



TM Szent László SZKIK TISZK  
Vályi Péter SZKIT

Készítette:

**Füredi János**

**Lipovits Ágnes**

**Pozsgai Tamás**

# **CSLEARNING**

# **C# OKTATÓANYAG**

**Számítógép-programozás tanításához**

**<http://cslpr.blogspot.hu>**

**2012**

## Tartalomjegyzék

1. Bevezetés.....	3
2. A keretprogram használata .....	5
3. A keretprogram szerkezete .....	7
4. Konzolalkalmazás készítésének folyamata .....	11
5. Windows alkalmazás kezelése .....	15
6. A leckék felépítése.....	16
7. Az alapsomagban kidolgozott témakörök.....	17
8. A csLearning használata hálózati környezetben.....	20
9. Fejlesztési lehetőségek .....	20
10. Összegzés .....	21
11. Felhasznált irodalom .....	23
12. Mellékletek .....	24

## 1. Bevezetés

„A múltban izomerőnket megsokszorozó  
eszközöket használtunk, ma pedig olyanokat,  
amelyekkel az elménket sokszorozzuk meg.”

Bill Gates

Az idők folyamán a programozás, a programnyelvek, a programozási technológiák jelentős változásokon mentek át. „Mindmáig a programozási nyelvek ezreit alkották meg: már 1969-ban a számítógépek mindennapi eszközzé válása előtt is 120 széles körben használt nyelv létezett. A legtöbb programozó azonban munkája során alig néhányat használ ebből. Kérdésként merülhet fel, hogy *mi az értelme újabb nyelvek megismerésének* és elsajátításának. Nos, a nyelvek alaposabb ismerete lehetővé teszi a feladathoz jobban illeszkedő nyelv kiválasztását, megkönnyíti az új programozási nyelvek elsajátítását, továbbá lehetővé teszi hatékonyabb algoritmusok készítését azáltal, hogy *növeli a programozók szókincsét.*”<sup>1</sup>

A mai középiskolás korosztály tanulási segédeszközei és információforrásai között már természetes módon előkelő helyet foglal el a számítógép és az internet. A feladatiak megoldásához programot választanak és optimális esetben törekednek minél igényesebb megoldásokra. A számítógépet használó emberek csillapíthatatlan szoftver-éhsége pedig arra ösztönzi a jó érzékű üzletemberek, és lelkes programozók hadát, hogy ezt a Kánaánt fenntartsák.

„A szoftver fejlesztési környezeteket tekintve az *utóbbi idők egyik legnagyobb és legtöbb újdonságot hozó* egységcsomagját kapták meg a fejlesztők az új *Microsoft .NET* rendszerrel. Ebben a környezetben, bármilyen területre gondolunk, új technológiák, új lehetőségek segítik a fejlesztők munkáját.”<sup>2</sup>

A C# programozási nyelv a Microsoft Visual Studio programcsomag része, egy modern, objektumorientált nyelv, amellyel a .NET keretrendszer alá készíthetünk alkalmazásokat. *A C# a .NET keretrendszer bázisnyelve*, teljesen komponensorientált. A fejlesztők a C++ hatékonyságát, és a Visual Basic fejlesztés gyorsaságát és egyszerűségét ötvözték ebben az eszközben. Éppen ez a sokszínűség és változatosság az, ami indokoltá teszi, hogy a számítógép-programozás oktatásában már a középiskolában a C# nyelvet használjuk. A programozás alapjainak tanítását konzolalkalmazások készítésével indíthatjuk, amelyekből megismerhető a program szerkezete, illetve felfedezhetők az

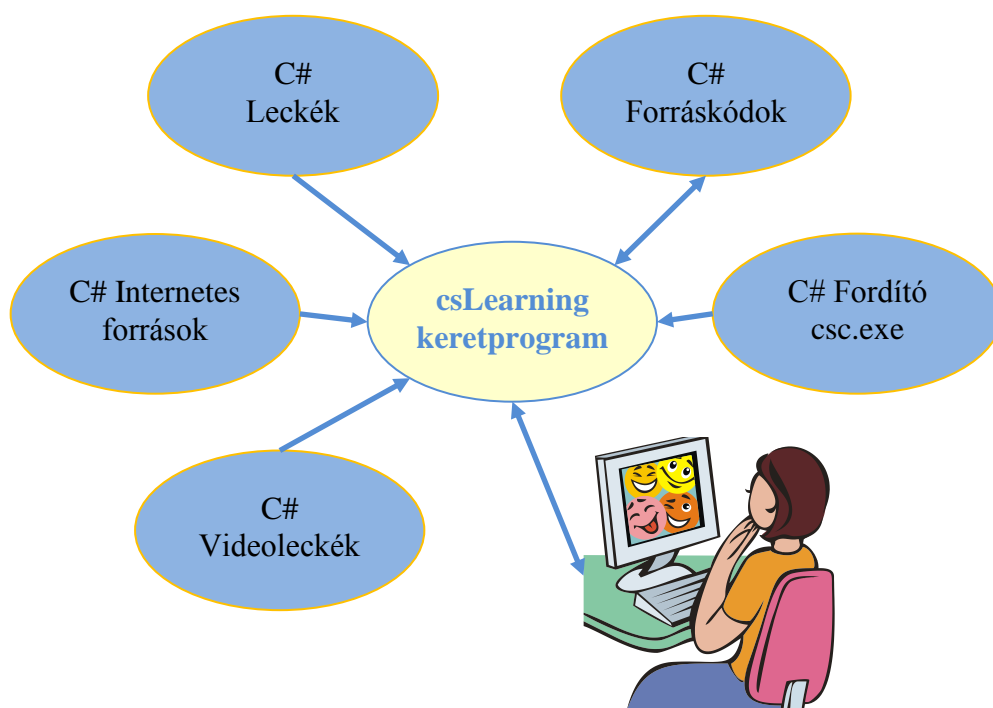
---

<sup>1</sup> NYÉKYNÉ GAIZLER JUDIT: Programozási nyelvek, Budapest, Kiskapu Kft. 2003. 7 p.

<sup>2</sup> ILLÉS ZOLTÁN: Programozás C# nyelven, Budapest, JOS Bt. 2008. 11 p.

adattípusok, a vezérlőszervezetek, a függvényhívások, a programozási tételek, majd ezt követően az OOP technológia ismeretében már korszerű Windows alkalmazások születhetnek.

A C# programnyelv tanulásához egy felől rendelkezésre áll az ingyenes Visual Studio C# Express fejlesztőkörnyezet, másfelől az oktató anyagok arzenálja. Ebben az igazán lenyűgöző kavalkádban szerettem volna egy szerény oázist kialakítani a csLearning keretprogram megírásával, a tematikus, fokozatosan bővíthető interaktív leckék megszerkesztésével, és az elérhető digitális publikációk beintegrálásával.

