kodolosuli.hu: Interaktív, programozást tanító portál

BALLA TAMÁS, DR. KIRÁLY SÁNDOR

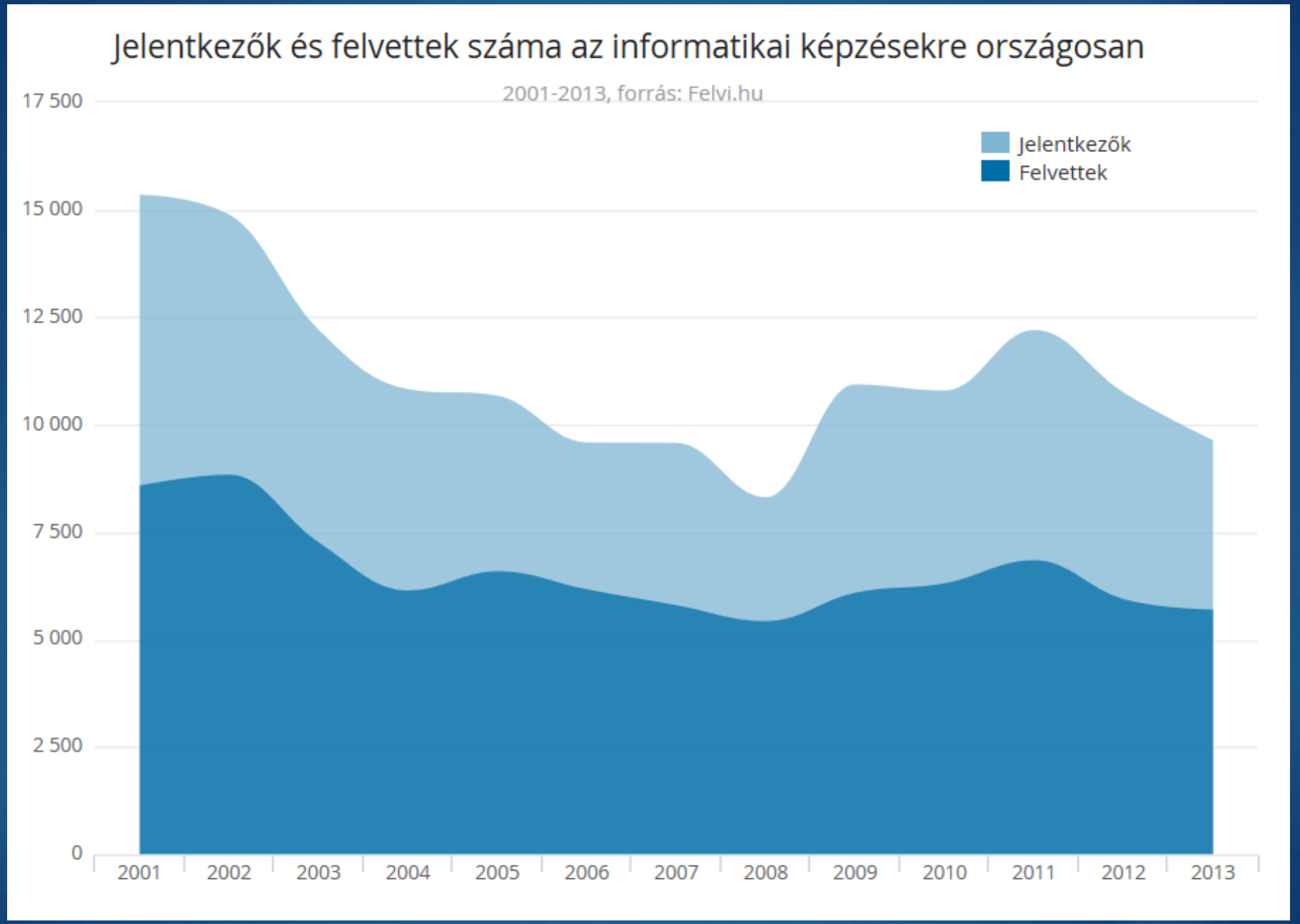
NETWORKSHOP 2017, SZEGED

A közoktatásban folyó informatika oktatásával kapcsolatos elvárások

- **Állami szereplő: „***Az informatikaoktatás célja a praktikus alkalmazói tudás, a készség és képességfejlesztés mellett a logikus, algoritmikus gondolkodás és a problémamegoldás tanítása.*” (Magyar Közlöny 2012. évi 66. szám: Nemzeti Alaptanterv 10813 (181.o)).
- **Gazdasági szereplő:**
  - **Jól képzett** munkaerő.
  - Az informatikai szféra munkaerőpiacára áramló **szakképzett fiatalok számának radikális növelése** (22 000 fő hiányzik Magyarországon).

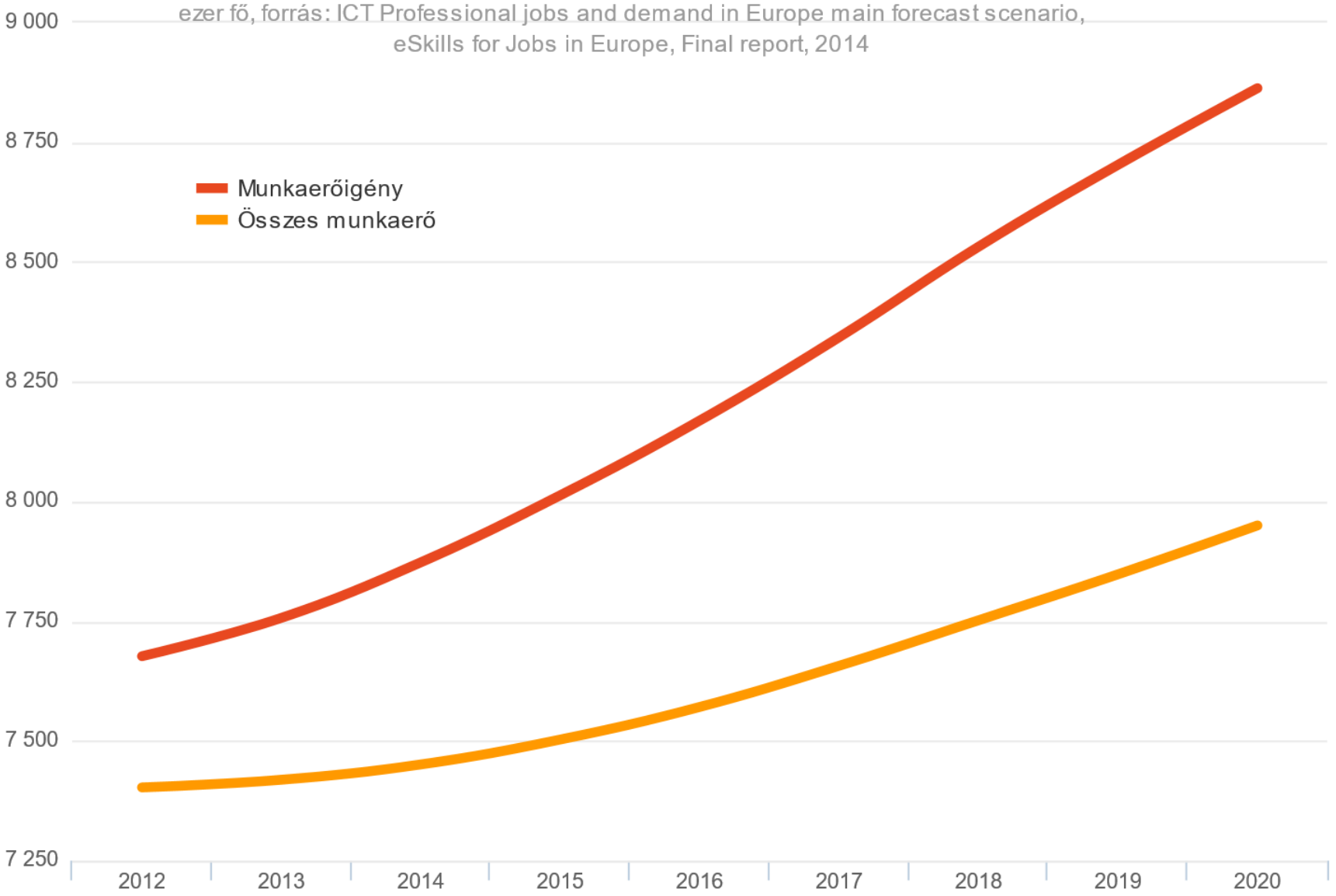
A közoktatásban folyó informatika oktatásával kapcsolatos elvárások

