A C# nyelv elsajátítása

Hivatalos: https://learn.microsoft.com/hu-hu/dotnet/csharp/

# Bevezetés a C# nyelvbe és környezetbe

* Mi az a C# és .NET?
* A Visual Studio / Visual Studio Code telepítése
* Első "Hello World" program létrehozása
* Fordítás, futtatás, hibakeresés alapjai

# Alapvető programozási elemek

## Változók és elemi adattípusok (int, string, double, char, stb.)

## Operátorok és kifejezések

### Hozzárendelő operátorok

* = (értékadás)
* +=, -=, \*=, /=, %=

### Aritmetikai operátorok

Numerikus értékekkel végeznek műveletet és numerikus értéket állítanak elő

* + (összeadás)
* - (kivonás)
* \* (szorzás)
* / (osztás)
* % (maradékos osztás)

### Összehasonlító operátorok

Numerikus értékekkel végzik a műveletet és logikai (bool) értéket adnak vissza.

* == (egyenlő)
* != (nem egyenlő)
* < (kisebb)
* > (nagyobb)
* <= (kisebb vagy egyenlő)
* >= (nagyobb vagy egyenlő)

### **Logikai operátorok**

Logikai értékeken végeznek műveleteket és logikai értéket állítanak elő

* && (logikai ÉS - mindkét feltétel igaz, akkor igaz az eredmény)
* || (logikai VAGY - legalább az egyik feltétel igaz, akkor igaz az eredmény)
* ! (logikai NEM - tagadás, invertálja az értéket)

## Bemenet és kimenet

(Console.ReadLine, Console.WriteLine)

## Feltételes szerkezetek

* if,
* if else,
* switch

## Ciklusok

* for ciklus.
* while ciklus.
* do-while ciklus.
* foreach ciklus
* Ciklusvezérlő utasítások: break, continue.

# Adatstruktúrák (Gyűjtemények)

* **Tömbök:** Egydimenziós és többdimenziós tömbök, inicializálás, elérés.
* **Listák (List&lt;T>):** Dinamikus méretű gyűjtemények, alapvető műveletek (hozzáadás, törlés, keresés).
* Egyéb gyűjtemények bemutatása (röviden)
  + **Dictionary**<TKey, TValue>,
  + **HashSet**<T>.
* Enum-ok, struct-ok
* Delegáltak és események
* **Lambda** kifejezések
* **LINQ** alapjai (Lekérdezések kollekciókon)

# Függvények és metódusok

* Metódusok létrehozása és hívása
* Paraméterek, visszatérési értékek (in, out
* static vs. példány metódusok

# Fájlkezelés és kivételkezelés

* Fájl olvasás / írás (System.IO)
* Kivételkezelés: try, catch, finally, throw

# Objektum-orientált programozás (OOP)

* Osztályok és objektumok
* Metódusok, tulajdonságok (properties), konstruktorok
* Kapszuláció és hozzáférés módosítók
* Öröklés, polimorfizmus, absztrakt osztályok, interfészek
* Encapsuláció (adatelrejtés)

# .NET és könyvtárak használata

* Alapvető .NET könyvtárak
* A System, Collections, Text, IO namespace-ek használata

# Egységtesztelés alapjai

* NUnit / MSTest használata
* Egységtesztek írása, tesztelési szemlélet

# Aszinkron programozás és párhuzamosság

* async, await, Task
* Többszálúság alapjai (Thread, Parallel, ThreadPool)

Grafikus vagy webes fejlesztés bevezető

# WinForms

# WPF GUI

# ASP.NET webes backend alapjai