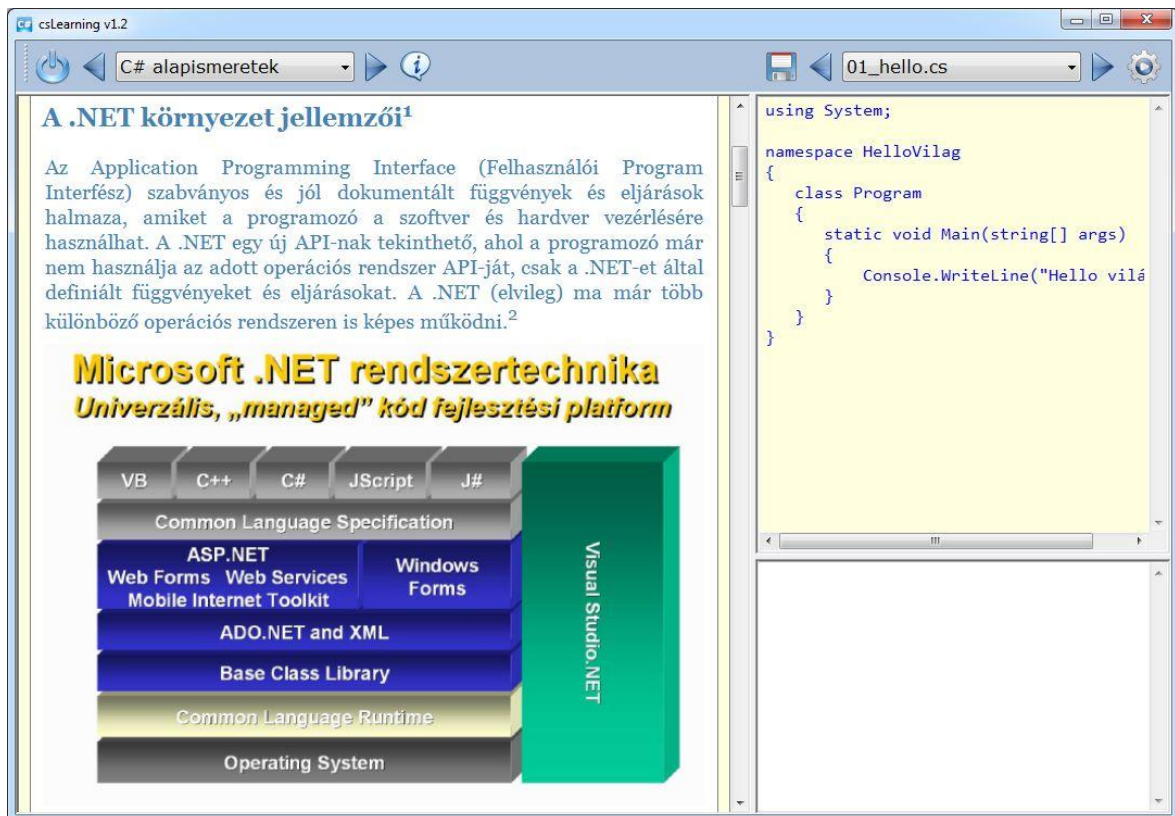


1. ábra: A csLearning környezet

A *keretprogram természetesen C# nyelven íródott* és lehetővé teszi, hogy a leckékben tárgyalt utasításokat, és az ebből készített forráskódokat közvetlenül a kezelőfelületén keresztül szerkeszthessük és futtassuk. A programozás tanulása során gyakran készítünk olyan kisebb mintaprogramokat, amelyekkel bizonyos utasítások használatát saját magunk számára próbáljuk érthetőbbé tenni. A csLearning lehetővé teszi, hogy tanulás közben leckénként magunk is bővíthessük a mintaprogramokat, így a későbbi ismétléseknél nem kell külön könyvtárakat böngészni egy-egy programozási ötlet felelevenítéséhez.



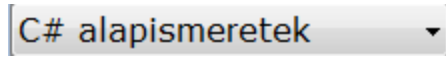




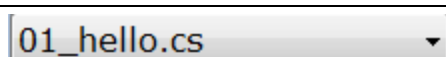


## 2. A keretprogram használata

A csLearning kezelőfelülete egy osztott alkalmazásablak, melynek a bal oldalán a leckék és a külső információforrások jelennek meg, míg a jobb oldali felső panelen a forráskódok szerkeszthetők, amelyek fordítási illetve futási üzeneteit a jobb oldali alsó panel mutatja meg abban az esetben, ha a program nem önálló ablakban kommunikál. A **tanulási folyamatot** az alkalmazásablak **címsora alatti eszköztárról irányíthatjuk**. A kezelőfelület tervezésekor az átláthatóság kedvéért egyfajta minimalista törekvés érvényesült, amelyet a leckék és forráskódok változatossága igyekszik otthonossá tenni.



2. ábra: A csLearning kezelőfelület

A bal oldali panel gyakorlatilag egy webBrowser, amelyben a HTML kódban elkészített leckéket tudjuk elolvasni. A felette lévő vezérlőkkel pedig a leckék közötti váltást kezdeményezhetjük. Ebben az esetben a jobb oldali panel is frissül a kiválasztott leckéhez tartozó programokkal. Egy leckéhez természetesen több mintaprogram is kapcsolódhat, amelyek között az eszköztár jobb oldali vezérlőivel mozoghatunk.

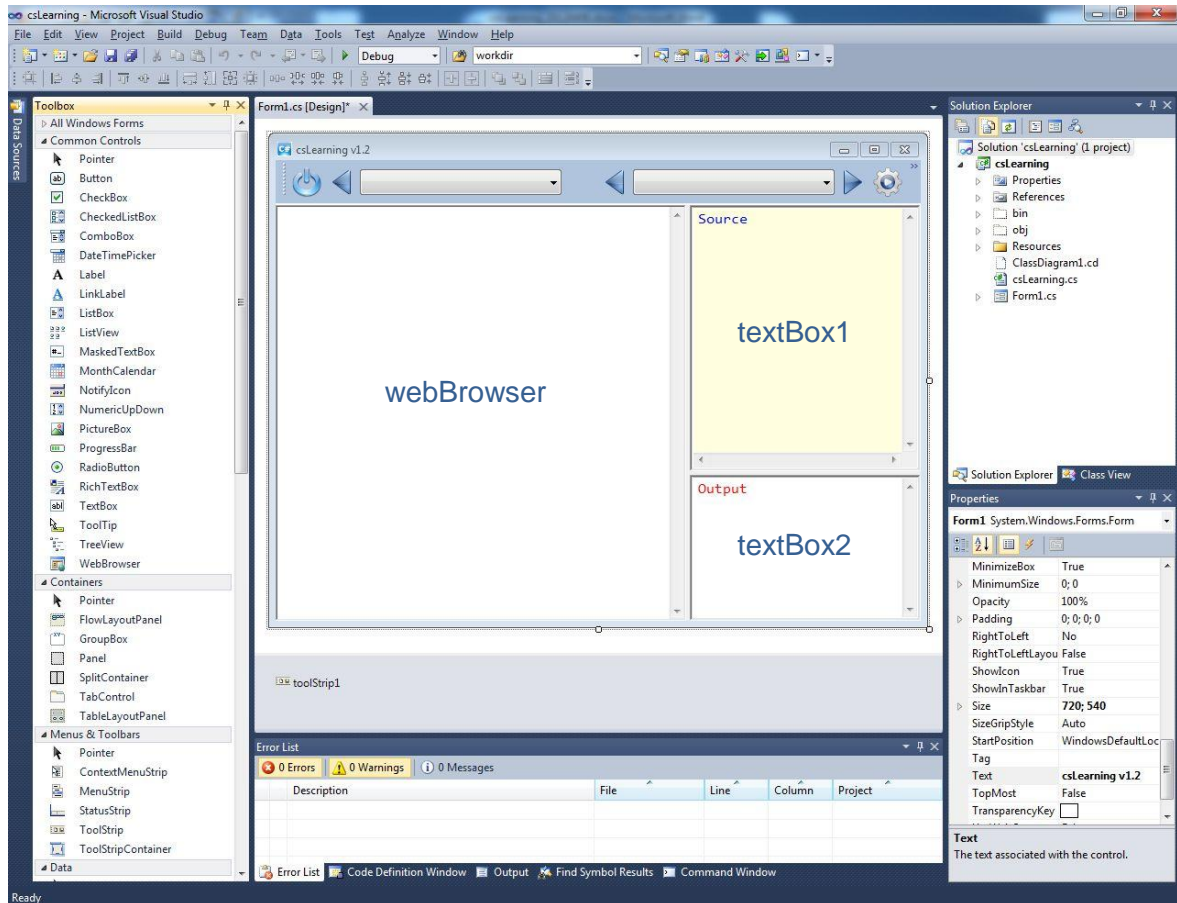
Ikon	Hozzárendelt feladat
	<i>Kilépés</i> a csLearning keretprogramból a munkaállományok törlésével.
	<i>Előző</i> lecke betöltése
Lecke	 Tartalom, lecke kiválasztása listából
	<i>Következő</i> lecke betöltése
	<i>Névjegy</i> , információ a keretprogram verziójáról.
[3,1]	<i>Kurzor sor és oszlop pozíció</i> . Csak a forráskód szerkesztésekor jelenik meg.
	<i>Forráskód mentése</i> .
	<i>Előző</i> mintaprogram betöltése
Forrás	 Forráskódok listája
	<i>Következő</i> mintaprogram betöltése
	<p><b>Forráskód fordítása és futtatása.</b> <b>F5</b></p> <ul style="list-style-type: none"> <li>A vezérlőre kattintással a keretprogram lementi a <code>textBox1</code> tartalmát (forráskódot) a TMP könyvtárba, majd meghívja <code>csc.exe</code> fordítót.</li> <li>Hiba esetén a fordítási hibaüzenet a jobb alsó panelen jelenik meg. Sikeres fordítás esetén pedig elindul a C# program.</li> <li>Ha a konzolalkalmazás nem kér be adatokat, akkor az elsődleges output szintén a jobb alsó panelen jelenik meg. Egyéb esetben önálló ablakban fut a programunk.</li> </ul>