- **Szülő, gyermek:**
  - Piacképes, praktikus, a gyakorlatban hasznosítható tudás.
  - Sikeres felkészülés a közép, illetve emelt szintű érettségire.
  - Pályakép.
- **Felsőoktatási intézmények:**
  - A tanulók felkészítése az informatikai pályára.
  - A jelentkezők számának növekedése.
  - A motiváció és az elköteleződés kialakítása.



Forrás: http://index.hu/gazdasag/2016/03/04/informatikushiany_munkaeropiac_oktatas_informatika/

Az évtized végére nagyjából 900 ezer informatikus fog hiányozni az EU-ból



Forrás: http://index.hu/gazdasag/2016/03/04/informatikushiany_munkaeropiac_oktatas_informatika/

Az informatikaoktatás helyzete Magyarországon

- Az állami szabályozás az informatika műveltségi területek oktatására **alacsony óraszám**ot biztosít.
- A **digitális analfabéták száma** egyre **növekszik**.
- **Sok tanuló kapcsolatba sem kerül a programozással,** legfeljebb a praktikus alkalmazói tudást sajátítja el.

Megoldást kell találni a problémára, választ kell adni a kérdésekre.

Célok

1. Olyan **magyar nyelvű** portál fejlesztése, amely **ingyenesen hozzáférhető** mindenki számára.
2. A portál **interaktív legyen,** képes legyen a kurzusokat feldolgozó tanulókat aktívan bevonni a tanítási-tanulási folyamatba.
3. A kurzusok **felkészítsék a diákokat az emelt szintű informatika érettségi programozási feladatára.**

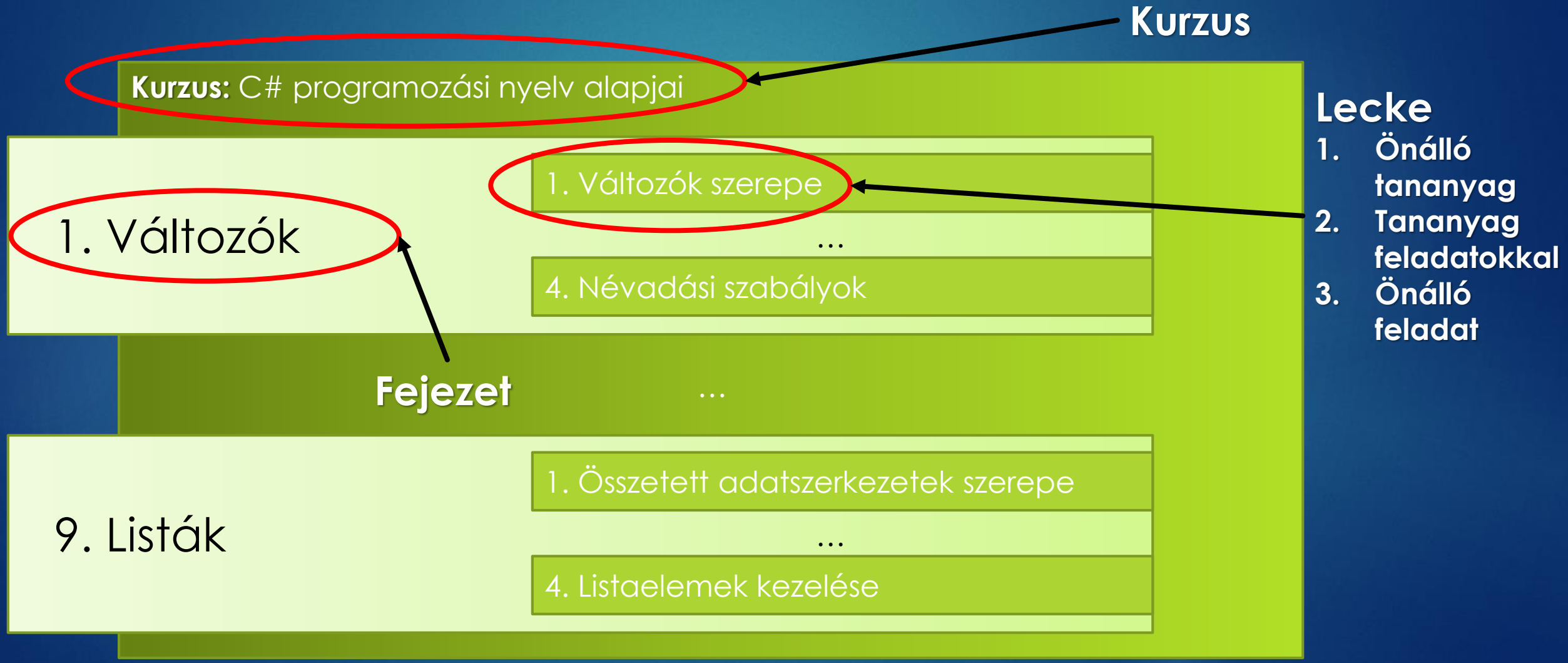
A fejlesztés bemutatása: kodolosuli.hu

- A portál **PHP5** nyelven, saját keretrendszerben került fejlesztésre. ➔ A keretrendszer, portál dinamikusan fejleszthető, bővíthető.
- Az adatok tárolása **MySQL** adatbázisban, 19 táblában történik. ➔ Teljes felhasználói aktivitás naplózása.
- A tananyagok kialakítása során készített játékokat **HTML5**, illetve **JavaScript** nyelvek felhasználásának segítségével készültek.

