Alkfejl 2

A C # nyelv bemutatása

-A C# a Microsoft által a .NET keretrendszer részeként kifejlesztett objektumorientált programozási nyelv.

Referencia és értéktípus

-minden típus ősosztály a System.Object

-referenciatípus: a heapen foglal helyet

-string, saját és beépített osztályok, delegate

-amennyiben referenciatípust adunk egy másiknak, akkor nem az értéke másolódik, hanem ugyanarra a memóriaterületre (objektumra) fog mutatni.

-new operátorral hozhatók létre

-megszüntetésükről a Garbage Collector gondoskodik

-értéktípus: a veremben foglal helyet

-char, enmu, bool, struct, float, int, double (minden beépített numerikus típus)

-közvetlen hozzájuk rendelt értéket tartalmaznak

-konstruktor alapértelmezett 0 vagy null értéket ad

Propertyk:

Egy speciális adattagja az osztálynak, mely lehetővé teszi a privát változók hozzáférését, getter és setter.

Indexerek

Az indexer egy speciális tulajdonság C#-ban, amely lehetővé teszi az objektumok tömbszerű hozzáférését. Az index függvényt definiálva, az osztály index segítségével tudjuk kezelni a keresett értéket. Az indexer függvény vectorhasználat és tulajdonság függvény kombinációja.

-szintaxis: szögletes zárójelek []

-az aktuális típust azaz az osztály példányát indexeli

-nem lehet statikus mivel egy adott példányt indexel

-get blokk definiált csak olvasható

-set blokk definiált csak írható

-mindkettő definiált írható és olvasható is

-override: lehet megegyező nevű indexer különböző paraméterlistával

-nem használhatunk ref és out paramétreket

-nincs automatikusan definiálva

-nem lehet ksaját neve, az osztály nevén keresztül hivatkozunk rá a „this”-sel

```
class IntList
{
    private List<int> numbers = new List<int>();

    // Az indexer definiálása
    public int this[int index]
    {
        get
        {
            // Visszaadja a kívánt elemet az index alapján
            return numbers[index];
        }
        set
        {
            // Beállítja a kívánt elemet az index alapján
            numbers[index] = value;
        }
    }
```

Delegate-ek

A C#-ban függvénymutatók, amelyekkel olyan függvényekre tudunk referálni, amelynek megegyezik a deklarációja (paraméterlista és visszatérési érték)

-alkalmazható callback függvények létrehozására, mert megadhatunk más függvényeket argomentumként

-először definiálni kell egy delegate típust

-majd hozzárendelhetünk függvényeket amiknek megegyezik a deklarációja

-delegateket használhatjuk más függvények argomentumaként

-hivatkozhatunk több metódusra is egyszerre, feliratkozási sorrendben hívódnak meg

-feliratkozhatunk (+=)

-leiratkozhatunk(-=)

Func:

-speciális delegate

-lehetnek paraméterei

-egy visszatérési értéke van

-utolsó paraméter a visszatérési érték

Action:

-pontosan egy bemenő paramétere van

-nincs visszatérési értéke

Event-ek

Olyan mechanizmus, amely lehetővé teszik egy osztály számára hogy értesüljenek más osztály állapotváltozásairól. Az események segítségével a megfigyelő osztályunk feliratkozhat a megfigyelt osztály eseményeire, hogy reagáljanak azokra.

-speciális delegate

-taralmazza az event kulcsszót

-definiálható interfészben, míg a delegate nem

-csak deklaráló osztályból lehet előidézni (külsőleg nem)

-feliratkozás: add

-leiratkozás: remove

-érdemes használatakor érdemes beépített EventHandler-t használni, két paramétere van egy object típusú referencia a feladóra és egy EventArgs típusú objektu, amely az eseményhez kapcsolódó paramétereket követi

-érdemes Eventhandlert használni

Initializer

Objektum-inicializálók lehetővé teszik, hogy értéket rendeljünk az objektum adattagjaihoz vagy tulajdonságaihoz anélkül, hogy konstruktort hívtunk volna.

-szintaxis: kapcsos zárójelbe írjuk az adattak nevét és értékét

-különösen hasznos a LINQ lekérdezéseknél, hiszen gyakran használnak névtelen típusokat, amelyek csak objektum-inicializációval inicializálhatók

A C# programok fordítása és futtatása.

-Visual Studio build menüből futtatható le „Build Solution”

-lefordítja az összes projektet a Solution-be

-egy projekt esetén -> Solution Explorer

-Output ablak ->fordítási eredmény

-Debug menüből indítható el a project

-hibakereső futtatás

-hibakeresés nélküli futtatás

-parancssorból: dotnet run és fordítás -> dotnet build