**1. táblázat: Az eszköztár vezérlői**

A jobb oldali felső panelra kattintáskor a szövegdobozban megjelenik a szövegkurzor, és ekkor szabadon szerkeszthetővé válik a forráskód, így kipróbálhatunk különböző utasításokat és megoldásokat. Az eredmény megtekintéséhez az eszköztár jobb szélső „Forráskód fordítása és futtatása” vezérlőre kell klikkelnünk, vagy az **F5** billentyűt kell lenyomnunk.

### 3. A keretprogram szerkezete

A keretprogram maga is egy Windows alkalmazás, amely a szükséges konténer és megjelenítő komponenseken kívül még néhány vezérlőt tartalmaz, melyek mögött megbúvó eseményvezérelt felhasználói logika tervezése és leprogramozása jelentett komolyabb kihívást.



3. ábra: A csLearning fejlesztés

Az alkalmazás Form1 ablaka két splitContainert tartalmaz, amelyekbe egy webBrowser és két textBox épül be. A kezeléshez használt vezérlők pedig egy toolStrip komponens Button, ComboBox és Label vezérlői. Az egér és billentyűzet eseményeket a 2. táblázatban összefoglalt metódusok kezelik le. A funkcionálisan önálló részfeladatok illetve a több metódusban is használt utasítássorozatok önálló eljárásokba kerültek.

Metódusok - Eseménykezelés	
public Form1()	
	Form1 inicializálása, az ideiglenes állományok elérési útjának, az ideiglenes forrásfájl nevének, a csc.exe elérési útjának beállítása, a lekecímek feltöltése.
private void toolStripButton1_Click(object sender, EventArgs e)	
	Környezeti változó, munkaállományok törlése
private void leckeinfo(string path)	
	Lekecímek begyűjtése
private void toolStripComboBox1_SelectedIndexChanged(object sender, EventArgs e)	
	Lecke betöltés
private void toolStripButton3_Click(object sender, EventArgs e)	
	Előző lecke
private void toolStripButton4_Click(object sender, EventArgs e)	
	Következő lecke
private void forrasinfo(string path)	
	Forráskód nevek begyűjtése
private void forraskod(string forrasnev)	
	Forráskód betöltés
private void toolStripComboBox2_SelectedIndexChanged(object sender, EventArgs e)	
	Választott forráskód betöltése
private void toolStripButton5_Click(object sender, EventArgs e)	
	Előző forráskód
private void toolStripButton6_Click(object sender, EventArgs e)	
	Következő forráskód
private void toolStripButton7_Click(object sender, EventArgs e)	
	Forráskód mentése
private string atkodolo(string str, Encoding encodingSrc)	
	Átkódoló a standard outputhoz
private void toolStripButton2_Click(object sender, EventArgs e)	
	Fordítás és Futtatás
private string cursorpos(TextBox tbMain)	
	textBox1 sorszám, oszlopszám kiírás



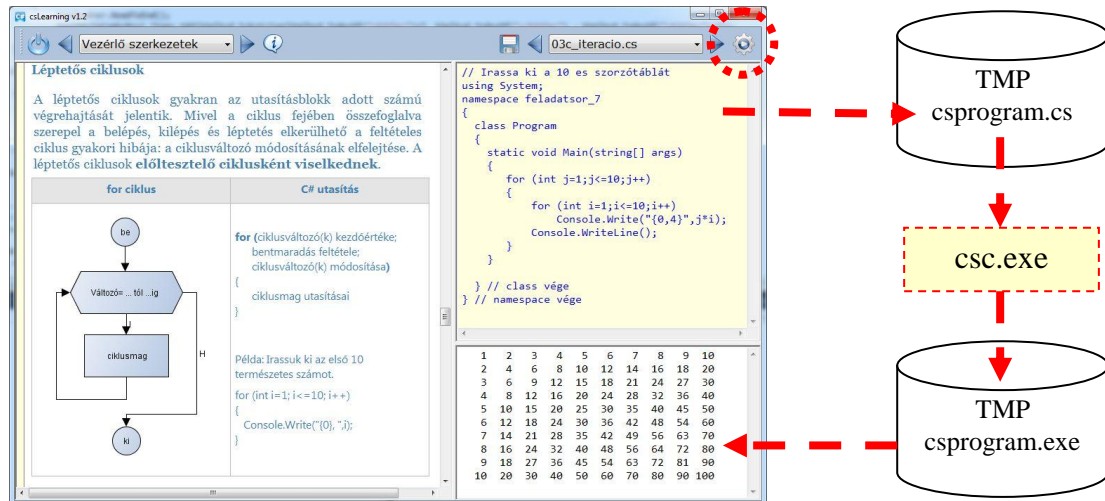
Metódusok - Eseménykezelés
private void textBox1_KeyPress(object sender, KeyPressEventArgs e)
textBox1 események a sor- és oszlopszámhoz
private void textBox1_KeyDown(object sender, KeyEventArgs e)
textBox1 billentyűlenyomás esemény lekezelése
private void textBox1_TextChanged(object sender, EventArgs e)
textBox1 szövegcsere esemény lekezelése
private void textBox1_Enter(object sender, EventArgs e)
textBox1 belépés esemény lekezelése
private void textBox1_Click(object sender, EventArgs e)
textBox1 egérekattintás esemény lekezelése
private void textBox1_KeyUp(object sender, KeyEventArgs e)
textBox1 billentyű elengedés esemény lekezelése
private void textBox1_MouseUp(object sender, MouseEventArgs e)
textBox1 egérgomb felengedése esemény lekezelése
private void textBox1_Layout(object sender, LayoutEventArgs e)
textBox1 layout esemény lekezelése
private void textBox1_Leave(object sender, EventArgs e)
textBox1 elhagyása esemény lekezelése
private void textBox2_KeyDown(object sender, KeyEventArgs e)
textBox2 billentyűlenyomás esemény lekezelése
private void toolStripButton8_Click(object sender, EventArgs e)
Névjegy, információ a keretprogram verziójáról

**2. táblázat: A keretprogramban használt metódusok**

*A környezeti változók a fordítóprogram és a forrásprogram elérési útjának meghatározásban játszanak szerepet.* A PATH változóhoz célszerű hozzákapcsolni a .NET keretrendszer legújabb csc.exe fordítójának elérési útját a saját gépen, míg a TMP változó azt a munkakönyvtárat azonosítja, ahová a keretprogram a forráskód fordítása és futtatása esemény bekövetkezésekor a `textBox1` tartalmát ideiglenesen lementi. Itt a `tmpfile` programváltozóban beállított nevet használja, ami jelen esetben a **csprogram** állománynevet jelenti.