A portál bemutatása: kodolosuli.hu

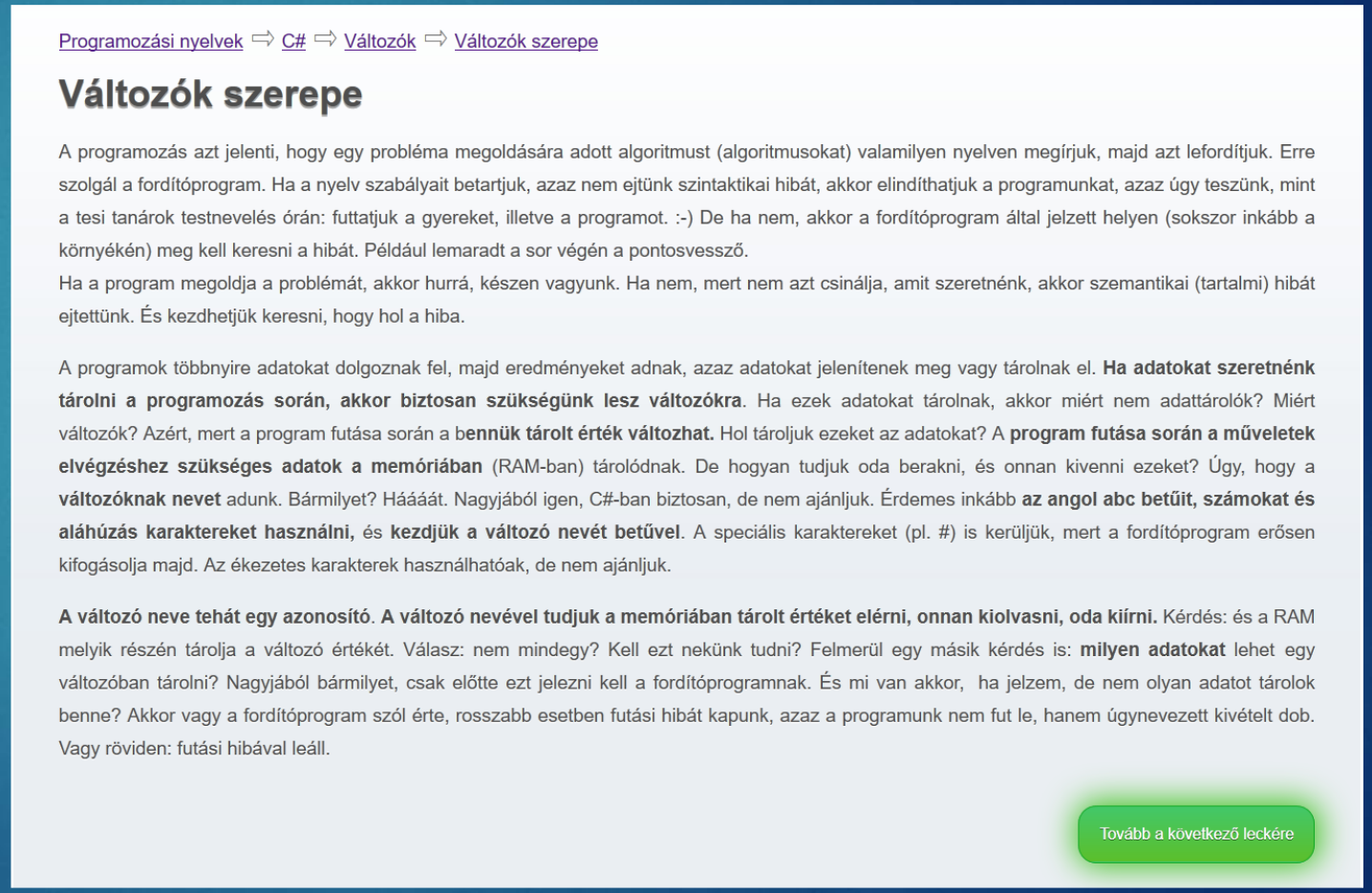
- A portál által nyújtott szolgáltatások regisztrációt és bejelentkezést követően érhetők el.
- **Három jogosultsági szint**et különít el a rendszer
  - **Nem bejelentkezett** felhasználó.
  - Bejelentkezett **tanuló** felhasználó.
  - Bejelentkezett adminisztrátor (**tanár**) felhasználó.

Kurzusok szerkezete



Önálló tananyag

- Csak **elméleti ismereteket** tartalmaz, programozási feladatot nem.
  - **Szöveg**es elemek.
  - **Audiovizuális** elemek.
  - **Játék**ok.



Programozási nyelvek ⇨ C# ⇨ Változók ⇨ Változók szerepe

Változók szerepe

A programozás azt jelenti, hogy egy probléma megoldására adott algoritmust (algoritmusokat) valamilyen nyelven megírjuk, majd azt lefordítjuk. Erre szolgál a fordítóprogram. Ha a nyelv szabályait betartjuk, azaz nem ejtünk szintaktikai hibát, akkor elindíthatjuk a programunkat, azaz úgy teszünk, mint a tesi tanárok testnevelés órán: futtatjuk a gyereket, illetve a programot. :-) De ha nem, akkor a fordítóprogram által jelzett helyen (sokszor inkább a környékén) meg kell keresni a hibát. Például lemaradt a sor végén a pontosvessző.
Ha a program megoldja a problémát, akkor hurrá, készen vagyunk. Ha nem, mert nem azt csinálja, amit szeretnénk, akkor szemantikai (tartalmi) hibát ejtettünk. És kezdhetjük keresni, hogy hol a hiba.

