: {1}",x,y);`

A {0} helyére az x változó értékét, a {1} helyére az y változó értékét írja, a többi szövegrészt szó szerint.

# Formázási utasítások

---

1. Az x (szám típusú) változó értékét 2 tizedesjeggyel írja ki:  
`Console.Write("{0:0.00}",x);`
2. Az s változó tartalmát 5 helyen ábrázolja jobbra igazítva (az üres helyeket szóközzel tölti fel):  
`Console.Write("{0,5}",s);`
3. Az s változó tartalmát 5 helyen ábrázolja balra igazítva (az üres helyeket szóközzel tölti fel):  
`Console.Write("{0,-5}",s);`
4. Az x (szám típusú) változó ezres tagolással és 1 tizedesjeggyel írja ki:  
`Console.Write("{0:N1}",x);`

# Változók

---

A program végrehajtása során az adatokat változókbán tároljuk.

(Fizikailag a memória egy adott területén tárolódik az adat.)

A változónak van:

- **Neve**
- **Típusa**
- **Hatóköre**
- **Élettartama**



# Változó nevek a C#-ban

---

- max 32 karakter hosszú
- az 1. karakter betű vagy alulvonás
- a 2. karaktertől számok, betűk és alulvonás szerepelhet
- Ékezetes betűket is használhatunk, de nem számít túl elegánsnak
- Konvenció: kisbetűvel kezdjük, több szóból álló változóneveknél a szóhatáron nagybetűt írunk  
Pl: maxSebesseg, haviFizetes
- A „foglalt” szavak, ún. kulcsszavak (keywords), nem használhatók. Ezek megtalálhatók az alábbi linken:  
<https://msdn.microsoft.com/en-us/library/x53a06bb.aspx>

# Változók típusa

---

A típus határozza meg, hogy :

- egy változó milyen értékeket vehet fel
- mekkora helyet foglalhat el a memóriában
- milyen műveletek végezhetők vele

# Változók deklarációja

---

A **változókat** használat előtt **deklarálni kell**, azaz meg kell adni a változó

- Típusát
- Nevét

A C# ban a változó **deklaráció alakja** elemi típusú változók esetén:

**Típusnév változónév (= kezdőérték);**

A kezdőértékkadás nem kötelező a deklarációkor!

Példa: `int a=12;`  
          `string nev="Pali";`  
          `double x;`  
          `x=1.5;`

# Elemi típusok a C#-ban:

---

1. Karakter típusok
2. Egész típusok
3. Valós típusok
4. Logikai típus

# Karakter típusok

---

<b>char</b>	2 byte	1 karakter lehet char c='A'
<b>string</b>	tetszőleges hosszú	Karakterlánc string s=„alma”

# Egész típusok

Előjeles (negatív és pozitív)			Nem előjeles (csak pozitív)		
Típus	Méret	Érték	Típus	Méret	Érték
sbyte	8 bit	-128...127	byte	8 bit	0...255
short	16 bit	-32768... 32767	ushort	16 bit	0...65635
int	32 bit	-2 milliárd ... 2 milliárd	uint	32 bit	0 ... 4 milliárd
long	64 bit	$-10^{20}...10^{20}$	ulong	64 bit	$0...2*10^{20}$

# Valós szám típusok (törtek tárolására)

Típus	Méret	Pontosság
float	32 bit	6-7
double	64bit	15-16
decimal	128 bit	28-29

# Adatbekérés a billentyűzetről

---

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

Ha nem `string` típusú változóba akarjuk tárolni a beolvasott értéket, akkor típus átalakítás (konverzió) szükséges:

1. `int n=int.Parse(Console.ReadLine());`  
`double d=double.Parse(Console.ReadLine());`  
**VAGY**
2. `int n=Convert.ToInt32(Console.ReadLine());`  
`double d=Convert.ToDouble(Console.ReadLine());`



# A `Math` osztály konstansai és függvényei

---

1. A matematikai  $\pi$  szám 14 tizedesjeggyel: `Math.PI`
2. Hatványozás, pl.  $x^3$  : `Math.Pow(x, 3)` ;
3. Gyökvonás, pl.  $\sqrt{a + b}$  : `Math.Sqrt(a+b)` ;
4. Kerekítés, pl. az x számot 2 tizedesjegyre: `Math.Round(x, 2)` ;

Stb.