A keretprogram process-ként indítja el **csc.exe** fordítót, ami az imént tárgyalt TMP könyvtárban lefordítja a forráskódot. A fordító standard outputon megjelenő üzenetei a

textBox2 komponensre vannak átirányítva, tehát fordítási hiba esetén felhasználhatók a textBox1 forráskódjának javításához. Ha fordítási hiba nem keletkezik, akkor a következő lépésben a *csprogram.exe* futtatása történik egy önálló process-ben. A „program a programban” technika tehát kihasználja a *C# process-kezelési szolgáltatásait*, amihez külső munkaállományokat vesz igénybe.



4. ábra: A forráskódok fordítása és futtatása

A fejlesztés közben a mintaprogramok írásakor jelentkezett egy átmentési probléma abban az esetben, amikor a mintaprogramok egy külső szöveges állományra vagy képre hivatkoztak, akkor az abszolút elérési út használatával a relokálhatóság indokolatlanul korlátozódott. Ennek feloldására vezettem be a *cslhome* nevű környezeti változót, ami mindig az *aktuális lecke elérési útjára mutat*, következésképpen a mintaprogramban a külső adat- és képállományokra hivatkozásnál felhasználva feloldódik az áthelyezhetőségi dilemma, és áttekinthető módon a mintaprogramot is tartalmazó lecke könyvtárában helyezhetjük el a feldolgozni kívánt külső adatokat.

```
// Példa a cslhome környezeti változó használatára

string home = Environment.GetEnvironmentVariable("cslhome");

StreamReader sr = new StreamReader(home+"Teszt.txt");
string sor;

while ((sor = sr.ReadLine()) != null)
{
    Console.WriteLine(sor);
}
```

Környezeti változó	Beállítása
PATH	<ul style="list-style-type: none"> <li>A keretprogram a C# compiler elérési útját először a PATH környezeti változóban keresi, ha itt nem találja, akkor a</li> <li>"C:\Windows\Microsoft.NET\Framework\v?.*" könyvtárak közül a legnagyobb verziószámút választja, és figyelmeztet a PATH környezeti változó kiegészítésére.</li> </ul>
TMP	A forráskód valamint a fordító munkaállományainak elérési útja.
cslhome	<p>Az állományelérés megkönnyítése érdekében a keretprogram a cslhome környezeti változóban mindig az aktuális lecke elérési útját tárolja.</p> <p>Alkalmazása a forráskódban:</p> <p>filePath = <i>Environment.GetEnvironmentVariable("cslhome")</i>+"pictures\\pic.jpg"</p>

**3. táblázat: A keretprogram által használt környezeti változók**

#### 4. Konzolalkalmazás készítésének folyamata

A konzolalkalmazásként előállított program parancssori környezetben működik, végrehajtása esetén automatikusan használja a System névtér Console osztályát, amely a beolvasás (standard input, billentyűzet), kiírás (standard output, képernyő) és hibairási (standard error, képernyő) műveleteket nyújtja. Ezek a Console osztály In, Out és Error tulajdonságaiban vannak hozzárendelve a be- és kiviteli eszközökhöz. Az In egy TextReader, míg az Out és az Error TextWriter típus lesz. (A TextReader, TextWriter a karakteres adatfolyam osztályai.)

A konzolalkalmazások forráskódja a csLearning keretprogram textBox1 komponensében szerkeszthető, illetve a sikeres fordítás és futtatás után leckénként eltárolható, és a későbbiekben az egyéni tanuláshoz vagy a tanórai szemléltetéshez is felhasználható. Ha a diákok a C# programszerkezetét már megismerték, akkor a tanult utasításokat közvetlenül a csLearning beviteli panelén kipróbálhatják, miközben a keretprogram baloldalán éppen az új leckét pásztázzák. Tehát nem szükséges átváltani a Visual Studio ablakára csak azért, hogy a leckében bemutatott forráskóddal kísérletezzenek.

csLearning v1.2

Programozási tételek

05\_maximum.cs

5. A program logikája:

A vizsgálat elején az első elemről feltételezzük, hogy ő a legnagyobb, ezért a sorszámát a maxidx nevű változóba írjuk. A második elemtől kezdve sorban összehasonlítjuk az értékeket a maxidx sorszámu elemmel, és ha nagyobb találmunk, akkor megjegyezzük annak sorszámát a maxidx változóban. Így az algoritmus végén a maxidx változó tartalmazza a legnagyobb elem sorszámát.

Maximumválasztás

Start

tomb, maxidx változók deklarálása

for (int i=1; i<tomb.Length; i++)

tomb[i]>tomb[maxidx]

maxidx=i

return maxidx

Visszatérés a hívó programba

C# forráskód

```

using System;
namespace névterem
{
    class prog...
    {
        //-----
        static ...
        {
            int maxidx;

            for (int i=1; i<tomb.Length; i++)
                if (tomb[i]>tomb[maxidx]) maxidx=i;

            return maxidx;
        }
    }
}

```

CTRL +  
CTRL -  
CTRL 0

3. elem a legnagyobb értéke: 777

5. ábra: Panel és betűméret beállítás

A tanórai minta és rávezető programok ugyancsak elkészíthetők a `textBox1` komponens segítségével. Ebbe az esetben a beviteli panel méretét érdemes megváltoztatni és a betűméretet úgy beállítani, hogy a tanterem hátsó soraiból is olvasható legyen a forráskód.

[3,1]

### 1. Forráskód bevitele a textbox1 panelbe

```

/*
Írjon programot, mely egy rendezetlen számsort a buborékos rendezés módszerével
növekvő sorrendbe rendez!
*/

```

```

using System;
namespace névterem          //----- névtér kezdete
{
    class prog...           //----- osztály kezdete
    {

```

```

//----- Buborékos rendezés
static void rend_bub(ref int[] tomb)
{
    int i=tomb.Length-1;
    int x;

    bool rendezett=false;
    while (i>0 && !rendezett)
    {
        rendezett=true;
        for (int j=0;j<i;j++)
            if (tomb[j]>tomb[j+1])
            {
                x=tomb[j];
                tomb[j]=tomb[j+1];
                tomb[j+1]=x;
                rendezett=false;
            }
        i--;
    }
}

//----- Tömb kiírása
static void kiir(ref int[] tomb)
{
    for (int i=0;i<tomb.Length;i++)
        Console.Write("{0}, ",tomb[i]);

    Console.WriteLine("\n");
}

//===== FŐPROGRAM
static void Main()
{
    int[] szamok = {20, 15, 345, 36, 777, 152, 353, 34, 534};

    kiir(ref szamok);

    rend_bub(ref szamok);

    kiir(ref szamok);
}

} // ----- osztály vége
} // ----- névtér vége