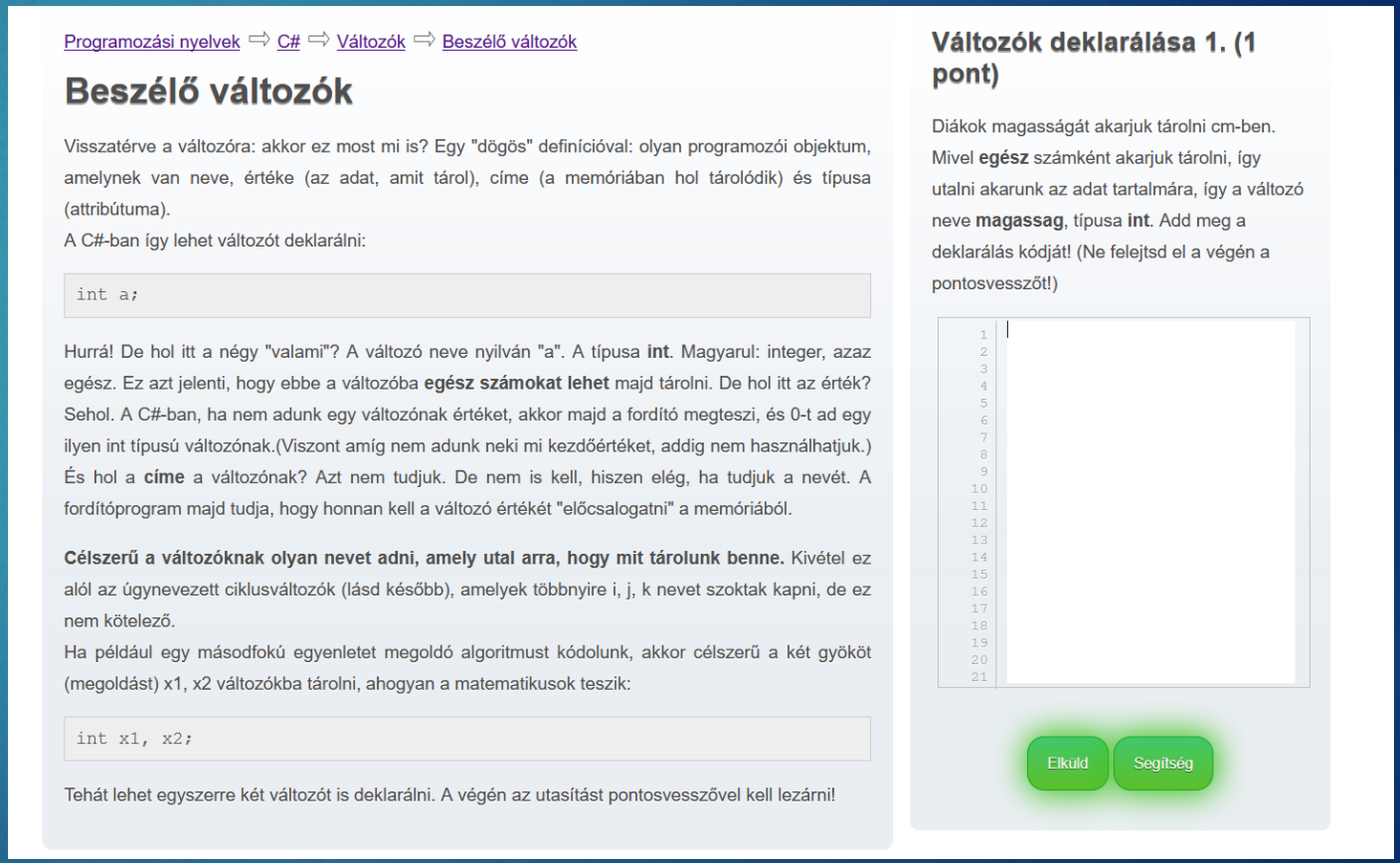
A programok többnyire adatokat dolgoznak fel, majd eredményeket adnak, azaz adatokat jelenítenek meg vagy tárolnak el. **Ha adatokat szeretnénk tárolni a programozás során, akkor biztosan szükségünk lesz változókra**. Ha ezek adatokat tárolnak, akkor miért nem adattárolók? Miért változók? Azért, mert a program futása során a b**ennük tárolt érték változhat.** Hol tároljuk ezeket az adatokat? A **program futása során a műveletek elvégzéshez szükséges adatok a memóriában** (RAM-ban) tárolódnak. De hogyan tudjuk oda berakni, és onnan kivenni ezeket? Úgy, hogy a **változóknak nevet** adunk. Bármilyet? Hááát. Nagyjából igen, C#-ban biztosan, de nem ajánljuk. Érdemes inkább **az angol abc betűit, számokat és aláhúzás karaktereket használni,** és **kezdjük a változó nevét betűvel**. A speciális karaktereket (pl. #) is kerüljük, mert a fordítóprogram erősen kifogásolja majd. Az ékezetes karakterek használhatóak, de nem ajánljuk.

A változó neve tehát egy azonosító. **A változó nevével tudjuk a memóriában tárolt értéket elérni, onnan kiolvasni, oda kiírni.** Kérdés: és a RAM melyik részén tárolja a változó értékét. Válasz: nem mindegy? Kell ezt nekünk tudni? Felmerül egy másik kérdés is: **milyen adatokat** lehet egy változóban tárolni? Nagyjából bármilyet, csak előtte ezt jelezni kell a fordítóprogramnak. És mi van akkor, ha jelzem, de nem olyan adatot tárolok benne? Akkor vagy a fordítóprogram szól érte, rosszabb esetben futási hibát kapunk, azaz a programunk nem fut le, hanem úgynevezett kivételt dob. Vagy röviden: futási hibával leáll.

Tovább a következő leckére

Tananyag feladatokkal

- **Elméleti ismeretek**et, illetve a begyakorlást, elsajátítását segítő **gyakorlófeladatok**at tartalmaz.
  - **Szöveges** elemek.
  - **Audiovizuális** elemek.
  - **Játék**ok.
  - Gyakorló **feladat**ok.



Programozási nyelvek ⇨ C# ⇨ Változók ⇨ Beszélő változók

Beszélő változók

Visszatérve a változóra: akkor ez most mi is? Egy "dögös" definícióval: olyan programozói objektum, amelynek van neve, értéke (az adat, amit tárol), címe (a memóriában hol tárolódik) és típusa (attribútuma).
A C#-ban így lehet változót deklarálni:

```
int a;
```

Hurrá! De hol itt a négy "valami"? A változó neve nyilván "a". A típusa **int**. Magyarul: integer, azaz egész. Ez azt jelenti, hogy ebbe a változóba **egész számokat lehet** majd tárolni. De hol itt az érték? Sehol. A C#-ban, ha nem adunk egy változónak értéket, akkor majd a fordító megteszi, és 0-t ad egy ilyen int típusú változónak.(Viszont amíg nem adunk neki mi kezdőértéket, addig nem használhatjuk.) És hol a **címe** a változónak? Azt nem tudjuk. De nem is kell, hiszen elég, ha tudjuk a nevét. A fordítóprogram majd tudja, hogy honnan kell a változó értékét "előcsalogatni" a memóriából.