```



## 2. Forráskód fordítása és futtatása

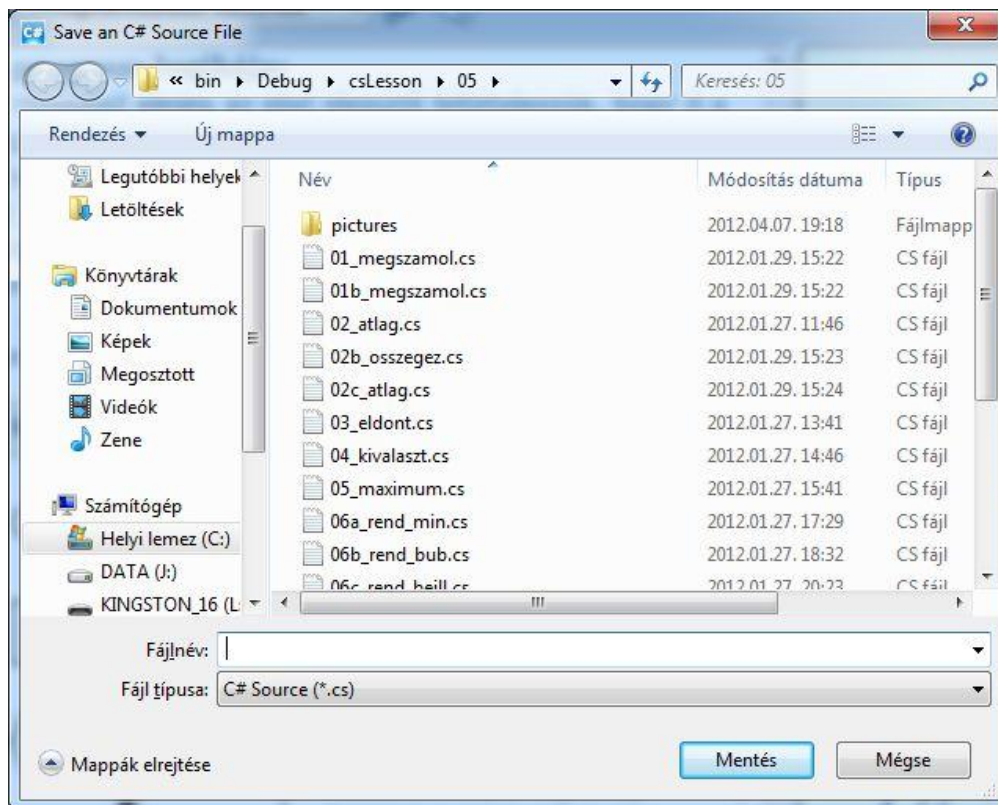
- Fordító hibaüzenete piros színnel jelenik meg a `textBox2` panelen:  
`csprogram.cs(10,11): error CS0103: The name 'alfa' does not exist in the current context`
- A hibaüzenet felépítése:
  - a forráskódot tartalmazó állomány neve, a hiba sor és oszloppozíciója
  - a hiba kódja és leírása
- A fordítási hiba esetén természetesen nincs mit futtatni, ezért a hiba **javítását kell elvégezni** a hibaüzenet alapján a `textBox1` panelen.

- **Sikeres fordítás után elindul a konzolalkalmazás:**
  - Ha csak a **standard output-ra ír** akkor a program futási eredménye a `textBox2` panelen jelenik meg.
  - Ha a **standard input-ról kér be adatokat a program**, akkor egy önálló parancsoros ablak jelenik meg, amelyen keresztül történik az adatbevitel.



## 3. Forráskód mentése

- A már működő konzolalkalmazás forráskódja, az aktuális lecke könyvtárába elmenthető, és a tanítási-tanulási folyamatban felhasználható. A forráskód nevének első két karaktere a megjelenítés logikai sorrendjét szabályozza.

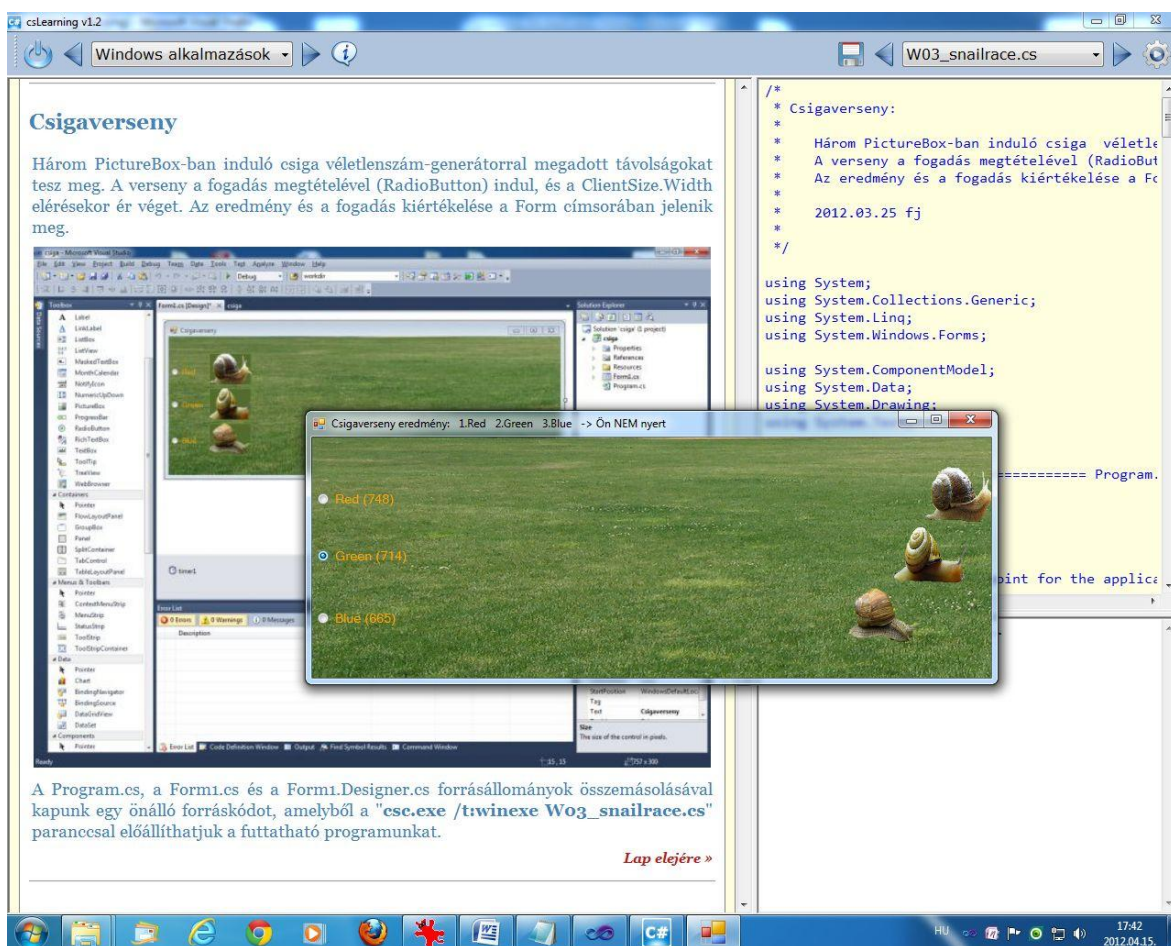




## 5. Windows alkalmazás kezelése

A feladatkitűzés alapvetően a programozás tanulás bevezető szakaszára és a konzolalkalmazások készítésére fókuszált, de a keretprogram lehetőséget biztosít egyszerűbb Windows alkalmazások teszteléséhez is, mintegy felkeltve az érdeklődést a grafikus környezetben történő programfejlesztés előnyeire.