Célszerű a változóknak olyan nevet adni, amely utal arra, hogy mit tárolunk benne. Kivétel ez alól az úgynevezett ciklusváltozók (lásd később), amelyek többnyire i, j, k nevet szoktak kapni, de ez nem kötelező.
Ha például egy másodfokú egyenletet megoldó algoritmust kódolunk, akkor célszerű a két gyököt (megoldást) x1, x2 változókba tárolni, ahogyan a matematikusok teszik:

```
int x1, x2;
```

Tehát lehet egyszerre két változót is deklarálni. A végén az utasítást pontosvesszővel kell lezárni!

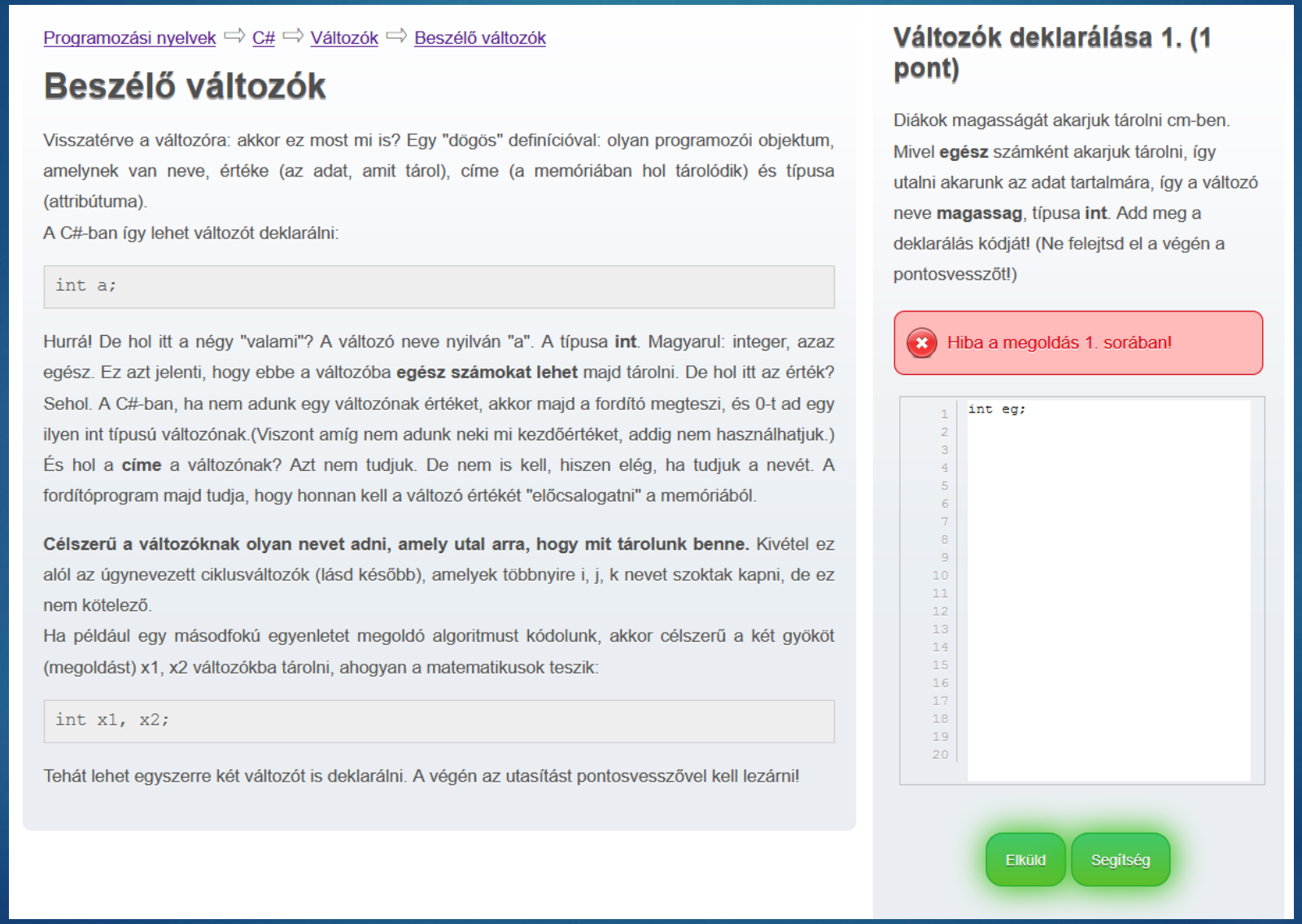
Változók deklarálása 1. (1 pont)

Diákok magasságát akarjuk tárolni cm-ben. Mivel **egész** számként akarjuk tárolni, így utalni akarunk az adat tartalmára, így a változó neve **magassag**, típusa **int**. Add meg a deklarálás kódját! (Ne felejtsd el a végén a pontosvesszőt!)

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21

Elküld Segítség

Hibás megoldás beküldése esetén…



Programozási nyelvek ⇨ C# ⇨ Változók ⇨ Beszélő változók

Beszélő változók

Visszatérve a változóra: akkor ez most mi is? Egy "dögös" definícióval: olyan programozói objektum, amelynek van neve, értéke (az adat, amit tárol), címe (a memóriában hol tárolódik) és típusa (attribútuma).
A C#-ban így lehet változót deklarálni:

```
int a;
```

Hurrá! De hol itt a négy "valami"? A változó neve nyilván "a". A típusa **int**. Magyarul: integer, azaz egész. Ez azt jelenti, hogy ebbe a változóba **egész számokat lehet** majd tárolni. De hol itt az érték? Sehol. A C#-ban, ha nem adunk egy változónak értéket, akkor majd a fordító megteszi, és 0-t ad egy ilyen int típusú változónak.(Viszont amíg nem adunk neki mi kezdőértéket, addig nem használhatjuk.) És hol a **címe** a változónak? Azt nem tudjuk. De nem is kell, hiszen elég, ha tudjuk a nevét. A fordítóprogram majd tudja, hogy honnan kell a változó értékét "előcsalogatni" a memóriából.

Célszerű a változóknak olyan nevet adni, amely utal arra, hogy mit tárolunk benne. Kivétel ez alól az úgynevezett ciklusváltozók (lásd később), amelyek többnyire i, j, k nevet szoktak kapni, de ez nem kötelező.
Ha például egy másodfokú egyenletet megoldó algoritmust kódolunk, akkor célszerű a két gyököt (megoldást) x1, x2 változókba tárolni, ahogyan a matematikusok teszik:

```
int x1, x2;
```

Tehát lehet egyszerre két változót is deklarálni. A végén az utasítást pontosvesszővel kell lezárni!

Változók deklarálása 1. (1 pont)

Diákok magasságát akarjuk tárolni cm-ben. Mivel **egész** számként akarjuk tárolni, így utalni akarunk az adat tartalmára, így a változó neve **magassag**, típusa **int**. Add meg a deklarálás kódját! (Ne felejtsd el a végén a pontosvesszőt!)