A Visual Studio fejlesztő rendszere a forráskódokat több önálló fájlban tárolja. A Program.cs, a Form1.cs és a Form1.Designer.cs forrásállományok összemásolásával azonban egy önálló forráskódot kaphatunk, amelyből a `"csc.exe /t:winexe csprogram.cs"` paranccsal előállíthatjuk a futtatható Windows alkalmazásunkat. Amennyiben a csLearning keretprogram a forráskódban "Application.Run" szöveget talál, akkor automatikusan a `/t:winexe` argumentummal fordítja le a mintaprogramot, majd a MSIL kódban elkészült Windows alkalmazást egy önálló ablakban indítja el.



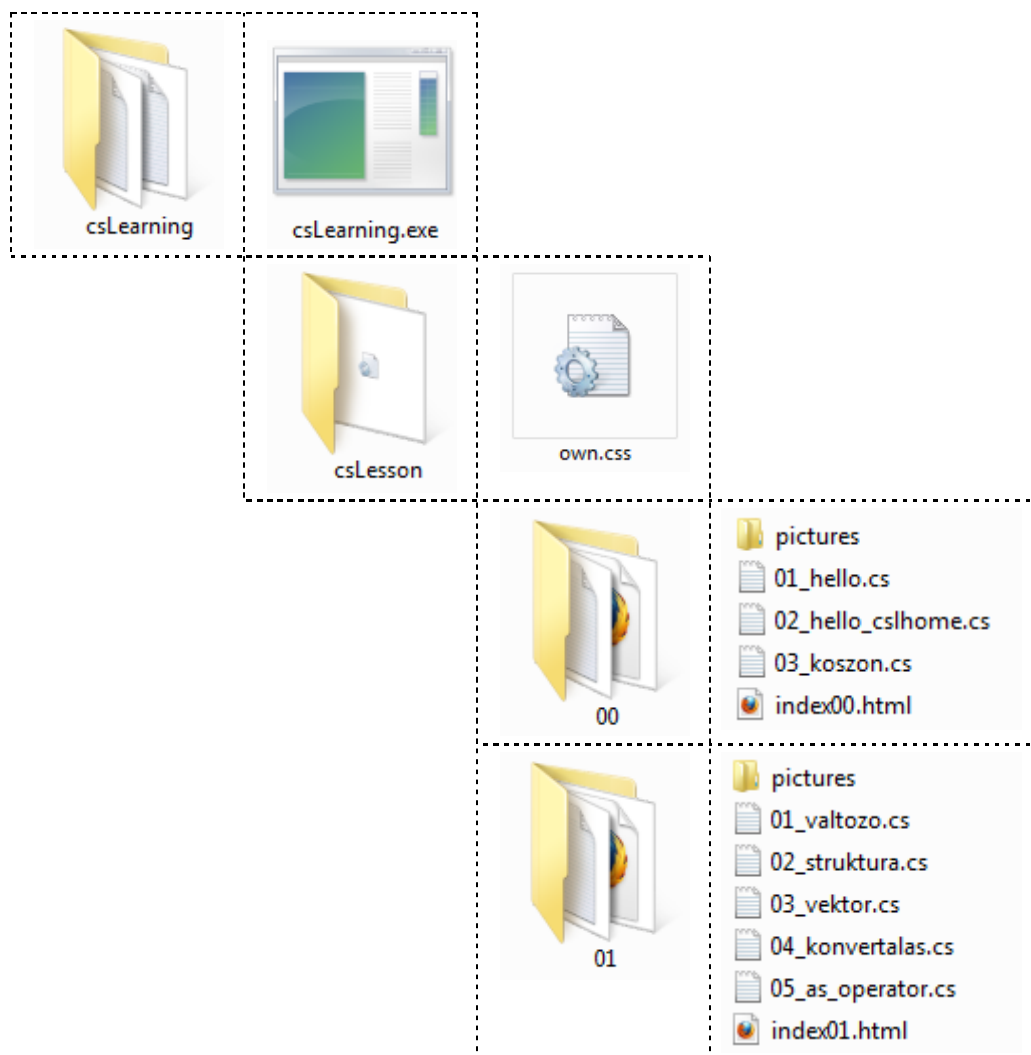
6. ábra: Windows alkalmazás kezelése a csLearning keretprogrammal

Ezzel a módszerrel lehetőség nyílik a valamelyest összetettebb grafikus kezelő felületű alkalmazások struktúrájának és komponenseinek a megismerésére, de természetesen a haladó szintű programozási feladatok kódolása már a Visual Studio keretrendszerben

történik. Az elmúlt féléves tapasztalatok azt bizonyítják, hogy ez az átmeneti megoldás felkelti diákok érdeklődését az OOP újabb lehetőségei iránt, amit egyszerűbb ötletes programok tervezésével és írásával tovább élénkíthetünk.

## 6. A leckék felépítése

A tananyag leckéi a keretprogramot tartalmazó könyvtár *csLesson alkönyvtárában* vannak elhelyezve a megjelenési sorrendjüket meghatározó sorszámmal azonosított alkönyvtárban. Tehát a könyvtár két számjegyes neve (99) a lecke sorszáma, a benne található `index99.html` file pedig maga a lecke lesz. A fejezet címeként a HTML kód `<title>` utasításában megadott szöveget jeleníti majd meg a keretprogram. A leckék és a forráskódok átírhatók és folyamatosan bővíthetők. A tanórán elkészített mintaprogramok közvetlenül a tárgyalt leckéhez csatolhatók, így a továbbiakban már a lecke kiválasztásakor megjelennek a forráskódok listájában.



7. ábra: A csLearning leckék felépítése



A `pictures` alkönyvtárban tárolhatjuk a HTML kódban használt külső kép file-okat, illetve a forráskódokban használni kívánt külső adat vagy kép file-ok is ide kerülhetnek. A forráskódok első két karaktere szintén szám, amivel a keretprogramban történő megjelenítési sorrendjüket lehet befolyásolni. A fentiek figyelembevételével újabb leckék kapcsolhatók a már meglévő alapsomag tananyagaihoz, amelyeket a keretprogram automatikusan fel fog ismerni.

A HTML dokumentumok az *own.css* állományban tárolt stílus beállításokat használják. A címsort követően egy főmenüt tartalmaznak a lecke egyes témáinak gyors eléréséhez. A HTML oldalon belül egy-egy témaegység után mindig megjelenik a „lap elejére” ugrás opció, ezzel biztosítva a gyors pásztázás lehetőségét, de természetesen a leckék folyamatosan is böngészhetők. Gyakran előfordul, hogy egy korábban tanult részinformációt szeretnénk visszakeresni valamelyik leckében, ilyenkor vehetjük a legnagyobb hasznát ez utóbbi megoldásnak. A HTML leckék végén a forráshivatkozásokat követően a lecke készítőjének neve (a honlapjára vagy e-mail címére hivatkozással), valamint a legutolsó frissítés dátuma jelenik meg, így a csoportmunkában történő tananyagfejlesztés is követhetővé válik.

A leckékhez tartozó forráskódok listáját az eszköztár jobb oldali `ComboBox`-a kezeli, melynek tartalma a leckék közötti váltáskor, illetve a forráskód mentésekor folyamatosan frissül, ezáltal a forráskódok elérése leegyszerűsödik, és nem vonja el a figyelmet a tanulástól.

## 7. Az alapsomagban kidolgozott témakörök

Az alapsomag a 10. évfolyamos számítógép programozás és gyakorlat tantárgyakhoz kapcsolódó témaköröket tárgyalja, amit több mint 80 feladtból álló feladatgyűjtemény egészít ki. A kezdő szinthez tartozó témaköröket az alábbi táblázat foglalja össze.

Témakör	Tartalom
C# alapismeretek	A .NET környezet jellemzői A program szerkezete Parancssori környezet használata Fejlesztőkörnyezet használata Az oktatóprogram Ellenőrző kérdések
Adattípusok, változók	Előre definiált típusok Összetett adattípusok

Témakör	Tartalom
	Típuskonverzió Megjelenítés Ellenőrző kérdések
Kifejezések, műveletek	Operátorok Értékadás Operátor precedencia Összetett logikai kifejezések Ellenőrző kérdések
Vezérlő szerkezetek	Szekvencia Szelekció - elágazás Iteráció - Ciklus Egyéb ugró utasítások Kivételkezelés Ellenőrző kérdések
Függvények	Felépítés Paraméterátadás Függvényhívás Ellenőrző kérdések
Programozási tételek	Megszámlálás Sorozatszámítás Eldöntés Kiválasztás Maximum - minimum választás Rendezés: Minimumválasztásos Buborékos Beillesztéses Keresés: Lineáris Logaritmikus Kiválogatás és másolás Szétválogatás Közös rész meghatározása, metszet Egyesítés, unió Összeválogatás Ellenőrző kérdések
Osztályok	Deklarálás Konstruktor Destruktor Tulajdonság Öröklődés Ellenőrző kérdések

Témakör	Tartalom
Input-Output	Standard input-output Fájl input-output Ellenőrző kérdések
Windows alkalmazások	Új Windows alkalmazás Ugráló gomb játék Képernyővédő Csigaverseny Videoleckék Ellenőrző kérdések
DataGridView bemutató	Demo alkalmazásról röviden Sorkezelés Oszlopkezelés Kijelölési módok Egyéb tulajdonságok Ellenőrző kérdések
Feladatgyűjtemény	A-N feladatsor (14*6=84 feladat) Z feladatsor érettségi példákkal
Felhasznált C# források	Digitális tan- és segédanyagok URL-je

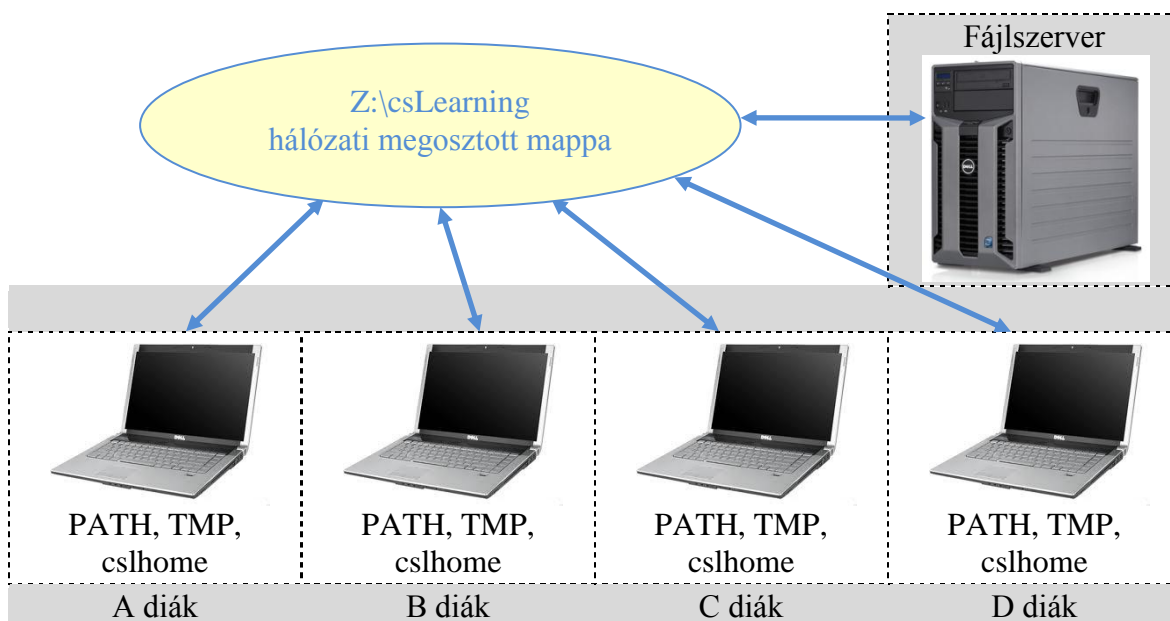
**4. táblázat: Alapcsomag témakörei és tartalma**

Alapvetően az adattípusok, a belső és külső értékadás, a vezérlőszervezetek és a programozási tételek ismerte alkotja a tananyag gerincét. A továbbhaladás minimális feltétele ezek alkalmazásának képessége, a maximum kritérium pedig mindezek alkalmazása a függvényhívásokkal és a szöveges állománykezeléssel kiegészítve. Az osztályok használta és a Windows alkalmazások készítése már a következő tanév előkészítése lehet.

A *programozási tételek kiemelt témakörként szerepel a tananyagban*, hiszen az érettségi feladatok megoldásánál elengedhetetlen az itt tárgyalt 12 típusalgoritmus ismerte. Éppen ezért az ide tartozó HTML lecke részletesen tárgyalja a programozási tételek elméleti sajátosságait, melyet a kidolgozott mintaprogramok illusztrálnak. Hasonlóan fontos szerepe van a *feladatgyűjteményben közre adott feladatsoroknak*, amelyek megoldása hozzájárulhat az algoritmikus gondolkodás képességének kialakulásához.

## 8. A csLearning használata hálózati környezetben

A csLearning keretprogram működésének feltétele a .NET keretrendszer előzetes telepítése a futtató számítógépen, így az iskolai hálózatos környezetben a **kliens oldali számítógépeken** rendelkezésre kell állnia a futtató környezetnek. Az iskolai számítógép-hálózatokban ma már természetes módon, fájlszervereken kialakított hálózati meghajtókon dolgoznak a diákok. Hálózati környezetben a csLearning oktatóanyag elhelyezhető egy megosztott közös hálózati mappában, amelyből a diákok elindíthatják a csLearnig.exe programot. Mivel a kliens oldalon a .NET telepítve van így nincs akadálya a kód futtatásának, a leckék és a mintaprogramok betöltésének.



8. ábra: A csLearning használata hálózati környezetben

A csLearning keretprogram azon a számítógépen hozza létre a munkaállományait, amelyiken éppen futtatják, így az oktatóanyag közvetlenül a fájlszerről is betölthető. Megfelelő jogosultságok birtokában pedig kialakítható egy csoportos munkavégzésre alkalmas hálózati környezet.

## 9. Fejlesztési lehetőségek

A C# oktatóanyag néhány hónapos használata során kapott visszajelzések és összegyűlt tapasztalatok arra engednek következtetni, hogy az alapötletben és a jelenlegi verzióban is vannak még tartalékok. A konzolalkalmazások készítéséhez már az alapsomag is elég sok segítséget ad, de újabb leckékben fel lehetne dolgozni az elmúlt évek érettségi feladatait, magyarázatokkal, rávezető feladatokkal. Érdekes kihívásokat tartogat a System.Net névtér

megismerése is, ám igazából a Windows alkalmazások készítésének oktatóanyagában érdemes gondolkodni.