Hiba a megoldás 1. sorában!

```
1  int eg;
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
```

Elküld Segítség

Segítség kérése esetén…

Programozási nyelvek ⇨ C# ⇨ Változók ⇨ Beszélő változók

Beszélő változók

Visszatérve a változóra: akkor ez most mi is? Egy "dögös" definícióval: olyan programozói objektum, amelynek van neve, értéke (az adat, amit tárol), címe (a memóriában hol tárolódik) és típusa (attribútuma).
A C#-ban így lehet változót deklarálni:

```
int a;
```

Hurrá! De hol itt a négy "valami"? A változó neve nyilván "a". A típusa **int**. Magyarul: integer, azaz egész. Ez azt jelenti, hogy ebbe a változóba **egész számokat lehet** majd tárolni. De hol itt az érték? Sehol. A C#-ban, ha nem adunk egy változónak értéket, akkor majd a fordító megteszi, és 0-t ad egy ilyen int típusú változónak.(Viszont amíg nem adunk neki mi kezdőértéket, addig nem használhatjuk.) És hol a **címe** a változónak? Azt nem tudjuk. De nem is kell, hiszen elég, ha tudjuk a nevét. A fordítóprogram majd tudja, hogy honnan kell a változó értékét "előcsalogatni" a memóriából.

Célszerű a változóknak olyan nevet adni, amely utal arra, hogy mit tárolunk benne. Kivétel ez alól az úgynevezett ciklusváltozók (lásd később), amelyek többnyire i, j, k nevet szoktak kapni, de ez nem kötelező.
Ha például egy másodfokú egyenletet megoldó algoritmust kódolunk, akkor célszerű a két gyököt (megoldást) x1, x2 változókba tárolni, ahogyan a matematikusok teszik:

```
int x1, x2;
```

Tehát lehet egyszerre két változót is deklarálni. A végén az utasítást pontosvesszővel kell lezárni!

Változók deklarálása 1. (1 pont)

Diákok magasságát akarjuk tárolni cm-ben. Mivel **egész** számként akarjuk tárolni, így utalni akarunk az adat tartalmára, így a változó neve **magassag**, típusa **int**. Add meg a deklarálás kódját! (Ne felejtsd el a végén a pontosvesszőt!)

int avaltozoneve;

```
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
```

Elküld Segítség

Önálló feladat

- Összetett feladat. Egy-egy **részegység gyakorlás**ára és a korábbi tudással való **integrálás**ra.
  - Fejlesztőkörnyezet használata.
  - Szintaktikai és szemantikai tudás fejlesztése, hibarendszerek megismerése.
- **Sandbox környezet** kialakítása: a felhasználó által megvalósított programforrások fordítása és futtatása ellenőrzött körülmények között.

Landolás ismeretlen bolygón

Űrhajóval kell landolni egy elhagyatott bolygón. A bolygón nagyon nagyok a hőmérsékleti különbségek, a hőmérséklet gyorsan változik, van, amikor nagyon hideg van (akár -200 Celsius is), van, amikor nagyon meleg (akár +400 Celsius is). Ezért akkorra kell a landolást időzíteni, amikor a hőmérséklet a 0-hoz a legközelebb van, hiszen ha változik a hőmérséklet, több idő van a felszállásra, mivel az űrhajó csak a +100,-100 tartományban képes felszállni.

írj egy programot, amely megadja, hogy hányadik adat van a legközelebb a 0-hoz, azaz mikor érdemes leszállni a bolygóra!
Ha 2 szám is legközelebb van a 0-hoz, akkor a pozitív sorszámát kell kiíratni. Például, ha a megoldás -2 és +2, akkor a +2 sorszámát kell kiírnia a programodnak. Ha több megoldás is van, akkor az első sorszámát írasd ki! Ha nincs hőmérséklet adat, akkor a program 0-t írjon ki!

A standard inputról kell beolvasnod a hőmérsékleti adatok számát (0<=N<10 000), a következő sorban N darab egész számot kell beolvasni, közöttük pontosan egy szóköz van. Ezek a számok a mért hőmérsékleti adatok.

Példa

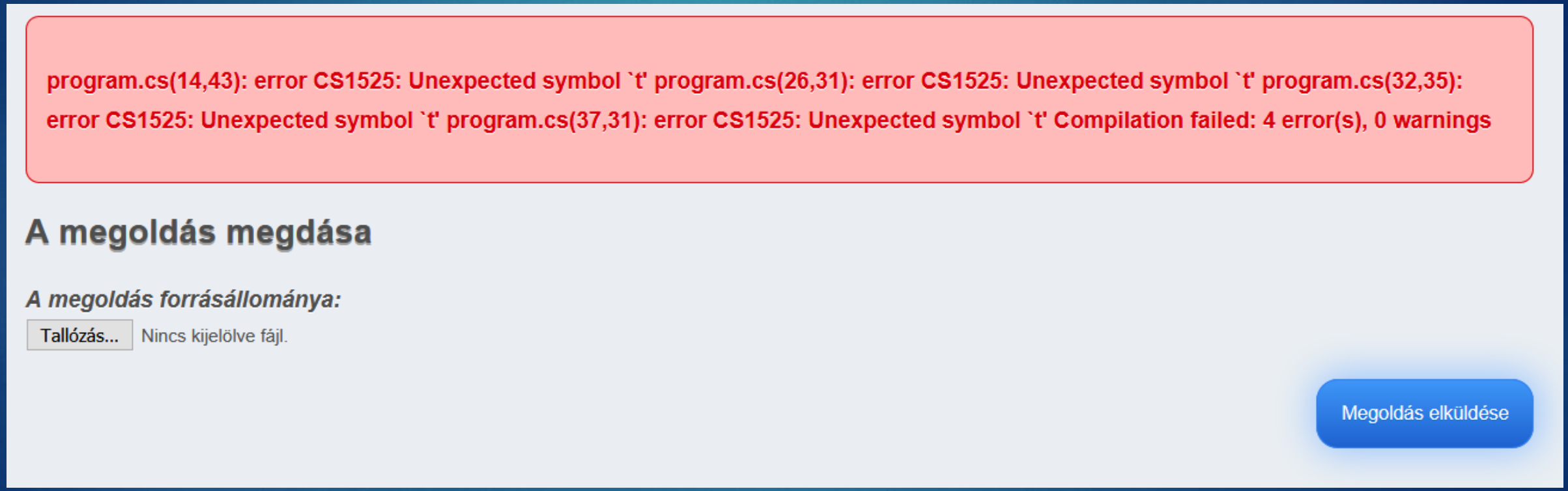
```
Input
6
12 -2 -8 2 3 2

Output
4
```

Azaz, a 4. adat (2) van a legközelebb a 0-hoz. (A 2. is jó lenne, de van +2 is. A 6. is jó lenne, de az első sorszámát kellett megadni.)

1. A feladat felhasználói beküldése.
2. Szintaktikai és szemantikai ellenőrzést.
3. A program futtatása, a feladat definiálása során megadott input adatok felhasználásával ➔ Minden inputhoz egy output fájl előállítása.
4. A feladat definiálás során megadott outputok és a felhasználói kimenetek összehasonlítása.

A program tesztelése I.
Szintaktikai hiba



A program tesztelése II.
Helytelen kimenet

Tesztek eredménye

Sorszám	Teszteset pontszám	Szerezett pont	Státusz
1	1	0	
2	1	0	
3	1	0	
4	1	0	

A program tesztelése II.
Helyes kimenet

Tesztek eredménye

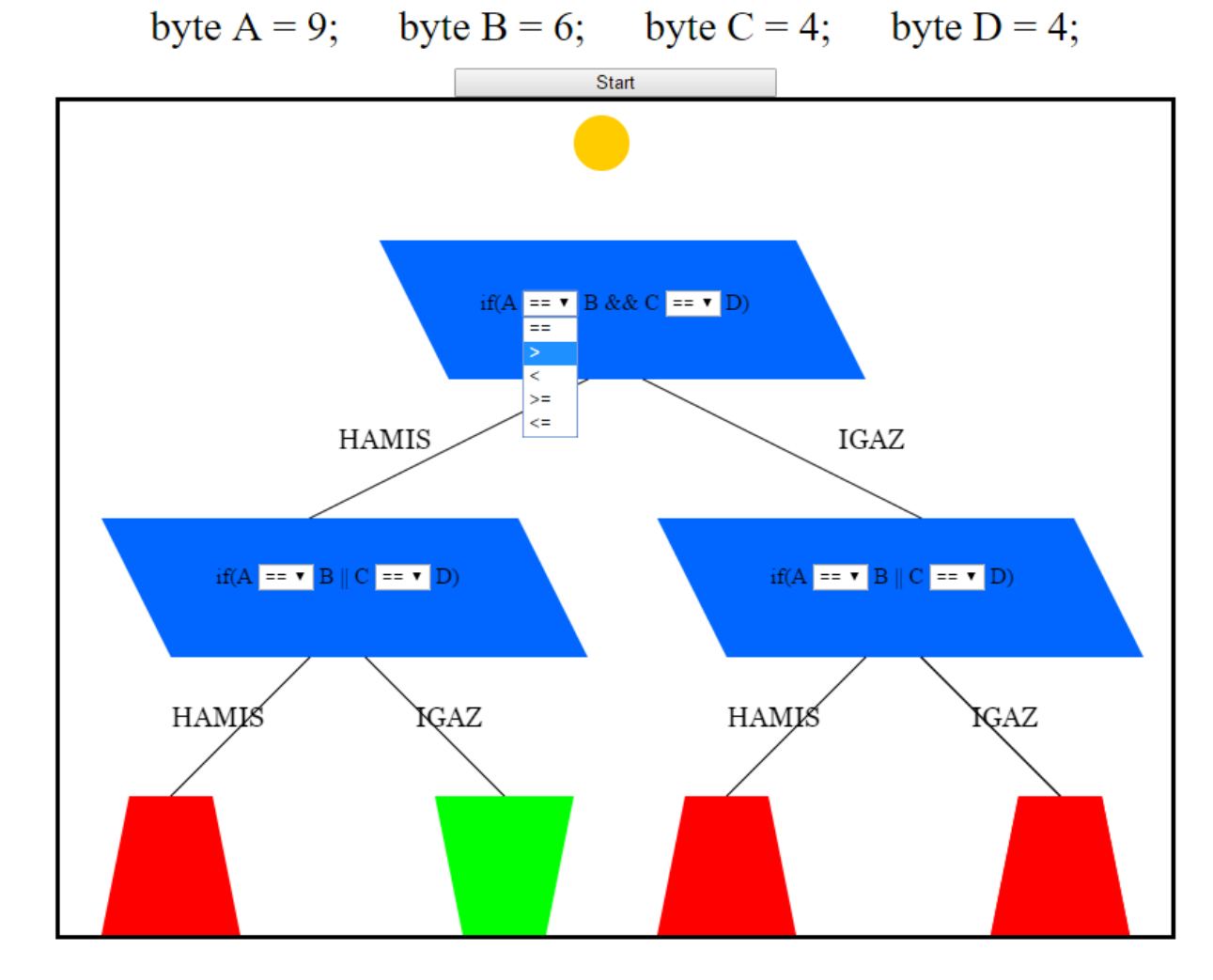
Sorszám	Teszteset pontszám	Szerezett pont	Státusz
1	1	1	
2	1	1	
3	1	1	
4	1	1	

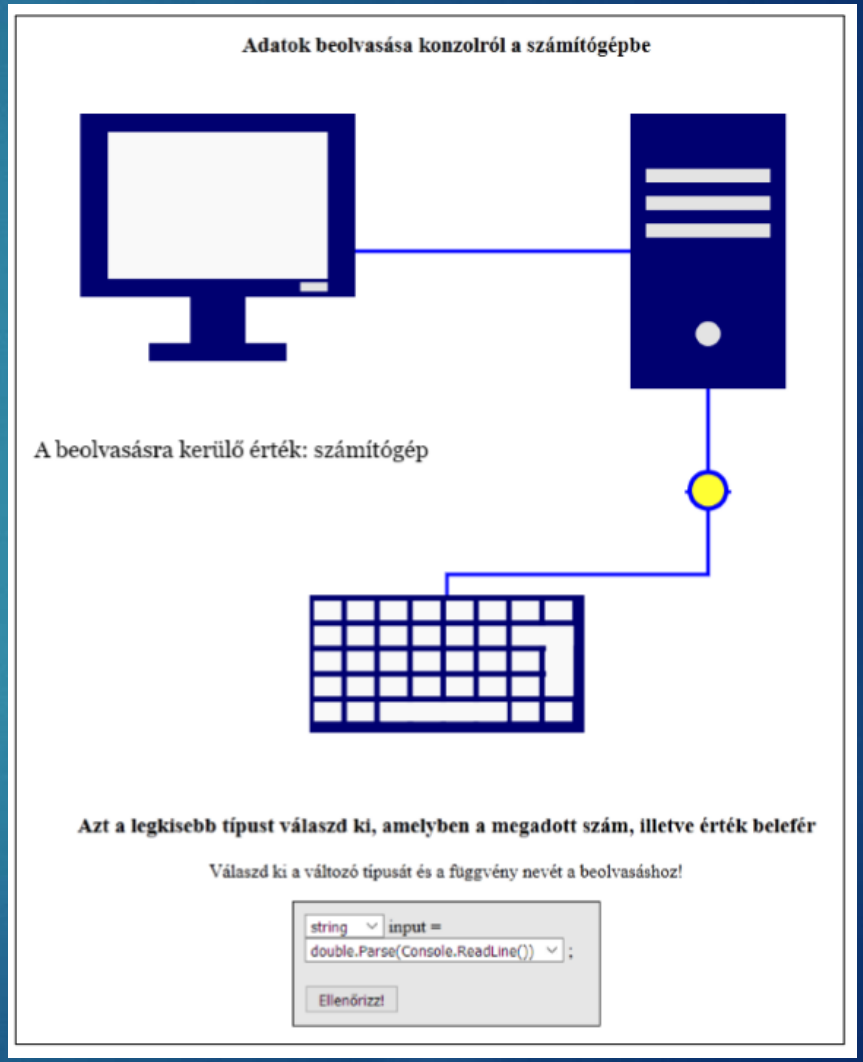
Tovább a következő feladatra...