Az esetleges következő verzióban a Windows alkalmazások vezérlőinek részletesebb bemutatása, a futásidejű komponensek kezelésének, illetve az adatállományok feldolgozásának megismerése lehetne a cél, amelyek segítségével már komfortosabb grafikus felhasználói felülettel működő programok készülhetnének. Hasonlóan érdekes munkát ígér egy a tanulók programozás tudását mérő diagnosztikai alkalmazás és adatbázis kialakítása is. Viszont innentől már kooperatív tananyag fejlesztési módszereket kellene bevetni. Ezt a módszert Csatlós László hallgatótársunknak köszönhetően ki tudtuk próbálni. Így készült el a DataGridView bemutató, ami szemléletesen ismerteti az említett vezérlő alkalmazási lehetőségeit.

Mindent egybe vetve a további fejlesztéseket a hatékonysági és minőségi szempontok figyelembevételével projekt munka keretében képelném el. A kooperatív munka közben kipattanó ötletek nyilvánvalóan újabb lehetőségeket nyithatnak meg az oktatóanyag fejlesztésében.

## 10. Összegzés

A C# oktatóanyag elkészítésének gondolata 2011 decemberében fogalmazódott meg. Alapvetően egy olyan keretprogram ötlete vetődött fel, amellyel ***mind az elméleti mind a gyakorlati programozási ismeretek egy felületen kezelhetők***. Így kézzelfogható megoldásnak tűnt egy osztott képernyős alkalmazásablak, melynek egyik oldalán a leckék jelennek meg, míg a másik oldalán a ***forráskódok szerkeszthetők és futtathatók***. Ehhez viszont a keretprogramból kell meghívni egy külső alkalmazást, jelen esetben a C# compilert.

A keretprogram készítése során jó néhány nyomtatott és elektronikus forrást átolvastam, melyek segítségével igyekeztem a C# lehetőségeit megismerni. A csLearning alkalmazás készítése közben a legtöbb gondot a process indítása, illetve a textBox karakterkódolási problémái okozták, de néhány hajnalig a számítógép előtt töltött éjszaka meghozta a megoldást. Az oktatóanyag optimális tárolási struktúrája viszonylag hamar tisztázódott, így a leckék feltöltését már január vége felé el tudtam kezdeni.

A HTML tananyag az elméleti ismereteket folyamatábrákkal, kép és videó anyagokkal illusztrálva mutatja be. Minden leckéhez készültek mintaprogramok, melyekkel a diákok kísérletezhetnek, sőt át is írhatják őket. Az alapsomag a 10. évfolyamos számítógép

programozás és gyakorlat ismeretanyagát tartalmazza, és kiemelten kezeli az adattípusok, a belső és külső értékadás, a vezérlőszervezetek és a programozási tételek témaköröket, továbbá megnyitja a Windows alkalmazások készítéséhez vezető utat is. A csatolt feladatgyűjteményben pillanatnyilag 14 feladatsor található egyenként 6 programozási feladattal, továbbá a „Z” feladatsorban már gyűlnek az érettségi példák, tehát a tervezett alapozási munkához minden rendelkezésre áll.

A csLearning keretprogram és oktatóanyag iskolai tesztelése a 2011/2012 tanév második félévében kezdődött el. A diákoknak tetszett az az elképzelés, ami szerint a programozás tantárgyhoz kapcsolódó ismeretek egy helyen megtalálhatók és kipróbálhatók, de természetesen az oktatási folyamat irányításában és ellenőrzésében a tanári elkötelezettség továbbra is kulcsfontosságú.

Az C# oktatóanyag készítése közben kapott *bíztatást és segítséget hálásan köszönöm* a diákjaimnak, évfolyamtársaimnak és Lipovits Ágnes tanárnőnek, aki újabb és újabb ötletekkel gazdagította a keretprogram által nyújtható oktatási lehetőségeket.

Tamási, 2012.05.03

Füredi János

## 11. Felhasznált irodalom

- 1) **NYÉKYNÉ GAIZLER JUDIT: Programozási nyelvek**, Budapest, Kiskapu Kft. 2003.
- 2) **ILLÉS ZOLTÁN: Programozás C# nyelven**, Budapest, JOS Bt. 2008.  
(pdf - 2,3 MB), <http://compalg.inf.elte.hu/~tony/Informatikai-Konyvtar/09-Programozas%20C-sharp%20nyelven/Programozas-Csharp-nyelven-Konyv.pdf>
- 3) **CZIGLÉCZKY GÁBOR: Bevezetés a C# programozásba**,  
<ftp://ganziskola.hu/Szakaly/C%23%20jegyzet%20CP.pdf>
- 4) **DR. KOVÁCS EMÓD - HERNYÁK ZOLTÁN - RADVÁNYI TIBOR - KIRÁLY ROLAND: A C# programozási nyelv a felsőoktatásban Programozás tankönyv**,  
(letöltés: pdf - 3,9 MB), <http://csharp.tk.ektf.hu/download/csharp-ekf.pdf>
- 5) **SIPOS MARIANNA: Programozás élesben**, Budapest, InfoKit, 2004.
- 6) **REITER ISTVÁN: C# jegyzet**, (pdf - 2,9 MB),  
<http://devportal.hu/Fajlok/Download.ashx?shareid=1&path=Konyvek%5ccsharp+jegyzet%5ccsharp.pdf>
- 7) **Standard ECMA-334 C# Language Specification**,  
<http://www.ecma-international.org/publications/standards/Ecma-334.htm>

## Ajánlott linkek

- 1) **A C# programozási nyelv (ELTE)**,  
<http://nyelvek.inf.elte.hu/leirasok/Csharp/index.php?chapter=5>
- 2) **Programozás bevezető C#-ban**,  
[http://www.berzsenyi.hu:8080/prog/View/szakkor/bdg\\_progbev/0809](http://www.berzsenyi.hu:8080/prog/View/szakkor/bdg_progbev/0809)
- 3) **MAJOR PÉTER: Fájlkézelés .Net-ben (BME)**,  
<ftp://ftp.inflab.bme.hu/pub/oktatas/Mobiltelefon%20programoz%E1s/Fajlkezel%E9s.pdf>
- 4) **Dot Net Perls (C#)**, <http://dotnetperls.com/>
- 5) **C# examples**, <http://www.csharp-examples.net/>
- 6) **C# Reference (Microsoft)**, <http://msdn.microsoft.com/en-us/library/618ayhy6.aspx>
- 7) **C# Programming Guide (Microsoft)**,  
<http://msdn.microsoft.com/en-us/library/67ef8sbd.aspx>

## Videoleckék

- 1) **You Tube C#**, [http://www.youtube.com/results?search\\_query=c%23](http://www.youtube.com/results?search_query=c%23)

## Letöltés

- 1) **Visual C# 2010 Express (Free tools to create .NET applications)**,  
<http://www.microsoft.com/visualstudio/en-us/products/2010-editions/visual-csharp-express>
- 2) **MonoDevelop (C# and .NET for Linux)**,  
[http://www.mono-project.com/Main\\_Page](http://www.mono-project.com/Main_Page)
- 3) **yEd Graph Editor**, [http://www.yworks.com/en/products\\_yed\\_about.html](http://www.yworks.com/en/products_yed_about.html)

## 12. Mellékletek

### 1. számú digitális melléklet: csLearning v1.2 C# oktatóanyag