Gamifikáció

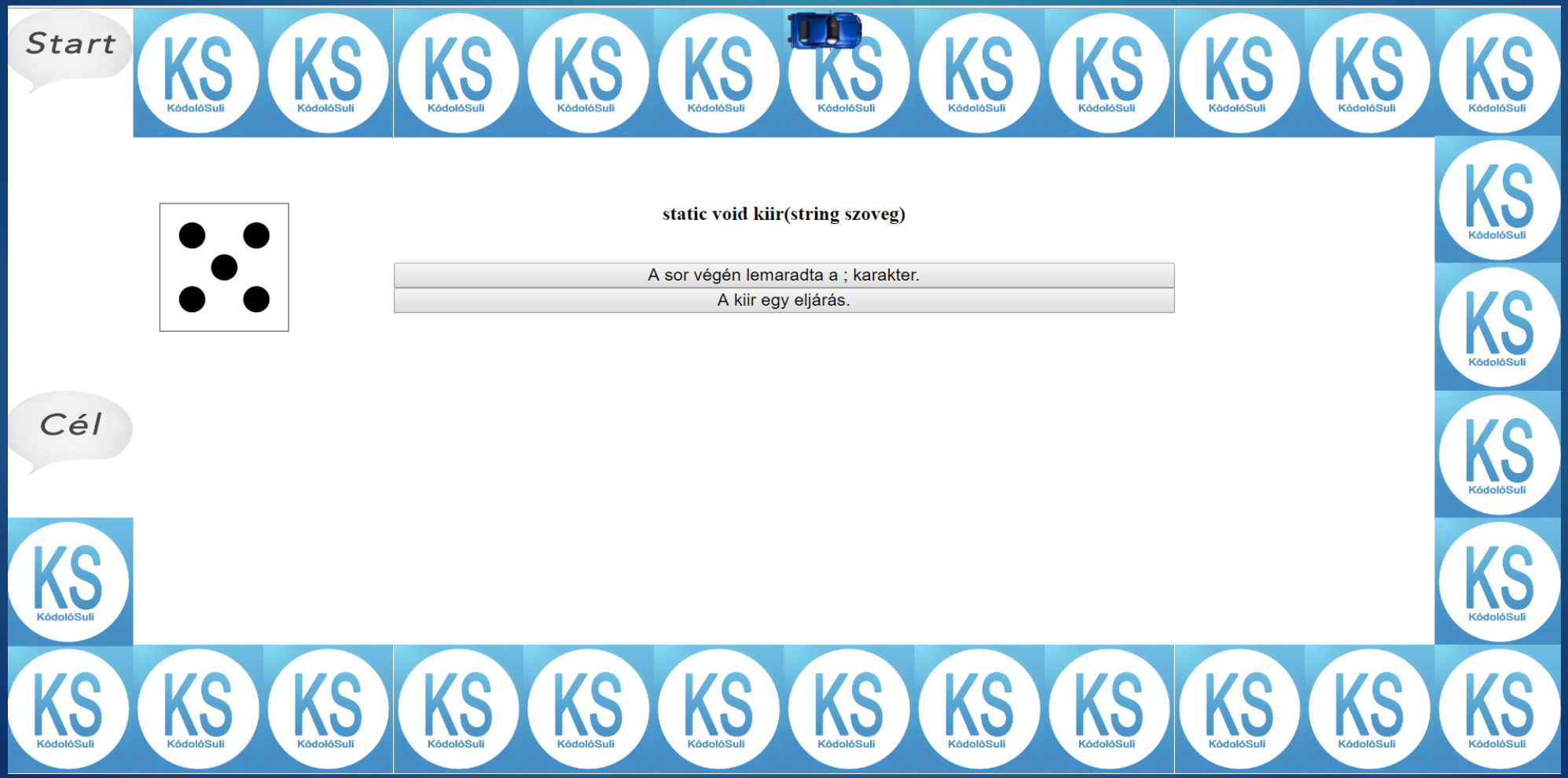
- Programozni tanulni merőben más, mint az alapvető informatikai hardver- és szoftverismeret.
  - Nehéz megbarátkozni a programozási struktúrákkal és a programfejlesztés- és végrehajtás során lezajló folyamatokkal.
  - Nem triviális a motiváció felkeltése és fenntartása.
- Gamifikció: 19 gamifikált tananyag
  - növeli a diákok elkötelezettségét,
  - elősegítik a diagnosztikus mérést és értékelést.

Játékok I.


byte A = 9;
byte B = 6;
byte C = 4;
byte D = 4;
Start
if(A == B && C == D)
==
>
<
>=
<=
HAMIS
IGAZ
if(A == B || C == D)
if(A == B || C == D)
HAMIS
IGAZ
HAMIS
IGAZ


Adatok beolvasása konzolról a számítógépbe
A beolvasásra kerülő érték: számítógép
Azt a legkisebb típust válaszd ki, amelyben a megadott szám, illetve érték belefér
Válaszd ki a változó típusát és a függvény nevét a beolvasáshoz!
string input =
double.Parse(Console.ReadLine());
Ellenőrizz!

Játékok II.


Start
static void kiir(string szoveg)
A sor végén lemaradta a ; karakter.
A kiir egy eljárás.
Cél

Jövőbeni tervek

- Értékelőrendszer fejlesztése
  - Pedagógiai mérési mutatók elemzésének biztosítása.
  - Item-alapú feladatbank építése.
  - A forráskód struktúrájának, felépítésének elemzése.
  - Unit tesztek segítségével történő tesztelés.
- Online osztálytermek kialakítása.
  - Dolgozatok megíratása, javítása, elemzése.
- Neurális hálózat beépítése.

Köszönjük a figyelmet!

BALLA.TAMAS@UNI-ESZTERHAZY.HU

KIRALY.SANDOR@UNI-ESZTERHAZY.HU